Thesis for The Degree of Doctor of Philosophy

Bridging Worlds: Integrating Human Factors in Agile Automated Vehicle Development

Amna Pir Muhammad

Division of Interaction Design and Software Engineering
Department of Computer Science & Engineering
Chalmers University of Technology and University of Gothenburg
Gothenburg, Sweden, 2025

Bridging Worlds: Integrating Human Factors in Agile Automated Vehicle Development

Amna Pir Muhammad

Technical Report No 5606
ISSN 0346-718X
Department of Computer Science & Engineering
Division of Interaction Design and Software Engineering
Chalmers University of Technology and Gothenburg University
Gothenburg, Sweden

This thesis has been prepared using LaTeX.
Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2025.

Cover Illustration: Cover image titled "Human-Automation Synergy" created by Amna Pir Muhammad using AI tools
Edited with the help of Beatriz Cabrero-Daniel using AI tools

Human-Automation Synergy

amnap@chalmers.se

Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2025.

Science is a wonderful thing if one does not have to earn one's living at it." - Albert Einstein.

Abstract

Context: Automated Vehicle (AV) technology has grown significantly in complexity and impact, promising to transform urban transportation. However, research shows that vehicle automation can only live up to this expectation if it is designed with human capabilities and limitations in mind. Integrating human factors knowledge into AV development is, therefore, essential. Traditionally, this integration has relied on upfront requirements during pre-development. The adoption of agile methodologies, which lack such upfront processes, necessitates new approaches for integrating human factors into agile AV development. This study addresses this challenge by exploring the integration of human factors knowledge within agile AV development from a requirements engineering perspective.

Objective: This thesis empirically examines how to efficiently incorporate human factors knowledge into large-scale agile AV development, identifying practical strategies to address this need.

Method: The research employs a mixed-methods approach, including interviews, workshops, document analysis, and surveys, to gather both qualitative and quantitative data. These methods provide insights into developing strategies for integrating human factors knowledge into agile AV development.

Findings: Initial findings highlight several challenges in integrating human factors knowledge, such as inadequate tools, methods, and expertise. It highlight the need for strategies to effectively capture and apply human factors requirements. Experiments emerged as a critical element, offering insights into human interactions with complex systems. As software-based systems grow increasingly complex, companies are not only adopting agile development methodologies but also placing greater emphasis on continuous software experimentation to adapt more effectively to evolving requirements. Building on these findings, a follow-up study examined the feasibility of using continuous experimentation to integrate human factors knowledge into agile AV development. Continuous experimentation alone proved insufficient to fully integrate human factors knowledge into agile processes. While it supports rapid feedback and iterative improvements, it does not accommodate the specific experiments required for addressing human factors effectively.

To address these gaps, the study applied a requirements engineering perspective. The concept of Requirements Strategies emerged, providing organizations with structured guidelines for defining and implementing effective approaches to manage their specific requirements in agile development. These guidelines emphasize three main components: structural, organizational, and work and feature flow perspectives. This concept was then used as a lens to collect best practices for the integration of human factors requirements in agile AV development.

In agile development, autonomous teams make localized decisions and discover new knowledge independently, often relying on implicit expertise. Effective integration of human factors requires teams to possess or have access to such knowledge. Given the scarcity of human factors experts, strategic placement of this expertise within organizations becomes critical. The study identifies optimal placements to enhance the management of human factors requirements and their integration into agile processes.

Conclusion: This research offers strategies, informed by practitioner feedback and study findings, to integrate human factors knowledge into agile AV development. These strategies are framed across structural, organizational, and work and feature flow perspectives. Additionally, the placement of human factors expertise within organizations is suggested to manage these requirements effectively and and maximize the impact of human factors considerations on final products. The findings contribute to the ongoing discourse on how to effectively incorporate human-centric considerations into the rapidly evolving field of automated vehicle development.

Keywords

Agile, Scaled Agile Development, Requirements Engineering, Human Factors, Automated Vehicles, AV Development, Requirements Strategy

Acknowledgment

Completing this PhD thesis has been a long and rewarding journey, and I would not have reached this milestone without the support and guidance of many individuals.

First and foremost, I would like to express my deepest gratitude to my supervisors, Eric Knauss, Jonas Bärgman, and Alessia Knauss. Your collective expertise, support, and encouragement have been instrumental in shaping this work. Despite your busy schedules, you always made time for me, helping me grow as a researcher and enhancing my critical thinking skills.

My main supervisor, Eric Knauss, provided unwavering support throughout the process and beyond. I was fortunate to receive not only top-notch scientific education and guidance but also more support and encouragement than I could have ever hoped for. Thank you from the bottom of my heart.

I extend many thanks to my co-supervisor, Jonas Bärgman, for his insightful advice and valuable feedback that sharpened my skills. Your keen eye for detail was invaluable in clarifying complex ideas.

Alessia, thank you especially for your academic insights and advice, which significantly contributed to the depth and rigor of this research. Your support extended beyond academic matters; starting my PhD in a new country during the onset of COVID-19 was challenging, but your guidance, kindness, and encouragement made me feel at home. You were not just a mentor but also a source of comfort and reassurance, for which I am deeply grateful.

I am also grateful to my examiner, Richard Torker, who gave me the freedom to conduct independent research and offered his support throughout this journey.

I would also like to thank Anne Hess, Daniel Amyot, Maya Daneva, and MariAnne Karlsson for accepting to serve on my committee and for their valuable feedback on earlier versions of the thesis. Your insights and suggestions have contributed to improving the quality of this work.

My heartfelt thanks go to my colleagues and friends. The list is too long to mention everyone here, but some feature later in the acknowledgments. The camaraderie, discussions, and support within the division greatly enriched my research experience. I am grateful for the engaging conversations and the enjoyable moments we shared over fika, lunch, and after work.

I also want to thank my officemate, Khan Habib, for listening, discussing, and being a great officemate. It was wonderful getting to know you better. I am also grateful to Wardah Mehmood for always being there to guide me and answer my questions. You were always available to talk, offer advice, and share your perspectives on various matters beyond work. For this, I am truly thankful.

I would also like to thank Hans Martin and Irum Inayat for their collaboration. Working together on papers was a rewarding experience, and I truly value your contributions and insights. Irum, although you joined recently, you have been invaluable. Thank you for your support.

Hina Saeeda, even though you joined earlier this year, I am truly thankful for your collegial support and kindness, as well as the moments we've shared beyond work.

Thank you to everyone I have crossed paths with over the last five years who made Gothenburg feel like home. I especially want to thank my colleagues and friends at the Interaction Design and Software Engineering division, including Bea, Tayssir, Malsha, Cristy, Chi, Mazen, Ranim, Krishna, Ricardo, Sushant, Teodor, Afonso, Babu, Hamdy, Linda, Sabina, and the entire IDSE division, for being around and making these moments wonderful. I am also grateful to all the Cabo players who helped me relax at the end of the day.

I am thankful to our industry partner, Zenseact, and colleagues there for sharing their resources and expertise on this topic. Moreover, I appreciate all the participants in my studies. Your invaluable time and input contributed to meaningful results.

Beyond university life, I also want to thank several friends who made this place feel like home. Marie-Ann, I am deeply grateful for our friendship. Cheers to our picnics, BBQs, and all the spontaneous days we spent shopping or just hanging out at the playgrounds. I also thank Attiqa Kiyani. You are truly amazing and fun. You made me feel closer to family despite being far from home. Your advice, hangouts, and all the time we spent together meant a lot to me. I am lucky to have you in my life and grateful for the memories we created.

I also thank all my friends who are far away but still in touch. A special thank you to Madiha Fatima, who kept checking in on me every day and motivating me. Our regular conversations, both light-hearted and encouraging, have been a great comfort.

Finally, I want to thank my family. I am deeply grateful to my parents, who have been my greatest role models and supporters throughout my life. You have shown me the value of hard work and perseverance. Your unwavering belief in me is the best gift I could have ever asked for.

To my husband, Babar, I am beyond words grateful to you. You have been the greatest support anyone could wish for—cooking the best meals, ensuring I relax and laugh every day, taking care of our little one, traveling with me, and patiently listening to my concerns about work. Your love and understanding have been my anchor through the ups and downs of this journey. Thank you.

A special note of gratitude goes to my daughter, Hafsa Farooq. Your presence has been a source of immense joy and motivation. Balancing research and parenthood has been challenging, but your smiles and laughter have made it all worthwhile. You inspire me to be the best version of myself every day.

Finally, I would like to thank SHAPE-IT for funding my work and providing great opportunities. I am also grateful for all the research visits I participated in as part of the SHAPE-IT project. These visits allowed me to meet many new and inspiring individuals. The training sessions, university visits, and fun activities were invaluable experiences, and I have fond memories of all these events.

This project received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement 860410.

List of Publications

Appended publications

This thesis is based on the following publications:

[A] Amna Pir Muhammad, Eric Knauss, Jonas Bargman "Human Factors in Developing Automated Vehicles: A Requirements Engineering Perspective"
Journal of Systems and Software, 2023.

[B] Amna Pir Muhammad, Eric Knauss, Jonas Bargman, Alessia Knauss "Continuous Experimentation and Human Factors: An Exploratory Study"
In the International Conference on Product-Focused Software Process Improvement, 2023.

[C] Amna Pir Muhammad, Eric Knauss, Odzaya Batsaikhan, Nassiba El Haskouri, Yi-Chun Lin, and Alessia Knauss "Defining Requirements Strategies in Agile: A Design Science Research"
In the International Conference on Product-Focused Software Process Improvement, 2022.

[D] Amna Pir Muhammad, Alessia Knauss, Eric Knauss, Jonas Bärgman "Requirements Strategy for Managing Human Factors in Automated Vehicle Development"
In 32nd IEEE International Requirements Engineering Conference(RE'24), 2024.

[E] Amna Pir Muhammad, Alessia Knauss, Eric Knauss, Jonas Bärgman "Integrating Human Factors Expertise into Development of Automated Vehicles"
In submission to Empirical Software Engineering Journal, 2024.

Other publications and research outputs

The following publication was published during my PhD studies. However, it is not appended, due to contents not related to the thesis.

[a] **Amna Pir Muhammad**, Eric Knauss, Jonas Bärgman, Alessia Knauss
"Managing Human Factors in Automated Vehicle Development: Towards Challenges and Practices"
In 31st IEEE International Requirements Engineering Conference(RE'23), 2023.

[b] Christian Berger, Chi Zhang, **Amna Pir Muhammad**, Eric Knauss.
"The use of AI in AV human-factors research and human-factors requirements in AI-based AV design"
Deliverable D2.4 in the EC ITN project SHAPE-IT. SHAPE-IT Consortium. 2023.

[c] de Winter, Joost, Siri Hegna Berge, Wilbert Tabone, Yue Yang, **Amna Pir Muhammad**, Sarang Jokhio, Marjan Hagenzieker.
"Design strategies and prototype HMI designs for pedestrians, cyclists, and non-automated cars"
Deliverable D2.5 in the EC ITN project SHAPE-IT. SHAPE-IT Consortium. 2023.

[d] Natasha Merat, Yee Mun Lee, Chen Peng, Nikol Figalova, Naomi Mbelekani, **Amna Pir Muhammad**, Liu Yuan-Cheng, Xiaolin He, Xiaomi Yang.
"Design guidelines for acceptable, transparent, and safe AVs in urban environments"
Deliverable 2.6 in the EC ITN project SHAPE-IT. SHAPE-IT Consortium. 2023.

[e] Hans-Martin Heyn, Eric Knauss, **Amna Pir Muhammad**, Olof Eriksson, Jennifer Linder, Padmini Subbiah, Shameer Kumar Pradhan, Sagar Tungal.
"Requirement Engineering Challenges for Ai-intense Systems Development"
In 2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN) (pp. 89-96). IEEE, 2021.

[f] **Amna Pir Muhammad.**
"Methods and Guidelines for Incorporating Human Factors Requirements in Automated Vehicles Development"
REFSQ Workshops. 2021.

[g] Natasha Merat, Yue Yang, Yee Mun Lee, Siri Hegna Berge, Nikol Figalova, Sarang Jokhio, Chen Peng, Naomi Mbelekani, Mohamed Nasser, **Amna Pir Muhammed**, Wilbert Tabone, Liu Yuan-Cheng, Jonas Bärgman.
"An Overview of Interfaces for Automated Vehicles (inside/outside)"
Deliverable D2.1 in the EC ITN project SHAPE-IT. SHAPE-IT Consortium. 2021.

[h] Nikol Figalova, Naomi Mbelekani, Chi Zhang, Yue Yang, Chen Peng, Mohamed Nasser, Liu Yuan-Cheng, **Amna Pir Muhammed**, Wilbert Tabone, Siri Hegna Berge, Sarang Jokhio, Xiaolin He, Amir Hossein Kalantari, Ali Mohammadi, Xiaomi Yang.
"Methodological Framework for Modelling and Empirical Approaches" *Deliverable D1.1 in the EC ITN project SHAPE-IT. SHAPE-IT Consortium. 2021.*

Research Contribution

The included papers in this thesis were published with co-authors. My specific contributions to each papers are as follows:

For Paper A, I was responsible for data collection and analysis. Additionally, I took the lead in conceptualization, methodology, and writing the original draft, as well as handling most of the writing and final publication. My co-authors contributed by supporting the conceptualization and methodology, providing valuable feedback through review and editing, and assisting in the investigation and validation of the findings. They also contributed to visualization of the results.

For Papers B, D, and E, I assumed the main responsibility for the design of the study, data collection, data analysis, discussion, and writing. The co-authors contributed to the research method, provided reviews, and suggested improvements.

Paper C is based on three case studies. I contributed to designing the overall study and took a leading role in data collection and analysis for the third case, which included a workshop, document analysis, and observations at a company. My co-authors collected the initial data for two of the cases. I also led the execution and writing of the final paper, synthesizing the findings from all three studies.

Ethical Considerations

This research adhered to Swedish law and GDPR requirements, ensuring that all participant data was handled ethically and securely. Full informed consent was obtained, anonymity was ensured, and no high-risk personal data was collected. Raw data was stored securely on personal computers and not shared publicly. Transcripts and findings were reviewed with participants to confirm accuracy and avoid revealing sensitive information. Further details on ethical considerations are discussed in Section 1.3.4.

Contents

Chapter 1

Introduction

The development of automated vehicles (AVs) represents a significant shift in urban transportation, promising numerous benefits such as reduced accidents, injuries, or fatalities [1, 2]. This potential has spurred competition within the automotive industry to create and market AVs with varying levels of automation [1], ranging from Advanced Driver Assistance Systems (ADAS) that support specific driving tasks to fully autonomous vehicles that can operate independently under certain conditions [4]. These advancements in AVs rely heavily on complex software and artificial intelligence, necessitating careful design considerations.

Despite their many advantages, AVs present several challenges, including over-trust in automation, increased workload for human operators, and issues related to driver engagement and re-engagement. These challenges often stem from inadequate consideration of human cognitive and physiological limits during the design process [5]. To effectively overcome these challenges and achieve the full potential of AVs, researchers emphasize the importance of incorporating human factors knowledge into their design [6, 7]. Integrating human factors knowledge into the design and engineering of AVs ensures their safety, usability, and public acceptance [8, 9, 10].

Human factors knowledge encompasses a broad spectrum of considerations. These include, for example, user experience, human-machine interaction, cognitive ergonomics, and human capabilities and limitations in relation to the design and operation of systems [9, 11]. These considerations are often translated into human factors requirements, which specify how systems must account for human capabilities and limitations to meet specific quality objectives. Unlike functional requirements, which describe what a system must do, or non-functional requirements, which address properties like reliability and scalability, human factors requirements focus on human-centered design principles. They emphasize quality attributes such as usability and safety.

For example, a functional requirement might state: "The vehicle must alert the driver when lane departure is detected." Implementing this functionality, however, involves significant human factors considerations, as outlined in UN

[1]Definitions of levels of automation, such as those proposed by SAE [3], are well-documented. However, the interpretation of these levels remains a topic of ongoing discussion. This work aims to address aspects relevant to all levels of automation.

Regulation No. 157 - Automated Lane Keeping Systems (ALKS) [12]. A corresponding human factors requirement could specify: “The system must provide a clear and precise warning to the driver when lane departure is detected. Sensory feedback, such as visual alerts on the dashboard or tactile vibrations on the steering wheel, should ensure immediate recognition and prompt corrective action. False or excessive warnings must be minimized to maintain trust.”

This example highlights how human factors requirements prioritize user comprehension, responsiveness, and the prevention of negative effects, such as over-reliance or reduced trust. By addressing human capabilities and limitations, these requirements ensure systems are safe, usable, and effective.

Despite the recognized importance of human factors requirements in AV development, current software development practices in the automotive industry often fall short in effectively incorporating human factors knowledge [13, 14]. Researchers suggest integrating this knowledge early in the design process to ensure human factors requirements are well-defined and actionable [15, 16, 17]. While traditionally integrated into upfront system requirements, the shift to agile methodologies—driven by the need for faster feature delivery and iterative development—has created new challenges for incorporating human factors knowledge effectively.

Agile development, characterized by its iterative and incremental workflows, is increasingly adopted in AV development due to its flexibility and efficiency [18, 19]. This shift, while enabling faster delivery, complicates the management and communication of human factors requirements, particularly in large-scale systems [20, 21]. Agile’s reduced emphasis on upfront documentation risks overlooking critical human factors considerations, highlighting a tension between the demands of agile development and the need for structured integration of human factors requirements.

To address this gap, this research adapts requirements engineering practices to align with agile methodologies. The goal is to investigate how human factors requirements can be effectively incorporated into large-scale agile AV development processes. Balancing agile’s focus on rapid delivery with the structured integration of human factors requirements is crucial to ensuring usability, safety, and human-centered design principles are not compromised [22].

In summary, the integration of human factors knowledge into agile AV development is critical for creating safe, user-friendly automated vehicles. However, significant challenges remain, particularly in aligning traditional human factors research methods with agile development practices. Investigating this integration through the lens of requirements engineering provides a structured approach to overcoming these challenges, ensuring that human factors considerations are effectively incorporated into the agile development.

1.1 Research Goal and Questions

1.1.1 Research Goal

The primary goal of this study is to provide empirical insights into the integration of human factors knowledge into agile AV development and to investigate solutions for enhancing this integration. The overall goal is:

Goal	Paper A: HF in agile	Paper B: HF Experimentation	Paper C: Strategies - Development	Paper D: Concrete HF Strategies	Paper E: HF experts Placement
	RQ1	RQ2	RQ1,RQ3	RQ3	RQ4
G1: Exploration	✓	✓	✓		
G2: Solutions			✓	✓	✓

Figure 1.1: An overview of the papers in relation to the research goals and research questions

Goal (G:) To investigate how to effectively bring requirements based on human factors knowledge to automated vehicle developers in agile development.

To achieve this high-level goal, it is further divided into two subgoals:

Goal 1 (G1): Domain Exploration. Explore the integration of human factors knowledge into agile development.

Goal 2 (G2): Solution Investigation. Identify solutions for integrating human factors knowledge into agile AV development.

Given the multidisciplinary nature of this work, it is essential to examine various aspects of the problem from multiple perspectives. G1 establishes the context by exploring key areas, specifically requirements engineering, agile development, and human factors in AV development. G1 identifies the relevant properties of agile methodologies and human factors, explaining the implications of agile ways of working, human factors, and requirements engineering. Additionally, G1 investigates if human factors experimentation can be integrated with continuous software experimentation, acknowledging the important role that human factors play in this process.

G2 builds on the insights from G1 to propose strategies for improving the integration of human factors requirements into agile AV development. This goal addresses the solution from two perspectives: the integration of human factors requirements and the integration of human factors experts. First, it aims to identify strategies for integrating human factors requirements into agile AV development. Second, it identifies the optimal placement options for human factors experts within an organization, allowing them to contribute most effectively. This optimizes their expertise to better manage human factors requirements and ultimately maximizes their impact on the product.

1.1.2 Research Questions

we aim to achieve the goal of this research by providing empirical insights into the integration of human factors knowledge in agile AV development and exploring solutions to enhance this integration. The focus is on incorporating human factors requirements into agile AV development. To achieve these goals, the following research questions were formulated:

G1: **Domain Exploration:**

To accomplish this goal, it is essential to understand practitioners' problems and needs. This leads to the following research question:

RQ1: To what extent can human factors knowledge be integrated into agile AV development?

This question aims to lay the foundation for including human factors knowledge in agile AV development. The first study explored how human factors knowledge can be systematically captured and managed, particularly in relation to agile practices and requirements engineering.

Through this investigation, key properties and implications for the integration of human factors requirements into agile AV development were identified. Specifically, the findings highlighted both the potential benefits of the flexibility offered by agile development and the significant challenges that arise, particularly in managing and communicating human factors requirements. The results indicated a significant gap in tools, methods, and expertise needed to effectively integrate human factors knowledge into the fast-paced, iterative nature of agile development. This gap underscores the need for targeted strategies and frameworks to address the complexities of integrating human factors knowledge into agile AV development.

Additionally, our findings revealed the critical role of human factors experimentation in AV development, leading us to formulate the second research question:

RQ2: **Can continuous experimentation help to integrate human factors knowledge into agile AV development?**

Human factors experimentation focuses on observing and analyzing cognitive, physical, and behavioral responses in realistic scenarios. It provides valuable qualitative insights into areas such as user experience, safety, usability, comfort, and trust, often requiring detailed observation of human behaviors that extend beyond technical metrics. In contrast, while software continuous experimentation also collects real-world data, its primary focus lies in evaluating system performance, feature effectiveness, or technical metrics such as response times and bug occurrences.

To explore RQ2, the similarities and differences between human factors experimentation and continuous experimentation were investigated. Additionally, the challenges involved in integrating and executing these two types of experimentation were examined. The results indicate that integrating human factors experimentation into continuous experimentation is not straightforward, necessitating the development of new methods and strategies to address these complexities effectively.

G2: Solution Investigation:

The focus then shifted to investigate potential solutions in the form of strategies (ranging from rather abstract guidelines to concrete solution spaces) for integrating human factors knowledge into agile AV development, addressing these through the following research questions:

RQ3: What strategies can improve integration of human factors knowledge as requirements within agile AV development?

In the context of requirements engineering, specifically within agile development, RQ3 focuses on defining solution strategies to enhance the integration of requirements in agile development.

Since this work explores the topic within the context of requirements engineering, specifically in agile development, RQ3 focuses on defining solution strategies to enhance the integration of requirements in agile development. To address this, the concept of a "requirements strategy" is introduced, offering a set of guidelines to mitigate challenges associated with requirements in agile development. Building on this concept, concrete solution strategies were explored to effectively incorporate human factors requirements into agile AV development.

RQ4: Where should human factors expertise be brought in to maximize its impact on the product?

In addressing RQ1 and RQ2, it became evident that beyond the need for new tools and methods, it is crucial to strategically decide where human factors expertise should be placed to maximize its benefits in integrating human factors knowledge into agile AV development. Furthermore, RQ3 highlighted the importance of defining clear roles and responsibilities for integrating human factors requirements and managing the requirements structure. This consideration led to an exploration of the optimal placement of human factors experts within organizations to effectively manage these strategies. RQ4 aims to determine where human factors knowledge should be positioned to maximize its impact on the product. This involves identifying the preferred placement options for human factors experts to ensure their expertise is utilized effectively.

This cumulative thesis is built on five publication papers (Chapters 2 – 6). Figure 1.1 illustrates the relationships between the research goals, the thesis research questions, and the included papers. It shows that Papers A, B and part of Paper C explore the domain in detail, contributing to achieving research goal G1. Paper C also ventures into the solution space, contributing to the achievement of goal G2: Solution investigation. Paper D investigates strategies for integrating human factors requirements in AV development. Paper E contributes by identifying and ranking different placement options for positioning human factors experts within organizations, aiming to pinpoint where these experts can have the maximum impact on the product by effectively managing human factors requirements. The relationship between the goals, the papers, and the research methods used in each paper is further illustrated in Figure 1.2.

The remainder of this chapter is organized as follows: Section 1.2 provides background information on requirements engineering, agile development, the automotive industry, and human factors. Section 1.3 presents the research

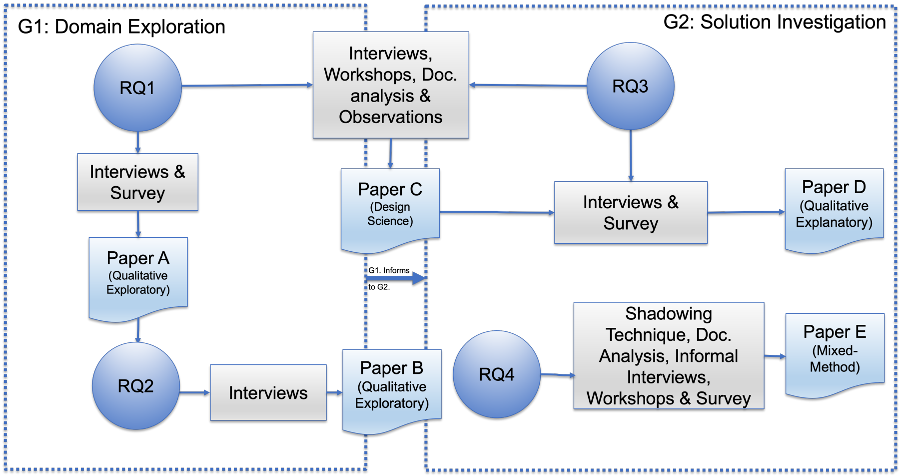


Figure 1.2: Overview of the relationships between the research goals (G1 and G2), research questions (RQ1–RQ4), and the corresponding papers (Papers A–E).

methodology, while Section 1.4 explores threats to validity. Summaries of the included papers are described in Section 1.5. Section 1.6 addresses the research question, and Section 1.7 discusses the findings. Finally, Section 1.8 concludes the introduction chapter.

1.2 Background

This section provides background information on the fundamental concepts used in this thesis. It introduces and clarifies terminologies such as agile development, requirements engineering, human factors, and automated vehicles.

1.2.1 Requirements Engineering

Requirements Engineering (RE) is the systematic process of identifying, documenting, and managing system requirements to ensure that the final system meets the needs and expectations of its stakeholders. It involves capturing both functional (what the system should do) and non-functional (e.g., usability, safety, and performance) requirements and serves as the foundation for subsequent development activities [23]. According to the International Requirements Engineering Board (IREB), RE plays a critical role in large-scale projects by addressing system decomposition, stakeholder management, and change management, with requirements often broken down and allocated to different teams or subsystems as they evolve over time [24].

Traditionally, RE was viewed as a set of sequential activities, including requirements elicitation, analysis, specification, and validation [25]. During elicitation, stakeholder requirements are gathered through various techniques, such as storyboarding, questionnaires, and prototyping. The requirements are then analyzed and any conflicts or redundancies are resolved through negotiation. In the specification phase, requirements are documented formally or informally, such as through diagrams or mathematical models, resulting in a comprehensive requirements document. Finally, these requirements are validated for consistency and completeness.

However, traditional RE methods, with their emphasis on upfront planning and detailed documentation, often struggle to integrate with modern, agile development practices. Agile environments prioritize flexibility, incremental development, and rapid adaptation to changing requirements. As a result, the rigid, sequential nature of traditional RE has become increasingly strained in such settings. This shift has led to the realization that, rather than being pre-specified, requirements are more effectively developed through ongoing use and interaction with the system, especially when user values and expectations are constantly evolving [26].

Recent research has focused on adapting RE to dynamic environments like agile, where continuous integration, frequent requirement changes, and stakeholder collaboration are crucial. The focus has shifted from producing a comprehensive, upfront specification to a more flexible approach that accommodates evolving needs and promotes ongoing refinement throughout the development process [26].

1.2.2 Agile Development

Agile development, commonly referred to as "agile methods," has become increasingly popular in development companies due to its flexibility and potential to improve product success rates, especially when compared to traditional development approaches [27]. This approach emphasizes adaptability, collaboration, and continuous customer feedback throughout the development process. Agile methods, such as Scrum and Kanban, divide projects into small, manageable increments called "sprints," typically lasting between two and four weeks. These iterative cycles allow teams to frequently reassess project goals and incorporate user feedback, leading to more responsive and user-centered development outcomes.

One of the key distinctions of agile development is its encouragement of continuous improvement and openness to changes, even late in the development process, which contrasts sharply with the fixed, upfront planning of traditional models like Waterfall. Agile development involves frequent feedback from users and continuously reflects customer values, leading to more user-centered outcomes [28].

Typically, agile methods are recommended for small teams (six to eight developers) [28, 29]. The Agile Manifesto (Table 1.1) [18] outlines the core values of agile methodologies, which prioritize individuals and interactions, creating functional software in close collaboration with clients. This approach deemphasizes reliance on processes, tools, extensive documentation, and contract negotiation, which are often the foundation of plan-driven approaches.

In agile development, traditional detailed comprehensive documentation of requirements is replaced by continuous communication with customers or product owners [28]. Teams typically start by writing user stories—brief descriptions of client needs—which guide development throughout each sprint. However, some shortcomings of agile development include the reduced focus on

Table 1.1: Manifesto for Agile Software Development [18]

Individuals and interactions	over	processes and tools
Working software	over	comprehensive documentation
Customer collaboration	over	contract negotiation
Responding to change	over	following a plan

upfront planning and the tendency to prioritize functional requirements over more comprehensive, system-wide needs, such as non-functional requirements (NFRs) [28].

The iterative nature of agile methods makes them well-suited for projects in fast-evolving fields like software and automated systems. Practices such as daily stand-up meetings, retrospective reviews, and continuous integration help teams address problems as they arise, integrate user feedback, and adjust priorities. However, the focus on quickly delivering functional features can create challenges when addressing NFRs—such as security, usability, and human factors—which often require more holistic planning and coordination across teams [30].

Research by Ramesh et al. [31] highlights these challenges, noting that the iterative, incremental nature of agile can make it difficult to maintain focus on system-wide qualities, especially when teams are under time and resource constraints. Similarly, Alsaqaf et al. [32] note that large-scale agile projects often struggle to align functional requirements and NFRs due to a lack of coordination between teams working on different system components.

To overcome these challenges, researchers recommend incorporating specific metrics for NFRs into agile development. This ensures that non-functional goals—such as performance benchmarks and security requirements—are clearly defined from the start and continuously validated through testing and stakeholder feedback throughout the development process [33, 34]. By embedding NFRs into agile development early on, teams can avoid the costly consequences of neglecting these critical system attributes during rapid development cycles.

1.2.3 Large-scale Agile Development

Agile methods were originally used by small development teams, but in recent years, they have been increasingly adopted by larger organizations [35]. The term *large-scale agile* describes agile practices applied in larger teams and multi-team projects, typically involving more than two teams, based on the scale taxonomy for agile development [19]. Organizations with more than nine teams are considered *very large-scale*. However, according to Dikert et al., large-scale agile development usually includes more than six teams [36].

Several guidelines and frameworks have been created to apply the agile development in areas beyond software development, such as business strategy and operations, as well as in larger organizations. One of the most popular frameworks for large-scale agile implementation, particularly in the automotive industry, is the Scaled Agile Framework (SAFe) [37]. SAFe structures teams into larger units known as agile release trains, which deliver their work on a regular basis to provide value to the end user [38]. Additionally, SAFe introduces a requirements information model that consolidates multiple user stories into an epic, representing medium- to long-term goals for team groups. This model also incorporates constraints, including quality (non-functional) requirements [38].

In AV development, the complexity of coordinating various subsystems (e.g., perception, decision-making, and control) underlines the importance of scaling agile effectively to ensure that the overall system meets stringent safety standards. Studies suggest that while frameworks like SAFe can provide a structured approach to handle such integration, challenges such as aligning development efforts across teams and managing interdependencies remain key concerns in large-scale projects [39, 40].

1.2.4 Requirements Engineering in Agile

Requirements Engineering (RE) in agile development, often referred to as *Agile RE* or *RE for agile*, can be broadly defined as an agile approach to performing RE, although there is no universally accepted definition [41].

Agile development can address some traditional RE challenges, such as communication gaps, but it also introduce new issues. These include the neglect of non-functional requirements, limited client availability, knowledge-sharing problems, insufficient documentation, and a lack of shared understanding of customer values [36, 42, 43].

As agile development become more prevalent, especially in domains that require safety, security, and regulatory compliance, the integration of RE into agile has become a critical area of research. Traditional RE methods, characterized by detailed upfront planning, can seem incompatible with the emphasis in agile development on adaptability and rapid iterations. However, as projects grow in complexity, structured approaches for capturing and managing requirements become essential [44].

Early attempts to address these challenges, such as the works by Inayat et al. [45] and Paetsch et al. [44], propose combining traditional RE with agile practices. These efforts led to the development of several strategies for managing both functional and non-functional requirements within agile projects. Lightweight documentation, modularization of requirements, and incremental refinement throughout the development cycle have emerged as key practices [44, 45, 46]. User stories, for example, are widely used to provide a flexible, adaptable way to capture requirements without burdening the process with extensive documentation [47]. Another approach is continuous experimentation, where requirements are iteratively refined based on real-time feedback from users, allowing agile teams to adapt to changing needs quickly [48].

By focusing on these approaches, agile teams can better manage requirements, including non-functional and human factors requirements, without overwhelming the development process with heavy documentation. As agile development continue to evolve, more studies are needed to address the ongoing challenges related to RE in agile.

1.2.5 Automated Vehicles

The development of automotive systems has historically been characterized by long lead times and sequential, plan-driven engineering methods [49]. However, the industry is now shifting towards more adaptive, continuous, and value-driven approaches, such as Agile development [28, 50]. According to Gren and Lenberg, the primary driver behind this shift is the need for flexibility in response to changing requirements, particularly in complex systems like automated vehicles (AVs) [51].

Despite the benefits of Agile methods, automotive developers face significant challenges in adopting these techniques, especially when it comes to designing, documenting, and managing the increasingly complex system requirements of AVs [37, 52]. This complexity arises not only from the technical complexities of AVs but also from the need to account for how humans will interact with and trust these automated systems. Given that AVs fundamentally change the role of the driver—from active controller to passive monitor at higher automation levels—developers must consider the cognitive, physical, and emotional aspects of human interaction with these systems.

The Society of Automotive Engineers (SAE) has established a six-level classification scheme (Levels 0-5) to define different degrees of vehicle automation (Figure 1.3) [53]. At Level 0, there is no automation; the vehicle may include features like automated emergency braking (AEB) or blind-spot warnings, but the human driver is fully responsible for operating the vehicle. Levels 1 and 2 introduce limited automation, such as steering or speed control, though the driver remains the primary decision-maker.

As automation progresses to Level 3, the human driver is no longer in full control, as the vehicle can handle most dynamic driving tasks within predefined Operational Design Domains (ODDs). However, the system may still require human intervention when it encounters situations beyond its design parameters. At Level 4, vehicles can manage driving tasks autonomously within specific ODDs, with no need for human input. Finally, Level 5 represents full automation, where the vehicle can operate independently in all conditions without human oversight.

As vehicles progress toward higher levels of automation, human factors become essential. Human factors are essential for ensuring AV systems are designed to accommodate the changing role of the driver, from active controller to passive monitor. This transition, along with issues like cognitive load and trust in automation, must be carefully addressed to prevent reduced safety and usability.

1.2.6 Human Factors

The *Journal of the Human Factors and Ergonomics Society* defines human factors as the "scientific study of human capabilities and limitations, encompassing cognitive, physical, behavioral, physiological, social, developmental, affective, and motivational aspects of human performance. This knowledge is used to inform design principles, enhance training, and improve selection and communication" [11]. Similarly, the same journal describes human factors as "concerned with the application of what we know about people, their

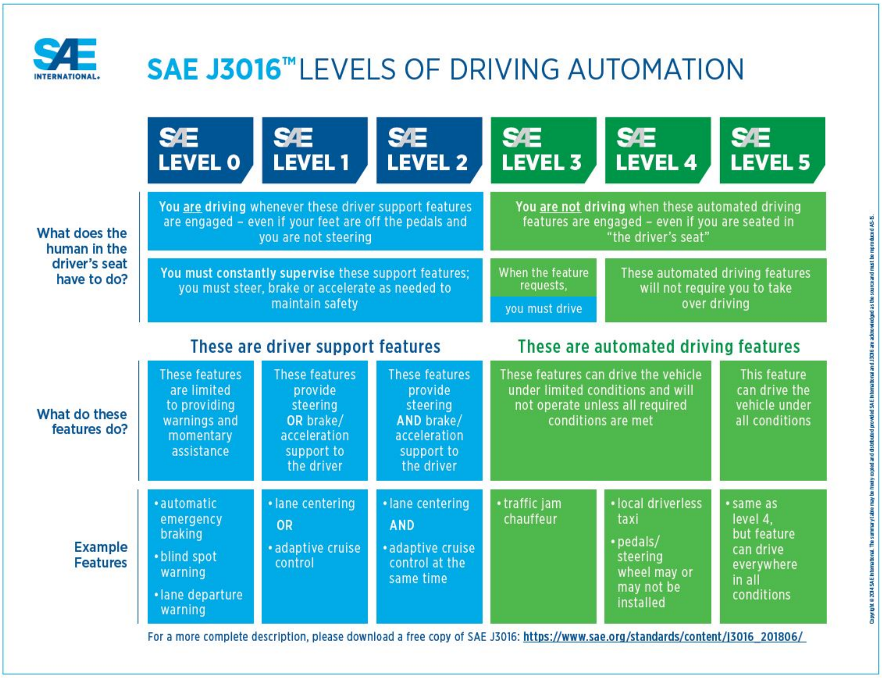


Figure 1.3: Description of levels of driving automation by SAE [53]

abilities, characteristics, and limitations to the design of equipment they use, environments in which they function, and jobs they perform."

The Health and Safety Executive (HSE) adds another perspective, defining human factors as "organizational, environmental, and job-related, as well as individual characteristics that affect the work environment and the quality of work" [54]. These varied definitions illustrate the complexity of the term and highlight how different professions interpret human factors based on their specific contexts. Consequently, this diversity can lead to challenges in communication, particularly when professionals from different fields use definitions that emphasize distinct aspects of human factors [11, 55].

Human factors play a critical role in system development, especially in safety-critical industries such as aviation, healthcare, and automotive design. Poorly designed systems can result in user errors, misuse, or even accidents [56, 57]. In this context, human factors aim to ensure systems are intuitive, safe, and meet user needs by considering elements such as cognitive load, situational awareness, and usability [58, 59, 60]. In autonomous vehicle (AV) design, these considerations are vital for shaping user interactions, fostering trust in automation, and ensuring effective communication of system status.

Given this multidisciplinary landscape, it is essential to adopt a definition of human factors that aligns with the specific goals and challenges of AV development. A unified understanding is particularly important in collaborative environments where requirements engineers, human factors specialists, and other engineers work together [59, 61]. To address this need, the following definition has been formulated for the purpose of this thesis:

> **Definition:** The field of *Human Factors in AV Development* aims to inform AV development by providing fundamental knowledge about human capabilities and limitations throughout the life cycle so the product will meet specific quality objectives.

This definition serves as a foundation for exploring how human factors inform AV design and development. Despite the variations in perspectives and interpretations, human factors consistently represent key values that enhance system performance and user experience. By emphasizing these principles, this thesis aims to contribute to the development of AV systems that are safe, intuitive, and effective for their intended users.

1.2.7 Human Factors and Automated Vehicles

Human factors are generally considered similar to soft factors, which describe characteristics that are not specific to technical skill sets but reflect non-technical and soft skills. Human factors need to be considered in software development—or any other work, such as AV development, where humans are involved.

The scientific study of human factors plays a crucial role in both the software development and hardware design of AVs. Software aspects related to human factors include how the vehicle maintains its lane position [62, 63], how it communicates with external road users [64, 65], and how software-based human-machine interfaces (HMIs) display information to the driver [66]. Moreover, it covers the broader communication between humans and AVs [64]. On the hardware side, human factors address aspects like seating ergonomics, which influence how AV capabilities can reshape automotive interiors [67], as well as the physical design and placement of HMIs within the vehicle.

As AV technology continues to evolve, the focus of human factors research has shifted towards the interaction between humans and AV systems. With the transition from active driving to monitoring, new challenges have emerged around driver engagement, trust in automation, and overall safety [60, 68]. For instance, studies have shown that drivers often over-trust AV systems, leading to slower reaction times when they are required to take control, which in turn increases the risk of accidents [69].

These examples show the extensive knowledge of human factors necessary for effective AV engineering. Several researchers have also emphasized the importance of integrating human factors into AV development to ensure safety and usability [10, 13, 57]. However, human factors research has not kept pace with the rapid development of AV technologies, and it remains uncertain how well engineers are incorporating these considerations into their design decisions. Consequently, finding effective strategies to incorporate human factors into AV development is critical to maintaining progress. Early studies suggest that human factors should be addressed in the initial stages of development [17, 70], but challenges remain, particularly in adopting an agile methodology that efficiently integrates this knowledge.

Incorporating human factors into agile development presents a unique challenge. Agile development, with its emphasis on rapid iteration and flexibility, has become widely adopted in software engineering. Sohaib and Khan [71] argue that continuous user feedback loops in agile development help align systems with user needs, while Ferreira et al. [72] demonstrate that usability can be integrated into agile teams without compromising speed or flexibility.

While many principles of human factors integration in agile development could be broadly applicable across domains, AV development presents distinct challenges, particularly for partially or conditionally automated vehicles. Unlike conventional systems, AVs require complex interactions between automated decision-making processes, artificial intelligence, and human operators. These complexities intensify the importance of addressing human factors such as trust, acceptance, user understanding, and driver-vehicle hand-overs (e.g., due to system limitations). For example, the unpredictability of urban environments places high demands on HMIs to provide timely, intuitive feedback to ensure users can respond effectively during handovers or failures. These unique challenges call for approaches tailored to integrating human factors into agile AV development.

In summary, while individual research areas offer valuable contributions, the intersection of these domains (human factors, agile development, RE, and automated vehicle development) remains underexplored. Our work seeks to bridge this gap by developing strategies to embed both human factors and technical aspects into the agile AV development, ensuring that user needs are continuously met without compromising the agility and speed of development.

1.3 Research Approach

This research investigates multidisciplinary areas, including human factors, RE, and agile development in the context of AV development. Each area is explored in the specific context of incorporating human factors knowledge into the design of automotive systems or software, particularly within large-scale agile development.

To achieve this, various empirical research methods were employed, which are essential for understanding the practical, real-world challenges of integrating human factors knowledge into RE practices in agile AV development.

To achieve this, we employed various empirical research methods, which are crucial for understanding the practical, real-world challenges of integrating human factors knowledge into RE practices in agile AV development. Empirical methods are particularly suited to this research, as they allow for an in-depth exploration of interdisciplinary and industry-relevant topics [73]. This applied approach ensures that our findings are both academically insightful and practically applicable to the industry.

The studies included in this thesis expand our understanding of current practices and help develop potential solutions for integrating human factors knowledge in agile AV development. The research began by understanding the interplay of agile development, RE, and human factors in AV development, identifying industry challenges and needs to better integrate human factors knowledge in agile AV development, and proposing strategies to address them. Papers A and B lay the foundation for the subsequent studies (Papers C-E), which focus on defining and refining potential solutions to these challenges.

Overall, these studies provide an in-depth understanding of the problems, propose concrete solutions, and align with the high-level goal of this thesis: to investigate how human factors requirements can be effectively integrated into agile AV development.

Empirical studies can employ both qualitative and quantitative methods [74]. In our research, we primarily relied on qualitative research methodologies to support our exploratory research goals, however in the last study we also used quantitative methods. Table 1.2 provides an overview of the research methodologies used across the five included papers, along with the data collection methods and references to the papers that contributed to this thesis.

1.3.1 Research Focus

To lay the foundation for our investigation, Paper A presents the properties and implications of human factors, agile ways of working, and RE in the context of AV development. It uses thematic analysis in a qualitative exploratory study, combining interview studies and a validation survey. The paper concludes that existing methods are ineffective in bringing human factors knowledge to AV developers in agile environments.

Paper B explores integration of human factors experimentation into continuous experimentation, highlighting the challenges of integrating human factors experiments within continuous experimentation and providing best practices for better management. Like Paper A, it identifies a lack of effective tools and human factors experts. This paper also uses interviews for the data collection. The findings from Papers A and B motivated further research to identify strategies for better integrating human factors requirements in agile AV development.

Paper C investigates how the challenges related to requirements in agile development can be addressed, providing initial solutions and introducing the concept of a *requirements strategy*. This strategy is presented as a comprehensive approach to resolving issues related to requirements in agile development. This strategy includes three building blocks: structuring requirements, organizing work, and integrating RE into agile development. Using a design science research approach and a mixed-methods data collection, this work provides initial solutions and sets the stage for integrating human factors requirements in agile AV development.

Paper D builds on the *requirements strategy* to find concrete solutions for integrating human factors requirements into RE in agile AV development. It identifies solution spaces for practices to better integrate and manage human factors requirements, using interview studies analyzed with an a priori coding method.

In Papers A and B, a lack of human factors expertise within organizations is identified, highlighting the impracticality of having experts in every team due to resource constraints. This motivated us to investigate the strategic placement options for human factors experts to maximize their impact on the product. Paper E explores strategic placement options for human factors experts to maximize their impact, considering the impracticality of having experts in every team. Using a mixed-methods approach, incorporating both qualitative and quantitative data analysis, this paper identifies preferred placement options for human factors experts.

Table 1.2: Included papers with their research methods

Paper	Research Method	Data Source
A	Qualitative exploratory study	12 interviewees & 28 survey respondents
B	Qualitative exploratory study	8 interviewees
C	Design science study	20 interviewees, document analysis, observation & 2 workshops
D	Qualitative research design, utilizing a priori coding	13 interviewees
E	Mixed methods approach	Shadowing technique, document analysis, informal interviews, workshop & 31 survey participants

1.3.2 Research Method

This section discusses several potential research methods and explains the specific subset chosen for the empirical studies. The five empirical studies employed different research approaches, including qualitative exploratory research, design science research, mixed-methods research, and a priori coding research. Table 1.2 shows the research methodologies applied across the five included papers and references the papers used in compiling this thesis.

1.3.2.1 Qualitative Research

Qualitative research makes it possible to look into and understand the meaning that people or groups make of a social or human problem. This research usually involves emergent questions and processes, data collection, data analysis proceeding inductively from particular to general themes, and data interpretation. In contrast to quantitative research that relies on large amounts of data to establish statistic significance, qualitative research supports a way of looking at research that emphasizes individual meaning, an inductive approach, and the need to depict the complexity of a situation accurately [75]. It represents an inductive approach based on interviews, observations, studying relevant working documents, and other relevant data to understand the subject of study and the context.

Qualitative research can serve different purposes, such as being exploratory or explanatory. Exploratory research is primarily concerned with investigating new areas, identifying patterns, and proposing potential hypotheses for future, more in-depth studies. This type of research is particularly valuable when the researcher is uncertain about which factors are key or when existing theories do not apply to a novel or complex domain. Exploratory research often employs emergent coding, a process where codes and themes are developed inductively from the data, allowing patterns to naturally emerge. This approach was applied in Papers A and B, which are exploratory in nature and employed qualitative methods based on emergent coding [76], providing insights into the role of human factors in agile AV development.

In contrast, explanatory research focuses on understanding the underlying causes and mechanisms of a phenomenon. It often examines established variables and relationships, aiming to explain why something happens. Explanatory research commonly uses a priori coding, where predefined codes based on existing theories or literature guide the analysis. This approach was applied in Paper D to systematically investigate concrete solutions for integrating human factors requirements into agile AV development. By employing a structured coding scheme, data from interview studies were categorized and interpreted, yielding clear and actionable insights. These insights guided the identification of solutions for better managing human factors requirements in agile development.

1.3.2.2 Design Science Research

Design science research aims to develop an artifact—such as a (software) tool, guidelines, or templates—through multiple cycles of creation and evaluation. According to Wieringa [77], this research strategy combines the study of one or more epistemological questions to solve a design problem. Aspects of the design problem include real-world change, context awareness, and proposals for one or more solutions. Design science studies are always concerned with the environment (consisting of individuals, organizations, and technology), and the output advances knowledge in one or more research fields [78].

Knauss suggests that design science should be iterative and touch on three aspects in each cycle: the problem, the solution design, and the evaluation of the extent to which the solution addresses the problem [79]. He also suggests that while each cycle should touch on all three aspects, early cycles can focus more on investigating the problem, middle cycles could focus more on developing the solution and artifact, and the final cycles could focus more on evaluating the solution against the problem.

As a result of these suggestions, the design science research method was adopted in Paper C. The research aims to develop appropriate methods for creating requirements strategies (the design artifact) for organizations using large-scale agile development. These requirements strategies should address real-world needs, incorporate state-of-the-art information, and undergo empirical evaluation in real-world settings Hevner et al., [78] state that it is crucial that the underlying issue be relevant and that the solution be thoroughly assessed. Because of this, we examined the existing solutions closely. To our knowledge, no other studies related to our design artifact ("requirements strategy") are available.

Commonly, a mix of methods is used in each phase of design science research to develop a design science artifact. Our study relied on interviews, workshops, document analysis, and observations.

1.3.2.3 Mixed-methods Research

Mixed-methods research combines the potential of multiple research methods [80], offering a more complete picture of the phenomenon under study. Methods can be combined in sequential or concurrent designs and can have an exploratory or explanatory focus. While this approach provides a comprehensive understanding, it poses challenges in extensively collecting, analyzing, and triangulating the required data [73, 81].

A sequential mixed-methods design was employed in Paper E, incorporating shadowing, document analysis, discussions, workshops, and surveys. Thematic analysis was applied to the qualitative data, while Bayesian analysis was used for the quantitative survey data.

1.3.3 Data Collection Methods

We employed a range of data collection methods across five empirical studies, focusing on companies within the automotive industry. Data for this thesis was gathered through interviews, surveys, document analysis, observations, workshops, shadowing, and discussions. These methods were selected to address specific research questions and support our overall research strategy. The selection of data collection methods was guided by the specific needs of the diverse research studies. For instance, interviews were essential for exploratory studies to gain in-depth insights, while document analysis provided crucial context and a deeper understanding of existing practices. By combining these diverse approaches, we were able to thoroughly explore the challenges and solutions for integrating human factors knowledge in agile AV development.

The studies involved professionals from various disciplines in the development process, including human factors, requirements engineering, and software development. Participants (also) represented a variation of actors in the automotive value chain, including original equipment manufacturers (OEMs), suppliers, and academic institutions. Given the multidisciplinary expertise required and the exploratory nature of certain studies, random sampling was not appropriate. Instead, we used convenience sampling, purposeful sampling, and snowball sampling.

The participant pool included professionals from leading companies such as Volvo Cars, Volvo Trucks, Mercedes-Benz, and Microsoft, as well as academic experts with substantial experience in automotive and human factors domains. While we aimed for theoretical saturation to guide the sample size in our studies, the multidisciplinary scope of this thesis presented limitations in terms of the availability of experts with the required expertise. As a result, demonstrating complete saturation was challenging. However, the analysis of our results indicated that later interviews often echoed sentiments expressed in earlier ones, demonstrating data saturation, as discussed in Paper D.

Table 1.2 outlines the research methodologies and data collection methods used in the five papers included in this thesis. The following sections provide an overview of the key methods used.

1.3.3.1 Interviews

Interviews are one of the most commonly used methods in qualitative research. There are three primary types of interviews: structured, unstructured, and semi-structured [82]. Structured interviews involve asking a set list of predetermined questions, with minimal variation between interviews and no allowance for follow-up questions based on responses. In contrast, unstructured interviews do not follow any preconceived theories or set questions, allowing for a more open-ended conversation. Semi-structured interviews combine elements of both, with some predetermined questions asked of all participants while also allowing spontaneous questions to emerge during the discussion. According to Smith [83], this approach facilitates a dialogue where the interviewer can adapt scripted questions based on responses and explore new topics as they arise.

The research aimed to maintain flexibility in interviews while providing enough structure to ensure replicability. This approach allowed for the exploration of new areas as they emerged while adhering to a framework that supported consistent and reproducible findings. To achieve this balance, semi-structured interviews were used in all studies.

Qualitative exploratory interviews were conducted to capture the personal opinions of experts in the field in all studies involving interviewees. These open-ended, semi-structured interviews provided flexibility while maintaining focus.

Paper E took a different approach by utilizing unstructured (conversational) interviews to maximize flexibility and adapt to the specific circumstances of each interaction. These unstructured interviews were referred to as informal interviews in Paper E. Informal interviews with industry experts and team members provided additional context and helped clarify findings from other data collection methods. Unlike formal interviews, these informal conversations were more flexible, allowing for spontaneous dialogue and the exchange of insights.

1.3.3.2 Questionnaire Survey

Survey research is usually a quantitative method in which a researcher presents a set of predetermined questions to a sample of the population. This approach is particularly valuable for describing the characteristics of a large group [84].

In this research, questionnaires were used in Papers A and E, enabling access to a larger sample compared to interviews. Both papers primarily utilized forced-choice questions with Likert scale response options [85], where answers were rated on an ordinal scale (e.g., from 'strongly agree' to 'strongly disagree'). Paper E also included one general ranking question. Furthermore, Paper E provided an open-ended option for all questions, allowing respondents to add additional information, feedback, or comments if they wished.

1.3.3.3 Document Analysis

Document analysis involves the systematic examination of existing documentation such as project reports, process guidelines, and technical specifications. This method helps to triangulate data from other sources, adding context and depth to the research findings. Document analysis was used in Papers C and E, providing additional layers of insight that complemented the other data sources in these papers [86].

1.3.3.4 Workshops

Workshops facilitated collaborative discussions and brainstorming sessions with various stakeholders [87]. These interactive sessions supported the development of solutions and strategies, especially in Papers C and E. Workshops enabled us to gather diverse perspectives and encourage a collective approach to problem-solving, enhancing the practical relevance of our research findings.

1.3.3.5 Observations

Observations were conducted to gain firsthand insights into the actual practices and interactions within agile AV development teams. This method allowed us to observe meetings, development sessions, and user interactions without actively following a specific individual throughout their entire workday. Observing meetings, development sessions, and user interactions provided practical insights into requirements management within these teams. This method was prominently used in Papers C and E to capture real-time behaviors and processes [88].

1.3.3.6 Shadowing Technique

The shadowing technique, in contrast to general observations, involved closely following and observing a specific participant throughout their workday. This method provided deep insights into the daily challenges and practices. Shadowing was particularly effective in Paper E, where it allowed for an immersive understanding of participant activities and interactions [89].

1.3.4 Ethical Considerations

According to Swedish law and the General Data Protection Regulation (GDPR), only human subject studies that collect the following data are considered high-risk (sensitive personal data) and in need of an explicit ethics review [90]:

- Personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs
- Trade-union membership
- Genetic data, biometric data processed solely to identify a human being
- Health-related data
- Data concerning a person's sex life or sexual orientation

As this research did not involve collecting any such high-risk data, explicit ethics review was not found necessary for similar studies under the Swedish law. Since June, 2024, Chalmers has an institutional ethics advisory board that could provide as statement on whether a study likely falls under Swedish law with respect to ethics, to which we should have sent our studies, if they were not published earlier. However, to ensure ethical rigor, the following practices were implemented throughout the studies included in this thesis:

- Participants were provided with information outlining the purpose and scope of the study before there participation in our studies.
- Participation was entirely voluntary, and full, informed consent was obtained from all participants before data collection began.

- Collected data was managed securely and responsibly. Raw data was not shared publicly and was stored on personal computers accessible only to the researchers involved in the study. Efforts were made to anonymize participants and their affiliations in all reporting to safeguard privacy.
- The anonymity of participants was prioritized in all studies, and care was taken to ensure that sensitive information was not disclosed at any stage of the research.
- To ensure accuracy and ethical integrity, transcripts were shared with participants for review and verification in many cases. Furthermore, manuscripts were reviewed with participants to confirm that no sensitive or identifying information was inadvertently revealed.

1.4 Threats to Validity

Ensuring research validity is a cornerstone of high-quality research. However, in contrast to quantitative studies, qualitative studies have no universally accepted framework for assessing validity [91]. Generally, the primary concern when assessing validity is to make sure the research correctly reflects reality. However, qualitative studies do not tend to describe reality directly, as they are based on perspectives, observations, and understanding. Moreover, qualitative researchers cannot rely on pre-planned comparisons, strategies, or statistical analyses to improve validity, as is typically done in quantitative studies [92].

While it is generally challenging to validate the results of qualitative studies, there are a few methodologies that can be used for the mitigation of threats to validity [93]. The four perspectives of validity threats, as outlined by Runeson and Höst [94] and Easterbrook et al. [73], are considered in this research.

1.4.1 Construct Validity

Construct validity relates to how well the operational measures align with the primary concerns of the researchers and the data collection methods employed, including interviews, observations, workshops, and surveys [94]. In exploring the integration of human factors knowledge in agile AV development, threats to construct validity can arise from using concepts with different interpretations. For instance, during interviews, terms like “human factors” and “agile” may be misunderstood. When multiple domains and disciplines are involved, achieving a shared understanding of terms can be challenging. To address this, we introduced relevant terms at the beginning of each data collection and established a common understanding of terms at the start of interviews, workshops, and surveys.

Moreover, we relied on the complementary knowledge and experience of the co-authors, who have worked closely with industry experts within the relevant domains. In addition, the interview guides were improved in several iterations. For example, in Paper A, in the first version of the interview guide, there were many related questions that were difficult to cover in a short time period, and some questions were difficult for the interviewees to understand. Since all three authors participated in most interviews, none was conducted by a single author.

This collaborative approach allowed us to resolve difficulties during interviews and revise the guide to ensure consistent interpretation of the questions by all interviewees. Similarly, the interview guides and data were refined over multiple iterations in the other studies.

1.4.2 Internal Validity

Internal validity considers the design of a study and whether the findings are derived from the collected data and investigates if external factors could impact the findings [73].

To minimize this threat, we carefully collected data about the topics and their contexts and provided detailed descriptions of our findings [73]. For studies relying on interviews, we ensured the accuracy of interview transcripts through member checking or audio recording. Analyzing the data over multiple iterations and reporting these iterations in the paper also helped reduce internal reliability threats. Additionally, follow-up questions in interviews and discussions enhanced our understanding of participants' explanations. In surveys, although limited to predefined questions, we included comment fields to capture additional aspects.

Collecting data from multiple sources (with a variety of roles in different contexts) facilitated the triangulation of the findings. We carefully selected participants by first understanding the companies they represented, ensuring a diverse range of roles to avoid overly narrow findings. However, since interviewees were selected through industry contacts, there might still be selection bias. To mitigate this, we discussed our results in workshops (Papers C and E) and validated them through a survey (Paper A) to gather more opinions on our findings. Our design science study (Paper C) and mixed methods study (Paper E) employed several methods to gather rich data from various sources, further validating our findings.

1.4.3 External Validity

External validity focuses on how well our findings can be generalized and the extent to which they can be applied to other companies, individuals, and situations beyond those examined in this thesis. Our research aims to understand the phenomena under study and to represent specific contexts rather than to establish statistical generalizations. Easterbrook [73] states that the purpose of qualitative research is to understand and explain a particular phenomenon rather than to generalize. However, understanding the researched phenomenon in one context may facilitate understanding in other contexts.

External validity is higher in some studies because they involve a broader range of companies and individuals. For example, in Paper C, we identified the building blocks for defining the requirements strategy in multiple iterations using interviews and two workshops at three case companies. We found common perspectives on solution strategies in each case company. Given that we found the same building blocks for each company, we expect them to be applicable to other companies or large-scale agile development projects in related domains. This generalizability ensures reduced threats to external validity. Still, further validation in other domains is the subject of future research.

Moreover, we thoroughly validated our findings in Paper A with several participants from different companies. We also included participants from different domains in Paper B, increasing external validity through triangulation. In Paper E, we used a mixed-methods approach, including surveys sent to a broader audience globally, to enhance the generalizability of our findings across different contexts. Paper D, which focused on integrating human factors requirements into agile AV development, included 13 individuals from 11 different companies across multiple countries.

To enhance transferability, we provided detailed descriptions of the characteristics and contextual factors of our participating companies and individuals. These descriptions help readers determine if the findings might be applicable to other cases. However, some studies, such as Papers A, D, and E, were conducted with experts only from the automotive industry. This allowed us to derive specific insights that are directly relevant and applicable to the AVs. We anticipate that fields sharing similar characteristics with AV development may also benefit from our findings. Nevertheless, further research is necessary to confirm the applicability of these results to other areas of application than the automotive industry.

1.4.4 Reliability

Reliability refers to the degree to which other researchers would arrive at the same conclusions if they replicated the study under the same conditions and methods [73]. In qualitative research, researcher biases and reactions can threaten the reliability of the results due to interactions with participants or interviewers.

To mitigate these threats, we involved multiple researchers in our studies. For instance, during interviews, more than one researcher was present. We also carefully documented our methods to make them as replicable as possible. To enhance the replicability of our studies, we made our tools available, including interview guides, survey questions, and documentation of our analysis methods. We aimed for high transparency and consistency in our evidence chain. By using quotes and specifying participant roles, we increased the transparency of our research findings. Member checking allowed us to ensure we correctly understood participants' statements and validated our findings. Peer debriefing further improved reliability, ensuring our methods were less reliant on individual researchers.

Whenever possible, we also shared the raw data used in our analyses. However, due to non-disclosure agreements, we couldn't publish the raw transcripts of interviews. For Paper E, we provided the data and replication package for statistical analysis, allowing others to replicate our study.

Detailed descriptions of our analysis processes in the papers allow other researchers to understand and replicate the findings, even if their results differ, by identifying potential reasons for discrepancies. While we provided detailed descriptions of our research methods, the qualitative data analysis through coding remains researcher-dependent.

Table 1.3: Overview of papers with research contributions

Paper	Paper Title	Contributions/ main findings	Thesis RQs
A	Human factors in developing automated vehicles: A requirements engineering perspective	Establishes the context by highlighting the importance and challenges of integrating human factors knowledge into agile development of AVs. Identifies the properties and implications for integrating human factors into agile development.	RQ1
B	Continuous Experimentation and Human Factors: An Exploratory Study	Explores the integration of human factors considerations within continuous experimentation in the context of agile development. Investigates practical challenges associated with managing human factors experiments in continuous software experiments.	RQ2
C	Defining Requirements Strategies in Agile: A Design Science Research	Investigates critical challenges in agile RE and proposes strategies for managing requirements in agile contexts. Provides guidelines for defining requirements strategies.	RQ1 and RQ3
D	Requirements Strategy for Managing Human Factors in Automated Vehicle Development	Focuses specifically on managing human factors requirements. Examines ownership, structure, and workflow aspects of human factors requirements and provides insights from industry professionals.	RQ3
E	Integrating Human Factors Expertise into Development of Automated Vehicles	Explores the optimal placement of human factors expertise within organizations and AV development teams to maximize its impact.	RQ4

1.5 Summaries of Studies

This section briefly outlines the five papers on which this thesis is built. Full papers can be found in Chapters 2-6. Table 1.3 provides an overview of the papers and related research questions with each paper's main contribution to each research question.

1.5.1 Paper A

Automated vehicles are growing in number, but still require human interaction and involvement. This study is motivated by the need to learn how to capture human factors knowledge as requirements. We aim to (i) understand the term human factors, (ii) explore the properties of human factors and agile, and (iii) provide implications for human factors, the agile way of working, and RE. The study's research questions are:
RQA-1: How do human factors experts and AV engineers characterize human factors in relation to AV development?
RQA-2: Which properties of human factors and agile ways of working impact AV development?
RQA-3: What are important implications when aiming to better integrate human factors into AV development?

To operationalize the goal, we conducted a qualitative exploratory study. We interviewed ten industry experts, including both AV developers and human factor experts, and two international academic leaders in human factors research. All of the interviewees were experts and had years of experience. Industry experts were from different Swedish companies, including Volvo, Veoneer, Zenuity, and Autoliv. We relied on semi-structured interviews with a predefined interview guide to collect qualitative data. Semi-structured interviews allowed us to adjust the questions and ask follow-up questions to satisfy the emergent information needs.

We began by looking at the definition of human factors. Several definitions of human factors are available [11, 55], even on the homepages of significant journals in the field, depending on the specific research context (e.g., [11]). Clearly, communicating requirements and knowledge could be challenging when people use different definitions [95].

Even when the definitions seem straightforward, different people may have different opinions about what human factors involves [55], which may influence how they interpret human factors in their line of work. Therefore, it is crucial to look into how people actually feel about human factors in the workplace, especially when researching the role of human factors in the development of automated cars (as in the current study), since a variety of different engineers are involved in addition to human factors experts [96].

Because it is critical to have a shared understanding of the core concepts in order to investigate the systematic capture and management of human elements in AV development, it is necessary to create a definition specific to this context (here, AV design). Thus, RQA-1.2 aimed to synthesize different interpretations from practitioners' perspectives into a definition of human factors in AV development.

In our study, we expanded one of the already existing definitions [11], making it more precise about the relationship between human factors and AV. Based on our data and literature, we define the term 'human factors' in relation to AV development as described below.

> **Definition:** The field of *Human Factors in AV Development* aims to inform AV development by providing fundamental knowledge about human capabilities and limitations throughout the life cycle so the product will meet specific quality objectives.

This definition is derived from existing definitions of human factors (derived from [11]); however, our main contribution is adding the *design cycle* part of the definition. It is essential to discuss the relationship of human factors to AV development throughout the design cycle particularly for automation. Human factors must have an impact on the design cycle and in a way it is more suitable for software engineering. Hence, we added this part in the definition.

Then, we defined the properties of human factors and agile development in relation to AV development. Our result indicates that agile promotes iterative, incremental work to help organizations deliver fast and increase responsiveness towards changing requirements. It advocates accountability by shifting responsibility from planning managers at the system level to autonomous teams that can make their own local decisions. Moreover, these autonomous teams often dislike static, detailed requirements. Instead, agile teams prefer being responsible for discovering knowledge, relying on face-to-face communication just-in-time by themselves rather than on extensive documentation.

Although agile approaches suggest that requirements rapidly change and may not describe the users' real needs at the time when the product is finished, they still focus on quality in use. Human factor experts also focus on quality in use, but they are concerned with human interactions with the system; it should be safe to use, pleasurable, and so on.

Human factors properties reflect on the importance of including human factors knowledge while performing experiments and testing the system. In agile development, iterative work demands continuous testing to avoid regression problems and to address changing requirements. Human factors experts aim to run experiments on the system with human subjects (e.g., how humans react in certain situations and how they get distracted) while considering human variability. It is important to consider human variability to improve performance and make it usable for a diverse set of customers. Depending on their background, humans have different capabilities and limitations. Human factors experts play an important role in ensuring that the developed systems are suitable for all humans, e.g., with different characteristics, ages, cultures, and visual/cognitive capabilities. For AV, users must have enough situational awareness (e.g., decision-making capability) to respond correctly, avoiding the system's misuse/disuse. However, not all users read the manual or attend training, so they may not be aware of a system's capabilities and limitations. Therefore, human factors experts prioritize ensuring that HMIs are transparent and self-explanatory for all kinds of users.

Implications of the agile way of working highlight the need to adjust it with human factors. As we know, agile AV developers perform iterative experiments with their teams. Even as experimental designs are created and lessons are learned, subsequent experiments risk overwriting the knowledge acquired. We might have these experiments with different quality, so we need to find a way to manage this knowledge effectively. In the case of human factors knowledge in agile development, the appropriate experts must be included in the development teams. Given the lack of human factors expertise, we need to identify a strategy for agile AV development that considers human factors. As the automotive value chain is transforming to agile ways of working as well as continuous integration and delivery, new collaboration models with suppliers are emerging that are integrated into incremental work for specific purposes [97]. Our final implication for agile is, therefore, to systematically decide whether and how to include a supplier in the scaled-agile development of AVs.

Human factors implications imply that human factors experts should be part of agile teams to raise awareness, enable relevant questions to be asked (regarding human behavior and capabilities), and guide teams. Human factors experts should also provide basic human factors knowledge as checklists and design principles [98] for development teams.

We believe that RE can support this effort effectively by managing the acquired knowledge from experiments and by expressing design decisions as they relate to human factors requirements in the backlog. A second implication for RE is to increase the capability for prototyping for requirement elicitation and validation within agile teams, based on the identified needs and human factors checklists. The third, and last, implication is to express the relationship between design decisions and human factors knowledge (e.g., via tracelinks), which means that system requirements must be created at the same time as the system/software—not before. Thus requirements would be provided (in the form of stories) during development rather than at the beginning of development.

The study's findings were validated in a workshop with academic and industrial professionals. We anticipate that these findings will help to improve the integration of human factors expertise into agile development and increase the impact of human factors research.

1.5.2 Paper B

Experimentation is a crucial aspect of human factors, which can aid in integrating human factors into agile AV development. In Paper B, we investigate how human factors knowledge can be incorporated into continuous software experimentation within agile AV development. Our focus is on determining whether it is easier to integrate human factors experimentation into continuous software experimentation. We approach this by uncovering the distinctive characteristics of human factors experiments and continuous software experiments, identifying practical challenges, and suggesting best practices for effective continuous human factors experimentation.

Concretely, we focus on the following research questions:

RQB-1: What are [the] main differences when comparing human factors experiments with continuous software experimentation?

RQB-2: What are [the] main practical challenges when managing human factors experiments in continuous software experimentation?

RQB-3: What are [the] best practices for managing human factors in continuous experimentation?

To collect data, we conducted interviews with eight professionals experienced in human factors and continuous experimentation, including experts from renowned organizations like Microsoft and highly cited academic professionals (e.g., h-index > 35 in four cases).

This study reveals significant differences between human factors experiments and continuous software experimentation (RQB-1). Human factors experiments focus on understanding user behavior and experiences through qualitative data, while software experiments often emphasize technical performance with quantitative metrics. Despite these differences, both aim to improve product quality and user satisfaction. However, methodological differences make integrating human factors experimentation into continuous software experimentation challenging.

Moreover we also identified the challenges for integration human factors experiments in continuous experimentation. Results show that conducting human factors experiments within agile AV development faces several practical challenges. These include ensuring GDPR compliance to protect participant data privacy and gathering reliable. While GDPR compliance is crucial for both types of experimentation, it poses greater challenges for human factors experiments due to the need for insights into how users with diverse capabilities and limitations interact with technology. This often requires the collection of personal characteristics, such as cognitive or physical abilities, to tailor the system design effectively.

While both human factors experiments and continuous software experiments often occur in real environments with real users, human factors experiments tend to be more expensive and time-consuming. One reason for the higher costs is the need to recruit and compensate human participants, whereas software experiments can often rely on automated data collection or remote observation without significant participant involvement. Additionally, human factors experiments require capturing detailed data on user behavior, cognitive load, and physical interactions, often needing specialized equipment and longer observation periods. The extended execution time in human factors experiments is due to the complexity of analyzing human interactions, as well as the need for repeated observations across diverse participant groups to account for varying capabilities and limitations. In contrast, software experiments generally involve shorter, more controlled iterations and can be repeated with fewer resources.

Moreover, there is often inadequate infrastructure to support complex setups for human factors experimentation, and many companies struggle with insufficient human factors expertise.

To address these challenges and integrate human factors knowledge effectively, this paper suggests several best practices (RQB-3). For example, prioritizing research questions and aligning experiments with product development timelines is essential. Developing meaningful metrics beyond basic interaction data, for example, those based on diverse human capabilities among the user base, helps capture deeper user insights. Ensuring a strong experimental setup with thorough documentation aids in replicating and validating experiments. Facilitating collaboration between human factors experts and software developers promotes better integration of human factors considerations. Securing management support is crucial for allocating necessary resources and infrastructure for human factors experimentation. In summary, the study concludes that current tools and methods are insufficient for easily integrating human factors experiments into agile continuous experimentation, necessitating the development of new concepts to better integrate human factors knowledge.

1.5.3 Paper C

When agile methods are applied to systems development on a large scale, it is not entirely clear how to manage complex stakeholder landscapes, system requirements, and systems engineering disciplines. It is true that RE approaches are strong in these aspects, since they have traditionally played a crucial role in systems and software engineering. However, because these approaches are rigorous, time-consuming, and extensively documented, it is hard to integrate them into agile methods—they actually contradict the agile development approach.

This paper is motivated by that contradiction. The study identifies specific RE-related challenges and related solution strategies in agile development. Based on this knowledge, we derive different viewpoints that should be considered when thinking strategically about RE in agile development. Thus, Paper C aims to identify the necessary building blocks of requirements strategy and establish the concept of requirements strategy for agile development (RQ3). We argue that defining a requirements strategy for RE can be critical for (successful) large-scale agile development. Multiple factors influence how requirements strategy can be built when attempting to define a strategy to address these challenges. We call these factors as building blocks of requirements strategy. The research questions in this paper are as follows:

RQC-1: Which challenges arise from an undefined requirements strategy?

RQC-2: How do companies aim to address these challenges?

RQC-3: Which potential building blocks should be considered for defining a requirements strategy?

The research method for this study is based on design science research with three industrial cases. We derived the guidelines for the requirements strategy model (the design artifact) from 20 interviews, two workshops, participant observation in two cases, and document analysis in all three cases. The guidelines helped us understand work & feature flows and concrete challenges in agile development. Case 1 was a telecommunications company with very large-scale agile software development. The focus was on creating a strategy to achieve a shared understanding of customer value in a large-scale agile project. A key concern was the trade-off between the risk of sharing too much information and overloading developers on the one hand, and not sharing important information on the other hand. To balance this trade-off, we aimed to determine who needed to know what and how much to share with whom.

We followed up with Case 2, a company producing smart security alarms and services. In this case, the focus was on a more general requirements strategy that covers stakeholder and system requirements (e.g., how to document user stories and qualitative requirements). The aim was to refine our design artifact into guidelines for a requirements strategy. Case 3 was an automotive supplier focusing on safety-critical and software-intense systems. We utilized our experience from the previous two cases to investigate whether it was feasible to define a requirements strategy and what the value of such a strategy would be in terms of systematically supporting continuous improvement. Our focus was to refine the design artifact by discussing, applying, and improving our understanding of the building blocks of a requirements strategy.

For all three cases, we started by listing challenges; since we particularly targeted agile development, we aimed to investigate requirements challenges independent of process phases or specific documents. Instead, we used the concept of shared understanding as a lens [99]. A shared understanding may target how an understanding is initially enabled in an organization, how it is built, and how it is assessed. Then, we discuss those challenges with respect to potential mitigation strategies. Based on the identified challenges and solutions, we systematically developed building blocks for the requirements strategy. Through building three (quite different) strategies, we can see that the model captures relevant information and provides a useful overview. We found that the following perspectives each play a decisive role in describing the requirements strategy: (i) structural, (ii) organizational, and (iii) work and feature flow.

We suggest starting with a structural view, defining the requirements structure in order to create a shared language; then defining the organizational responsibilities and ownership of requirements knowledge; and finally mapping both structure and organizational responsibilities onto the agile workflow. In this context, workflow refers to the structured sequence of activities, tasks, or processes that enable teams to achieve specific goals within a development cycle. To design a requirements strategy to solve RE challenges in agile development from a structural view, we need to know what kinds of requirements we have, on what levels of abstraction, and whether we have templates for those requirements. For example, do we have high-level requirements? Can we decompose these requirements into lower-level requirements? There might also be traceability demands.

The organizational view focuses on roles and responsibilities (which must somehow be combined with the structural items). We need to address questions such as who owns requirements, which roles exist in the company and what their responsibilities are, and how these roles relate to requirements. It is necessary to consider the organizational view to ensure that things do not fall between the cracks—otherwise, it is possible that everybody assumes that someone else is taking care of them.

The third perspective integrates requirements strategy with agile work and feature flow, which refers to the management of development tasks (work) and the delivery of functionality (features) within agile processes. We need to map the structural and organizational perspectives to the work and feature flow. This can partially be provided by defining done criteria. Further, the work and feature flows should be related to the roles, responsibilities, and ownership of requirements. A stakeholder map can provide valuable information by defining who owns an artifact, who should be informed, and who needs to review it. An explicit review strategy can be very useful, improving the requirements quality and keeping reviewers informed about recent changes.

1.5.4 Paper D

This study investigates the integration of human factors knowledge into agile AV development through the lens of requirements strategy. The aim is to provide concrete solution strategies for managing human factors requirements in agile AV development. We conducted semi-structured interviews with 13 professionals specializing in requirements and design, who also have knowledge of human factors and agile development. The research focuses on three key aspects of requirements strategy: organizational, structural, and work and feature flow. Organizational aspects involve addressing the roles and responsibilities for human factors requirements within the organization, defining how these roles are assigned and managed. Structural aspects examine the structure of human factors requirements, emphasizing the need for traceability and the use of information models to document, track, and maintain these requirements throughout the development. Work and feature flow aspects focus on how human factors requirements are incorporated into development processes, detailing the specific workflows and mechanisms used to manage and integrate features influenced by human factors. The goal is to investigate the human factors requirements integration in AV development from all these three aspects, as captured in the following research questions:

RQD-1 Organizational perspective: How do ownership and responsibility for human factors requirements impact their integration in product development?

RQD-2 Structural perspective: How does the structure of requirements and information models impact the integration of human factors requirements in product development?

RQD-3 Work and feature flow perspective: How does defining a work and feature flow related to human factors requirements influence their integration in product development?

We found that clear ownership and responsibility positively enhance the integration of human factors requirements (RQD-1). This clarity was found to improve communication, ensure proper responsibility allocation, and prevent tasks from being overlooked. However, there were mixed views on whether human factors expertise should be embedded within critical roles like product owners or distributed across teams, with some participants advocating for specialized human factors teams and others suggesting shared responsibilities across all team members.

Regarding the structural aspect (RQD-2), results show that a clear human factors requirements structure positively impacts integration. Participants noted that structured documentation practices, facilitated by tools such as Jira and Confluence, help maintain clarity and organization in specifying human factors requirements. While many interviewees agreed that a clear requirements structure aids in integrating human factors knowledge into product development, there was some debate over whether existing tools are sufficient or if more explicit documentation for human factors requirements is needed. The diversity in responses indicates that while structure is beneficial, its implementation may need to be tailored to the specific organizational context and project requirements.

In terms of work and feature flow (RQD-3), the study found strong support for a robust lifecycle model for human factors requirements, which ensures that these requirements are considered throughout all stages of the development process. Interviewees emphasized the value of iterative development models like agile, which allow for continuous human factors considerations and adjustments. Regular review and reflection on human factors requirements, through practices such as retrospectives and sprint reviews, were also highlighted as critical for continuous improvement. However, the effectiveness of these practices was seen as varying based on project specifics and organizational dynamics. Despite the general support, some participants noted that the approach to integrating human factors requirements might need to be flexible, adapting to the unique needs and methodologies of different projects.

1.5.5 Paper E

This study aims to investigate the role of human factors requirements across different levels of abstraction, from high-level goals to detailed requirements, and identifying the optimal placement of human factors experts within organizations.

A mixed-methods approach was employed in this research, involving shadowing techniques, document analysis, informal interviews, workshops, and surveys. Data collection was conducted in three stages: gaining an in-depth understanding of requirements decomposition and human factors requirements at an automotive company, conducting a workshop with industry and academic professionals, and designing a survey to explore the strategic integration of human factors expertise within organizations.

This paper answers the following two research questions:

RQE-1: To what extent can human factors requirements be confined to specific levels of abstraction of requirements?

RQE-2: Where should human factors experts be positioned within an organization?

Our study indicates that human factors requirements cannot be confined to specific levels of abstraction within requirements (RQE-1). Human factors considerations are relevant at all levels, including feature, system, and detailed software requirements. Through our work with an automotive company, it became clear that details often reveal how a system can support human factors requirements.

To illustrate this, we used a fictive example of a vehicle lane-keeping feature, demonstrating how human factors requirements appear at all levels of requirements. At the highest level (Feature Requirements), we have the overall functionality desired by stakeholders. This is broken down into specific requirements at the System Requirements level, and further into detailed Software Requirements that describe the behavior and functionalities of the software components needed. The refinement from high-level goals to detailed requirements is useful for integrating human factors requirements across all levels of abstraction [100]. This process ensures that high-level objectives are iteratively refined into actionable requirements aligned with human capabilities.

To determine the favorable placement for human factors experts (RQE-2), we identified eleven strategic options, each with its advantages and challenges. The potential placements include feature requirements, system requirements, software requirements, dedicated human factors teams, user-experience teams, product owners, non-functional requirements teams, system/feature evaluation teams, safety teams, and teams responsible for the overall system.

A quantitative survey was conducted to determine the optimal placement of human factors experts within organizations among above identified placement options, focusing on general ranking, effectiveness, and ease of implementation. Bayesian analysis was applied to the survey data to assess the optimal placement options. The results indicate that *user experience teams* and *feature requirements* are the most preferred placements for maximizing product impact, while *user experience teams* and *dedicated human factors teams* are the top choices for ease of implementation. In terms of overall ranking, *user-experience teams* and *feature requirements teams* consistently scored high.

1.6 Answering the Thesis' Research Questions

1.6.1 G1: Domain Exploration

To achieve research Goal 1, we addressed the two questions RQ1 and RQ2.

RQ1: To what extent can human factors knowledge be integrated into agile AV development?

Papers A and C contributed to answering this question. Paper A lays the foundation by investigating the connections between human factors, agile practices, and requirements in AV development. It explores the relationship in detail, identifying the properties and implications of integrating human factors into agile workflows. Part of Paper C builds upon this by identifying the concrete challenges of integrating requirements in agile AV development

Properties of human factors and agile ways of working impact AV development: In Paper A, we looked into the properties of human factors and agile ways of working in AV development. We learned that agile development calls for iterative incremental work and shifts responsibilities to autonomous teams, which usually dislike detailed, static requirements; instead, they are responsible for discovering knowledge by themselves [28]. Human factors experts highlight the importance of considering human variability while developing and testing the system. Human factors experts also focus on the importance of making HMIs and automation transparent. Both agile development and human factors focus on quality in use (for details, see Section 2.4.2).

We observed that human factors knowledge is closely related to agile development. For example, agile development supports iterative incremental work, and human factors properties highlight the importance of experiments. In agile development, iterative work demands continuous testing to address changing requirements, as do human factors. However, certain conceptual differences exist between human factors and agile development. For example, agile development typically implements fast, iterative increments, which do not usually allow time for the rigorous experiments that human factors experts may need in order to ensure user-centered quality.

Agile development prioritizes producing a working product while rejecting extensive up-front analysis and secondary documents (such as requirements, architectures, or human factors studies) [28]. In contrast, human factors emphasize having extensive knowledge and detailed system evaluation before release.

We conclude that the properties of agile development and human factors complement each other in principle. Thus, the inclusion of human factors in agile development can positively affect AV development. However, it is a challenge to fit human factors knowledge (and the corresponding requirements) into the agile way of working that the automotive industry is moving towards, with its fast pace of change.

Paper A also reveals several implications in three themes, i.e., agile work, human factors, and RE. Table 1.4 gives an overview of the implications. These implications can be useful for any organization that aims to consider human factors requirements explicitly during agile AV development.

Our implications suggest that agile teams need to find a way to include

Table 1.4: Overview of Implications

Implications of agile way of working	
(I1)	AV developers must run human factors experiments
(I2)	Experiment design & lessons learnt must be created, re-used, and updated efficiently
(I3)	Human factors expertise must be included on the teams
(I4)	The role of suppliers in agile AV development that integrates human factors must be defined strategically
Implications for human factors	
(I5)	Raise awareness among engineers
(I6)	Put questions on teams, not requirements (and: storytelling over technical requirements)
(I7)	Provide basic human factors knowledge as checklists and design principles
Implications for RE	
(I8)	Epics and user stories to express need for learning in the backlog
(I9)	Increase capability to use prototypes for requirement elicitation and validation
(I10)	Express the relationship between design decisions and human factors as system requirements during development

human factors knowledge in their work in a way that allows them to run human factors experiments while conserving accumulated knowledge. However, humans are adaptive and unpredictable, which makes the formalization of testing procedures and thresholds complex. Another core challenge is that agile frameworks do not offer specialized support for teams to conduct human factors experiments. Because of the large number of autonomous agile teams and the wide range of situations in which human factors considerations may need to be addressed, it is frequently not possible to find dedicated human factors experts and resources to plan and carry out human factors experiments for the team. Also, templates and guidelines which would allow teams to perform their own human factors experiments when experts are unavailable, are not yet mature enough to fully describe human factors experiments in the context of AV development.

Engineers could be trained in multidisciplinary work, making it easier to incorporate human factors knowledge into agile teams. However, further study is needed to determine how agile teams may better manage open questions and their infrastructure for experimentation [101]. Our findings suggest seeking assistance in specialized areas from people outside the team, release train, or even suppliers with the required expertise. Thus, we encourage future research to improve the integration of tests and experiments from a human factors perspective into AV development and to ensure that human factors experts are part of the experimental setup.

Moreover, our implications suggest a shift in the roles that human factors and RE play in agile AV development. In the agile setting, the roles of human factors knowledge and RE become less clearly defined. Human factors experts should play a strategic role rather than an operational one. Instead of designing and conducting experiments themselves, they are needed to mentor and support agile teams.

Since backlog management and increment planning have partially replaced the breath of RE, it appears that the role of RE is waning. As with human factors, the implications for RE demand that requirements engineers take on a new role to better adapt to agile development needs, while supporting the integration of human factors into agile development. Considering that agile teams are responsible for finding and managing a large portion of requirements just-in-time, we anticipate an RE role focused more on assisting developers as they discover, record, and reuse requirements-related data rather than on dictating requirements to them.

Previous work shows how crucial it is to incorporate human factors into the RE process. Our results support this finding, but also identify that actually doing so is more difficult with agile development. Thus, there is a need for additional study in order to integrate the knowledge of human factors (and related concepts) across all the systems engineering disciplines engaged in AV development.

Our exploratory research provides the foundation for future studies that could improve RE in AV development and increase communication about the human factors perspective within agile development. It shows the importance of establishing a culture that integrates human factors knowledge throughout the engineering development cycle. Redefining the roles of human factors and RE specialists so that they support and facilitate agile teams, rather than providing comprehensive and detailed knowledge, would be beneficial.

We believe that these implications provide beneficial knowledge to those who are responsible for developing design procedures and tools—as well as to human factors professionals looking to have a more substantial impact on AV progress. It is anticipated that future research in agile work will formalize efficient procedures for handling human factors studies and their findings.

Critical Challenges with RE in Agile Development: These findings focus on the challenges of managing requirements in agile AV development

Since we mainly focus on agile development in this thesis, we examined requirements challenges irrespective of process phases or particular documents. Instead, we looked at several RE activities (i.e., elicitation, interpretation, negotiation, documentation, and general issues) through the lens of Fricker and Glinz's shared understanding [99]. According to Fricker and Glinz [99], an analysis of shared understanding may focus largely on how it is enabled, how it is built, and how it is assessed in an organization.

We identified several challenges related RE activities in agile development in three case companies. Although the companies have different software development domains, we still found many similar challenges related to RE activities. Table 1.5 gives an overview of the challenges, grouped by RE activity and key factors of shared understanding. We explain all of these identified challenges in Paper C.

There exist RE challenges with large-scale agile development both in the scope of knowledge management and the shared understanding of requirements. Along with many other challenges, our study identified that coordination across teams, a shared understanding of user values, ownership of requirements, and traceability in agile development are all hard to maintain in practice. Some challenges are inherent to large-scale agile development, such as decentralized knowledge building. Some challenges are related to managing requirements,

Table 1.5: Overview of Challenges. Indices (1,2,3) show in which case study (company) a challenge was encountered.

	Shared Understanding		
RE	*Enable*	*Build*	*Assess*
General issues	a) Teams struggle to integrate RE in their agile work efficiently[1,2,3]	b) No formal event to align on customer value[1]	c) Insufficient customer feedback[1,2]
Elicitation	d) Lack of communication with customer[1] e) Who owns customer value[1]	f) Inconsistent elicitation[2]	g) Lack of feedback on elicitation[2]
Interpretation	h) Unclear why requirement is needed[2]	i) Wrong assumptions about customer value[1]	j) Unclear and volatile customer needs[2]
Negotiation	k) Decentralized knowledge building[3]	l) Focus on technical details[1,2] m) Req. open for comments[3]	n) No time for stakeholder involvement[2]
Documentation	o) Customer value description lost between systems[1] p) Lack of knowledge about writing requirements[1,2,3] q) No dedicated time for requirements[1,2,3]	r) Too much/not enough document.[1,2] s) Trace the requirements to all levels, (test, and code)[3]	t) Inconsistency b/c of requirements change[3]

i.e., communicating and documenting requirements.

We found several challenges with requirements management in large-scale agile development. Many of our identified challenges are studied in conjunction with other studies, such as [36, 102, 103]. However, some aspects were not studied in the related work, for example decentralized knowledge building, requirements open for comments (means anyone can open an issue related to any requirements, who have access to system), etc. Many challenges were observed in all three cases, and we are confident that they can also be observed in other large-scale agile companies. However, there may still be many unknown challenges and further study is needed. While the implications identified in Paper A show what needs to happen, Paper C explores the barriers to making it happen. As we moved from exploring implications to addressing challenges, it became clear that integrating human factors into agile AV development is not just about acknowledging its importance. It is about overcoming practical barriers in how requirements are communicated, managed, and shared within agile teams. Addressing these barriers is essential to realizing the potential benefits of human factors, as identified in Paper A, and ensuring their incorporation into agile workflows.

RQ2: Can continuous experimentation help to integrate human factors knowledge into agile AV development?

The exploration of RQ1 led to looking into the integration of human factors experiments in continuous experimentation (RQ2). To address this question, we relied on Paper B, where we examined the similarities and differences between human factors experiments and continuous experimentation. The focus was on understanding how continuous experimentation (typically used in agile software development) could be adapted to integrate human factors knowledge.

Continuous experimentation allows for iterative testing and feedback collection from real users, which is crucial for understanding how users interact with AV systems. This iterative process helps refine systems to better meet human needs, aligning well with the principles of human factors research. By continuously observing user interactions, developers can gather valuable insights to inform design decisions and system refinements. This iterative process aligns well with the priorities of human factors research, which focuses on understanding user behavior, needs, and experiences. The integration of human factors experiments into continuous experimentation allows AV developers to ensure that their systems are not only technically sound but also aligned with the cognitive and behavioral characteristics of their users.

However, integrating human factors experiments into the rapid, iterative cycles of continuous experimentation presents several challenges. Human factors research often requires controlled, thorough experiments that focus on the cognitive, physical, and psychological interactions between humans and systems. These in-depth studies can clash with the fast-paced nature of agile development, which prioritizes speed and adaptability. While continuous experimentation enables rapid feedback, human factors experiments need more time to develop a deep understanding of user behavior and capabilities, potentially slowing down the development process.

Challenges in Integrating Human Factors Experiments into Continuous Experimentation: The integration of human factors experiments into continuous experimentation in agile AV development is not straightforward. Several challenges arise from the differences in approach between these two domains, as identified in the Paper B. Some of the key challenges include:

- Complexity of Human Factors Experiments: Human factors experiments differ significantly from traditional software experiments. They require a focus on human behavior, which involves numerous uncontrollable variables such as individual learning effects and interpersonal communication. This complexity makes it difficult to achieve the same level of control and predictability that is typical in software experiments.

- Sampling and Participant Scarcity: Human factors experiments often require a diverse and substantial number of participants to observe their behavior to obtain meaningful results. However, recruiting participants, especially in sufficient numbers, can be challenging. This scarcity of eligible participants can limit the scope and reliability of the experiments.

- Regulatory and Privacy Concerns: Collecting detailed personal data is often necessary for understanding user interactions with AV systems. However, this raises significant privacy and regulatory challenges, particularly concerning compliance with GDPR and other data protection laws. These concerns add complexity to the management and execution of experiments.

- Resource Constraints: Continuous experimentation in an agile environment can be resource-intensive, and integrating human factors research into this process may require additional resources. They often require specialized facilities, participant compensation, and extensive time commitments, making them more difficult to integrate into the rapid cycles typical of agile development.
- Integration into Agile Workflows: Continuous experimentation in agile development emphasizes quick development and iteration, often requiring decisions to be made quickly based on the latest data. Human factors experiments, however, often require longer durations to produce meaningful data, which can be difficult to reconcile with the fast-paced nature of agile methodologies.
- Lack of Human Factors Expertise: There is a shortage of human factors experts in many development teams, which limits the ability to effectively integrate this knowledge into continuous experimentation. Without sufficient expertise, teams may struggle to design and execute human factors experiments that yield actionable insights.
- Infrastructure and Tooling Needs: Conducting effective human factors experiments requires robust infrastructure, including the right tools and environments for testing. Many organizations may lack the necessary infrastructure to support the integration of human factors experiments into continuous experimentation fully.

In summary, our findings in Paper B show that while continuous experimentation has the potential to integrate human factors knowledge into agile AV development, it is clear that this integration is not without its challenges. The differences in approach between human factors experiments and continuous experimentation require new methods, strategies, and a willingness to adapt these practices.

1.6.2 G2: Solution Investigation

In the following, we start elaborating on solutions and answer RQ3 and RQ4.

RQ3: What strategies can improve integration of human factors knowledge as requirements within agile AV development?

There are many challenges related to RE that can be solved through RE approaches. In RQ3 of this thesis, we introduce the concept of a "requirements strategy" as a method to define RE practices to tackle challenges related to RE in agile.

In the following, we propose several solution strategies that address the challenges related to the needs identified in RQ 1 (see Table 1.5).

[a] Provide tools that allow developers to take ownership of requirements

[b] Have regular meetings with customer representatives

[c] Initiate on-demand meetings with customer representatives

[d] Establish fast feedback cycles

[e] Aim to have requirements templates that includes customer value & goals

[f] Define team responsibilities for different parts of requirements and review updates regularly

[g] Provide rationale

[h] Establish just enough documentation

[i] Plan time for requirements updates

[j] Educate and train the development teams

[k] Ensure to have tools, to support traceability

We observed that the proposed solution strategies can be grouped into three categories, i.e., structural, organizational, and work and feature flow.

For example, a solution strategy for challenge *l) focus on technical details* might be *[e] aim to have requirements templates that include customer value & goals.* According to this strategy, the requirements templates should include particular fields that promote a clear understanding of customer value. This response demonstrates the need for structural improvement.

On the other hand, a solution strategy for the challenge *g) lack of feedback on elicitation* which may lead to misunderstandings later on in an agile workflow is to establish the ability to *[c] Initiate on-demand meetings with customer representatives.* Accessing limited and expensive resources, like a customer representative, is related to the organizational perspective. Moreover, it is challenging to properly integrate stakeholder roles and responsibilities into the business when there is *b) no formal event to align on customer value.* In order to address challenge *b)*, we propose solution strategy *[b] have regular meetings with customer representatives*, which considers both the organizational and work & feature flow.

Another solution strategy *[d] establish fast feedback cycles*, for the challenge *j) unclear and volatile customer needs*, falls under the category of work and feature flow, since it organizes events where individuals can communicate, sharing customer values and feedback.

Similarly, the challenge *s) trace the requirements to all levels* can be addressed with the structural solution strategy *[k] ensure to have tools to support traceability.* The challenge *k) decentralized knowledge building* can be addressed by the organizational solution strategy *[f] define team responsibilities for different parts of requirements and review updates/comments regularly.* Finally, an example of a work and feature flow related solutions strategy is *[i] plan time for requirements updates* in agile sprints to counter the challenge of having *q) no dedicated time for requirements.*

In each case, we defined a solution strategy in collaboration with process managers and experienced engineers. Large-scale agile companies facing similar challenges can adopt these solution strategies (presented in Paper C) to mitigate their RE-related challenges . From this experience, we extracted guidelines for defining requirements strategies in agile.

In summary, specific solution strategies fall into three categories: structure, organization, and work and feature flow. Each category handles unique issues connected to enabling, establishing, and assessing a shared understanding of requirements in agile. Thus, these three viewpoints should be covered by a requirements strategy that groups the solution strategies that apply to a certain situation.

We therefore developed a template for the requirements strategy along these categories, outlined below. We anticipate that with this knowledge, we can learn how companies aim to address RE challenges and use this information to help us address the management of human factors knowledge as requirements in agile development. The term "meeting with customers" reflects the terminology and practices observed during data collection at the participating company. This phrasing represents their specific approach to engaging stakeholders and may not encompass all potential perspectives, such as those of end users. While balancing customer and user requirements could be beneficial, the choice of focus ultimately lies with each company based on their priorities and strategic objectives. This paper does not aim to prescribe a specific approach but rather highlights the general need for a requirements strategy that aligns with organizational goals.

Requirements Strategy: To define *requirements strategy*, our inspiration comes from *test strategy* [104, 105], which focuses testing efforts on achieving quality assurance goals and requires a plan document that defines the scope, strategy, resources, and timetable for testing activities [106, 107]. To our knowledge, this requirements strategy has not been described before.

We argue that developing a requirements strategy that is comparable to a testing strategy is critical for effective agile development. We have iteratively derived our artifact, which provides a template for defining a requirements strategy for agile development. This template is equipped with guidelines for creating a solution strategy to define RE activities in an agile development workflow.

The proposed requirements strategy provides three complementary perspectives: the building blocks. We provide the following building blocks for a requirements strategy: a structural perspective, an organizational perspective, and a work & feature flow perspective. Table 1.6 provides an overview of our proposed artifact, including instructions, typical examples, and best practices—drawn from the three case studies.

The purpose of a requirements strategy is to enable a shared understanding of requirements [108] among these perspectives, particularly in terms of developing a common language (i.e., the functional perspective in Table 1.5) and facilitating the flow of information (i.e., evaluating the building and approach in Table 1.5).

A requirements strategy should be created and systematically documented to ensure all objectives are properly addressed and understood by all stakeholders. It should include practices, tools, and templates that can help an organization address requirement engineering challenges strategically. It should be constantly evaluated, challenged, and revised as the organization, work methods, and products change over time. In addition, the requirements strategy should facilitate the aligning of different stakeholders in terms of terminology, types of requirements and their level of abstraction, roles and responsibilities, traceability, resource planning, etc. [109, 110].

Our guidelines for requirements strategies aim to support organizations

Table 1.6: Building Blocks of a Requirements Strategy: How shared understanding impacts decisions in the workflow [111]

Perspective	Support for shared understanding Common language	 Knowledge flow	Examples
Structural	Define reqts. levels	Define structural decomp.	Stakeholder, System, Component Requirements
	Define reqts. types	Define traceability demands	Requirements and Traceability Information Model
	Define templates		User stories include customer value and goal
Organizational	Define ownership of reqts. types	Define roles and responsibilities	Training plan per type/role; Team responsibility sheet
Work and feature flow	Define lifecycle of types	Map structure to workflow	Elicitation strategy, Definition of done
		Map organization to workflow	Stakeholder map, Requirements review strategy

as they incorporate RE activities more effectively into agile development. Many RE approaches lend themselves to a dedicated upfront requirements phase, which is discouraged in most agile approaches. However, in situations where requirements documentation is needed, agile methods fail to provide good mechanisms to cover it. A good requirements strategy should achieve a compromise that maps RE activities to agile workflow. This compromise should allow the effective management of requirements, but at the same time it should not contradict the organization's goals that led them to introduce agile workflows in the first place. We believe that our work on requirements strategies can be useful and inspiring for any organization dealing with similar challenges. Our guidelines were designed to be adjustable according to the needs of a specific development domain. Thus, any agile organization can create the strategy they need using the provided template. In addition, this artifact could be a base for building solutions for specialized areas, such as managing requirements related to human factors knowledge or AI development in large-scale agile development.

Strategies for Effective Integration of Human Factors Requirements in Agile AV Development: After establishing the general requirements strategy template, we explored specific strategies commonly employed for the integration of human factors requirements in agile AV development. In this section, we define the strategies identified for the three key building blocks of the requirements strategy.

Clear Ownership and Responsibility: A critical strategy for integrating human factors requirements in agile AV development is the clear assignment of ownership and responsibility. Our results emphasize that human factors requirements should be explicitly assigned to roles with the necessary expertise, such as product owners or dedicated human factors specialists. This ensures that human factors considerations are prioritized and consistently addressed throughout the development process. Clear ownership increases accountability and facilitates better decision-making, thereby enhancing the integration of human factors requirements.

Structured Requirements Documentation: Our results in Paper D show that the integration of human factors requirements can be significantly improved through the use of structured documentation tools like JIRA, DOORS, and other requirements management systems and specifically by explicitly mentioning human factors requirements in these tools. These will help in organizing and maintaining human factors requirements, making them more accessible and traceable. A well-structured documentation system ensures that human factors requirements are clearly defined and integrated into the broader development workflow, which is essential for their consistent application in product development.

Our findings also show that the decomposition of human factors requirements, which involves breaking down high-level human factors requirements into more granular system-level requirements, is not universally perceived as an effective strategy. The paper reveals that in innovative and rapidly developing fields like AV development, traditional decomposition methods may not be suitable. Professionals in the field express concerns that these methods could limit the ability to explore emergent properties and adapt to new technological capabilities. This indicates a need for more flexible and adaptive approaches to managing human factors requirements, rather than relying solely on traditional decomposition strategies.

Work and Feature Flow: Developing strong lifecycle models for human factors requirements is crucial in an agile context. The iterative nature of agile development, characterized by continuous feedback and improvement cycles, supports the integration of human factors requirements. Our results highlight the value of these iterative processes, which allow for regular updates and refinements of human factors requirements based on real-time insights and stakeholder feedback. This approach ensures that human factors considerations remain relevant and aligned with the project's evolving needs.

Moreover, a clear plan for regular review and reflection on human factors requirements is essential for their effective integration. Paper D shows that agile practices, such as sprint reviews and retrospectives, provide valuable opportunities for teams to assess and refine human factors requirements. These practices encourage continuous learning and adaptation, ensuring that human factors considerations are consistently evaluated and improved throughout the development process. This approach not only supports the integration of human factors requirements but also promotes a culture of ongoing improvement and responsiveness to human factors requirements in AV development.

In summary, we learned that the integration of human factors requirements in agile automated vehicle development is a multifaceted challenge that requires a combination of structured approaches and flexible, iterative practices. While clear ownership and structured documentation are foundational strategies, the effective integration of human factors requirements also depends on how well these elements are incorporated into the agile workflow. The paper's findings highlight the importance of cross-functional collaboration and the use of agile development to iteratively refine human factors requirements. However, it also highlights the limitations of traditional decomposition strategies, particularly in the context of innovative and rapidly evolving technologies like AVs. This implies that organizations must adapt their approaches to human factors re-

Table 1.7: Solution Propositions for placements of human factors experts.

ID	Solution Propositions
S1	At a higher level of requirements (Feature Requirements)
S2	At System Requirements
S3	At the lowest level of requirements (Software Requirements)
S4	One human factors expert in each team
S5	In a dedicated human factors team
S6	A person/team responsible for the overall system
S7	In user experience teams (interaction design, HCI, UX)
S8	At the Product Owner (PO) Level
S9	With non-functional requirements team
S10	With system/feature evaluation team
S11	With Safety team

quirements integration, balancing the need for structure with the flexibility required to navigate the complexities of agile AV development.

RQ4: Where should human factors expertise be brought in to maximize its impact on the product?

To maximize the impact of human factors expertise on the product, it is essential to strategically integrate this expertise across various levels and teams within the organization. Paper E identifies several key options for placing human factors expertise to maximize their impact. Initially, we identified eleven placement options, as presented in Table 1.7.

However, based on insights from Papers A and B, which highlighted a shortage of human factors experts to assign to every team, we eliminated one option: *One human factors expert in each team.* We then surveyed to determine the most favorable placement of human factors expertise among the remaining ten options, aiming to identify where their placement could maximize impact on the product. We assessed these options not only from an effectiveness perspective—i.e., where they would have the maximum impact on the product—but also in terms of ease of implementation and also in terms of general ranking. Our results show that, in terms of effectiveness, *user-experience teams* and *feature requirements*, *person/team responsible for the overall system* were ranked highest. Meanwhile, in terms of ease of implementation, *user-experience teams*, *dedicated human factors teams*, and *safety teams* were preferred. For the general ranking, the most preferred options for placing human factors experts are *feature requirements* and user *experience teams.*

1.7 Discussion

This thesis investigates the integration of human factors knowledge into agile AV development, with a focus on adapting RE practices to support this integration. This section will summarize our findings, discuss them in relation to existing literature, and describe implications for research and practice.

Agile Development and Human Factors (RQ1): Agile development prioritize rapid iterations, incremental progress, and adaptability to changing requirements, as demonstrated in various studies [19, 112]. The principles of accountability and autonomy within agile teams, which our study highlights, are also echoed in the broader literature [19], particularly the emphasis on teams taking ownership of their own local decisions without relying on extensive documentation, in favor of more flexible, ongoing discovery.

The integration of human factors into AV development is recognized as critical for ensuring safety, usability, and public acceptance [113, 114, 115]. However, the literature [22, 116] indicates that current practices often fall short, particularly in agile development. Agile development, with its emphasis on short development cycles and data-driven approaches, often lead to insufficient consideration of human factors in the development process. The agility and speed of this process contrast with the deeper, more detailed work of human factors, which typically require detailed consideration of the needs of users [61], especially those with different capabilities, ages, or cognitive skills [6]. Our findings align with this perspective, showing that while human factors are recognized as important, their systematic integration into agile AV development is still limited, necessitating for new tools and strategies.

Moreover, our results identified a lack in human factors expertise within organizations. Without sufficient human factors knowledge embedded in development teams, there is a risk that critical insights into human-system interaction may be overlooked. Our findings also highlight that experiments are integral to human factors. While continuous experimentation in agile development provides opportunities to iteratively integrate human factors knowledge through ongoing user feedback, it presents challenges. These challenges, are discussed in RQ2.

Continuous Experimentation and Human Factors Knowledge (RQ2): Continuous experimentation, as a means of iterative user testing and feedback, aligns with agile principles and has been widely recommended in software and product development [117]. The ability to iterate rapidly and test with real users aligns with human factors research, which prioritizes understanding human behavior, cognition, and usability through real-world testing [118]. However, our findings indicate significant challenges in integrating human factors experiments into this process. Unlike software experiments, which are often conducted in controlled environments, human factors experiments involve greater variability and complexity, making it harder to maintain strict control over conditions.

The complexity of human factors experiments, especially when considering user variability, mirrors the observations made by Hancock et al. [119], who stress the importance of understanding diverse human behaviors and interactions in AV systems makes it difficult to apply human factors knowledge within the short feedback loops of continuous experimentation. Furthermore, issues such as participant recruitment, the need for diverse samples, and regulatory constraints, including privacy concerns under GDPR, add further layers of difficulty to the integration process. This is consistent with the findings of

Goodall, who argues that the collection of personal data for user behavior analysis in AV systems raises significant ethical and legal challenges [120]. Agile teams need to navigate these regulatory frameworks while still maintaining the rapid pace of development that is characteristic of their workflow. The need for such a balance has been a recurring theme in literature, particularly with the growing focus on user data and privacy in system design [121].

While this research did not explicitly focus on ethics, these concerns underline the importance of incorporating ethical frameworks alongside regulatory compliance—such as participant consent, data usage transparency, and the broader societal implications of experiments.

Although Paper B also identified best practices—such as refining metrics, improving team collaboration, and aligning experimental goals with agile sprints—these practices alone were insufficient to address the fundamental challenges of integrating human factors experiments into continuous experimentation. Continuous experimentation, while promising, requires a more mature foundation for managing human factors knowledge in agile development.

Therefore, we decided to shift our focus from continuous experimentation to establishing a requirements strategy as a more immediate solution. A well-defined requirements strategy would provide the necessary structure and clarity for handling human factors knowledge within agile processes.

Note that this thesis discusses human factors experiments broadly and does not focus on any specific type of experiments. However, we acknowledge that these experiments can range from controlled simulator studies via closed-track and on-road tests, to large naturalistic field operational tests and production-fleet data collection, depending on the specificity of the requirements being evaluated [122]. The chosen approach should align with the developmental context and desired insights. For example, requirements that necessitate precise, controlled measurements, such as reaction times to system warnings, are best suited for environments like simulators or test-track studies, while broader goals like user trust may require iterative on-road testing or naturalistic field operational tests to capture long-term adaptation and behavior [123, 124].

Strategies for Integrating Human Factors Requirements (RQ3): To address the challenges of integrating human factors requirements in agile development, we developed a requirements strategy inspired by the concept of test strategies [104, 105]. Our approach identifies three essential building blocks that serve as guidelines for overcoming challenges related to requirements in agile development: clear roles and responsibilities, a well-defined structure for requirements, and a transparent workflow.

A key strategy proposed in this work is to define a clear assignment of ownership for human factors requirements. This aligns with the principles of responsibility assignment in agile teams discussed by Moe et al. [125], where the delineation of roles can help ensure that certain aspects of the system—such as human factors—are not neglected. By establishing clear ownership and responsibilities, this strategy seek to prevent tasks from being overlooked, while also improving communication across teams, which aligns with Smith and Reinertsen's observations on the importance of role clarity in complex development [126].

Moreover developing a structured approach to human factors requirements helps in maintaining clarity and better traceability. In traditional development, requirements decomposition and traceability are well-established practices [127, 128]. However, our findings indicate that these practices are not fully supported in the emerging field of AV technology, where requirements can evolve. While structured approaches to requirements management are valuable, they must be flexible to accommodate the dynamic nature of agile development. The diverse perspectives on requirement structuring and traceability highlight the need for adaptable strategies that balance the benefits of structured approaches with the flexibility required for agile development [129]. Moreover, to better integrate human factors in agile AV development, the use of tools for improved documentation has been proposed, though it remains unclear whether existing tools are sufficient or if new ones are needed. Explicitly mentioning and labeling human factors requirements would likely enhance their integration.

Our findings suggest that clear human factors workstreams and regular reviews are useful to better integrate human factors requirements in agile development. Agile development, by design, offer opportunities for reflection through sprint reviews and retrospectives, which can be leveraged to ensure continuous improvement in how human factors requirements are integrated [130].

The effectiveness of these approaches may vary depending on the context, organizational culture, project nature, and team dynamics resonating with the insights of Cockburn [131]. For human factors requirements integration in automotive product development, our findings suggest that tailored strategies are necessary to address the diverse needs of different teams and projects, reinforcing the notion that "one size does not fit all" in the application of agile practices [132]. Additionally, requirements strategies must be regularly updated to remain effective.

Placement of Human Factors Expertise (RQ4): The strategic placement of human factors expertise within an organization is critical to maximizing its impact on AV development. Our results provide several perspectives on this issue, suggesting different options for placing human factors expertise in organizations, including within feature requirements teams, non-functional requirements teams, or in dedicated human factors teams. Each placement strategy presents distinct advantages and challenges.

Our analysis indicates that most participants selected *user experience (UX) teams* as the most effective and feasible option for integrating human factors experts. This aligns with human-centered design principles, emphasizing the importance of UX in creating user-friendly systems, as noted by Norman [59] and the IEA [133]. While placing human factors experts at the feature requirements level also ranked high in effectiveness, its implementation is not seen as easy.

Similarly, integrating human factors expertise at the *person/team responsible for the overall system* level offers high potential for effectiveness but is difficult to implement, as it requires individuals with both technical and organizational oversight. Establishing a *dedicated human factors team*, however, was seen as one of the easiest options, providing a focused resource to ensure that usability is consistently prioritized, though its effectiveness ranked lower, as development teams may not frequently consult this specialized team. Finally, placing human factors expertise within safety teams offers value in ensuring compliance with safety standards (e.g., ISO 9241) [134], but this approach alone is too narrow, and integrating human factors expertise into broader contexts is recommended.

This study focused on identifying preferred and feasible strategies for maximizing the impact of human factors expertise in resource-limited scenarios. While the findings provide valuable insights into participant preferences, it is important to note that the identified strategies are neither exhaustive nor universally optimal. The optimal placement of human factors expertise can vary significantly across organizations, depending on specific contexts, objectives, and resource constraints.

Moreover, the proposed strategies are not mutually exclusive. In fact, integrating human factors expertise across multiple teams—such as UX, feature requirements, and safety teams—may offer a more comprehensive solution to address diverse organizational needs. Considering the potential interdependence among these options is also crucial, as organizational structures often necessitate collaboration across teams. However, this study did not explore such interdependencies.

Future research could investigate additional configurations and examine how varying organizational contexts influence the effectiveness of different strategies, further enhancing our understanding of practical applications in this domain.

1.7.1 Implications

This thesis investigates the integration of human factors knowledge into agile AV development, an area that is generally more straightforward in traditional methods. Recent studies underscore the importance of addressing human factors in AV development and highlight the challenges that arise when they are not properly considered [7, 113]. The research presented here aims to enhance the consideration of human factors knowledge in large-scale agile environments through various strategies, supported by empirical findings that address these challenges.

To effectively incorporate human factors knowledge in agile AV development, we introduced the concept of a *requirements strategy.* This strategy offers guidelines for developing concrete approaches tailored to specific organizational needs and includes three key perspectives or building blocks: organizational, structural, and work & feature flow. The strategy emphasizes the importance of assigning clear roles and responsibilities, structuring human factors requirements, and maintaining transparent workflows to ensure their integration into agile development.

Our proposed strategies and concepts offer potential for better integrating human factors requirements into this field. Our results suggest that more tailored approaches are needed for each building block, considering the project's context and organizational structure. This thesis may inspire researchers to develop customized methods and tools for human factors requirements integration in large-scale agile development projects.

The findings from this research have significant implications for both practice and research.

For practitioners, the findings provide a roadmap for integrating human factors requirements into agile AV development. This includes clearly defining roles and responsibilities, clear requirements structure and clear workflows. Organizations should also invest in educational initiatives to raise awareness and understanding of human factors knowledge among development teams through targeted training and the development of interdisciplinary skills. Additionally, there is a need to develop and implement new tools that facilitate the better integration of human factors knowledge into agile workflows, supporting continuous improvement and adaptability. Establishing collaborative environments where human factors experts and development teams can work together is essential to enhance integration efforts, aligning with human-centered design principles. Moreover, adopting principles of continuous improvement and iterative evaluation will ensure that human factors requirements are regularly reviewed and updated based on feedback and new insights, maintaining their relevance and effectiveness throughout the development lifecycle.

For researchers, the findings highlight the need to explore and validate these strategies across diverse organizational contexts. Future research should focus on developing and empirically evaluating new approaches to managing human factors knowledge, integrating human factors experiments into continuous development, and refining RE processes to better support agile methodologies. Additionally, examining the long-term impact of different strategic placements of human factors experts on the quality and usability of AV systems is crucial, as it will provide insights that can further optimize the integration of human factors knowledge in large-scale agile development.

1.7.2 Future Work

This thesis has identified several challenges and opportunities in integrating human factors knowledge into agile AV development. While the proposed solutions lay a solid foundation for future practices, some areas require further investigation.

One important direction involves aligning human factors experimentation with the rapid cycles of continuous software experimentation characteristic of agile workflows. Research should focus on developing flexible frameworks that integrate human factors evaluation into these cycles while maintaining the speed and adaptability that agile methodologies demand.

Another critical avenue for research stems from the findings of Paper E, which revealed varying stakeholder preferences for the placement of human factors expertise within organizations. Understanding the contextual, organizational, and cultural factors that influence these preferences could provide valuable insights into how to tailor placement strategies for maximum effectiveness.

Moreover, longitudinal studies could explore the long-term implications of these placement strategies. Such research would examine how different approaches impact team dynamics, product quality, and user satisfaction over time, offering evidence-based guidance for sustainable integration practices.

Finally, there is a need for further development of tools and training mechanisms that enable effective integration of human factors expertise across agile teams. For example, tools that embed usability evaluation into continuous integration pipelines or training programs that foster cross-disciplinary collaboration could significantly improve integration outcomes.

Addressing these research directions will advance the integration of human factors knowledge into agile development, ultimately contributing to the creation of safer, more user-centric AVs.

1.8 Conclusion

Human factors knowledge in AV development is important, human factors knowledge ensure that these vehicles meet user needs for safety, usability, and overall satisfaction. The integration of human factors knowledge in agile AV development is an emerging area of interest due to the increasing adoption of agile methodologies by automotive companies. This shift is primarily driven by the need for flexibility, speed, and improved collaboration, which agile practices promise [112]. However, the integration of human factors knowledge, which traditionally relies on thorough documentation and extensive planning, poses significant challenges within agile frameworks. Agile development favors rapid, iterative cycles and minimal upfront requirements, which can conflict with the comprehensive approach required for human factors considerations [84]. This thesis aims to address how effectively human factors knowledge can be brought to AV developers in agile development. The solution has been approached from a RE perspective

Starting with exploratory studies, we identified the current challenges and interrelations between the agile way of working, human factors, and RE. We found that human factors requirements should be integrated from the early stages and continuously adjusted through iterative cycles. The critical role of human factors experimentation emerged as a key finding, prompting us to further investigate how human factors experiments can be integrated with continuous software experimentation. Our results indicate that currently integrating human factors experiments into continuous experimentation is not without challenges, highlighting the need for new methods and strategies to achieve better integration.

Moreover, we identified challenges related to RE in agile development and proposed solutions to overcome these challenges. Based on these solutions, we developed the concept of a *requirements strategy*. We also developed a template for requirements strategy that provides guidelines to help practitioners formulate a strategy for addressing challenges specific to their context. The requirements strategy template can be tailored by different teams and organizations to meet their individual needs while following the provided guidelines. It is essential to regularly review and update the requirements strategy to ensure its effectiveness. The strategy is built on three key aspects: structural, organizational, and work and feature flows.

We then presented various solution spaces for the integration of human factors requirements in agile AV development. We see that the clear ownership and responsibilities, regular retrospectives, and explicit mention of human factors requirements enhance the integration of human factors into product development.

In addition, we identify the strategic placement of human factors experts within organizations, emphasizing the importance of considering human factors requirements at all levels of requirements abstraction and identifying effective placements for maximizing impact. Results suggest that, in case of fewer resources, we should start placing human factors expertise with the *user experience teams* followed by *feature requirements teams* to have their maximum impact on the product.

In conclusion, this research contributes to the understanding of how human factors knowledge can be better integrated into agile AV development from a RE perspective, offering practical solutions. By addressing the identified challenges and proposing concrete strategies for better integration of human factors knowledge in agile AV development, this study aims to enhance the safety, usability, and overall success of automated vehicles in an agile development environment.

Chapter 2

Paper A

Human Factors in Developing Automated Vehicles: A Requirements Engineering Perspective

Amna Pir Muhammad, Eric Knauss, Jonas Bärgman

Journal of Systems and Software, 2023.

Abstract

Automated Vehicle (AV) technology has evolved significantly both in complexity and impact and is expected to ultimately change urban transportation. Due to this evolution, the development of AVs challenges the current state of automotive engineering practice, as automotive companies increasingly include agile ways of working in their plan-driven systems engineering—or even transition completely to scaled-agile approaches. However, it is unclear how knowledge about human factors (HF) and technological knowledge related to the development of AVs can be brought together in a way that effectively supports today's rapid release cycles and agile development approaches. Based on semi-structured interviews with ten experts from industry and two experts from academia, this qualitative, exploratory case study investigates the relationship between HF and AV development. The study reveals relevant properties of agile system development and HF, as well as the implications of these properties for integrating agile work, HF, and requirements engineering. According to the findings, which were validated in a workshop with experts from academia and industry, a culture that values HF knowledge in engineering is key. These results promise to improve the integration of HF knowledge into agile development as well as to facilitate HF research impact and time to market.

2.1 Introduction

The term automated vehicles (AVs) refers to an emerging technology that increasingly automates driving tasks and decision-making in transportation [135]. The society of automotive engineers (SAE) has defined six levels of automation (0–5) [3], starting from *no automation* at Level 0. Many automation features of Levels 1 and 2 (providing one or more automated driving assistance systems (ADAS) to the driver of the car) are already available to consumers. Level 3 features such as lane changing [136], steering control, and car parking [137] are becoming more common. Level 4 is known as high automation, and there are very few companies that have deployed Level 4 vehicles in real traffic (Waymo [138] is one example). However, several companies are promising Level 4 deployment [139], and prototypes of Level 5 vehicles (full automation that does not require human intervention and can perform driving under all circumstances) are under development.

Thus, the number of vehicles with medium to high levels of automation are increasing; according to Litman, half of all new vehicles will be autonomous (which the author defines as automation Levels 4 and 5) by 2045 [140]. As the number of AVs is increasing, so does the number of reported failures. Although fatal crashes of Teslas have been well publicized, [141, 142, 143], failures of AV technology are not limited to a single brand; for example, a pedestrian was killed by an Uber self-driving car in 2018 [144].

These examples, as well as more recent ones reported in scientific journals [5, 145] and the media [146, 147], show how human over-trust in and over-reliance on the automated systems can cause fatal failures of AV. Clearly, even if an engineered, automated solution works perfectly in theory, human factors (HF) must be accounted for to ensure perfect functionality on the roads. As a research field, HF considers humans' physical, physiological, social, and cognitive capabilities and limitations while designing a system [11]. Expanding on this characterisation, several definitions of HF are available, depending upon the context [11]. As part of our study, we extended one of these definitions to enable us to be more precise about HF in relation to AV (see Section 2.4.1).

Several HF researchers have emphasized the need to consider HF knowledge during AV development [8, 10, 148, 149]. For example, Hancock states that attention must be paid to the proper design of new vehicle automation technologies and warns that with the breakneck speed at which automated and autonomous systems are developing, HF perspectives might be overlooked [8]. According to Lee, HF aspects must be considered in order to increase the safety, trust, and acceptance of automated technology, as well as to avoid its misuse and disuse [149]. Currently, companies are trying out different ways to manage the integration of HF knowledge into their research and development (R&D).

In addition to the changes urged by HF researchers, agile development approaches to system engineering are also being introduced to AV R&D organizations. While initially agile approaches were focused on small software development teams [29, 150, 151], their success has led to their adoption in the development of large-scale [36, 152, 153] and mechatronic systems [51], where non-agile, plan-driven, and stage-gate-based processes have been the norm [154].

The agile ways of working adopted by these companies are primarily based on the scaled agile framework (SAFe) [155], which promises to provide "proven, integrated principles, practices, and competencies for achieving business agility using Lean, Agile, and DevOps". SAFe suggests distinguishing a number of abstraction levels, including the lowest level teams, a middle layer where different *solution trains* (a group of teams working on a coherent part of the product) are managed, and a portfolio level on top. Due to their iterative nature, agile approaches are suitable for building systems whose requirements may change; further, experience from early versions of a system can impact later versions [29, 51, 150]. Thus, in theory, agile approaches are well suited to the introduction of stakeholder concerns (such as those provided through HF knowledge) in automation development: Agile often reveals previously unforeseen requirements for a system under development, such as considerations of HF.

The process of eliciting, analyzing, describing, and validating requirements is called requirements engineering (RE) [156]. To date, it has been particularly challenging to apply RE to the agile development of systems at scale [150, 157]. Meyer highlights the rejection of upfront analysis as particularly problematic [150], but other challenges exist, particularly with managing and communicating requirements-related knowledge at scale [157].

The literature [8, 10, 148, 149, 158] leaves no doubt about the importance of considering HF in AV development. For example, an AV at Level 3 still requires humans to be able to take over control of the vehicle. Especially when it comes to switching control between the human and vehicle, human factors such as reaction time, comfort, fatigue, and understandability must be considered as requirements [159]. Yet, particularly in the light of well-known challenges for RE in scaled agile system development, it is unclear how to ensure their consideration. There is a lack of empirical research on how to communicate HF aspects of vehicle automation to AV engineers[1], particularly during agile development. This is a relevant research gap with practical implications: Automotive companies are moving towards scaled-agile system development. It is unclear how to introduce HF requirements into agile system development, which is the traditional way of managing knowledge in the development lifecycle. Thus, it is unclear how to ensure that HF knowledge is incorporated into agile system development, and practitioners struggle with a lack of clear guidelines.

We investigate this research gap in this exploratory qualitative study. Within the general research goal of determining how HF aspects of AV development can be communicated to AV engineers, this study specifically aims to investigate the essence of agile, HF, and RE in the context of large-scale agile AV development. The research goal is operationalized by addressing the following research questions (RQs):

RQ1: How do HF experts and AV engineers characterize HF in relation to AV development?

RQ1 is motivated by the broad spectrum of definitions offered by literature.

[1]We recognize that many HF experts can also be considered engineers in terms of AV development (the domain of HF engineering). However, in this study, we distinguish between HF experts and AV engineers in order to clarify how HF experts are currently communicating to AV developers and identify any communication gap.

In order to understand how HF aspects can be communicated, we first need to establish a working definition of HF in terms of AV development. We then explore the relevant properties of HF and agile work in RQ2:

RQ2: Which properties of HF and agile ways of working impact AV development?

In RQ3, we are particularly interested in implications for agile ways of working, HF work, and managing requirements in AV development:

RQ3: What are important implications when aiming to better integrate HF into AV development?

This work answers these research questions by qualitatively analyzing interviews with ten experts (HF experts and AV engineers who work in the automotive industry), complemented by two additional interviews with academic leaders in the field of human factors. Our results indicate that an important property of scaled agile is its way of working, which advocates responsiveness to change by shifting responsibility from managers who plan at the system level to autonomous teams that make local decisions. To support such local decisions, it follows that HF knowledge should be available to the agile teams to raise awareness, enable asking relevant questions, and guide them in the right direction. It also follows that agile AV teams should be able to produce HF knowledge on demand, e.g., by conducting HF experiments within their team's iterative work; further, RE should provide methods for effectively managing the knowledge gained from the experiments. We validated these findings in a workshop setting using a survey questionnaire, as well as in discussions with 28 expert participants from industry and practice. The validation study confirms that our findings are very relevant to the industry.

The paper is divided into seven sections. This introduction, Section 2.1, is followed by Section 2.2, which provides the background and reviews related work; Section 2.3 discusses the research methodology. The main findings are presented in Section 2.4. Section 2.5 presents the outcome of the survey performed to validate the findings of this study. In Section 2.6, we discuss our findings. Finally, Section 2.7 concludes the paper.

2.2 Background and Related work

The research presented in this paper is multidisciplinary, targeting both systems and software engineers as well as HF experts. Therefore, this section provides the background on which the argument of the exploratory qualitative analysis is built. This background may seem obvious and basic in parts. However, since the targeted readers belong to many disciplines, some basics need to be explained for completeness: many HF experts are not familiar with the agile way of working or RE, and many AV engineers are not familiar with the domain of HF.

2.2.1 Human Factors in Automated Vehicle Development

Human factors are an integral part of the development of road transport [57]. However, as the definitions of HF are many and diverse [11, 55], there may be a problem when people with different definitions are communicating requirements and knowledge [55]. Taking a scientific perspective of the definition of human factors,

The Journal of the Human Factors and Ergonomics Society describes the science of human factors as pursuing “fundamental knowledge of human capabilities and limitations – and the basic understanding of cognitive, physical, behavioral, physiological, social, developmental, affective, and motivational aspects of human performance” as a means ”to yield design principles; enhance training, selection, and communication; and ultimately improve human-system interfaces and sociotechnical systems that lead to safer and more effective outcomes.”[2]. Although this definition may seem clear and concise, individuals may have different views of what HF entails [55], and their views may impact how they consider HF in their profession. Thus it is important, when studying how HF is considered in the workplace, to investigate what their views of HF actually are. This may be particularly important when the subjects in a study have very different backgrounds, such as when studying the role of HF in the development of automated vehicles (as in the current study); HF experts, as well as a range of different engineers, are involved [57]. As a consequence, developing a precise definition related to a specific topic (here AV design) is warranted.

2.2.1.1 Human Factors and its Role in AV Development

In AV development, HF relates to aspects of both software development and physical AV design. Examples of HF aspects in AV development are many. Note that a common misconception by many non-HF experts is that HF is simply a list of factors, while it is actually a range of aspects that affect humans, or that humans affect (see, e.g., the definition by the Journal of Human Factors). Physical aspects range from seating ergonomics (as AVs are impacting vehicle interiors [67]) to the physical design and placement of human-machine interfaces (HMIs). Typically, humans are directly affected by software aspects of HF, including: how and when the (software-based) HMIs display information [66], how external road users are to be communicated with [64, 65], how the vehicle stays in the lane [62, 63], how it keeps its distance from a lead vehicle [95, 160, 161], how it overtakes other road users [162, 163], how humans and AVs communicate [164], and how AVs can avoid driver over-reliance on the AV performance and ensure that the trust in the AV is properly calibrated [165, 166]. These examples highlight the extent to which successful engineering depends on HF knowledge. Yet it remains an open question how engineers gain awareness of HF in their daily work and design decisions.

2.2.1.2 What HF Issues Impact AV Development?

Kyriankidis et al. highlight that as AV development in the industry keeps moving forward at a fast pace, the gap between research in academia and R&D in the industry continues to grow [167]. They stress the importance of more research on the interconnection of AVs with other road users, human trust in and acceptance of AVs, and how much (and which) information drivers will

[2]https://journals.sagepub.com/aims-scope/HFS

get and should be getting from AVs. The authors also discuss the need for more experiments to study how humans interactand control transitions between the driver and the AV. Similarly, Ian et al. [168] argue that the benefits of AVs (such as safety) can only be achieved if they are designed according to standards of human-system integration. The importance of integrating HF into the design and evaluation processes of autonomous vehicles to increase their safety and trust is also highlighted in this position paper [169] and in the book by Lee et al. [57].

The work by Saffarian et al. [170] lists six specific issues regarding HF in AV development: overreliance, behavioral adaptation, erratic mental workload, skill degradation, reduced situation awareness, and inadequate mental models of automation functions. The authors proposed a solution for these issues specific to CACC (Cooperative Adaptive Cruise Control), as well a proposing a mechanism of interaction between humans and CACC. However, the solution simply proposed a few different modes to keep the driver in the loop and facilitate cooperation between driver and vehicle.

Chen et al. [171] describe the importance of transparency between intelligent systems (e.g., robots or AVs) and humans. The authors developed a Situation awareness-based Agent Transparency (SAT) model to ensure an appropriate interplay between AVs and humans. Their study mainly targets human drivers' need for transparency of AV functionality in order to promote better understanding, trust, and interaction.

For each individual HF issue encountered during the AV development process, involved engineers may lack the experience or competence to include the appropriate HF aspects. However, no one can know everything. Communication about HF among stakeholders is therefore crucial. The AV development process must include many stakeholders from different domains, making it interdisciplinary.

2.2.2 AV Development: Processes, Approaches, Recent Developments

In the automotive sector, the R&D required to create cars and trucks and offer related services is a complex affair, involving many disciplines such as mechanics, electrical hardware, and (increasingly) software. Whereas electronics and software in cars were originally introduced simply to optimize control of the engine, their development now drives 80% to 90% of the innovation in the automotive industry[3]. This subsection provides an overview of AV development in the context of requirements engineering(RE).

This section provides an overview of AV development in the context of requirements engineering (RE).

2.2.2.1 Requirements Engineering

International standardization and certification bodies provide valuable insights into the fundamental concepts of requirements engineering. The IEEE defines a requirement as either (i) a condition or capability needed by a user to solve a problem or achieve an objective; (ii) a condition or capability that must be

[3]According to industry experts: `https://tinyurl.com/y9jnoupd`

met by a system or component to satisfy a contract, standard, specification, or other formally imposed document; or (iii) a documented representation of a condition or capability as in (i) or (ii) [172]]. The International Requirements Engineering Board (IREB) describes requirements as representations of the needs and desires of customers and users for new things to be built or old things to be upgraded [173]. Accordingly, requirements can be of three types: functional (a result or behavior to be provided by a function), quality (a quality concern not covered by functional requirements, such as performance, availability, security, or reliability), and constraint (a further limitation on valid solutions beyond what is necessary to fulfill functional and quality requirements). IREB characterizes Requirements Engineering as specifying and managing "requirements for systems such that the systems implemented and deployed satisfy their stakeholders' desires and needs" [173].

Activities of RE typically include elicitation, analysis, specification, validation, and management of requirements [174]. In addition, requirement prioritization becomes a key RE activity in agile development, supporting elicitation and analysis by identifying the requirements with the highest stakeholder value [175]. Research emphasizes the multidisciplinary aspects of requirements engineering [174]; however, we are not aware of any works that explore how HF research can be integrated into requirements engineering activities for agile system development at scale.

2.2.2.2 Development Practices

Traditionally, the automotive environment has been characterized by long lead times [49] and stable, sequential engineering practices [154]. Eklund et al. [50] argue that the industry is currently transitioning from plan-driven, stage-gate processes [154] to more value-driven, continuous approaches [176, 177] (often referred to as agile methods [150] or agile transformation [178]). Gren and Lenberg argue that the main motivation for such a transformation is to be able to respond to changing requirements [51].

Agile methods have traditionally been proposed for small teams (six to eight developers) [29, 150, 179]. The core values of agile methods as described in the agile manifesto [18] are: Focusing on individuals and interactions to develop working software in close collaboration with customers with an emphasis on embracing change while de-emphasizing processes, tools, extensive documentation, contract negotiation, and following plans. In fact, agile methods have been presented as the antithesis of previous plan-driven approaches. In its original form, an agile team would take notes about customer needs in the form of user stories on small index cards. Often, these are described as boilerplate statements: "As a <role> I want <feature> so that <value>" [38]. The much more detailed requirements of plan-driven approaches are omitted; instead, agile methods push for a continuous dialogue with customer representatives or product owners and comprehensive sets of tests, which are ideally automated [150]. On the other hand, agile methods have been criticized for limiting requirements engineering to functional requirements described through (exemplary) scenarios and discouraging upfront planning [150].

2.2.2.3 Development Approaches at Scale

Automotive R&D work is typically a collaboration between an OEM (Original Equipment Manufacturer) and suppliers in several tiers. The OEM owns the vehicle brand and orders mechanical, electrical, and software components from suppliers. Thus, the ability to specify requirements for the vehicle and break them down into component specifications is a core competency for an OEM.

In order to improve their responsiveness to changing requirements, OEMs have started to bring more development in-house and to identify new collaboration models with suppliers [97, 180]. As a result, OEMs struggle to maintain effective ways of structuring, documenting, and managing requirements for increasingly complex systems [37, 52]. While software teams may have quickly learned to adopt agile methods, company-wide adoption is usually slow, mostly due to skepticism [181]. Thus, new ways of managing requirements must be conceived for OEMs and their supplier value chains [157].

Moreover, for complex products such as cars, it is important to scale agile methods beyond individual teams, since if the overall plan for the complete vehicle cannot be changed there is limited value in an individual team's ability to respond to change [51]. SAFe is the most commonly used framework for scaling agile [155], especially in the automotive domain [157]. SAFe describes a requirements information model that groups several user stories into epics. Epics can then describe mid-to-long-term goals for groups of teams. The model also describes non-functional requirements as a way to present quality requirements as constraints for user stories and epics [38].

Previous works have described inadequacies in the SAFe framework [157] and its requirements information model [182]. Of particular relevance to this paper is the fact that scaled-agile methods struggle to provide alignment among many software teams [157, 182]; we need to consider the effects of scaling agility beyond individual software teams since questions about agile ways of working must be part of our exploration of HF. For example, for a given automated driving function, several teams must align on how to address HF. For brevity, hereafter we refer to *scaled agile* or *large scale agile* simply as *agile*.

2.2.3 Related work: Communicating Human Factors Knowledge and Requirements to AV Engineers

Interdisciplinary communication is often difficult. However, many fields such as aviation, transportation, and medicine, acknowledging the importance of HF knowledge, have worked to integrate HF design principles and techniques into the design and development of products and systems. Vincent et al. [183] suggested that the communication gap between HF knowledge experts and other developers is due to a lack of common ground; they proposed the use of mediating representations of boundary objects [184] for effective communication. Bruseberg [185] introduced a novel methodology that feeds HF knowledge into an architectural framework. However, the author mainly discusses HF from a cognitive perspective. Alternatively, Chua and Feigh [70] suggest including HF in an early design stage. While HF can provide significant input to improve the communication between HF experts and system engineers, it is unclear exactly how to include HF knowledge in these stages of development. Other authors

[186, 187, 188, 189] advocate including HF in system design via SysML, using activity, block, and sequence diagrams.

Van Maanen et al. [190] have discussed how HF can be integrated with AI for better human-machine cooperation (HMC). Whereas current customization is limited to static interfaces, improved HMC could provide customized support to users. However, knowledge about both HF and artificial intelligence (AI) and how to integrate them is lacking. To bridge this knowledge gap, van Maanen et al. [190] have proposed a methodology based on multidisciplinary cognitive engineering (CE+). In CE+, HF experts provide the relevant information (such as the support concepts and rules) and strategies for the specification and evaluation of HMC. The authors concluded that HF and AI must be integrated into the early stages of the development process. In fact, the User Centered Ecological Interface Design (UCEID) [191] method proposes a combination of techniques (e.g., data collection and task and cognitive task analysis) to include HF considerations in the early stages of the overall system design processes. The main finding of UCEID is that it is important to meet the dual requirements of demographically diverse clients and technology delivery. It remains unclear how these requirements can be integrated into the (agile) development cycle; however, considering the importance of the issues mentioned above, a way must be found to design AVs with HF in mind [192, 193].

Adopting this design practice proves to be challenging, not the least because of the adoption of agile development [194]. Processes have become more iterative, putting more emphasis on a continuous understanding of requirements. It is unclear how the above-mentioned methodologies would work for the communication of HF knowledge in today's large-scale agile AV development. For example, Kashfi shows how difficult it is to align user-centered design and UX in agile development [195].

In summary, although communicating HF knowledge to engineering teams is challenging, research provides ample motivation to explore how this challenge can be overcome in practice. To our knowledge, no systematic approach exists that make sure that HF are adequately represented in agile system development.

2.3 Research Method

Our exploration of the role of HF in developing automated vehicles is widely based on the epistemological stance of critical realism, a research philosophy that distinguishes between the 'real' world and the 'observable' world. With respect to this study, we made this distinction by observing and analyzing expert opinions about how HF aspects are addressed in engineering, rather than assuming that we can analyze those aspects directly. Critical realism relies on a common ontology or sociological theory, which we provide through our detailed assumptions about the role of HF, RE, and agile methods based on related work in Section 2.2. In our study design, however, we were also inspired by the school of pragmatism, focusing on particular causalities of pragmatic relevance (i.e., the implications that follow from particular properties of agile AV development and HF). Finally, we also drew inspiration from constructivism, considering that anyone willing to learn a complex topic may need to construct knowledge. In fact, we added continuously to our knowledge as we learned

Table 2.1: Interviewees' roles and work experience (Experience level: Low= 0–5 years, Medium=5–10 years, High= More than 10 years)

ID	Role	Experience Level
S1	HF Expert (Specialist)	High
S2	HF Expert (Strategy, Specialist & Research)	High
S3	AV Engineer (Strategy & Architecture)	High
S4	AV Engineer (Requirements & Research)	Medium
S5	HF Expert (Management & Research)	High
S6	HF Expert (Specialist)	High
S7	HF Expert (Specialist & Design)	High
S8	AV Engineer (Safety & Research)	Low
S9	AV Engineer (Strategy & Specialist)	High
S10	HF/AV Engineer	High
***Special**	**Interviews**	
S11	HF Expert (Specialist)	High
S12	HF Expert (Specialist)	High

new items that did not previously fit into our mental model. Given this mix of epistemological stances, we decided that an exploratory, qualitative inquiry was the most appropriate to address our research questions [75].

Our case consists of a number of automotive companies, including manufacturers and suppliers, collaborating not only within the value chains needed for building automated vehicles but also beyond, to build and maintain excellence in the field. We relied on semi-structured interviews with experts to provide the primary data, since we were specifically interested in applying the personal views of experts in the field (who collaborate within and across value chains and concrete products) to chart the landscape of HF in relation to AV development.

In this section, we describe the collection and analysis of the data and discuss the validity of our results, given the qualitative nature of our research.

2.3.1 Data Collection

Our strategy for recruiting interviewees for our study [196] relied primarily on convenience sampling. That is, we tried to identify interviewees who possess the relevant expertise and were willing to participate. Our results confirm that such experts are rare among companies, and that it is important to protect their time. In recruiting interviewees, we relied both on the personal and professional networks of the authors, built through years of research with participating companies, and on recommendations from the interviewees themselves.

We aimed for a mix of similarity and variation in our sampling in order to cover different perspectives (HF vs. engineering and OEM vs. supplier) in sufficient depth. We interviewed ten experts from five Swedish companies: four from Volvo Cars, two from Volvo Trucks, two from Zenuity, one from Veoneer,

Table 2.2: Interview questions, mapped to the research questions.

Interview Questions	**Research Question(s)**
1. Background of Interviewee (Demographic Data) • What is your role? • What is your experience in that role? • What is your experience with HF / Requirements? Reminder: We will take notes during the interview, which we will send later for confirmation.	Demographics
2. How would you characterize what HF is and how it relates to requirements for AV development (or AI-based systems)?	RQ1
3. In your experience, how does engineering work with or without HF? What is missing?	RQ2
4. How does HF knowledge come to engineers?	RQ3
5. What are the main challenges in conveying requirements from HF to engineers that design automated vehicles (or from engineers to HF experts)? • Follow-up: what about conveying knowledge from HF/behavior as input into the AI-based AV-design process? • Think about comfort zones as an example, safety aspects, software requirements aspects (e.g., AI based control of the vehicle) compared to traditional physical "user experiences" of AV.	RQ2 & RQ3
6. What scenarios related to AV in urban environment are the most difficult (and/or important) to convey requirements to AV-engineers?	Not used
7.Do you have recommendations on how to optimize communication between human factors experts and engineers of AI-based AVs? Any guidelines for incorporating human factors into AI-based AV design guidelines?	RQ2 & RQ3
8. How should the process (or: way of working) for system design look like? Particularly in agile development how we do that?	RQ3
9. Thanks you for the interview, next steps. • Whom else should we interview? • Anything we forgot to ask?	All

and one from Autoliv. In addition, we conducted two more complementary interviews with international academic leaders in the field (S11 and S12), to get additional perspectives on the definition of HF and emerging themes. All of the industry interviewees have been working with AV companies for years, often more than ten (see Table 2.1). The experience level of the participants is classified as *low* if they have less than five years of experience, *medium* if they have between five to ten years, and *high* if they have more than ten years. In Table 2.1, S11 and S12 are separated from the other participants because these interviews were conducted in a slightly different style and the preliminary results from the other interviewees were kept in mind.

We relied on semi-structured interviews because they are especially suitable for exploratory studies [75]: depending on the course of the interview, questions can be adjusted to mitigate the risk of asking the wrong questions, and follow-up questions can be created to satisfy emergent information needs. This approach allowed the participants to articulate their individual and valuable views, concerns, and expectations. Consequently, interviews tended to resemble guided discussions and were engaging both for interviewees and interviewers.

Each interview took between 60 and 80 minutes. In most interviews, all three authors were present; at least two authors were present in every interview, which allowed us to keep extensive, often verbatim, notes. The second author took notes, and the first author conducted the interview. The third author has extensive experience working with the automotive industry along with a knowledge of HF, in addition to being an engineer. Given his multidisciplinary background, he was there to ask follow-up questions and provide clarification.

Notes ranged from 700 to 1750 words and contained, on average, 1325 words. We did not record the interviews. We did, however, show our notes to the interviewee during the interview. While we were interested in the perspectives of experts on the role of HF knowledge in AV development, the discussion could have touched on examples of perceived or real shortcomings in processes, which would be very sensitive information. It was thus deemed better not to record the interviews; after a sensitive discussion, any such content was eliminated from the meeting notes or, if necessary, more suitable examples/formulations were substituted. Before the interviews, we prepared a guide[4] to help us cover the same topics in each interview. Each interview included the nine open-ended questions and detailed follow-up questions contained in the guide. We designed the interview guide with the intention of getting an HF perspective on the design and development of AV technology. The map between the interview guide questions and the research questions is shown in Table 2.2.

2.3.2 Data Analysis

In order to analyze the data obtained from the interviews, we relied on the common set of principles [197] used for qualitative analysis of interview data. Specifically, these principles include: transcribing the interviews, familiarizing ourselves with the data to attain a deep understanding of the phenomena being investigated, coding, generating initial themes, and finalizing the themes and overarching concepts.

The extensive interview notes were a good starting point for further analysis. To familiarize ourselves with the data, we read the interview notes thoroughly while creating memos describing those ideas that the notes inspired [198]. Then we highlighted parts of the text related to our research interest and assigned them labels (so-called "codes"). In parallel, we continued to create and discuss memos to capture any noteworthy aspects as they surfaced. For these activities, we relied on both generic word processors (MS Word) and specialized qualitative analysis tools (NVivo[5]). Through these steps, we identified the main ideas as well as common perspectives.

After formalizing and coding the data, we further classified all the relevant

[4] We provide the interview guide as well as an overview of our themes in relation to codes and example quotes as data set at Zenodo, DOI: 10.5281/zenodo.5562487

[5] https://www.qsrinternational.com/nvivo-qualitative-data-analysis-software/home

codes into candidate themes. For example, the following quotes were coded as "validation test" and "test dilemma," respectively.

> *"Perhaps put more emphasis on validation tests, that is, not only automated tests but also test the quality in use."* — S4 - AV Engineer

> *"I have seen people spend three person-years on things they have never tested with real humans. Then they claim they have never had time to do so."* — S2 - HF expert

By analyzing and categorizing these and other relevant quotes, we came up with a theme called "Testing".

The themes and codes were then re-analyzed to check if there was any missing or extra theme with respect to our interview notes or any mismatch in the code classifications. In this way, we refined the set of themes until all authors agreed that it provided complete coverage of all aspects of the data, without redundancy, on a meaningful level of abstraction.

Finally, we renamed our themes to better align with research questions. Section 2.4 describes the outcomes of our data analysis

2.4 Findings

This section presents our findings, with each subsection addressing one RQ. We start by defining HF in AV development, based on our interviews and the literature (RQ1). The second research question focuses on the properties of HF and agile ways of working (RQ2). These properties raise important questions (discussed in our interviews) about the interplay of both disciplines. Then, we present the implications that emerged from these discussions in three themes related to research question RQ3: implications for agile ways of working, implications for HF work, and implications for managing requirements.

For each theme, we start our report of results with a box that shows in which interviews we have identified related codes. We provide a table with the assigned codes per interview as an external resource[6].

2.4.1 Human Factors in Relation to AV Development (RQ1)

> HF Definition based on codes from interviews with:
> S1-3, S5-6, S11-12

In order to explore the systematic capturing and managing of human factors in AV development, it is important to share a common understanding of the key concepts. Therefore, our first question aimed to understand each interviewee's perspective on human factors and their relation to AV development. Our interviews show a broad and diverse usage of the term 'human factors', which is also reflected in the literature.

For example, the following quote shows a rather broad definition of the term,

[6]We provide an overview of our themes in relation to codes and example quotes as data set at Zenodo, DOI: 10.5281/zenodo.5562487

assigning responsibility for considering human factors to the complete development cycle:

> *"How to safely develop an AD function (without killing humans in the process) so that in case of a crash, people will say that the car was behaving reasonably."* — S1 - HF Expert

In other examples, interviewees had a more technical, outcome-oriented view of the term and how it feeds into other engineering processes:

> *"Learning the user preferences, should it be race driving, comfort, safety, or speed."* — S3 - AV Engineer

> *" HF was 2WW system ergonomics, then CS brought up HMI. Those have merged since. You have physical interfaces, but also services, but also how users are adopting new functions and whether or not they continue using. HF and HMI are intertwined. Ergonomics is included and overlaps with the cognitive side, e.g., external communication with other road users. Understanding the warnings and so on."* — S6 - HF Expert

Human factors knowledge, such as preferences about the level of comfort, safety, and speed, is instrumental for the development of AV. The role of HF in providing input to design and development is also reflected in another interviewee's quote:

> *"Understanding the interactions between people and all other elements within a system, and designing in light of this understanding."* — S5 - HF Expert

However, considering HF requires more than one-way communication with engineers. As the following quote reveals, HF sets limitations on both engineers and users.

> *"How to communicate the limitations of behavior so that people understand what they are allowed to do and what they are not allowed to do... [This is easy to do with] HF related to safety. [With other] HF [e.g., those] related to a sense of calm or serenity that is a bit more difficult."* — S2 - HF Expert

Given the broad use of the term 'human factors', we aimed to integrate different interpretations from practitioners' perspectives into a definition of human factors in AV development.

As part of this process, we relied on the two international experts to provide more insight. They confirmed that a working definition is indeed needed and might need to be compiled from various sources and then matched with comments from our other interviewees:

> *"So one definition is from the journal of human factors, which is about knowledge of human capabilities and limitations. I think that would be good, but there is one on this other page...The goal to design safe, comfortable, effective [systems for] human use is almost describing what you are trying to achieve, so I am just wondering whether you could start by saying this is what we believe HF is and then add more with your work."* — S12 - HF Expert

In addition, our international experts confirmed that a good working definition must be related to the engineering cycle

> *"Understand, create and evaluate cycle. HF plays a role in each component & understand is about identifying requirements, human capabilities, limits, needs & describing those in ways that influence the design."* — S11 - HF Expert

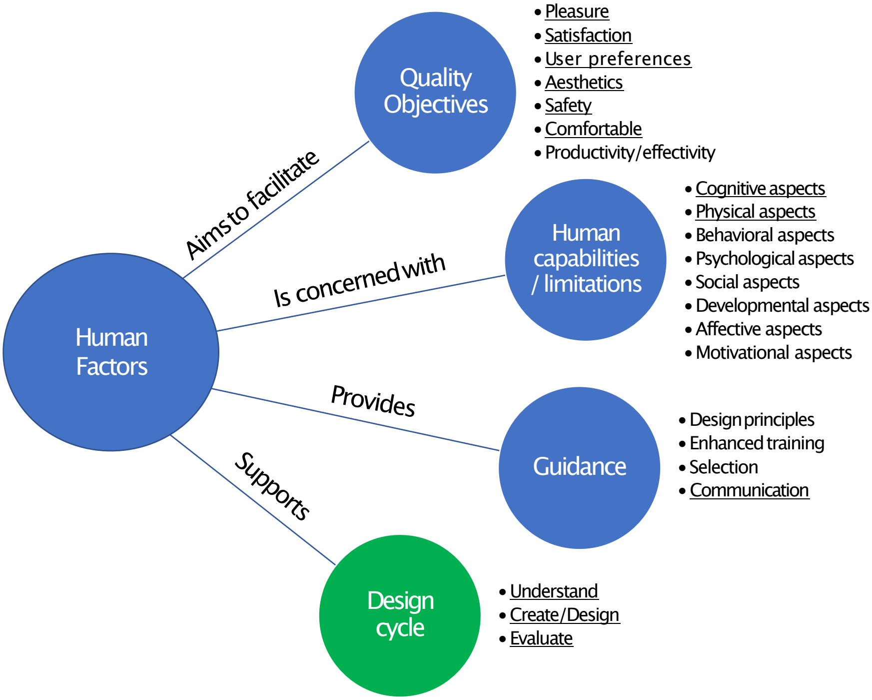


Figure 2.1: A mind-map of aspects that define Human Factors in the context of the design and development of automated vehicles.

> *"AV has physical considerations regarding how you get in a vehicle, make the seats big enough to accommodate the people, there are certain design issues. But it also that people trust the AV to be safe, how do they perceive the important risk, do they feel comfortable with the algorithms, do the algorithms behave as expected, does it enhance end goals: pleasure, satisfaction, aesthetics?"* — S11 - HF Expert

This is a cross-cutting theme that is also visible in the other subjects' quotes above.

In summary, we note that multiple definitions of HF exist, even on the homepages of key journals of the field (e.g., [199]), depending on the specific research context. In our research context, it is crucial to link HF to AV design and development, as well as the development cycle. As suggested (by S12 above, for example), we start from a generic established definition of HF (taken from [199]]), and relate it to the development cycle. Figure 2.1 represents our working definition graphically: added aspects are shown in green, and the most important aspects from our interviews are underlined (both in the Figure and in the quotes above).

> **Definition:** The field of *Human Factors in AV Development* aims to inform AV development by providing fundamental knowledge about human capabilities and limitations throughout the life cycle so the product will meet specific quality objectives.

Based on our interviews, we can highlight some critical aspects of this definition that shape the relationship between human factors and agile AV development. Firstly, it is important to relate human factors to *AV development and its product quality objectives.* These objectives usually include an AV design result that is pleasurable, satisfactory, user-preferred, comfortable, aesthetic, effective, and safe for stakeholder interaction [57].

Another component of the definition, *human capabilities and limitations*—which include cognitive, physical, behavioral, psychological, social, affective, and motivational aspects, is the core concern of human factors experts [199]. It is critical to effectively manage these capabilities and limitations during AV development. Therefore, it is a crucial role of HF in AV development to provide *fundamental knowledge* about human capabilities and limitations and their relation to quality objectives for AV design. Typically, this knowledge is provided in the form of design principles, training, selection, and communication. In this paper, we will focus on the implications of knowledge transfer in the context of agile AV development.

This fundamental knowledge is needed throughout the *design cycle of AVs*. While various design cycles have been proposed, we refer to the phases that Jacobson et al. found to be essential when building software-intense systems [200]: understanding the requirements; shaping, implementing, testing, and evaluating the AV system; and putting the AV system to use. Note that in modern AV development, these phases are iterative and incremental. Relating HF to AV development throughout the design cycle is of paramount importance for discussing the relationship between HF and AV development. Yet, it is missing from many established definitions of HF and therefore highlighted in green in Figure 2.1.

Thus, to answer RQ1, we noted that AV development is suffering from the lack of a working definition of HF. From our interviews with industry HF experts, we extracted the core aspects that such a working definition should have and triangulated it with definitions found in the literature. We further validated our suggested definition with interviewees S11 and S12. Thus we have established a common language for addressing RQ2 and RQ3.

2.4.2 Properties of Human Factors and Agile (RQ2)

In order to lay the foundation for improving the way that HF knowledge and development work are integrated into agile AV development, we first focus on the properties of HF and agile ways of working. We provide an overview of our findings for RQ2 in the left part of Figure 2.2.

2.4.2.1 Properties of Agile.

When we started our investigation, we were aware of the role of agile in transforming companies and the challenges this puts on requirements. Initially, we mainly included questions about agility to investigate its influence. However, all interviewees highlighted certain properties of agile that are important when considering the interplay of HF and AV development. In order to mirror the emphasis that our interviewees put on agile methods, we begin by describing the properties of agile that influence the management of HF knowledge most.

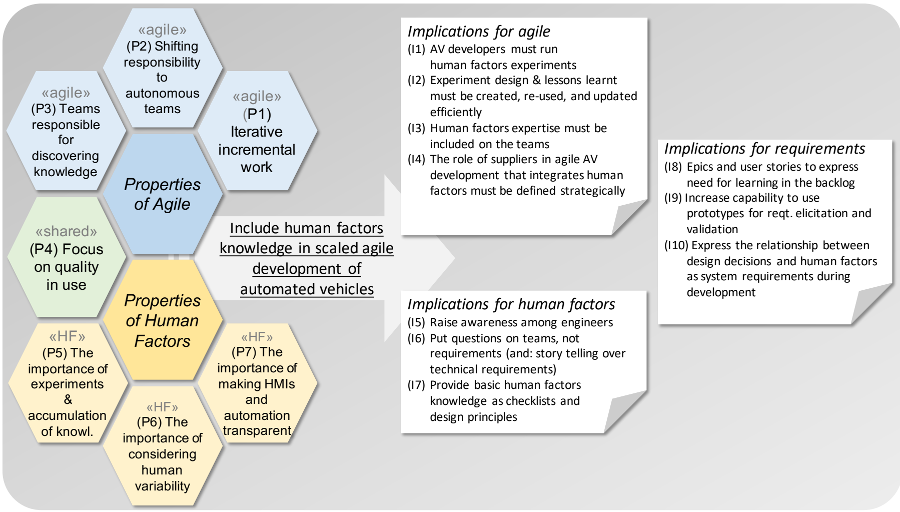


Figure 2.2: Taking a requirements engineering perspective, our qualitative study on Human Factors for Automated Vehicles revealed themes relating to properties of human factors and agile system development, as well as implications for human factors and agile system development and requirements.

The following themes emerged from the data analysis of these properties (shown as P1–P4 in Figure 2.2).

(P1) Iterative incremental work

P1 based on codes from interviews with: S2, S4, S5, S6, S7

Agile promotes *iterative incremental work*, to help organizations deliver fast and often as well as increase their responsiveness to changing requirements. For example, Subject 4 mentions that a property of agile work is an incomplete specification early on, combined with iterative work:

> *"[...] But we are working in an agile way, so the specification is not complete in the beginning, but we iterate, and changes might come later."* — S4 - AV Engineer

Subject 2 suggests that this has completely changed how HF are communicated to development teams:

> *"We had requirements, but that has changed with the agile transformation. We now see it mainly as knowledge transfer, how to move HF knowledge to the teams. The game has completely changed. It is much more a social kind of setting."* — S2 - AV Expert

Our interviewees mainly expressed this as a positive change, as expressed by Subject 5:

> *"At least not in the very old way, where high-level aspects are very much disconnected. Waterfall will not be the solution. But better integration and iterative work sound very promising."* — S5 - HF Expert

Yet, it is important to complement the perspective of teams with a full system view and make sure that HF (for example) fit into the big picture, as Subject 7 mentions:

> *"Agile teams tend to get small bits of tasks and work with these for a short period and then leave it because it is not in the backlog anymore. If it was only for the teams to develop, then nobody would take full system view. What kind of language do we use, when to use knobs, touch screens, ... if it was only up to the teams, you would not have that holistic picture. That is our most important part right now."* — S7 - HF Expert

(P2) Shifting responsibility to autonomous teams

P2 based on codes from interviews with: S3, S6, S7

Agile methods aim to achieve fast, incremental delivery and responsiveness to change by *shifting responsibility to autonomous teams*. These teams can then make local decisions quickly. As a result, agile teams dislike static, detailed requirements, which limit the team's autonomy and, therefore, its effectiveness. This property of agile is mentioned by Subject 3 (for example):

> *"[...] they are then responsible for the topic. T-shaped teams."* — S3 - AV Engineer

This property of agile teams has advantages and disadvantages. Subject 6, for example, highlights the transparency that this approach generates.

> *"I like the way we work now with agile trains. Things are very visible; you see all the stories created by the different teams, and you have clear goals. It is in the method that you promote what each team is doing."* — S6 - HF Expert

However, Subject 7 repeats their concern about the potentially missing system level view as a result of increased team responsibilities.

> *"Ideas come up internally that developers and hardware designers should know their requirements by themselves. I feel like that is difficult."* — S7 - HF Expert

(P3) Teams responsible for discovering knowledge

P3 based on codes from interviews with: S1, S3, S6, S7, S9

Instead of receiving detailed requirements, agile teams prefer being *responsible for discovering knowledge* themselves, relying on face-to-face communication rather than on extensive documentation.

This preference is implied by a number of our interviewees. Subject 7, for example, explains how the role of HF experts has changed:

> *"[...] It is less about handing over requirements, and instead being there for discussion or to evaluate the concepts."* — S7 - HF Expert

The responsibility of agile teams to discover knowledge is also evident from how S9 describes the need for agile teams to seek expertise:

> *"Then, as an engineer, you should have enough awareness to know when to seek out that expertise. But it is, of course, only one competence area of many."* — S9 - AV Engineer

Similarly, Subject 1 shares their view on how to guide agile teams to discover knowledge about the right concepts, not by defining requirements but by relating high-level stories that then can be explored:

> *"[...] Do the guerilla requirements. Do not write requirements, but tell interesting stories based on empirical data, getting the right concepts into the brains of engineers (where it then stays because they are so bad at forgetting things)."* — S1 - HF Expert

(P4) Focus on quality in use

P4 (agile) based on codes from interviews with:
S3, S4, S6, S10

One of the differences highlighted by our interviewees between agile and traditional approaches is the different concept of quality. The quality of software-based systems is commonly divided into internal quality (structural properties such as maintainability of the software) and external quality (the fulfillment of user requirements—i.e., providing the desired functionality) [201]. In contrast, agile approaches suggest that requirements rapidly change and those provided initially may not describe the users' needs by the time the product is finished. Therefore, according to agile approaches, it is not sufficient to fulfill (potentially outdated) requirements to obtain user satisfaction; it is necessary to address the users' actual needs and focus on *quality in use*. Agile practices with this focus include, for example, the on-site customer [29] and sprint demos [179].

Thus, agile teams take responsibility for regularly demonstrating a working product, putting it to use in the intended context, and enabling feedback by end users and customers.

A good description of this property was given by Subject 3:

> *"[...]Working agile means being able to test what you are doing and improve the quality continuously."* — S3 - AV engineer

It is, however, important to not rely solely on automated tests. Subject 4 highlights the need to push for acceptance tests.

> *"[...] Put more emphasis on validation tests, that is, not only automated tests but also test the quality in use."* — S4 - AV Engineer

This is generally a good fit for HF, as our interviewees mentioned—for example:

> *"Understanding that problem is crucial, as well as getting experience about what users like. How do people want to be addressed?"* — S10 - HF/AV Engineer

> *"[...] If you have a nice mindset and an open point of view, the iterations, increments, and multi-disciplinary work will fix many of these things. User-centered design."* — S6 - HF Expert

There are, however, a number of conceptual mismatches between the HF and agile AV development domains. Examples include agile focusing on delivering a working product and rejecting big up-front analysis and secondary documents (for example, requirements, architectures, or HF studies)—and even removing those documents after implementation is complete [150]. These practices may lead a team to decide on a particular design based on requirements and HF studies, and to maintain only the actual work product. In future iterations, therefore, the rationale for a design decision is no longer available, potentially leading to duplicate or sub-optimal work (since previous requirements and HF knowledge cannot evolve).

2.4.2.2 Properties of Human Factors.

In order to represent the relevant properties of human factors, the following themes were derived from certain characteristics referred to by the interviewees.

(P4) Focus on quality in use

P4 (HF) based on codes from interviews with: S2, S9, S10

HF experts also focus on *quality in use*, since they are concerned with deriving knowledge from human interactions with the system (here: the AV). Clearly, with that focus, the internal, structural properties of the software are of little relevance. Even external quality does not sufficiently describe a system's quality from a human factors perspective: A system that fulfills all requirements on paper but is not pleasurable, satisfactory, or safe to use in the real world will fail to win over an end user. As a result, with agile, HF experts and AV engineers are much closer to each other than they were in traditional development approaches (which broke HF quality considerations down into internal and external quality indicators for implementation). This concordance is implied by the following response from S9:

> *"[...]Not sure we are good with agile yet, but ideally, through improved testing, we should get even more improvements. As long as you can include an HF expert, then all should be fine in the larger picture."* — S9 - AV Engineer

Incremental, agile work can actually be ideal for addressing HF. For example, S2 points out that it allows the quick generation of feedback and an understanding of HF in relation to the system under construction.

> *"[...] Could be really interesting to see how an HF requirement changes with time. How and why does it change? You change it because of some feedback. Why did it not work? Because of this test. Then assess the quality of the test (formal or just friends trying it out). Then also heuristic evaluations, defining usability errors. For those, you do not need a lot of subjects. This is not a statistical approach; it can generate a lot of problems at a low cost. But are these the right problems? The key problem is that HF experiments are expensive."* — S2 - HF Expert

(P5) The importance of experiments

P5 based on codes from interviews with: S2-4, S7, S9, S11

HF experts highlight the importance of experiments and testing the system. In agile development particularly, iterative work demands continuous testing, both to avoid regression problems and to address changing requirements.

HF experts aim to perform experiments with the system under assessment using human subjects who are not on the engineering team developing the product. Thus, HF experts might test how humans react in specific situations, how they get distracted, how they feel about the system, and how the system affects their behavior (e.g., over-reliance), while considering human variability. S2, for example, relates the importance of experiments to the need to identify assumptions:

> *"You need to identify assumptions. ... Start from someone's idea and explore it (from engineering), or you can take your own knowledge (HF) and bring it in. And then you create the experiment and the conditions."* — S2- HF Expert

Again, the shift to agile work has significantly changed the work with experiments. As S3 points out, it requires continuously finding ways to test assumptions.

> *"[...] Before it was easier: Just ask this department to come up with requirements from HF perspective, then push it into the development teams. Then, have test methods in place. What we have done... working agile means to be able to test what you are doing and improve the quality continuously. That also well matches with HF."* — S3 - AV Engineer

Even though it might have been *easier* in plan-driven development, our interviewees confirm that agility promises to be more effective, as stated by S7:

> *"Agile promotes these things; you need to demo regularly. [but are there enough HF people?]."* — S7 - HF Expert

S11 reasons that short, quick experiments with quick feedback cycles should be preferred. The short feedback cycle would help to identify challenges and notify the organization while the topic is still hot. This could enable bringing in the right expertise (e.g., HF or control theory) at the right time, and consequently make the team "fluid and agile".

Perhaps experiments to check assumptions could become a continuous source of input to agile development, since assumptions will always come up. S4, for example, speculates about a shared service to provide support for such continuous experimenting:

> *"You could treat this as a shared service for everyone, support to set up such experiments. It should be quick and easy. It is also related to dealing with assumptions in a more structured way than we currently do."* — S4 - AV Engineer

(P6) The importance of considering human variability

P6 based on codes from interviews with: S1, S4-5, S8, S12

HF play an important role in ensuring that the developed systems are suitable for all humans (with different user characteristics such as age, culture, experience, and visual and cognitive capabilities). Depending on their backgrounds, humans have different capabilities, limitations, and behavior, as for example stated by S4:

> *"Requirements are very different depending on the country and customer company. How does culture change how people think about HF?"* — S4 - AV Engineer

HF knowledge can help design the system to improve its performance, while considering human variability makes the system usable for a diverse set of users. For example, S5 confirms:

> *"Yes. Humans are complex, with strengths and weaknesses that are very different from artificial systems, there is a lot of variability in the performance of a human."* — S5 - HF Expert

This leads to a high level of complexity that must be managed during AV development.

> *"In many cases, the empirical data set is very complex."* — S1 - HF Expert

Bringing the complexity of human aspects into the development of AV also poses technical challenges to engineering, as S8 suggests:

> *"We need the car to handle random walks with these parameters or with those parameters. Can we even model all human traffic in this way?"* — S8 - AV Engineer

The challenge of modeling complex human traffic behavior could also be seen as an argument for the iterative development of AV systems and HF experiments: it would not only allow the incremental verification of assumptions that are relevant for the current development, but it would also allow the accumulation of knowledge about the bigger picture.

(P7) The importance of making HMIs and automation transparent

P7 based on codes from interviews with: S4, S7, S10-11

It is critical for users of vehicle automation to have a proper understanding of the system's capabilities and limitations (i.e., the decisions the AV makes must be understandable and the user must understand what the system's limits are) in order to respond correctly and avoid misuse or disuse of the system. Yet not all users read the manual or attend training. Therefore, the system's capabilities and limitations must be completely transparent, through HMIs and kinematic cues; the AV's capabilities and limitations should be obvious as a result of proper HF design. S10, for example, frames important HF questions around this theme:

> *"Who even checks the manual? Will you even be able to find the button that activates an assisting system? With new functionality in a car, how do you introduce it to users?"* — S10 - HF/AV Engineer

If a feature is not transparent to users, they might deactivate it (potentially reducing their safety, but even more problematically, resulting in over-reliance and over-trust).

> *"Try to find ways so that users do not switch off active safety systems. It is about the methods, how you use them, and their purpose, HF and RE."* — S4 - AV Engineer

It is through the effective interplay of systems and users that the overall safety goals are reached. Making sure that typical users sufficiently understand new features is, therefore, an integral HF part of developing AV.

> *"[...]There are certain design issues there, but there is also [the fact that] that people trust the AV to be safe; how they perceive the risk is important. Do they feel comfortable with the algorithms? Do the algorithms behave as expected? Does it operate reliably?"* — S11 - HF Expert

Aligning trust and understanding between users and automated systems is of critical importance—but also hard to do. HF expertise is needed, which could, as S7 points out, be obtained from experts on the team or from the results of surveys (or other sources):

> *" ... 8/10 people can make sense of the new function in the first attempt. We need either to be there with our expertise or bring in the end users, e.g., in a clinic or survey, have test drivers."* — S7 - HF Expert

2.4.3 Implications (RQ3)

This section presents the implications that emerged from interview notes on the three themes related to research question RQ3 ((shown as I1–I10 in Figure 2.2). Each theme (implications for agile ways of working, implications for HF work, and implications for managing requirements) is presented in a separate subsection.

2.4.3.1 Implications for Agile

Given the set of properties of agile and HF discussed above, there are certain implications for any organization that aims to take HF knowledge explicitly into consideration during agile AV development. These implications are not currently provided by agile methods, nor are they easily achieved. This section, therefore, highlights the need to adjust agile ways of working and presents, where available, potential approaches indicated by interviewees.

(I1) AV developers must run human factors experiments

I1 based on codes from interviews with: S2, S5, S7-8, S12

> *"[. . .]Holistic view, ideas come up internally that developers and h/w designers should know requirements by themselves. I feel like that is difficult."* — S7 - HF Expert

> *"[. . .] it is less about handing over requirements, and instead being there for discussion or to evaluate the concepts."* — S7 - HF Expert

Thus, when integrating HF knowledge into agile AV development, it follows that agile teams must be able to run HF experiments themselves. This ability is the first implication for agile that we derived from our interview data. For example, S8 clearly states that the engineers are ultimately responsible for the implementation of a function:

> *"[. . .] Engineers should make sure that those (requirements) are implemented and tested."* — S8 - AV Engineer

This generally includes extensive testing. However, as S12 points out, tests that only focus on technical aspects and ignore HF will not fully cover the actual needs.

> *"You know engineers will test and retest and retest, but not really with a human in mind. . . "* — S12 - HF Expert

Agile teams know best what specific knowledge is needed at any given time. Yet, those teams usually lack the HF expertise and knowledge, which must then be provided in a different way (see Implication I3).

> *"For an engineer without HF training, the fundamental thing in HF is to test your assumptions. How do you communicate to engineers that to get HF knowledge, you need to test it with human subjects? Experiments."* — S2 - HF Expert

A core challenge is that agile frameworks do not offer dedicated support for teams to run HF experiments. Due to the large number of autonomous agile teams and the wide variety of situations in which HF considerations may have to be made, there are often no dedicated HF resources available to take on the role of designing and running HF experiments for the team.

> *"[. . .]I do realize that the teams need such HF knowledge."* — S5 - HF Expert

Based on *(P2) shifting responsibility to autonomous teams*, *(P3) teams are responsible for discovering knowledge* and *(P5) the importance of experiments*, we conclude from our data that *AV developers must run human factors experiments.*

(I2) Experiment design & lessons learnt must be created, re-used, and updated efficiently

I2 based on codes from interviews with: S2, S8-9

If agile teams are to take responsibility for running HF experiments (Implication I1), the teams should also be responsible for decisions about which *experiment design & lessons learnt must be created, re-used, updated efficiently.* S8, for example, suggests the need to aim for re-use.

> *"[...] We must have a generic model for such experiments, that can be reused in different products."* — S8 - AV Engineer

In particular, the re-use and updating of designs and lessons require additional attention in agile ways of working. Agile setups must support a single team as it creates HF experiment designs and generates results, which are then re-used by other teams. If a particular change to the system invalidates the results of a study (e.g., by changing how a user interacts with the system), the team must understand the change and, for example, run a new, updated experiment. In short, teams must be able to judge the validity of experimental designs and results and re-run the experiments if needed, as mentioned by S9:

> *"Create new knowledge on demand but also use the accumulated knowledge from previous projects. Several levels of tests, even with customers."* — S9 - AV Engineer

AV development therefore must integrate discovery and reuse of HF knowledge into agile methods, where the focus is on maintaining tests and deploying working versions of the product iteratively. S2 provides thoughts on how this could work in principle:

> *"With the agile approach, you continuously test. It allows you to fake a finished product. Then you can put an experienced user in the car and see how they react. You can go in both directions: Start from someone's idea and explore it (from engineering), or you can take your own knowledge (HF) and bring it in. And then, you create the experiment and the conditions and then update it."* — S2 - HF Expert

Our second implication therefore follows from our data, specifically considering *(P1) iterative incremental work*, *(P4) focus on quality in use*, and *(P5) the importance of experiments.*

(I3) Human factors expertise must be included on the teams

I3 based on codes from interviews with: S1-2, S6-9

Agile teams should have the expertise that allows them to take ownership and responsibility for identifying HF needs and relevant HF knowledge. Interviewees suggested *including HF expertise in the agile teams* (for example, in the form of T-shaped teams), with each team member having a certain area of expertise.

> *"Not sure we are good with agile yet, but ideally, through improved testing, we should get even more improvements. As long as you can include an HF expert, then all should be fine in the larger picture."* — S9 - AV Engineer

While there is a lack of availability of HF expertise in most companies, there are, different ways of ensuring teams have the necessary expertise.

S8, for example, wonders whether HF experts should be involved in creating abstract, reusable models, or instead be part of the teams which are deriving operational requirements.

> *"[...]But this requires a good model of the HF. We must have a generic model for such experiments that can be reused in different products, or do we need to create those models within the operational requirements specification? In that case, HF experts must be included in the teams."* — S8 - AV Engineer

Similar considerations were also discussed with S6. In typical scaled-agile frameworks, such as SAFe, HF experts could be assigned as a shared resource or within a particular release train. S6 suggests that as a shared resource, HF experts would lack visibility and would thus not be able to have an impact on agile design decisions.

> *"I like the way we work now with agile trains. Things are very visible; you see all the stories created in the different teams, you have clear goals... The problem is, if you are not on the train, you are not able to promote yourself. If you are a shared resource team, you have less visibility. So it will be better to be on the train."* — S6 - HF Expert

For the same reasons, S1 also considers adding HF experts to the release trains; but in line with S8 above, advances the alternative consideration of having HF experts as part of the individual development teams within an agile release train.

> *"You cannot be everywhere. But having your requirements and hand them over and then wait, that is not going to work. Being a part of the team or an agile train to some extent is the way forward."* — S1 - HF Expert

S7 indicates a clear preference that the HF expert should be involved with the teams directly.

> *"[...]The way you communicate your requirements is within the teams. You need to be there. If you are not in the teams, it will be a challenge."* — S7 - HF Expert

In summary, successful AV development relies on HF experts who can guide developers with respect to how to set up an experiment, run it, and interpret its results—as well as judge its credibility (and identify when a change invalidates previous experiment results, requiring another experiment iteration). This fits well with agile T-shaped teams (i.e., a team in which each member can work on all tasks but also possesses particular, specialized expertise, so that leadership is assumed by the most suited member).

While there are clear advantages to including HF experts directly in agile work (i.e., within the teams or in larger release trains that combine a number of teams working on a specific product area), there are also challenges with this setup: for example, lacking HF experts as S2 indicated.

> *"But we are lacking HF people."* — S2 - HF Expert

I3 is established based on *(P2) shifting responsibility to autonomous teams, (P3) teams responsible for discovering knowledge, (P5) the importance of experiments* and *(P6) the importance of considering human variability.*

(I4) The role of suppliers in agile AV development that integrates human factors must be defined strategically

I4 based on codes from interviews with: S3-6, S10

Given the lack of HF expertise, one has to identify a strategy that ensures that HF are taken into account in agile AV development. The strategy may consist of getting support in certain specialized areas from outside the team or release train, or even from suppliers with expertise in the area. As the automotive value chain is increasingly transformed into agile ways of working and continuous integration and delivery, new collaborative models are emerging that integrate suppliers tightly into incremental work for specific purposes. In fact, large suppliers already do a substantial amount of research on HF related to their current and future product portfolios.

> *"Currently, we are working more on component level. This is even more challenging since it depends on system level engineering decisions, so you should ideally work with an OEM to define the particular requirements for the component and its context."* — S6 - HF Expert

A particular impediment is the access of suppliers to users of a specific AV, which limits the supplier to relying on their more general expertise and specific requirements from the manufacturer, as discussed by S4.

> *"Yes, but we do not often have access to the users, we get the requirements from the OEM, and we rely on them to tell us what is really needed. So perhaps, it is good that things are then indeed separate (HF, RE)."* — S4 - AV Engineer

Still, we conclude that the role of suppliers is significant for two reasons: a) they often possess HF expertise that could be valuable to their customers and b) as agile development includes increasingly large parts of the value chain, our previous reasoning about the need for HF expertise in agile teams also holds for suppliers.

Our final implication for agile is, therefore, to *systematically decide whether and how to include (or get engaged as) a supplier* in the agile development of AVs, including the supplier's HF expertise in the teams when collaboratively designing, developing, and integrating AV components. It is based on *(P2) shifting responsibility to autonomous teams, (P6) the importance of considering human variability* and *(P7) the importance of making HMIs and automation transparent.*

Summary and important questions. The four implications for agile lead to the following important questions for future research in agile AV development:

[a] How can developers be encouraged to run HF experiments?

[b] How can we efficiently create, re-use, and update HF experiment designs and lessons learnt?

[c] How can HF expertise be included in agile teams, given that few experts are available?

[d] How can suppliers be involved strategically in working with human factors?

2.4.3.2 Implications for HF

(I5) Raise awareness among AV developers

I5 based on codes from interviews with: S5-7, S9

Through our interviews, we learned the need to *raise awareness among engineers* about HF and the implications for the final product of not including HF in the development process.

> *"It is a lot about marketing yourselves internally. For example, we are part of PI planning for different trains, talk to the teams, explain what we need at which point."* — S7 - HF Expert

Although conducting extensive experiments and communicating their results are part of agile development, engineers often do not have enough time to acquire the needed information (e.g., due to short, agile development cycles). Moreover, engineering companies may have engineering cultures; generally, engineers prefer gathering information through data rather than HF, which may be considered less important than simply getting the technology working. This culture is implied in the following quote from S5:

> *"[...]Sometimes, engineering sometimes just seems to think that HF is about putting nice wallpaper on the wall. They don't understand how early [how fundamentally] HF needs to be taken into account."* — S5 - HF Expert

S6 points out that for managers, it is often easier to bring a particular expert onto a team than to work on changing the mindset of the engineering department (although it is much less effective):

> *"They like to bring in a UX engineer rather than work on the mindset. "* — S6- HF Expert

A shift of the overall company mindset would be needed so that HF knowledge can be integrated into the AV development more effectively, as S9 hopes for:

> *"[...] Then, as an engineer, you should have enough awareness to know when to seek out that expertise."* — S9 - AV Engineer

I5 is based on *(P2) shifting responsibility to autonomous teams, (P5) the importance of experiments, (P6) the importance of considering human variability* and *(P7) the importance of making HMIs and automation transparent.*

(I6) Provide teams with questions, not requirements

I6 based on codes from interviews with: S2-4, S7, S11

As AV engineers adapt to work in an agile way, communication about HF must be adjusted as well. One of our interviewees formulated this implication clearly:

> *"Put questions on teams, not requirements."* — S3 - AV Engineer

Agile teams do not like detailed requirements, which are often too detailed and too static, interfering with their autonomy as they seek appropriate solutions and adjust to change.

> *"We had requirements, but that has changed with the agile transformation. We now see it mainly as knowledge transfer, how to move HF knowledge to the teams. The game has completely changed. It is much more a social kind of setting."* — S2 - HF Expert

It might be better, therefore, to raise important questions and allow the agile team to find answers that fit their current state of development.

> *"[...] it is less about handing over requirements, and instead being there for discussion or to evaluate the concepts."* — S7 - HF Expert

A complementary approach (to raising questions for the team) relies on storytelling. By using stories that highlight the critical concepts while considering questions that point to the critical information needed, agile teams are enabled to take responsibility for HF knowledge. This empowerment is the consequence of *(P3) teams responsible for discovering knowledge*, and *(I3) human factors expertise must be included on the team.*

(I7) Provide basic HF knowledge as checklists and design principles

I7 based on codes from interviews with: S1, S6-7, S12

A key impediment to providing HF expertise to agile teams is the availability of experts, as mentioned by S7:

> *" We have tried different things. We had one HMI expert in each team, but that did not scale, we do not have enough experts to have one in each team for 100%. Maybe HF experts should provide checklists to engineers."* — S7 - HF Expert

We, therefore, add implication *(I7)*: HF experts should *provide basic HF knowledge as checklists and design principles* to development teams. S1, for example, points out that HF experts should work on a higher abstraction level to increase their reach. They should provide guidelines and other reusable knowledge, rather than specific, system-related requirements:

> *"From an HF perspective, it is important to prioritize the human experience. Better to talk about guidelines than about requirements."* — S1 - HF Expert

The availability of such reusable guidelines would be an asset, as S5 confirms:

> *"Ideally, one would need some guidelines, to coordinate between application projects that must be communicated. Those guidelines can be in PowerPoint or other company standards."* — S5 - HF Expert

According to S12, this could be done via checklists:

> *"I think we need to make engineers aware of the typical HF limitations and capabilities... You know, how is the mental model affected, or, you know, what's the relationship between the system and our mental model, or fatigue, distraction, situation awareness, workload, all of this everyday stuff that we as people suffer from when it comes to interacting with systems. So, you know, it is almost like a checklist... I guess we need to have a certain checklist."* — S12 - HF Expert

Several of our interviewees agreed that this could lead to a better utilization of the available HF experts' skills. This implication is supported by *(I1) AV developers must run human factors experiments*, *(I3) human factors expertise must be included on the teams*, *(P5) the importance of experiments*, *(P6) the importance of considering human variability*, and *(P7)the importance of making HMIs and automation transparent.*

Summary and important questions. The implications for HF indicate a strategic, rather than operational, role for HF experts. Instead of designing and running experiments themselves, these experts are increasingly mentoring and supporting agile teams. This raises important questions:

[a] How can awareness of HF be raised in agile AV development?

[b] How can agile teams be enabled to effectively create and maintain HF knowledge?

[c] Which guidelines and design principles can provide basic HF knowledge to agile teams?

2.4.3.3 Implications for Requirements Engineering

(I8) Use epics and user stories to express a need for learning requirements in the backlog

I8 based on codes from interviews with: S1, S3, S6

Agile methods provide only a limited view of requirements, focusing mainly on epics and user stories in various backlogs. This shortcoming introduces new challenges for the decomposing high-level concerns into different backlog items and distributing them over the different release trains and value streams, as S3 pointed out. While interviewees mention that there is still a lot to learn, advantages and best practices slowly become manifest, as mentioned by S6:

> *"Things are very visible, you see all the stories created in the different teams, you have clear goals... We should likely start documenting them as part of epics in JIRA. We have HF streams, active safety streams, ... The work is cross-functional, so I am both in HF and active safety streams. The recommendations/ functions should be written in a user-friendly way and which value it provides to customer and user."* — S6 - HF Expert

This, in particular, affects strategies to get cross-cutting and interrelated requirements such as those related to HF into the system, as the following practice from S1 suggests:

> *"[...] Do the guerilla requirements. Do not write requirements, but tell interesting stories based on empirical data, getting the right concepts into the brains of engineers (where it then stays because they are so bad at forgetting things)."* — S1 - HF Expert

For RE experts in the automotive domain, the change in focus from providing a comprehensive requirements document to managing continuous learning with respect to certain goals is challenging. From our interviews, we conclude that an RE expert should enable teams to approach and document this learning systematically, instead of writing requirements for them. This implication is based on *(P1) iterative incremental work* and *(P4) focus on quality in use.*

(I9) Increase capability to use prototypes for requirements elicitation and validation

I9 based on codes from interviews with: S2, S4, S10-11

Prototyping was suggested by S4 when discussing requirements engineering:

> *"Prototyping for requirements engineering, so one can find specific details about a problem, and use them to discover new requirements."* — S4 - AV Engineer

This is not only a good way for agile teams to discover requirements, but also a necessary way for HF experts to uncover new HF knowledge, as S2 suggests:

> *"Then I like to ask them to help me build a prototype, a wizard of oz car. Then I can test it. Because prototyping is a good way for requirements elicitation and validation."* — S2 - HF Expert

Consequently, prototypes are key for aligning HF experts and agile teams as well as facilitating synergies:

> *"Prototype adds a set of requirements, but also how the requirements are manifest in terms of interaction or physical design. Then HF expert get involved in evaluating that in usability testing and heuristics evaluation."* — S11 - HF Expert

The infrastructure for constructing prototypes has become quite sophisticated, as mentioned by S10—allowing a huge variety of tests to be run and collecting large amounts of data.

> *"What do we need for hardware to succeed in ADAS or AD platform(first question from the system team)? We have this box full of things we can measure in our prototype.–¿ which of these tools do we need...It is fun to work with everything. But we need to find the key sensor outputs for good collaboration. If we have new sensor inputs, how can we put a value on those for a collaboration? How can we structure that kind of work?"* — S10 - AV Engineer

We summarize our interview data in this theme as an implication for RE to *increase the capability to use prototypes for requirements elicitation and validation*, based on the identified needs and HF checklists within agile teams. This implication resonates well with *(P3) teams responsible for discovering knowledge* and *(P5) the importance of experiments*, and might offer good support for *(I2) experiment design & lessons learnt must be created, re-used, and updated efficiently*. This is also in line with *(P6) the importance of considering human variability*, as prototype validation must take into account the range of human variability.

(I10) Express the relationship between design decisions and human factors as system requirements during development

I10 based on codes from interviews with: S3, S5, S9, S10

While it makes sense to describe stakeholder requirements as epics or user stories (see Implication I8), it is important to document the desired capabilities of components and subsystems, which follow system requirements; otherwise, it is not sufficiently clear how HF related to essential requirements for automated vehicles can be managed, as implied by S9:

> *"How does this relate to requirements? It is even tricky to define what a safety requirement is. For safety analysis, the human aspects are critical input to system design and testing. That is with safety as a purpose of design. In particular, the person in the car. In trucks, it is mainly the safety of other road users. But that is very different from functional safety requirements."* — S9 - AV Engineer

However, it is difficult to clearly define these requirements, as well as architectural decisions, in agile projects, as indicated by S3:

> *"Architectural decisions are taken all over the place. The architect must go around and collect them to raise those aspects that should be treated globally. The decisions now are made differently than they were made before. The design decisions should follow system requirements."* — S3 - AV Engineer

Thus, there is a need to document system requirements, which describe how the different parts of the system under construction will address the stakeholder needs. While these requirements are valuable to manage the knowledge about the system with respect to stakeholder needs and HF, they are not suitable input to agile development work. As S10 implies, one needs to closely investigate collaboration in agile system development to identify system requirements.

> *"To be able to create requirements or needs, one has to understand what is the problem with collaboration today."* — S10 - HF/AV Engineer

From the interviews, we derive the implication of using *system requirements to express the relationship of design decisions to HF knowledge.* In the context of the other implications, we suggest that this implies the need to allow teams to *create system requirements together with the system*, i.e. while developing its software and the corresponding tests, and not before. Requirements would be provided during development (in the form of stories) rather than at the beginning. This sequence allows the requirements to remain up-to-date with the current implementation, rendering them useful for informing future system evolution.

Thus, a general approach that fits our interview data is as follows: teams would run experiments during a sprint and then modify the system accordingly for the next release. They would also, at the same time, describe the updated capabilities of the system and *trace system requirements to related existing/future HF experiments, in order to provide rationales for the decisions.*

This implication is based on *(P1) iterative incremental work.*

Summary and important questions.
As with HF, the implications for RE call for a changed role for requirements engineers. A high percentage of requirements will be discovered and managed just-in-time by agile teams. RE experts, therefore, will increasingly provide infrastructure and coaching, which raises important questions:

[a] How can epics and user stories be positioned as a means to learn rather than to specify?

[b] How can agile teams be enabled to use prototyping to perform HF experiments and discover and manage requirements?

[c] How can system requirements be used to efficiently express the relationship between design decisions and HF in continuous development?

2.5 Validation Study

We validated the results using a questionnaire-based survey in a workshop setup. By presenting the topic to the audience and directly answering their questions, the workshop format allowed us to ensure that participants understood the topic

The questionnaire started with basic demographic questions to assess basic response behavior differences between participants based on their background. We then provided the context, introduced the main topic in the presentation form, and explained the research questions. Next, we explained the research results so that participants could better understand the topic. Keep in mind that the context and description of the outcome of the paper were also provided to participants before the session. Afterward, we asked participants to indicate their level of agreement (on a 5-point Likert scale) with the stated impacts of the properties of agile and HF on AV development. Finally, we asked for the participants' agreement on the implications of the agile way of working, HF, and managing requirements in AV development.

2.5.1 Participants' Demographics

Figure 2.6 depicts the demographic data of the workshop participants. For this survey, participants were invited from different automotive companies and research institutes, mainly based in Sweden. There were 28 participants in the workshop and we asked three basic demographic questions. We did not include participants from the original interview study, to avoid bias. On average, 17 participants responded to each question, and the rest (on average 11) chose not to answer. Subfigure (a) of Figure 2.6 depicts the overall results and shows that the majority (56%) of participants work for automotive OEMs, 7% work for automotive suppliers, 25% work in research institutes, and the rest were from academia.

Overall, the fact that all participants were from Sweden limits the generalizability of the results. However, the survey aimed to evaluate our already identified findings (which were obtained using industry experts in Sweden) rather than arriving at a general conclusion or discovering new implications/properties. At the end of the workshop, we asked the participants if we had missed any critical topics.

2.5.2 Evaluation of Properties of Agile and HF

On the next questionnaire page, we started with RQ2 and explained the properties of agile and HF which can impact AV development. We then asked the participants to indicate their level of agreement with our interview study findings (on the 5-point Likert scale), in order to assess whether the participants identified the same properties as important.

The survey results are shown in Figures2.7 and 2.8 for the properties of agile and HF, respectively. The blue bars on the left indicate the percentage of participants who agreed (light blue) or strongly agreed (dark blue) with the findings. The grey bar in the middle shows the percentage of neutral participants, the light orange bar depicts the percentage of participants who showed disagreement, and the dark orange bar on the right shows strong disagreement.

Figure 2.7 shows that the majority of participants agreed with *(P1) iterative incremental work*. Five participants were neutral, and nobody disagreed with *(P1)*. For *(P2) shifting responsibility to autonomous teams*, 13% of participants were slightly in disagreement. 40% and 20% of participants strongly agreed or agreed, respectively, while the rest were neutral. Fifteen participants rated *(P3) teams responsible for discovering knowledge*, majority of participants showed their agreement (54% agreed and 13% strongly agreed) with *(P3)*. For *(P4) focus on quality in use*, 54% of respondents agreed with the statement. Two participants strongly disagreed, and the rest were neutral.

Most participants indicated their agreement with identified properties for both agile and HF. Thus we can say that our initial impression that these

Figure 2.3: Area of work - (20/28 Respondents)

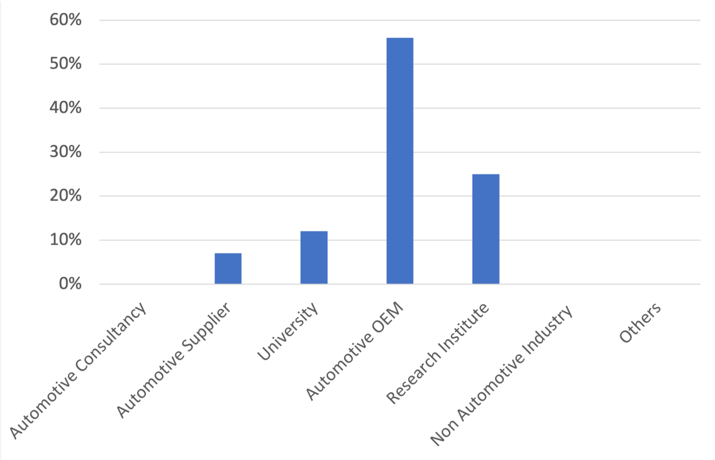


Figure 2.4: Work Perspective -(20/28 Respondents)

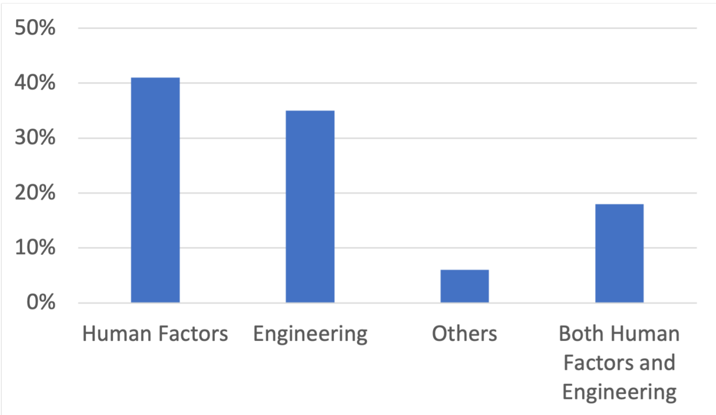


Figure 2.5: Work Experience - (22/28 Respondents)

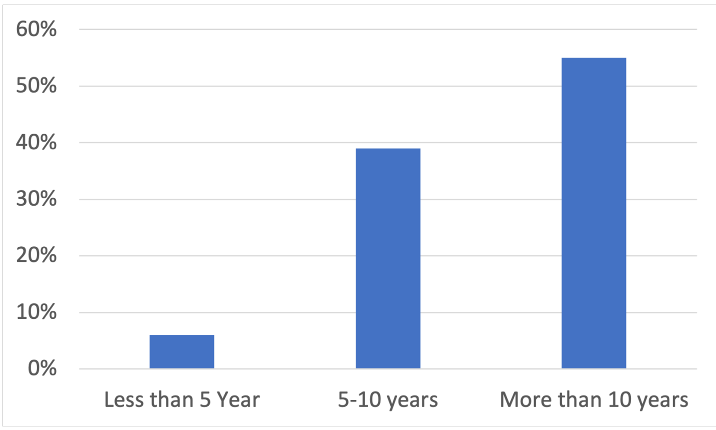


Figure 2.6: Demographics

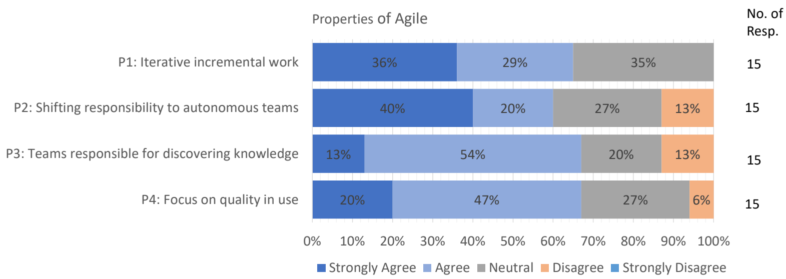


Figure 2.7: Level of agreement regarding the impacts of the properties of Agile on AV development.

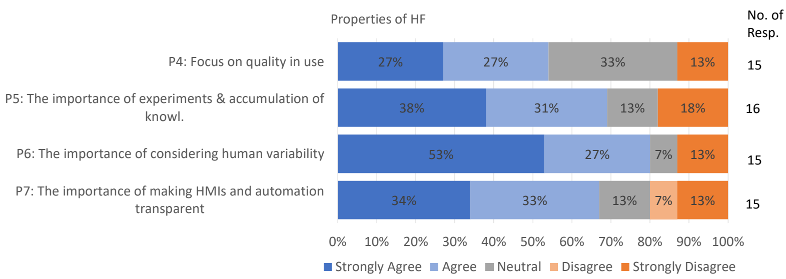


Figure 2.8: Level of agreement regarding the impacts of the properties of HF on AV development.

properties are critical for defining HF requirements in agile AV development is supported by the participants.

2.5.3 Evaluation of Implications

With respect to RQ3, the questionnaire presented Likert scale statements about the implications of combining the relevant properties of HF and scaled agile into the agile way of working, HF, and managing requirements in AV development.

We started with the agile implications, asking the participants to rate their level of agreement for each implication. Figure 2.9 shows the findings for each implication of scaled agile on the agile way of working. For both *(I1) AV developers must run human factors experiments* and *(I2) experiment design & lessons learnt must be created, re-used, and updated efficiently*, 50% of participants showed strong agreement, and 72% in total expressed their agreement with the stated implications. The majority (57%) of participants strongly agreed with *(I3) human factors expertise must be included on the teams*, and 64% of participants agreed with *(I4) the role of suppliers in agile AV development that integrates human factors must be defined strategically.*

Generally, more than 50% of respondents agreed with the stated implications of HF and RE (presented in Figures 2.10 and 2.11, respectively). An exception was *(I6) put questions on teams, not requirements.* An equal number of participants agreed and disagreed; however, as 40% of participants were neutral, there was no clear-cut disagreement. This result suggests that *(I6)* should

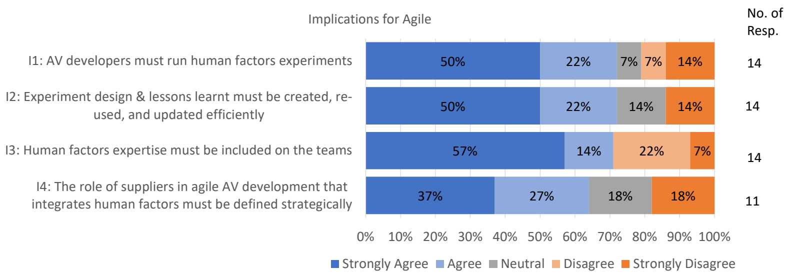


Figure 2.9: Level of agreement for the implications for the agile way of working on AV development.

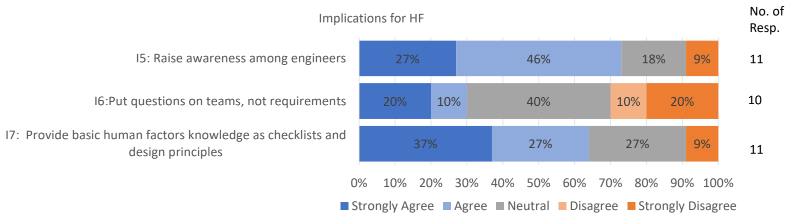


Figure 2.10: Level of agreement for the implications for the HF on AV development.

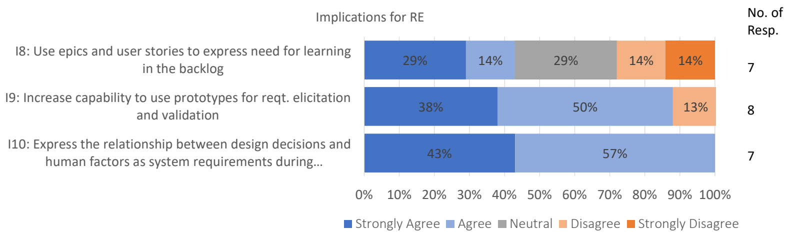


Figure 2.11: Level of agreement for the implications for RE on AV development.

be investigated further. *(I8) use epics and user stories to express a need for learning requirements in the backlog* also showed mixed agreement, indicating the need for extended research on how to represent the need for HF knowledge to AV developers.

For *(I10) Express the relationship between design decisions and human factors as system requirements during development*, all the participants indicated their agreement (57% agreed strongly and 43% agreed).

The results for HF and AV Engineers were similar for most of the questions. However, two HF experts (one with more than ten years of experience and one with less than five) rated the implications for HF very low. On the other hand, all AV engineers rated them highly.

Generally, the majority of participants agreed that all the implications that we derived from the interview notes were relevant and important for bringing HF knowledge into an agile way of working for AV development.

2.6 Discussion

Based on an exploratory interview study with ten experts from the industry and two experts from academia, this paper charts the landscape of human factors (HF) in relation to the agile development of automated vehicles (AVs). We adopted a Requirements Engineering (RE) perspective, since requirements are traditionally the mechanism for notifying automotive engineers about conditions that should be met by their systems as well as capabilities that the systems should possess [172, 202]. Recently, the traditional approach to RE has been challenged by the success of agile methods [150] and their adoption in systems engineering [202].

2.6.1 Implications for Practice

We argue that our findings can provide valuable insights for both HF experts and AV engineers in the automotive industry. Particularly, our findings on how to communicate HF requirements during the development should be useful for guiding practitioners. Previous work shows how crucial it is to incorporate HF into the RE process. Our results support this finding, but also identify that actually doing so is more difficult with agile development. We also acknowledge that areas of AV development that are relatively new, such as AV functionality development based on artificial intelligence (AI, including machine learning) [203], may require specific focus in the integration of HF. Otherwise, the impact on humans (drivers, occupants, and surrounding traffic) of the (typically highly data-driven [204]) AI approaches can easily be overlooked.

New roles for HF and RE. Our study took place at a pivotal time in the automotive industry. The automation of driving tasks is proceeding rapidly, adding significant complexity to automotive systems. Automotive companies are transitioning to agile approaches in order to enable shorter development times despite this increased complexity. We were surprised by the strong focus on agile methods in all of our interviews. At the same time, the role of HF knowledge and requirements becomes less clear in the agile setting. RE appears to play a smaller role, partially replaced by increment planning and backlog management. Moreover, RE often focuses on specific technical aspects such as functional safety.

This adds to another trend: Automated systems development often prioritizes the technology, without much consideration of HF [205]. In fact, HF is rarely considered in the early phases [206], although our results highlight the importance of doing so. We suggest that this change be enacted through RE, which may help to identify a role for HF in organizations that seek "to inform AV development by providing fundamental knowledge about human capabilities and limitations throughout the life cycle so the product will meet specific quality objectives".

We also suggest further refining the role of RE so that it can better adjust to the needs of agile development, while also improving the support required to integrate HF knowledge into agile development. We envision a role that is less prescriptive and focused on setting requirements for developers, and instead more supportive: enabling developers to explore, document, and reuse requirements-related knowledge. This role will be particularly useful for identifying HF knowledge (e.g., results from experiments) that is no longer valid due to system/software changes—thus, calling for new experiments.

Testing and experiments. The field of HF highly prioritizes experimenting and testing. With agile's fast, iterative way of working, there is a need to test regularly and quickly while keeping accumulated knowledge in mind. In contrast with fields such as software testing, in which tests are very formalized and mature, HF has a few substantial challenges:

[a] In the context of AV development, formalization of (most) experiments is not mature enough.;

[b] Humans are adaptive and unpredictable, making the formalization of experimental protocols and passing thresholds difficult.

Thus, we encourage future research to improve the integration of tests and experiments from an HF perspective into AV development, keeping accumulated knowledge and ensuring that HF experts are part of the experimental setup.

2.6.2 Implications for Research

Common understanding of terms. Definitions provided by our interviewees differed substantially, not only between the HF experts and the AV engineers, but also among the HF experts. This ambiguity identifies a critical communication gap [185]. In this work, we propose a slightly refined definition of HF, geared towards the development of AVs (see Definition in Section 2.4.1) and relating specifically to the essential phases of system engineering. Our results, however, call for future research to achieve an aligned understanding of HF and related concepts through all the systems engineering disciplines involved in AV development.

Raise awareness and develop mindset in agile engineering. It is important to raise awareness and develop an HF-friendly mindset in development teams, in order to improve the communication of HF requirements. A suitable mindset would consider not just the user experience or HMI, but all aspects of human interactions with a system. Many HF experts agree with this assessment [57, 118, 207]; however, to our knowledge, there is little awareness in systems and software engineering, areas where research is highly encouraged. Awareness could be raised by training engineers in multidisciplinary work so that it becomes easier to integrate HF experts in agile teams (as in I3). In addition, research is needed to determine how to increase the ability of agile teams to manage open questions (see I6) as well as their experimentation infrastructure (see I2) [101, 117, 208].

Need to develop and empirically evaluate new approaches to manage HF knowledge. This qualitative study presents several implications which human factors experts, AV agile teams, and requirement engineers can adopt to communicate the knowledge of HF during the agile AV development process.

> *"Or should the team explore the HF? But then we would need a really good model that the team can explore and a lot of expertise that the team can assess. On the high level, we may only have a very crude understanding."* — S8 - AV Engineer

In particular, the need to have AV developers participate in (or even run) HF experiments *(I1)* requires the attention of researchers. In continuous software development, there is a trend towards data-driven decision making and experimentation [101, 208, 209, 210, 211].

It could be exciting to compare such experiments on variants of software with HF experiments and investigate possible synergies, which might provide insights into how HF experiments can be integrated into the fast-paced agile development environment.

In Summary. We believe that our exploratory study provides a foundation for future research that could improve RE in AV development, as well as refining communication about the HF perspective within the agile way of working. Both HF and RE experts should re-interpret their roles, enabling and facilitating agile teams seeking knowledge—instead of providing comprehensive and detailed knowledge themselves. We anticipate that future research in agile work will formalize ways to manage HF experiments, as well as their results, efficiently. Being able to keep knowledge across design cycles will contribute to a mature synergy between these formerly disparate ways of working.

2.6.3 Discussion of quality

Our particular epistemological stance (critical realism with influences from pragmatism and constructivism) and choice of method for data collection and analysis also influence the discussion of research quality. In particular, the predominant positivist approach to validity—in terms of construct validity, external validity, internal validity, and generalizability—fits this study poorly. Instead, for this qualitative inquiry, we followed advice from Leung [212], discussing validity, reliability, and generalizability in terms that are a better fit in the context of this study.

2.6.3.1 Credibility

The credibility of our study is supported by the diverse background of the researchers and our ability to interview the leading experts in the domain. This was the first joint interview study of the authors, which allowed us to bring in complementary perspectives by recruiting interviewees from each author's personal network. Further, we asked each interviewee to suggest additional candidates to mitigate a potential selection bias. By inviting such a diverse group of interviewees and collecting their potentially contradicting perspectives on the topic, we had to challenge and overcome pre-conceptions. We described our background assumptions in detail in Section 2.2 and challenged them throughout the analysis of our data. This approach has led us to construct our mental model about HF in agile system development. For example: We learned that most participants had only recently moved to agile development approaches. We understand that in such approaches, teams are expected to generate knowledge as needed to implement features. We learned that high-quality HF knowledge stems from experiments and relies on a high level of HF expertise. Thus, we conclude that HF expertise must be included in agile teams to facilitate agile system development that includes HF knowledge, an implication that fits the data from our interviews well and resonated also with participants in the validation workshop.

2.6.3.2 Resonance

Through our data collection and analysis, we aimed to establish resonance, e.g., by asking for clarification when we felt that our assumptions were challenged. One such example occurred when we learned that something that was described as relatively easy to accomplish by one participant was described as very difficult by another participant. We learned that in the first case (driver monitoring), a rich set of models, checklists, and design principles exists, which was missing for the second (monitoring of cyclists). The lack of these resources made communication considerably more difficult in the second case. Thus the apparent contradiction was explained, providing us with a richer understanding of possible challenges. Implication I7 (provide basic HF knowledge as checklists and design principles) and our definition of HF in the context of the design and development of automated vehicles both reflect the lesson learned.

2.6.3.3 Usefulness

We believe that our study, albeit a preliminary exploration, is significantly useful. Integrating the design cycle into our working definition of HF in the context of the design and development of automated vehicles is one example of its utility, since the new definition makes it possible to specify where in the design cycle HF knowledge becomes useful. In addition, we believe that our implications provide useful knowledge to those who are tasked with the design of methods and tools for development, as well as to HF experts who aim to increase their impact on AV development. We derive confirmation of these conclusions through feedback received after presenting the study results to the participating companies.

2.6.3.4 Transferability

Case studies aim to investigate a phenomenon in depth within its natural context. They do not generally aim for generalizable findings in the same way, as for example, an experiment would. Instead, as qualitative research, case studies should lead to theoretical generalizability: concepts that are transferable in principle. Wieringa and Daneva, for example, highlight the ability to provide a causal or structural (architectural) explanation as a theoretical generalization [213], which then can be transferred to other contexts.

In our study, we provide such explanations through the properties of agile and HF, and the implications for agile, HF, and requirements. Figure 2.2 provides an overview of these findings in a qualitative model, specifically relating the concepts (implications) to assumptions (properties) that we have identified through our interviews with experts. In this way, we provide both causal explanations (properties of agile and HF generate implications) and structural explanations (integrating HF into large-scale agile system development will benefit from addressing implications in the area of agile, HF, and RE). This knowledge is transferable, allowing experts from different domains to judge how our concepts apply to them.

Our results stem from the automotive industry, including considerations of automated cars and trucks, and should be applicable to other cases in that domain. We further believe that our concepts are transferable, not only beyond the national AV hotspot where we recruited most of our interviewees, but also to other automated vehicles such as aircraft or ships. It would be harder to transfer beyond the realm of automated vehicles, and even more so if no physical product is created. For example, we would assume that a web application will have very different constraints on prototyping and testing.

2.7 Conclusion

In this paper, we present an exploratory, qualitative inquiry into how to manage HF knowledge during AV development. Our investigation revealed the fundamental role that large-scale agile development plays in the automotive sector. From our data, we derived a working definition of Human Factors for AV development, discovered the relevant properties of agile and HF, and defined implications towards agile ways of working, managing HF knowledge, and managing requirements.

Experiments and experience are integral parts of HF. It is a challenge to fit HF knowledge (and the corresponding requirements) into the agile way of working that the automotive industry is moving towards, with its fast pace of change.

As our properties and implications reveal (e.g., P3 and I3), there is an increased need to bring HF expertise to the development teams, caused by the team-based approach and team responsibilities inherent in agile AV development. The paucity of HF experts and the intermittent need for HF expertise in many agile AV development teams makes the inclusion of HF expertise in teams a challenge. In addition, fast, iterative increments do not typically allow time for the rigorous experiments that HF experts may need in order to ensure user-centered quality. In general, reflections from this study and responses from (especially but not exclusively) the HF experts indicate that it is important to push for an HF culture in companies, in the same way that many automotive companies have a safety-first culture. Why not safety and human factors first? Our exploratory study, admittedly limited in scope, relies on 12 interviewees, mainly recruited from a national hotspot of AV development. We believe that our study demonstrates the relevance of this research topic, as well as the value that additional interviews (beyond the scope of this study) could provide.

Chapter 3

Paper B

Continuous Experimentation and Human Factors An Exploratory Study

Amna Pir Muhammad, Eric Knauss, Jonas Bärgman, Alessia Knauss

In the International Conference on Product-Focused Software Process Improvement, 2023.

Abstract

In today's rapidly evolving technological landscape, the success of tools and systems relies heavily on their ability to meet the needs and expectations of users. User-centered design approaches, with a focus on human factors, have gained increasing attention as they prioritize the human element in the development process. With the increasing complexity of software-based systems, companies are adopting agile development methodologies and emphasizing continuous software experimentation. However, there is limited knowledge on how to effectively execute continuous experimentation with respect to human factors within this context. This research paper presents an exploratory qualitative study for integrating human factors in continuous experimentation, aiming to uncover distinctive characteristics of human factors and continuous software experiments, practical challenges for integrating human factors in continuous software experiments, and best practices associated with the management of continuous human factors experimentation.

3.1 Introduction

In today's fast-paced software development environments, characterized by competitive and unpredictable markets, there is a need to deliver and improve products rapidly [214]. This urgency is intensified by complex customer requirements and rapid technological advancements. Consequently, many software companies have embraced or are transitioning toward continuous experimentation [215, 216].

Continuous software experimentation [1] involves iteratively gathering user feedback and observing user interactions[117]. With the growing significance of software in complex and automated systems, continuous experimentation has become increasingly prevalent across various industries. These systems require robust and continuously evolving software [217]. Researchers have acknowledged that the design for such systems is inherently complex and that a more comprehensive understanding of the real world can be achieved by actively looking at the system from a human factors perspective and not only a technical perspective [119, 218].

In order to ensure the effectiveness, safety, and reliability of systems, particularly complex software systems, it is desirable to provide more holistic knowledge on human factors in continuous experimentation. Especially for safety-critical systems, a human factors perspective may provide crucial in-depth insights. Therefore, integrating human factors experimentation into the continuous experimentation process promises to be a game changer [22, 113]. Human factors refer to the various aspects of individuals, including their physical, cognitive, social, and emotional elements, all of which can significantly influence their performance and interactions with systems [11]. Human factors experiments prioritize studying user behavior and involve experiments with humans as participants [219]. We acknowledge that the concepts of continuous software experimentation and human factors experiments overlap to some extent (i.e., the latter can be a component of the former, and vice versa), but in this study, we discuss them as separate entities as they come from different domains and are likely to complement each other. However, to understand whether HF experiments fit the continuous software experiment practices, one needs to understand in detail where they differ, where they overlap, and in what they can be integrated.

While the significance of human factors has been widely recognized [9, 59] and continuous software experimentation methodologies are widespread in industry and have received extensive research attention [220, 221], there remains a research gap when it comes to incorporating human factors experiments into the well established continuous software experimentation processes [22]. Consequently, further investigation is required to bridge this gap [222].

This research aims to address differences, associated challenges, and best practices for integration of human factors experiments within the context of continuous experimentation. The following research questions (RQs) are used to guide our research:

RQ1: What are main differences when comparing human factors experiments with continuous software experimentation?

[1]Key terms of this study are defined in Table 3.1.

RQ2: What are main practical challenges when managing human factors experiments in continuous software experimentation?

RQ3: What are best practices for managing human factors in continuous experimentation?

The findings for RQ1 reveal that while both human factors and software experimentation emphasize the significance of understanding user behavior and needs, they differ in their approach. RQ2 highlights the challenges in managing human factors experiments, pointing to complexities like GDPR compliance, data collection issues, additional costs, and an industry scarcity of experts. RQ3 focuses on best practices in this domain, emphasizing the need to prioritize research based on product timelines, invest in actionable metrics, maintain robust experimental infrastructure and documentation, and including or transferring human factors knowledge.

The rest of this paper is structured as follows: We start with an overview of definitions for key terms used in this paper in Section 3.2, which covers background knowledge and related work as well. Section 3.3 presents the research methodology, and Section 3.4 outlines the findings. Section 3.5 presents the discussion and potential threats to validity. Finally, Section 3.6 concludes the paper.

3.2 Background and Related Work

Key terms of this study can be interpreted differently depending on the domain. Hence, for the scope of this study, we use the definitions provided in Table 3.1.

Continuous Software Experimentation. Agile development methodologies have gained widespread popularity in software development due to their iterative and collaborative nature [226]. These methodologies emphasize continuous experimentation, which involves constantly testing and validating hypotheses to make data-driven decisions throughout development [216]. This approach has proven effective in optimizing software products and services.

Continuous experimentation is primarily applied in web-based systems, allowing developers to analyze and deploy changes based on real-world data and user preferences, rather than relying solely on simulations or the opinion of the highest-paid person's opinion (HiPPO) [227]. Leading technology companies like Microsoft, Google, Facebook, and Booking.com utilize online controlled experiments, also known as A/B tests, to evaluate the impact of changes made to their software products and services [220, 221, 228].

Despite the numerous advantages of wide-ranging continuous software experimentation, there are still several challenges that need to be addressed during its implementation. Some of the major hurdles include cultural shifts within development teams, slow development cycles, product instrumentation, and the identification of appropriate metrics for measuring user experience [229, 230]. Rissanen and Münch [231] confirmed these challenges and also found that capturing and transferring user data becomes challenging due to legal agreements.

Human Factors and Experimentation. By including human factors experiments from the outset, it becomes possible to ensure system reliability

Table 3.1: Definitions of key terms used in this study

Term	Definition
Continuous (software) experimentation	An approach to support software development, where research and development activities are guided by iteratively conducting experiments, collecting user feedback, and observing the interaction of users with the system or services under development. The goal of continuous software experimentation is to evaluate features, assess risks, and drive evolution [117, 216, 221].
Human factors in development	The field that aims to inform developers by providing fundamental knowledge about human capabilities and limitations throughout the design cycle so that products will meet specific quality objectives. These capabilities and limitations include cognitive, physical, behavioral, psychological, social, effective, and motivational aspects [11, 223].
Human factors experiments	Investigations that focus on how human capabilities and limitations affect specific quality objectives during the interaction between humans and the system, service, or product under development. Thus, humans are part of human factors experiments and their behavior and perception/opinions (of, e.g., the system, service, or product under assessment) can impact the result and consequently the design of the system [224, 225].
Continuous human factors experimentation	An iterative approach in software development that evaluates how human capabilities and limitations impact specific quality objectives during user interactions. It involves ongoing experiments, user feedback, and observations to inform the design process and enhance user experience.

and evaluate the system considering real-world human constraints [224]. Human factors experiments aim to understand how people interact with technology, products, and systems to optimize usability, user experience, and overall performance [11]. They commonly evaluate aspects such as user interface design, cognitive workload, situation awareness, and user behavior [232, 233, 234, 235].

Continuous Human Factors Experimentation. In terms of testing and experiments, there have been some initial efforts to integrate usability testing and user-centered design practices into agile development, like for example the approach proposed by Nakao et al. [236] to incorporate usability testing throughout the agile development process. Despite these efforts, research has emphasized the need for new processes and tools that empower practitioners of human factors to promote usable and effective products in the agile development environment [22] and the integration of human factors into the well established continuous software experimentation practices used in agile development [222].

Note that our research does not center around the impact of human factors on employees or developers involved in the development processes, as mentioned in [237]. Instead, our focus is primarily on the product itself. By conducting and analyzing a series of semi-structured interviews, we aim to explore the integration of human factors experiments within the context of continuous experimentation in software development.

3.3 Methodology

Sampling: We conducted interviews with eight professionals (P1-P8). We aimed for a broad sample of expert participants with high experience in human factors, continuous experimentation, ideally in both fields. This criteria however limits the number of available subjects. Thus, we accepted lower participant numbers than initially planned and focused on interviewing a smaller selection of leading experts in their respective fields for this exploratory study.

We focused on recruiting industry participants from renowned organizations such as Microsoft. Targeting those known for their impactful success stories, to ensure a significant impact and obtain high-quality input. Our academic interviewees have extensive experience collaborating closely with industry, and their credentials include thousands of citations (h-index > 35 in four cases), providing them with a good overview of practices in the field that supports our exploratory study goal.

Table 3.2 presents each participant's role and experience level.

Data Collection: To gather comprehensive information for our study, we used a qualitative study design inspired by Maxwell [238]. Our data collection involved conducting a series of semi-structured interviews, following a predefined set of open-ended questions while allowing flexibility to include

Table 3.2: Interviewees' roles and relevant work experience (Experience level: Low= 0–5 years, Medium=5–10 years, High= More than 10 years).

ID	Role	Main Domain	Continuous experimentat. experience	Human Factors experience
P1	SE Researcher	Academia	High	Low
P2	Human Factors Researcher	Academia	Low	High
P3	Human factors Engineer	Industry	High	High
P4	UX Expert	Industry	High	High
P5	Data Scientist	Industry	High	Low
P6	SE/Human Factors Researcher	Academia	High	High
P7	CS Researcher and IT Consultant	Industry & Academia	High	High
P8	Human Factors Researcher	Academia	Low	High

additional follow-up questions when necessary. The interview questions used can be found here.

The interviews were conducted online through Zoom, with each session lasting around one hour. We obtained permission from the interviewees to record the sessions, which we later transcribed and anonymized for analysis.

The interview questions were organized into three main categories. The first set aimed to collect demographic information from the interviewees, as well as confirming their experience working with continuous experimentation and human factors. The second set focused on exploring the management of experimentation in both software and human factors contexts. We used these question to get a better understanding of the participants background, how and which experiments they use and generally of the topic under study. Finally, we asked specific questions related to human factors in continuous experimentation. We used the entire data in our analysis and to answer our research questions.

We initiated each interview by providing a brief overview of the study to establish a shared understanding and create a comfortable environment. We also presented the basic terms and definitions relevant to the study topic, seeking agreement from the interviewees. This approach aimed to establish a common foundation for our discussions, minimize potential confusion, and ensure a consistent standpoint when gathering participants' perspectives. Notably, all participants expressed agreement with our definitions, offering no suggestions for improvement or indicating any discrepancies between their own understanding and our proposed definitions as outlined in Table 3.1.

Data Analysis: For the qualitative analysis, we employed the thematic analysis approach [239] to identify themes and analyze the content. This approach consists of six key steps. Initially, we comprehensively reviewed all the interview notes and generated research-related memos. To facilitate the process, we employed Nvivo initially and later transitioned to using the Miro board for enhanced visualization. These tools allowed us to assign codes or labels to the text. Through an iterative process, we refined the coding scheme to uncover significant ideas and viewpoints. The codes were then analyzed and grouped together to identify common patterns, thereby defining the themes. Subsequently, we thoroughly reviewed and verified the themes that emerged from the coding process, ensuring clarity, consistency, and addressing any ambiguities, contradictions, or omissions.

3.4 Findings

We present our findings for each research question with primary themes and their related sub-themes. Figure 3.1 gives an overview of the main themes.

RQ1: What are main differences when comparing human factors experiments with continuous software experimentation?

F1.1: Contextual Factors

Human Behavior vs. Technical Aspects: Both software developers and human factors professionals recognize the importance of an intuitive user perspective. They acknowledge that users have varying levels of technical proficiency and may not be inclined to explore complex features. However, human factors experts go a step further by emphasizing the need to understand the underlying reasons for potential user challenges. For example, these challenges could include over-trusting software or avoiding it altogether due to fear or apprehension. To address these concerns, human factors experiments are conducted to gain insights into human behaviors, needs, and experiences. These experiments prioritize the user perspective and strive to optimize user satisfaction and safety. On the other hand, software experiments typically have a more technical development-centric focus. This discrepancy in approach highlights the importance of adopting a human-centric understanding of user behavior and needs, which may differ from the primary focus of developers on technical functionalities.

> *"They can develop and test and design and maybe it doesn't need to involve human, then it works fine, as soon as you add human, a whole set of questions & requirements come into place which needs to be considered."* — P8

Human factors experts primarily focus on observing and analyzing human behaviors to collect data using different interaction metrics. Such an environment poses inherent challenges due to numerous uncontrollable variables at play. For instance, humans exhibit a learning effect that can significantly impact the experimental results. Moreover, interpersonal communication and feedback loops among participants may also influence their responses to the experiments.

Conversely, continuous software experimentation primarily focuses on monitoring system behavior rather than directly observing human behavior. Such experiments collect data from performance indicators, system logs, issue reports, or user interactions documented by the software. They are often conducted

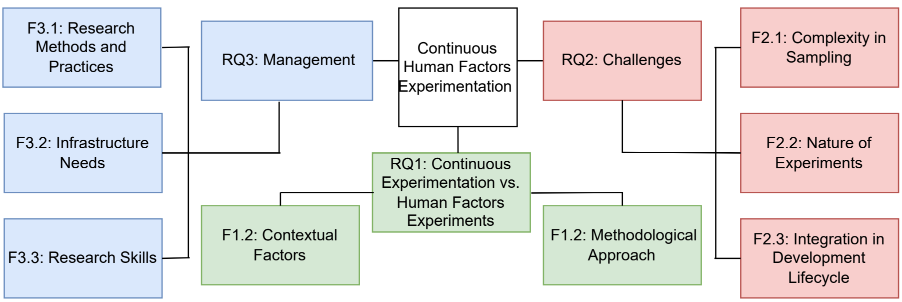


Figure 3.1: Overview of key high-level themes identified from the interview analyses.

under controlled conditions, emphasizing variables like reaction time, resource usage, scalability, or software stability. We believe that these differences are brought to a point by the following exemplary quote:

> *"The main difference between human factor and traditional experiments, for instance, is that humans have much more of a learning effect."* — P7

F1.2: Methodological Approach

Diverse Approaches in Experimentation: The methodology for both human factors and continuous software experiments varies depending on the nature and scope of the feature being tested. Various techniques can be employed for both software and human factors experiments.

> *"If it's a very small audience, then product teams can also choose actually to do some surveys and interviews they invite customers in. So it really depends on like what is the scope of the feature that you're testing."* — P5

While some methodologies, such as surveys and interviews, can be utilized for both software and human factors experiments, there are some notable differences in how the results are analyzed and interpreted. We found that while A/B experimentation is a dominant method in continuous software experimentation, it is often only one of many methods used in human factors experiments.

Qualitative and Quantitative: Much like software experiments, human factors involve qualitative and quantitative data analysis. However, the analysis of human factors experiments leans more towards qualitative methods due to the complexity of measuring and interpreting human behavior. Therefore, conducting effective human factors experiments necessitates practitioners with a strong foundation in qualitative methodologies and empirical work involving human participants. Such practitioners are able to capture the rich and nuanced aspects of human behavior and user experience. In contrast, continuous software experiments often adopt a more quantitative approach, aiming to establish causal relationships between independent and dependent variables, allowing for statistical analysis. That said, a substantial part of human factors experiments still involve collecting quantitative data, such as eye-tracking data and performance data (e.g., in the automotive domain in terms of measures of lane keeping, time gaps, etc., or task completion times considering desktop software tools).

> *"If you have a background in quantitative experiments with technical systems, I would think you cannot do [human factors experiments] in a good way. You need some kind of background in doing empirical work with humans."* — P6

RQ2: What are main practical challenges when managing human factors experiments in continuous software experimentation?

F2.1: Complexity in Sampling

Controlled vs. Uncontrolled Variables: One aspect is the presence of a higher amount of uncontrolled variables in human factors experiments.

Numerous contextual factors cannot be fully controlled, which poses challenges in ensuring comparability and measuring variables. Lack of control over contextual factors also complicates the analysis, as there may be numerous variables that cannot be fully controlled or accounted for in the experiment.

> *"The other issue is control. I think you will look at situations where there are just a lot of context factors, there is just no way to control everything."* — P1

Statistical Analysis: One challenge lies in the statistical analysis of the data. In certain cases, conducting a rigorous statistical analysis may not be feasible due to the nature of the human factors experiment. For instance, the research goal might involve observing how people react in a particular situation without quantifiable metrics, so conducting a traditional statistical analysis becomes challenging.

> *"It might not be possible to do a proper statistical analysis because you might want to expose people to a certain situation and see what happens."* — P2

Participant Scarcity: Another challenge in human factors experiments is the limited availability of participants. Getting enough people to participate can be difficult, and the scarcity of eligible participants further complicates the process. In contrast, continuous software experiments, especially those conducted online, can be performed on a larger scale. While involving as many participants as possible is generally advised, practical limitations may hinder this goal.

> *"Often these studies are fairly small regarding the number of subjects."* — P6

F2.2: Nature of Experiments

Personal Information and GDPR Issues: When conducting experiments, the collection of personal information can be crucial for understanding human behavior and software performance. In experimental research, collecting personal information is pivotal for understanding both human behavior and software performance. This is particularly evident in human factors experiments, where insights into how individuals from varied backgrounds interact with technology are essential. However, collecting this in-depth personal information presents challenges, mainly due to privacy and ethical issues. The requirements of GDPR regulations amplify these concerns, necessitating meticulous attention. While software experiments might occasionally need such information, the emphasis is much greater in human factors experiments.

> *"It is a bit hard. Like with the GDPR and everything. How to store stuff actually? It makes it a bit more complicated."* — P4

Prototype vs. Real Environment: Our interviewees mentioned that, although experiments are typically carried out using prototypes or simulators, human factors experts also advocate for conducting experiments in the actual environment where the product will be finally be used. Experiments conducted in real environments offer a more realistic and authentic representation of how participants interact with the product or system in their natural settings. Unlike prototype experiments, where external factors can be tightly controlled, real environment experiments expose participants to multiple variables and contextual factors that can significantly impact human performance and behavior.

> *"Having design prototypes is one approach so that people get the vision behind. But testing in real cars, it makes it so difficult, which is, but also important, to go in that direction or to get more research done."* — P3

Expensive: Human factors experiments are often perceived as more costly compared to continuous software experiments. This perception stems from the direct involvement of real humans participating in real-time scenarios. For instance, experts in human factors often need to recruit participants for their studies, compensating them for their time and effort, which can be a significant expense. On the other hand, many continuous software experiments can gather data online, reducing the need for physical presence and direct human interaction, and direct payment. While continuous software experiments do have associated costs—such as development, deployment, and server infrastructure—these expenses are generally lower than those of human factors experiments.

> *"We have to pay for this for facilities, we have to pay participants because we get people from the real world, and the preparations is quite prolonged."* — P8

F2.3: Integration in Development Lifecycle

Execution Time: Managing and executing human factors experiments in agile development can be challenging due to their inherent time-consuming nature. Unlike continuous software experiments that typically run for at least a week, human factors experiments often require more time to obtain meaningful results. The duration of such experiments is influenced by the desired change in a metric being measured. Obtaining timely results from human factors, that can be integrated into ongoing projects without significant delays can be difficult, especially in agile, short sprint-based, work flows.

> *"You do a sprint and then you need results to run it and assume you need these kind of results quickly. So not in three months. And that's, I would say that's the problem for integrating these kind of things."* — P2

Infrastructure Needs: One challenge involves obtaining the necessary tools and setup to conduct the desired tests. Ensuring that the basic infrastructure is in place to facilitate the experiments can be a significant hurdle.

> *"If there's getting the right tools and right setup, like the basics in place to even be able to test what you wanna test. That could be a challenge."* — P4

Too Few Human Factors Experts: Many companies struggle with insufficient human factors expertise and limited resources, which can hinder their ability to improve user experience. This deficiency often leads to a few outliers (or even the development team itself) having a disproportionate impact on the final product design. This concern arises from the fact that there are too few human factors experts available, which limits comprehensive evaluations and increases the risk of biased results.

> *"So I think that's what, what other companies are lacking actually: Enough human factors, people doing that kind of work."* — P3

Lack of Motivation: Another challenge is that many individuals with a technical mindset often overlook the importance of understanding human behaviors. This lack of motivation can hinder the collection of relevant data and make it difficult to address the complexities involved in studying human subjects.

> *"How can you influence people? I think that's the number one thing."* — P1

RQ3: What are best practices for managing human factors in continuous experimentation?

F3.1: Research Methods and Practices

Prioritizing Hypotheses/Research Questions: Prioritizing research questions and hypotheses based on the product timetable and development sprints is a crucial aspect in agile development. By identifying the experiments that have the most impact on design decisions and user experience improvement, organizations can allocate resources efficiently and gain valuable insights.

> *"The number of experiments that you can do is basically infinite. So the hardest part in running experiments is how do I prioritize running the most valuable experiments first. And, I think that's where many companies struggle."* — P7

Metrics and Measurement Instrumentation: Based on our interviewees, to enable informed decision-making it is essential to invest in the development of meaningful metrics that align with the desired outcomes. While simple interaction metrics like clicks or selections are useful, it is important to go beyond them and capture success metrics related to user sessions and product features. As one of our interviewees pointed out, the value of experiments ultimately relies on having good metrics and making significant investments in their development. Without such metrics, experiments become less valuable as they fail to provide actionable results for decision-making. It was also emphasised that developing and validating such metrics can (and must be allowed to) take substantial time.

Another critical aspect is the measurement of various metrics that provide insights into different aspects of the product under evaluation. It is worth noting that interviewees stressed the significance of using proper measurement methods to obtain valuable results for making informed decisions. To measure different aspects of the product or system being evaluated, multiple metrics should be computed simultaneously. These metrics should align with the goals of the experiment and help determine what is reasonable to measure and what constitutes a good outcome.

> *"But at the end of the day about experiments, it all boils down to metrics. If you don't have good metrics and you don't invest significantly into metrics, your experiments will not be valuable."* — P5

Results and Lessons Learned: When determining whether to reuse or evolve experiments, the organization may take several factors into consideration. These factors include the importance of the findings, potential influencing factors, and information indicating changes in the validity of previous results. The relevance of the results and their impact on decision-making are carefully evaluated when planning subsequent experiments. It was also mentioned that the decision to reuse experiments is often driven by the interest and initiative of individuals involved in the projects, rather than being a formalized process.

> *"There are sometimes factors that are influencing what's factors that may confound the outcomes from one experiment such that we need to rerun it in order to make sure that the thing is still true."* — P7

F3.2: Infrastructure Needs

Experimental Setup: The infrastructure should support the setup and integration of different components required for the experiment. This includes ensuring that the necessary tools and setups are in place to conduct the experiments effectively. It may involve creating prototypes, simulating scenarios, or integrating various hardware and software components to enable the desired testing environment. Careful planning of the experiment is crucial.

> *"If there's getting the right tools and getting the right setup or the right HMI, like the basics in place even to be able to test what you wanna test."* — P4

Traceability and Documentation: Maintaining traceability and documentation throughout the experimental process is important. This includes preserving initial design proposals that led to the ideas being tested. Having a clear traceability trail helps in understanding the decision-making process during the experiment and provides valuable insights for product teams. Utilizing an experimentation platform that incorporates this traceability is essential.

> *"So having some traceability on the decisions that led to what is being tested would be very helpful, I think, for product teams. And that should be part of the experimentation platform."* — P5

Collaboration and Management Support: Our interviewees highlighted that infrastructure should facilitate collaboration among different teams involved in the experiments. It should provide a platform for coordinating activities, managing participants, and ensuring the smooth execution of the experiments. Additionally, management buy-in, support, and drive are also important factors to overcome obstacles and successfully implement the infrastructure needed for human factors experiments.

> *"Main obstacle is kind of like management, high management buy-in, and support and then like knowledge on how to design and collect it. So, to me, infrastructure would be something they [practitioners] would know how to solve that."* — P6

F3.3: Research skills

Roles and Responsibilities: Our findings indicate that experiment management becomes a collaborative effort within cross-functional teams in an agile environment. These teams typically include data scientists, engineers, product managers, program managers, and user researchers. Our findings also highlight the pivotal role of data scientists in continuous software experiments and the need for technical support from engineers in human factors experiments. Moreover, considering a single role for responsibility, product managers are crucial in deciding which experiments to run and ensuring that relevant metrics are effectively measured. We learned that while the responsibility for managing continuous human factors experiments can be shared within a team or primarily held by the manager, it is crucial to recognize that specialized knowledge and expertise are often necessary. Having human factors specialists in human factors experimentation can greatly benefit the planning and management of human factors experiments. Human factors specialists bring specialized knowledge and expertise in research methodology, data analysis, and experimental design to guide the team and ensure precise and accurate experiments.

> *"I really think that it should be less of a single responsibility and more of a team responsibility."* — P7

Knowledge and Training: A solid foundation of knowledge, theory, and models is essential to design and evolve effective human factors experiments. Furthermore, establishing an infrastructure to disseminate this knowledge and provide comprehensive training to researchers and teams is crucial. Agile teams can conduct human factors experiments with appropriate training and methodologies.

> *"A bit with training. If you follow a specific procedure, then I think it's not a problem."* — P4

The training should cover experimental design, research methodology, human factors principles, biases, usability evaluation methods, and research methods. Although individuals inherently possess some understanding of human behavior, training will help broaden their perspective.

3.5 Discussion

Continuous experimentation for web-based systems has received extensive research attention [220, 221], however, the human factors aspect remains relatively underexplored. This study explores the idea to bridge this gap by discussing the integration of human factors experiments with continuous experimentation. This promises to enable continuous experimentation even in the domain of safety critical systems to a larger extent. Integrating human factors experiments into continuous experimentation presents both benefits and challenges [240]. For instance, these experiments can shed light on usability, user experience, and decision-making [224]. Yet, they also pose challenges, such as the need to execute experiments in real environments with real human participants [241].

We confirm challenges highlighted by previous studies [229, 230, 231] that have investigated challenges in continuous experimentation in general (e.g., cultural shifts and appropriate identification of metrics) also for the integration of human factors into continuous experimentation. On top of that, our findings introduce additional complexities when human factors are integrated into the mix.

Moreover, our findings indicate that the integration of human factors in continuous experimentation is currently lacking. One of the contributing factors to this gap is the shortage of human factors experts available to collaborate with teams engaged in continuous experimentation [223]. While these teams conduct experiments tailored to their specific system components, they often lack input from human factors specialists. Another factor is the usually higher complexity of human factors experiments. On the fast pace of continuous experimentation, this affects options for data collection and appears to cause human factors experiments leaning towards qualitative data collection in this context.

To effectively integrate human factors experiments into continuous experimentation, companies should consider including human factors experts within teams and raising awareness among developers about the importance of incorporating human factors. The successful execution of human factors experiments by teams requires developers to be skilled in empirical study methods, enabling them to conduct impactful human factors experiments.

Threats to validity:

The interdisciplinary nature and vast scope of the fields involved introduces a threat to **Construct Validity** in that various definitions exist for the same terms, such as "human factors". Consequently, different individuals may have different interpretations. We have included clear definitions of the key concepts in interviews and report to mitigate this threat and ensure a common understanding of the fundamental concepts used in this study. Additionally, experienced authors were involved in the study to address the risk of construct validity. Their expertise assisted the first author in developing an interview guide that effectively aligned with the study's research objectives. For **Internal Validity**, we implemented measures to reduce bias and confounding variables, such as having multiple authors conduct each interview to minimize personal bias. Due to the specialized scope and high demands on participant expertise (human factors and continuous experimentation), we had to rely on convenience sampling, taking into account both the profile and availability of potential subjects. Consequently, the low number of participants introduces a threat to **External Validity**. We aimed to mitigate this threat by aiming for covering a wide range of roles, domains, and cultural backgrounds. Finally, to ensure **Reliability,** we implemented various measures. Throughout the interviews, we had multiple researchers present to enhance the reliability of our data. Additionally, we provided used materials and a detailed analysis process, enabling other researchers to replicate our methodology in diverse contexts. Moreover, the authors actively engaged in discussions to maintain consistency in the coding results. However, despite our efforts, we acknowledge the possibility of some subjectivity in our analysis.

3.6 Conclusion

This qualitative exploratory study investigates the integration of human factors with continuous experimentation. To effectively integrate human factors experiments in continuous experimentation, there's a pressing need for upgraded infrastructure, improved developers' awareness about the importance of human factors, and training developers in empirical study methods essential for effective human-centric experimentation.

By fostering interdisciplinary collaboration and promoting the integration of human factors considerations into continuous experimentation, organizations can enhance the user experience, and improve the quality of software and systems. Future research should focus on developing frameworks and detailed guidelines for effectively incorporating human factors into continuous experimentation processes, leading to the creation of more user-centric, safe, and acceptable systems.

Chapter 4

Paper C

Defining Requirements Strategies in Agile: A Design Science Research

Amna Pir Muhammad1, Eric Knauss1, Odzaya Batsaikhan, Nassiba El Haskouri1, Yi-Chun Lin1, and Alessia Knauss

In the International Conference on Product-Focused Software Process Improvement, 2022.

Abstract

Research shows that many of the challenges currently encountered with agile development are related to requirements engineering. Based on design science research, this paper investigates critical challenges that arise in large scale agile development from an undefined requirements strategy. We explore potential ways to address these challenges and synthesize the key building blocks of requirements strategies. Our design science research rests on a multiple case study with three industrial cases. We relied on a total of 20 interviews, two workshops, participant observation in two cases, and document analysis in each of the cases to understand concrete challenges and workflows. In each case, we define a requirements strategy in collaboration with process managers and experienced engineers. From this experience, we extract guidelines for defining requirements strategies in agile.

4.1 Introduction

Agile development methodologies aim to shorten the time to market and incorporate maximum changes during the sprint to meet customer needs [28] and have been adapted at small-scale as well as large-scale organizations [35]. With its' focus on interactions and working software over rigid processes and extensive documentation, traditional well established Requirements Engineering (RE) processes have been neglected. Research shows that many of the challenges currently encountered with agile development are related to requirements engineering [21] for example, misunderstanding customer needs, missing high-level requirements, and difficulty to achieve having just enough documentation.

In this study, we identify specific RE-related challenges and related solution strategies in agile development. Based on this knowledge, we derive necessary building blocks as different viewpoints that should be considered when thinking strategically about RE in agile. In this, we are inspired by test strategies, which guide testing activities to achieve the quality assurance objectives [106] and which mandate that each project must have its test plan document that clearly states the scope, approach, resources, and schedule for testing activities [107]. We argue that defining a so called *requirements strategy* similar to a test strategy for RE can be critical for successful agile development.

In this paper, we aim to establish the concept of *requirements strategy for agile development* by investigating the following research questions based on iterative design science research in three industrial case studies.

RQ1 Which challenges arise from an undefined requirements strategy?

RQ2 How do companies aim to address these challenges?

RQ3 Which potential building blocks should be considered for defining a requirements strategy?

Since we particularly target agile development, we aimed to investigate requirements challenges independent from process phases or specific documents. Instead, we took the lens of *shared understanding*[99] to investigate different RE activities (i.e., elicitation, interpretation, negotiation, documentation, general issues). According to Fricker and Glinz, an investigation of *shared understanding* may primarily target how such shared understanding is *enabled* in an organization, how it is *built*, and how it is *assessed* for relevance [99].

Therefore, our contribution are guidelines on how requirements strategies should be described for agile development. Through building three complementary perspectives, we see that the requirement strategy guidelines capture relevant information and provide a useful overview. We suggest that a strategy defines the structure of requirements to create a shared language, define the organizational responsibilities and ownership of requirements knowledge, and then map both structure and responsibilities to the agile workflow.

In the next section, we provide the related work for our study. In Section 4.3 we elaborate on our design science research method before revealing our findings in Section 4.4 in order to answer our research questions. Then, in Section 4.5 we present our artifact - guidelines on how to define a requirements strategy for RE in agile. Finally, we discuss and conclude our paper in Section 4.6.

4.2 Related Work

Literature shows that many companies adopt agile methods [242, 243] due to its numerous benefits, for example, flexibility in product scope which improves the success rate of products [36], in contrast to traditional development methods [27]. Furthermore, agile methods incorporate maximum change and frequent product delivery [243], encourage changes with low costs, and provide high quality products in short iterations [28].

Due to its success, agile is become widely popular and adopted by both small and large companies [35]. The term “large scale agile” refers to agile development, which includes large teams and large multi-team projects [19]. Dikert K. mentioned that “large scale agile development” refers to agile development that includes more than six teams [36].

However, despite the success of agile methods, large scale companies also still face several challenges. Dikert et al. (2016) [36] conducted a systematic literature review of empirical studies. The authors identified several challenges and success factors for adopting agile on a large scale. The most mentioned challenges are change resistance, integrating non-development functions, difficulty to implement agile (misunderstanding, lack of guidance), requirement engineering challenges (e.g., high-level requirements management largely missing in agile, the gap between long and short term planning). Based on a literature review, Dumitriu et al. (2019) [102] identified 12 challenges of applying large scale agile at the organization level. The most cited challenge is the coordination of several agile teams. Kasauli et el. (2021) [21] identified 24 challenges through multiple case studies across seven large scale companies. Some of the identified challenges are building long lasting customer value, managing experimental requirements, and documentation to complete tests and stories. The authors conclude that strong RE approaches are needed to overcome many identified challenges.

When it comes to RE in agile, challenges that have been identified include lack of documentation, project budget, time estimation, and shared understanding of customer values [31, 43, 45, 244]. First attempts have been made to tackle some of the challenges of RE in agile, e.g., Inayat et al. and Paetsch et al [44, 45]. suggest combining traditional RE with agile and encounter challenges like how much documentation is just enough documentation [47] to have a shared understanding of customer values.

Considering that there are many challenges related to RE that can be solved through RE approaches, this paper proposes to use the concept of a requirements strategy as a method to define requirements engineering practices to tackle challenges related to requirements engineering in agile.

4.3 Design Science Research Method

Our research aims to design suitable ways of defining requirements strategies for organizations with agile software development. Such requirements strategies should be suitable for addressing real-world needs, incorporating state-of-the-art knowledge, and ideally being empirically evaluated in practice. Thus, we decided that design science research [77, 78, 245] is a good fit.

Table 4.1: Research questions in relation to cases and research cycles

	Case 1 Cycle 1	Case 2 Cycle 2	 Cycle 3	 Cycle 4	Case 3 Cycle 5
Challenges (RQ1)	■■■■■■	■■■■■■	■■■■□□	■■□□□□	■■■□□□
Solutions (RQ2)	■■■■□□	■■□□□□	■■■■■■	■■■■□□	■■■□□□
Blocks (RQ3)	■■□□□□	□□□□□□	■□□□□□	■■□□□□	■■■■■■
Data source	11 Interv., document analysis	9 Interviews, participant observation, document analysis, 1 workshop			Participant observation, document analysis, 1 workshop

4.3.1 Design Science Research

Our research questions are targeted towards design science research, with RQ1 focusing on the problem domain, RQ2 investigating potential solutions, and RQ3 targeting towards deriving the artifact. Our artifact are guidelines on how to define a requirements strategy in agile development. Refining on well-known challenges with RE in agile development, we needed to gain in-depth insights into those challenges related to a lack of a clear requirements strategy throughout the agile development organization (RQ1). Throughout our cases, we discuss those challenges with respect to potential mitigation strategies (RQ2) for those challenges. Finally, we systematically synthesize the building blocks of requirements strategies (RQ3) from solution strategies.

Inspired by the regulative cycle [77], the artifact (guidelines for defining a requirements strategy based on good practices from our cases) has iteratively evolved, allowing to refine the knowledge with respect to each research question. Table 4.1 provides an overview of our research method. As can be seen, we relied on three case studies over which we distribute a total of five research cycles. The cycles differ in how much focus is given to each of our three research questions:

Case 1 *- Exploring the problem through the lens of requirements engineering and shared understanding:* Case 1 focuses on a strategy to achieve shared understanding about customer value throughout the development organization. Our research aims were two-fold: understand the real world problem and conceptualize a design artifact that may address this problem. Within a Master's thesis [246], we developed an appropriate lens that combined both the concept of shared understanding (as expressed by Glinz and Fricker through enabling, building, and assessing shared understanding [99]) and commonly used RE activities (such as elicitation, interpretation, negotiation, and documentation). We then relied on 11 interviews to understand customer value and its common understanding, information sharing, tools and channels for sharing, and tools and methods for documenting. Since our first cycle focuses on the exploration of the problem we locally relied on the case study research method for our research with respect to Case 1 [246]. As Table 4.1 shows, we complemented the interviews with document analysis to produce an overview of challenges and related solution strategies.

Case 2 *- Refining the requirements strategy artifact iteratively:* We then followed up in *Case 2*, a company producing security smart alarms and services.

In this case, the focus was on a more general requirements strategy that covers both stakeholder and system requirements. Again, through a Master's thesis [106], we investigated concrete requirements challenges of an agile team, defined a requirements strategy along the lines of the result from Case 1, and investigated in depth to what extent it could help with the challenges in practice. At this point, we further focused on investigating whether there are reusable building blocks for a requirements strategy.

Case 3 *- Applying and evaluating the artifact:* Finally, we brought our experience from the previous two cases into *Case 3*, an automotive supplier, focusing on complex safety critical and software intense systems. Here, we focus less on challenges and solution strategies, in particular, since the case company already had compiled a good overview. Instead, our focus is to refine the artifact (guidelines for requirements strategies) by discussing, applying, and improving our understanding of the building blocks of a requirements strategy. At the time of the research, the first author of this paper did an internship with this automotive supplier and helped to make the requirements strategy explicit. The company did already identify some of the challenges and making a first step towards implementing their solution strategies. Thus, she was able to investigate the phenomenon as a participant observer, contrasting it with documents and ways of working in the practical context, allowing us to fine-tune our guidelines for requirements strategies to provide an overview of challenges and solution strategies for continuous process improvement.

4.3.2 Data Collection

We relied on a mix of different methods for data collection, including interviews, participant observation, document analyses, and workshops.

Interviews - We relied on interviews in Case 1 and Case 2, in particular, to understand the problem (RQ1) in each specific case. 20 interviews were conducted using interview guides (details in [106, 246]), relying on a mix of closed and open-ended questions. Interviews were recorded, transcribed, and then coded. In both cases, we recruited interviewees through purposeful sampling [196]. We relied on convenience sampling so that interviewees were both available and knowledgeable. We employed diversity sampling to capture the views of multiple relevant roles and stakeholders and similarity sampling to ensure that we received multiple views for each relevant perspective for triangulation.

Participant Observation - The fourth author was very familiar with Case 2, in which she had worked for several years before starting her Master thesis [106]. Her work included defining a testing strategy, which provided intimate knowledge about the agile ways of working in the case company, which were also helpful for understanding the requirements-related challenges, defining a requirements strategy, and conducting the evaluation in Case 2. The first and last authors did work with RE and the continuous improvement of requirements processes of Case 3. Through this work, we were able to verify that previously itentified challenges in Case 3, as well as initiatives to address them, were of similar nature and matched well with our recent work on requirements strategies. Both co-authors relied on our requirements strategy work to support the ongoing initiatives on requirements processes and on integrating RE practices into agile ways of working. This allowed us to evaluate the suitability of our requirements strategy concept. Knowledge from these activities was collected through field notes, presentations given at the case company, and discussions with other co-authors.

Document Analysis - In all three cases, a subset of the authors studied the documents related to the flow of requirements (Case 1: Author 3 and 5, Case 2: Author 4, Case 3: Author 1 and 6). Since all three cases embraced agile ways of working, we considered that not all relevant information might be found in formal documents. However, we ensured that documentation did match or at least not contradict our data. We found relevant documentation of requirements, e.g., as user stories, in all three cases to match and support our other data sources. Document analysis also allowed us to better understand the implied requirements strategy and processes.

Workshops - We relied on two workshops in cases 2 and 3 to evaluate the proposed requirements strategies and, by that, also our requirements strategy guidelines. In case 2, a workshop was conducted to present the challenges identified, the proposed solution candidates, and different versions of the specific requirements strategy of each case. In case 3, a workshop was used to understand the requirements strategy that was used to address certain challenges. Expert participants were sampled similarly as for interviews. They were asked to bring up additional challenges that we may have missed, give feedback on the criticality of the challenges that we had found, provide their opinion about the solution candidates, and evaluate the structure, presentation, and concrete advice on the requirements strategies. Depending on the circumstances in each case, we recorded and transcribed, took live notes for all participants to see, or shared our notes after the workshop for validation.

4.3.3 Data Analysis

In order to analyze interview transcripts, field notes from participatory observation and document analysis, and workshop notes/transcripts, we relied on typical coding approaches for qualitative research [247]. This allowed us to report on challenges that relate to a missing or undefined requirements strategy. For example, the following quote from Case 1 contributed to identifying the challenge *d) lack of communication with customers*: *`The thing that sometimes does not work as well as it should, is communication with some of the customer units. It heavily depends on the competence of the customer unit people."*

In each case, we had access to an industry champion from the respective company, who helped to suggest practical solutions. For example, the following quote suggested a solution for challenge above as *c) ability to initiate on demand meetings with customer representatives: "The right people to nail down a requirement should be put together in the meeting to have a requirement handshake."* In addition, the second author was involved as an academic supervisor in all three cases, providing pointers toward relevant published knowledge. We regularly presented and discussed our findings at the case companies, focusing on strong tracing between challenges, solution candidates, and the proposed requirements strategies. Together with iterative refinements, this allowed us to analyze the data in depth.

4.3.4 Threats to Validity

Internal validity aims to reveal factors that affect the relationship between variables, factors investigated, and results. It reflects that the studied results eliminate alternative explanations. Therefore, one of the main internal validity threats to this study is misinterpretations, particularly during the interviews and observations. *Construct validity* defines the extent to which the investigated measures describe what the researchers analyze and what is studied according to research questions [248]. We mitigated threats to internal and construct validity through interacting closely with industry partners and by iterating over our results. *External validity* relates to identifying to what extent our findings can be generalized [249] and is typically of lower priority in in-depth case studies and design science research. However, we identified some common challenges in all three case companies. Thus, we expect that in particular the structure and perspectives of our requirements strategy guidelines can be transferred to other contexts. Related to external validity, reliability reflects to what extent, other researchers can produce the same results repeating the same study methodology. In a qualitative study, it is always hard to achieve *reliability* since one cannot argue based on statistical significance. We mitigate this threat by elaborating our research method in detail to support other researchers in replicating our research and in recovering from any possible differences in results.

4.4 Findings

4.4.1 RQ1: Which challenges arise from an undefined requirements strategy?

The left column of Table 4.2 depicts RE challenges identified based on our three cases that are encountered without a clear requirements strategy existing for agile development.The challenges are categorized in RE practices and related to Glinz and Fricker's [99] practices of shared understanding, grouped in three categories, i.e., enable, build, and assess. Enable practices describe what is needed to form and establish a common foundation of knowledge. Building practices aim to provide the structured knowledge that can be communicated within the team or company through explicit artifacts or by constructing a body of implicit knowledge for shared understanding. Assessing methods determine how all team members have a shared understanding of a topic or artifact. Some methods can be used for both building and assessing practices. Indices indicate in which of the cases a challenge was relevant.

a) Teams struggle to integrate RE in their agile work efficiently[1,2,3] - Agile development enables organizations to respond to change. If there is a change in code and tests, the requirements should usually be updated. Or if requirements change, then the code and tests need to be adjusted accordingly. Teams struggle with this since requirements tools do not integrate well with agile software development work and do not support parallel changes from several teams. Thus, it is hard to integrate RE work into the agile work effectively.

b) No formal event to align on customer value[1] - There were no formal events to create awareness of customer value in Case 1. Even when the customer unit took the initiative and organized some events, there were only a few participants. Such events must be better integrated in the organization and workflow.

c) Insufficient customer feedback[1,2] - In Case 1 and 2, developers lack customer feedback, which is crucial for agile workflows.This can be due to a lack of formal events, or due to scale and distance to customers. It impacts the ability of an organization to assess whether shared understanding has been reached. Customer feedback should be integrated into the workflow across organizational levels and take into account the specific needs of product owners and developers.

d) Lack of communication with customer[1] - Customer-facing units have a key role and are on the boundary between development teams and customers. We encounter difficulties with communication in both directions: between customer-facing teams and development teams and between customer-facing teams and the customers. These challenges are mainly due to a lack of systematic guidance on how such communication should take place, thus depending completely on the individual skills of those involved. Companies would have to find a way to ensure good and transparent communication, for example by having product owners moderating direct meetings between developers and customers.

e) Who owns customer value[1] - Requirements enter the development organization mainly through the hierarchy of product owners (PO) in Case 1. However, a significant amount of requirements originate from other sources, e.g., development teams or system managers, and in those cases, it is less clear who is able to define or who owns the customer value.

f) Inconsistent elicitation[2] - POs or application specialists collect requirements when needed and apply techniques such as interviews. There is, however, no systematic strategy to elicitation integrated into the workflow.

g) Lack of feedback on elicitation[2] - Without a systematic validation of elicitation results, misunderstandings will only surface late in the agile workflow, e.g. during acceptance testing and result in additional costs and effort.

h) Unclear why requirement is needed[2] - Due to scale, distance to customers, or because a customer value description is not available for developers (see Challenge o), application specialists and POs may lack information on why specific low-level requirements are needed. This can result in a gap between what product owners want and how the development teams interpret their requirements.

i) Wrong assumptions about customer value[1] - Interviewees highlighted that one of the significant challenges is that people assume customer value based on their tacit knowledge, leading to the development of faulty assumptions.

j) Unclear and volatile customer needs[2] - Requirements change, for example when the customer changes their mind or did not have a detailed opinion in the beginning. When assessing the interpretation of requirements, this can cause friction, since the team tries to “hit a moving target”.

k) Decentralized knowledge building[3] - Different teams develop requirements, architecture, and also processes at the same time. This decentralized way of working is needed to yield the benefits of agile work at scale, but requires some infrastructure to enable knowledge sharing and alignment. Otherwise, conflicting decisions will be made throughout the organization.

l) Focus on technical details[1,2] - Often customer value is not explicitly

Table 4.2: Overview of Challenges in Relation to the Requirements Strategy Model. Indices ([1],[2],[3]) show in which case study a challenge or strategy was encountered.

	Shared Understanding			
RE	*Enable*	*Build*	*Assess*	**Solution Strategy**
General issues	a) Teams struggle to integrate RE in their agile work efficiently[1,2,3]	b) No formal event to align on customer value[1]	c) Insufficient customer feedback[1,2]	a) Tools that allow developers to take ownership of req.[1,2,3] b) Regular meetings with customer representat.[1,2]
Elicitation	d) Lack of communication with customer[1] e) Who owns customer value[1]	f) Inconsistent elicitation[2]	g) Lack of feedback on elicitation[2]	c) Ability to initiate on demand meetings with customer representatives[1,2]
Interpretation	h) Unclear why requirement is needed[2]	i) Wrong assumptions about customer value[1]	j) Unclear and volatile customer needs[2]	d) Fast feedback cycles[1,2]
Negotiation	k) Decentralized knowledge building[3]	l) Focus on technical details[1,2] m) Req. open for comments[3]	n) No time for stakeholder involvement[2]	e) Req. template includes customer value & goals[1,2] f) Define team responsibilities for different parts of req. and review updates regularly[2,3]
Documentation	o) Customer value description lost between systems[1] p) Lack of knowledge about writing requirements[1,2,3] q) No dedicated time for requirements[1,2,3]	r) Too much/not enough document.[1,2] s) Trace the requirements to all levels, (test, and code)[3]	t) Inconsistency b/c of requirements change[3]	g) Rationale must always be provided[1] h) Just enough documentation[1,2] i) Plan time for requirements updates[3] j) Educate and train the development teams[2,3] k) Tools need to be setup to support traceability [3]

described; instead, customer needs and technical solutions are more explicit. When we asked participants in Case 1 and 2 to describe the customer value of specific requirements, they explained the technical solutions rather than customer values. This finding is consistent with documentation, where often technical details are described instead of linking to a business reason for motivating the requirement.

m) Requirements open for comments[3] - In agile development, everyone who has access to the system can create issues related to requirements in the requirements management tool. While it is positive to include as many stakeholders as possible in discussions, without a defined process that respects the development lifecycle, this can result in an unstructured discussion and very late changes.

n) No time for stakeholder involvement[2] - Getting stakeholders' feedback after interpreting the elicited requirements is challenging since stakeholders do not have time for several meetings.

o) Customer value description lost between systems[1] - At the scale of Case 1, it is not unusual to use several different tools to manage requirements at various abstraction layers. Customer-facing units use one tool, in which they define stakeholder requirements and customer value. Development teams interact with different tools, and it is the task of the POs to refine and decompose the stakeholder requirements from tool 1 into work items for the agile teams in tool 2. At this step, documentation about customer value is often not transferred and thus not available to the developers.

p) Lack of knowledge about writing requirements[1,2,3] - Throughout our cases, we found that those who are responsible for documenting requirements often do not have the right training. In addition, we frequently saw a lack of structure and no requirements information model. Thus, teams mix stakeholder and system requirements and are challenged with writing high-quality user stories, system requirements, and in particular quality requirements. In particular, the quality requirements might not get documented at all and teams will work on them without making them visible on the sprint dashboard.

q) No dedicated time for requirements[1,2,3] - Since agile methods focus on reducing time to market, spending time on writing formal requirements is not considered. Instead, agile teams rely on verbal requirements. Dedicated time to work on requirements should be integrated in the agile workflow, e.g. each sprint.

r) Too much/not enough documentation[1,2] - Because agile focuses on less documentation, some essential parts could be missing if we do not do much documentation, such as the "why" part of the requirement. Thus, in agile, determining the right amount or sweet spot of documentation is challenging.

s) Trace the requirements to all levels, (test, and code)[3] - Due to ISO26262 and ASPICE compliance, the automotive company needs to guarantee full traceability between all requirements levels, (tests, and code). This places a big challenge on the entire company, since most teams work on something related to requirements, tests, or code.

t) Inconsistency because of requirements change[3] - Agile methods embrace change and, consequently, teams will make changes on requirements during their work. However, it is challenging to handle sudden change requests and opinions from different team members, especially at scale. The consequence can be that teams inconsistently change related requirements, or that the scope is increased without central control. The problem is known, yet there is a lack of guidance on how to handle this in large scale agile development to avoid expensive rework.

4.4.2 RQ2: How do companies aim to address these challenges?

The last column of Table 4.2 summarizes the answers to RQ2 on solution strategies associated with the challenges with each phase of RE in respective rows, derived from interviews, literature, or workshops and confirmed by experts in each case.

a) Tools that allow developers to take ownership of requirements[1,2,3] - In order to allow developers to take ownership of requirements, we need to find requirements tooling that integrates into the mindset and the development environment of developers to provide an efficient way of manipulating requirements. For instance, developers work closer to the code, so the requirements tool that supports commit/git is highly encouraged.

b) Regular meetings with customer representatives[1,2] - The customer-facing unit should arrange regular meetings with customers. These meetings should be well integrated in the agile workflow and mandatory for team members.

c) Ability to initiate on demand meetings with customer representatives[1,2] - There should be a setup to initiate meetings with customers whenever developers need feedback. Since access to customer representatives is a sparse and valuable resource, a strategy for such meetings should be well aligned with the organizational structure and the agile workflow.

d) Fast feedback cycles[1,2] - All teams use direct communication with stakeholders and fast feedback cycles as a baseline to get the correct interpretation. Customer insight is abstract knowledge and could be hard to write down. There is a need to arrange events where people can meet, interact, and share customer values and feedback.

e) A requirements template that includes customer value and goals[1,2] - To avoid challenges related to a lack of awareness of customer value, there should be specific fields or tracelinks that show how each requirement adds customer value. It is important to check their usage regularly.

f) Define team responsibilities for different parts of requirements and review updates/comments regularly[2,3] - In order to yield benefits from agile workflows, RE must be integrated into the agile workflow. This means that agile teams need to take responsibility of maintaining requirements and to monitor changes of requirements that are potentially related. This allows to manage requirements updates in parallel and at scale. However, responsibilities have to be carefully delegated and clearly assigned.

g) Rationale must always be provided[1] - The rationale for the requirement should mandatorily be provided by the role/person writing the requirement. Moreover, it should effectively be passed on from tool to tool.

h) Just enough documentation[1,2] - Balancing sufficient communication and documentation is crucial in agile. We should not spend too much time documenting; however, it should have all the necessary information. Developers need clear guidelines to achieve this balance.

i) Plan time for requirements updates[3] - Teams should plan (update, change, review) the requirements in time to align with the updated scope. Such a plan should consider that updating requirements in the scope of one team may imply also requirements updates in other scopes.

j) Educate and train the development teams[2,3] - If development teams should take more responsibility of requirements, they need to be trained in RE as well as in the specifics of the overall requirements processes in their organization. A clear requirements strategy can be a good starting point to plan such training.

k) Tools need to be setup to support traceability[3] - Requirements are usually represented in different forms (e.g., textual requirements, user stories) and on different levels (e.g., system level and software level). Teams could get requirements at higher level and then derive the lower level requirements (e.g., software/technical requirements). Tracing requirements could be hard in a large complex system. Tools are needed and they should be aligned with a requirements strategy for agile workflows, i.e. allow parallel work for many teams.

4.4.3 RQ3: Which potential building blocks should be considered for defining a requirements strategy?

This section systematically develops the building blocks of a requirements strategy from our findings in all three cases.

In Case 1, the company was challenged to establish a shared understanding. Proposed solution strategies for specific challenges in Case 1 can be categorized as ***structural***, ***organizational***, or related to the ***work and feature flow***. For example, for the challenge *l) focus on technical details)*, a related solution strategy is *e) requirements template includes customer value and goals.* This strategy explains that, to avoid the lack of awareness about customer value, there should be specific fields related to customer value in the requirements templates. This solution shows that there is a need for improvement at the ***structural level***. In contrast, *b) no formal event to align on customer value* is a challenge related to stakeholders' roles and responsibilities that needs to be well integrated into the ***organization***. The last column in Table 4.2 provides a solution strategy related to this challenge as *b) regular meetings with customer representative*, which relates not only to the ***organizational perspective***, but also to the ***work and feature flow***.

In Case 2, we found the same perspectives (***structural, organizational***, as well as ***work and feature flow***) in in solution strategies for their specific challenges. As in Case 1, the solution strategy to introduce *e) requirements templates that include customer value and goals* is a ***structural*** example. In contrast, the challenge *g) lack of feedback on elicitation* can lead to misunderstandings late in an agile workflow. The solution strategy is to establish the *c) ability to initiate on-demand meetings with customer representatives.* Providing access to a sparse and valuable resources such as a customer representative relates to the ***organizational*** perspective. Another related solutions strategy, *d) fast feedback cycles*, for the challenge *j) unclear and volatile customer needs* falls into the ***work and feature flow*** perspective, by arranging events where people can meet, interact, and share customer values and feedback.

After looking deep into the concrete solution strategies in Case 1 and Case 2, we found that many of these strategies were already successfully implemented in Case 3. However, the company still faced some RE challenges in agile development, allowing us to check whether the same building blocks are also applicable in Case 3. For example, the challenge *s) trace the requirements to all levels* can be addressed with the ***structural*** solution strategy *k) tools to set up traceability.* Similarly, the challenge *k) decentralized knowledge building* can be addressed by the ***organizational*** solution strategy *define team responsibilities for different parts of requirements and review updates/comments regularly.*Finally, an example of a ***work and feature flow*** related solutions strategy is to *i) plan time for requirements updates* in agile sprints to counter the challenge of having *q) no dedicated time for requirements.*

In summary, in order to address specific challenges related to enabling,

Table 4.3: Building Blocks of a Requirements Strategy: How does shared understanding impact decisions in the workflow

Perspective	Support for shared understanding: Common language	Knowledge flow	Examples
Structural	Define reqts. levels	Define structural decomp.	Stakeholder, System, Component Requirements
	Define reqts. types	Define traceability demands	Requirements and Traceability Information Model
	Define templates		User stories include customer value and goal
Organizational	Define ownership of reqts. types	Define roles and responsibilities	Training plan per type/role; Team responsibility sheet
Work and feature flow	Define lifecycle of types	Map structure to workflow	Elicitation strategy, definition of done
		Map organization to workflow	Stakeholder map, requirements review strategy

building and assessing shared understanding of requirements in agile, specific solution strategies fall into three distinct categories: ***structure***, ***organization***, as well as ***work and feature flow***. Thus, a requirements strategy that bundles such solution strategies for a concrete case should cover these three perspectives. Our data from all three cases indicates that challenges occurred since insufficient structure, organization, and integration to work and feature flows were provided. As this insight was considered useful in Case 3, we developed guidelines for requirements strategy presented in Section 4.5 along these categories by abstracting from similar concrete solution strategies in our three cases.

4.5 Artifact: Guidelines for Defining a Requirements Strategy

Based on our findings from the three cases, we have iteratively derived our artifact: the guidelines for defining a *Requirements Strategy* as a means to define RE activities in agile development workflows. At the time of research, the term "requirements strategy" has not been widely used. This is in contrast to, for example, "test strategy", which has quite widely been accepted to describe how testing practices can be integrated in development workflows, such as in agile ways of working. In our work, we refer to "requirements strategy" as a general strategy for including RE practices in agile ways of working.

Definition: Requirements Strategy. A requirements strategy is an outline that describes the requirements engineering approach in systems or software development cycles. The purpose of a requirements strategy is to provide a rational deduction from organizational, high-level objectives to actual requirements engineering activities to meet those objectives and to build a shared understanding about the problem space and requirements.

The creation and documentation of a requirements strategy should be done in a systematic way to ensure that all objectives are fully covered and understood by all stakeholders. It should also frequently be reviewed, challenged, and updated as the organization, the ways of working, and the product evolve over time. Furthermore, a requirements strategy should also aim at aligning different requirements stakeholders in terms of terminology, requirements types and abstraction levels, roles and responsibilities, traceability, planning of resources, etc.

Therefore, our contribution is a model of how requirements strategies should be described for agile development. Through providing three complementary perspectives, the proposed guidelines help to capture relevant information and provide an useful overview. Our guidlines are summarized in Table 4.3, including reoccurring examples and good practices abstracted from the three case studies. We propose that a requirements strategy should include the following building blocks: a structural perspective, an organizational perspective, and a work and feature flow perspective. Across these perspectives, a requirement strategy aims to support a shared understanding of requirements, in particular with respect to establishing a *common language* (i.e., enabling perspective in Table 4.2) and with respect to facilitating the exchange and *flow of knowledge* (i.e., building and assessing perspective in Table 4.2).

We suggest to start with a structural view to create a common language. A good starting point can be the artifacts in the development lifecycle model, for example the requirements information model in the Scaled-Agile Framework SAFe. It can be good to define templates for all types. Based on these initial definitions, refinements can be provided based on experience, e.g., after sprint reflections.

As a second step, we propose to make the organizational perspective explicit. Define the roles and responsibilities with respect to the definitions in the structural view. This can, for example, be done with a one-pager that describes the responsibilities of a team. Also, state who owns which part of requirements (e.g., requirements on certain subsystems) to determine specific training needs.

Finally, the work and feature flow perspective needs to be defined. A good starting point can be a lifecycle model for each critical type, which is then mapped to the intended workflow. In agile, this can partially be provided by defining done criteria. In particular, it needs to be defined when and by whom certain information must be provided. If requirements elicitation efforts are anticipated, guidance should be given on obtaining the information from stakeholders. The workflow should be related to the roles and responsibilities as well as ownership. A stakeholder map can provide valuable information: who owns an artifact, who should be kept informed, and who needs to review it. An explicit review strategy can be very valuable, affecting not only the requirements quality but also keeping reviewers informed about recent changes.

4.6 Discussion and Conclusion

In this design science research study, we identified challenges related to agile requirements engineering in three case companies. Based on these three case studies, we identified solution strategies for resolving the identified challenges and derived building blocks as substantial parts of a requirements strategy. For each case we investigated a concrete requirements strategy. The individual requirements strategies have been well received by experts in each case company. As our experience grew, we noticed reoccurring building blocks on what should be part of such a requirements strategy.

By embracing RE as a knowledge management problem, we now propose guidelines for creating requirements strategies that focus on aspects independent of documentation and waterfall phases in agile development as the artifact of our design science research. Instead, we recognize the need to enable, build, and assess shared understanding of requirements in agile development. To do so, a requirements strategy should describe how requirements are structured, how work is organized, and how RE is integrated in the agile work and feature flow.

Ideally, such a strategy will be documented concisely and made available to all stakeholders. Our requirements strategy can be interpreted as an instance of situational method engineering [250] where we focus on the context of agile system development and requirements methods in particular. By this, we aim to make it easier for practitioners to integrate RE in their agile workflows. This supports its evolution through the reflection opportunities built into agile methods. We hope that our requirements strategy model facilitates future research on how to manage knowledge related to requirements in agile development.

Chapter 5

Paper D

Requirements Strategy for Managing Human Factors in Automated Vehicle Development

Amna Pir Muhammad, Alessia Knauss, Eric Knauss, Jonas Bärgman

Requirements Engineering -(RE'24), 2024.

Abstract

The integration of human factors (HF) knowledge is crucial when developing safety-critical systems, such as automated vehicles (AVs). Ensuring that HF knowledge is considered continuously throughout the AV development process is essential for several reasons, including efficacy, safety, and acceptance of these advanced systems. However, it is challenging to include HF as requirements in agile development. Recently, Requirements Strategies have been suggested to address requirements engineering challenges in agile development.

By applying the concept of Requirements Strategies as a lens to the investigation of HF requirements in agile development of AVs, this paper arrives at three areas for investigation: a) ownership and responsibility for HF requirements, b) structure of HF requirements and information models, and c) definition of work and feature flows related to HF requirements. Based on 13 semi-structured interviews with professionals from the global automotive industry, we provide qualitative insights in these three areas. The diverse perspectives and experiences shared by the interviewees provide insightful views and helped to reason about the potential solution spaces in each area for integrating HF within the industry, highlighting the real-world practices and strategies used.

5.1 Introduction

Human factors (HF), a multidisciplinary field focusing on understanding the interactions between humans and other system elements, plays a pivotal role in the design and development of user-centric, safe, and effective automated vehicle (AV) technology [199], including supervised and unsupervised automation. User-centric here means, for example, that the AV technology is predictable, comfortable, acceptable, and trustworthy.

Human factors requirements refer to those requirements that are primarily introduced due to HF and are essential in this context. A HF requirement could be that an AV should move to the lane's edge when detecting an oncoming truck, addressing human comfort aspects.

Despite the recognized importance of HF in system development, its integration into the complex and multidimensional environment of AV development [6, 251, 252], especially within agile frameworks [253], remains challenging. For example, it has been reported that tech leaders and project managers might underestimate the impact of human-related factors, focusing instead solely on technical challenges [254]. In our experience, this oversight is partially due to the technology driven nature of AV development, where the driving task is increasingly perceived as disconnected from humans, focusing instead on computer based automation. Thus, companies tend to embrace data-driven approaches while also adopting more agile and iterative development paradigms. However, this may lead to underestimating the role of humans and their capabilities and limitations as constraints for the overall system. This underestimation and the inherent complexities of agile methodologies, which emphasize speed and flexibility, often lead to insufficient considerations of HF in the development process and threatens the safety and success of AV Systems. Recognizing this gap, there arises a critical need for a structured approach to ensure that HF requirements are consistently and effectively integrated. This is where the concept of Requirements Strategies comes into play, providing a systematic framework for aligning requirements integration with the dynamic and iterative nature of agile development [111].

The Requirements Strategy provides a guideline to integrate requirements engineering practices effectively within agile workflows [111]. The core purpose of this framework is to bridge the gap between high-level organizational objectives and the specific requirements engineering activities needed to achieve these objectives. This framework facilitates building a shared understanding of the problem space and requirements among all stakeholders involved and to integrate it into a development workflow. The Requirements Strategy framework consists of three main building blocks: an organizational perspective, a structural perspective, and a work and feature flow perspective. Each of these perspectives contributes to creating a holistic understanding and effective management of requirements.

In this qualitative study, we examine the integration of HF requirements through the lens of Requirements Strategies. Our objective is to evaluate the effectiveness of Requirements Strategies in facilitating this integration. For this purpose, we formulated three research questions, each focusing on a main building block of the Requirements Strategy.

RQ1 *Organizational perspective:* How do ownership and responsibility for HF requirements impact their integration in product development?

RQ2 *Structural perspective:* How does the structure of requirements and information models impact the integration of HF requirements in product development?

RQ3 *Feature and workflow perspective:* How does defining a work and feature flow related to HF requirements influence their integration in product development?

For each question, we propose propositions based on common believes in existing literature. These propositions serve as fundamental hypotheses that guide our analysis, shaping the way we approach and interpret our findings. This method helps us to thoroughly examine and apply different aspects of HF within the lens of Requirements Strategy. Using this theoretical framework, we conducted 13 semi-structured interviews with professionals from the global automotive industry.

Our findings reveal diverse perspectives on the proposed propositions. While some cases lend support to our propositions, others suggest that these methods (e.g., requirements decomposition, traceability, assigning HF expertise to critical roles) may not be as effective as anticipated. The universal acceptance of these requirements practices as described in the literature and within the framework of Requirements Strategy, is inconclusive in our context. These mixed views highlight the necessity for additional research to further explore these findings, but also the need for organizations to define a consistent Requirements Strategy within the identified solution spaces.

The remaining paper is organized as follows: Section 5.2 depicts the background and anchors our propositions in prior related research. Section 5.3 outlines the research methodology, also addressing potential threats to the validity of the results. The findings are presented in Section 5.4. Section 5.5 engages in a discussion of these findings. Finally, the paper concludes with Section 5.6.

5.2 Background

5.2.1 Human Factors in Automotive Industry

The concept of HF is pivotal in AV development. According to Muhammad et al., "the field of *Human Factors in AV Development* aims to inform AV development by providing fundamental knowledge about human capabilities and limitations throughout the design cycle so the product will meet specific quality objectives" [223].

HF experts concentrate on understanding the full spectrum of human attributes, including cognitive, physical, behavioral, psychological, social, emotional, and motivational aspects [199]. This understanding is vital for designing AVs that are trustworthy, comfortable, effective, and safe as well as deliver a satisfying and aesthetically pleasing experience for all stakeholders, as noted in [6]. The insights into human capabilities and the objectives of AV design are disseminated through various means including design principles, training programs, selection processes, and effective communication strategies. This dissemination is an ongoing necessity, integral to the comprehensive development and refinement of AV systems, throughout the AV design cycle.

In Table 5.1, we provide specific examples of functionality of (conditionally) automated vehicles together with some important HF aspects for which HF requirements are crucial. Note that these are examples relating to AV features of different levels of automation that are typically engineering and technology driven, and not to human machine interface (HMI) design. While the design of the latter typically includes HF experts, the former are usually designed and developed in agile teams without integrated HF experts. Also, note that studies have shown a misalignment between engineers' judgments and users' trust in automation [255].

Note that this study focuses on HF relevant to the product design and users, not the developers of these systems.

HF Integration in AV Development: The integration of HF in the automotive industry has emerged as a critical area of focus. Research, as highlighted in studies such as Lee et al. [6] and Hancock [119], underlines the vital role of HF in automated system design. This is especially relevant given the increasing complexity of technology and the imperative need for enhanced user safety. Researchers have studied various key aspects, including the interaction between AVs and human drivers [265], how humans engage with and disengage from these systems [266], the design of HMI [267], and the maintenance of situational awareness [268]. These studies are foundational in comprehending HF's role in elevating the user experience (UX), ensuring safety, and fostering trust and acceptance of AVs.

Despite this, the practical application of HF knowledge often remains limited to specific scenarios or design solutions, usually integrated into the design process or the requirements specification. A gap persists in understanding how HF research insights are effectively translated into the agile development processes of AVs, signaling an area ripe for further exploration and application.

Agile Methodologies and HF Integration: The challenge of integrating HF into agile development processes in the automotive sector remains a relatively under-explored area, with limited studies specifically addressing this context. Saghafian et al. [116] describe the conflict between the swift and adaptable nature of agile methodologies and the comprehensive, often time-intensive demands of HF practices in immersive visual technologies field. Steinberg and Grumman's study [22] offers valuable perspectives on modifying agile frameworks to better incorporate HF considerations. Muhammad et al. emphasize the role of cross-functional collaboration in successfully integrating HF [223, 253].

These research works highlight the challenges arising from inadequate collaboration and communication among engineering, design, and HF teams, which can hinder the effective integration of HF in development processes. They stress the necessity of establishing robust communication channels and collaborative structures within organizations to facilitate this integration. However, specific discussions on integrating HF in agile contexts, as outlined in our research, are relatively less explored and represent a contribution to the field.

Table 5.1: Examples on Human Factors and its impact on AV Design

Lane-Keeping: Designing and developing a lane-keeping feature presents substantial HF challenges. For instance, automated lane keeping should be designed to minimize the risk of motion sickness [256], while avoiding making interactions with other road-users uncomfortable - for the AV users as well as for other road-users. An example related to user discomfort is that the AV can position itself in the lane in a way that minimizes perceived risk [257]. Further, the implemented lane-keeping strategy (e.g., the level of lane-centering) affects the user's acceptance and trust in the system [258]. That is, HF requirements include ensuring the lane-keeping algorithm facilitates smooth and predictable movements [259] while incorporating algorithms to maintain safety during close encounters with other vehicles.
Lane change: Lane changes must be made in a safe, predictable, comfortable, and acceptable way, for both the users of the AV and for surrounding traffic [260]. Based on user studies, HF requirements may here be operationalized through functional requirements. That is, there may, for example, be specific timings of turn-signal initiations and trajectory shapes that make users (and surrounding traffic) feel that a lane change is predictable and feels comfortable, which in turn facilitates acceptance.
Collision Avoidance System Interaction: In scenarios where the collision avoidance system intervenes to prevent crashes, the timing and nature of the intervention are crucial. For example, nuisance interventions (i.e., when a user does not feel an intervention is warranted) should be kept to a minimum [261], while the intervention should be made early enough to avoid crashing. Considering driver comfort zone boundaries [262] in collision avoidance design can improve performance while keeping the nuisance intervention rate low. Therefore, the collision avoidance algorithm should strike a balance between early intervention (i.e., optimal crash avoidance) and minimizing driver nuisance (e.g., based on driver comfort zone boundaries). Examples of HF requirements in this case include seamless interventions to maintain passenger trust in the vehicle's safety capabilities, avoiding behaviors that may induce discomfort or distrust.
Driver Monitoring System: Driver monitoring systems can be used both to mitigate inattention and distraction in lower levels of automation [263], and to assess if the driver is ready to take over when an AV issues a take-over request (ToR) in higher levels of automation [264]. Both uses are intended to improve safety. For the former, designing warning or intervention strategies that minimize driver nuisance is crucial for acceptance, while it for both is important to consider driver variability and human capabilities. HF requirements on driver monitoring systems may therefore relate to how and when to provide information to drivers (e.g., warnings or ToRs).

5.2.2 Requirements Strategy and Proposition Formulation

The concept of a Requirements Strategy allows to reason on a high level of abstraction about how requirements engineering is covered in agile and hybrid software and systems development [111]. This is particularly applicable since all case companies are agile to some degree; thus, traditional processes do not suffice to express the concepts we need. Its primary purpose is to establish a clear and shared understanding of both the problem space and the corresponding requirements. A Requirements Strategy comprises three core building blocks: an organizational, a structural, and a work and feature flow perspective.

Organizational Perspective: This perspective in requirements management concentrates on defining teams' roles, responsibilities, and the ownership of different types of requirements. This aspect is critical for the effective integration and management of requirements in product development [111].

A crucial element of requirements management is the clear definition of roles and responsibilities. Wiegers and Beatty [269] emphasize the importance of clearly defining roles and responsibilities in the requirements management process. The significance of assigning clear ownership to each requirement is highlighted by Cagan [270], who argues that this clarity fosters accountability and greatly influences project success. Cockburn [271] discusses the importance of ownership in agile environments, arguing for a more collaborative and less 'siloed' approach. The literature vividly highlights the importance of role clarity and responsibility in requirements management. This understanding leads to the formulation of our first proposition, emphasizing the impact of clear ownership on HF requirements integration in product development:

- *Proposition 1:* Clear ownership and responsibility for HF requirements positively impact the integration of HF in product development

Paasivaara and Lassenius [272] discuss the integration of clear roles and responsibilities within agile methodologies, showing how agile practices can be adapted to improve the management of requirements in dynamic environments. Rubin [273] provides insights into the role of product owners in guiding agile teams, hinting to the importance of critical roles in clear ownership in product development. Aurum and Wohlin [274] contend that such assignments facilitate better decision-making and accountability, leading to more effective management of requirements and resource allocation, thereby contributing to the overall quality and success of the product. Thus, our literature review indicates that if organizations aim to support managing HF knowledge in a Requirements Strategy from an organizational perspective, HF expertise should be assigned to roles that are critical for the agile development workflow, such as product owners. We formulate our Proposition 2 as follows:

- *Proposition 2:* Clear ownership and responsibility for HF requirements requires to assign HF expertise to critical roles

Structural Perspective: This perspective focuses on defining requirements levels, types, and traceability demands, ensuring clarity and structure in how requirements are approached and documented [111].

Hull et al.[128] emphasize the pivotal role of clear and structured requirements for successful project outcomes. This approach was further refined by Davis [275], who differentiated user and system requirements, thus initiating a nuanced understanding of requirements levels. Wiegers and Beatty [269] expanded on this by categorizing requirements into business, user, and software types, offering a comprehensive framework for large-scale project management. The emphasis on traceability in the structural perspective aligns with the principles outlined by Gotel and Finkelstein [276] and Rupp et al. [277]. They advocate for bi-directional traceability, ensuring that each requirement is linked to its origin as well as the implementation. Literature underscores the importance of well-structured requirements, emphasises traceability, and requirements decomposition. This need for clarity and structure in managing requirements structurally informs our next set of propositions:

- *Proposition 3:* Clear HF requirements structure positively impacts the integration of HF in product development
- *Proposition 4:* Clear HF requirements traceability positively impacts the integration of HF in product development
- *Proposition 5:* Clear decomposition of HF requirements with respect to organizational levels positively impacts the integration of HF in product development

Work and Feature Flow Perspective: This perspective focuses on defining the lifecycle of requirement types and mapping them to the workflow, ensuring that requirements are effectively integrated into the agile development process [111].

The significance of adaptability and iterative development in managing complex requirements is a cornerstone in this context. Authors such as Van Der Vyver et al. [278] and Schwaber and Beedle [279] highlight the importance of being responsive to changing requirements. Cohn and Ford [280] further elaborate on this by discussing the dynamics of incorporating requirements into agile processes, setting a foundation for understanding the adaptability of agile frameworks to these changes. Schwaber and Beedle emphasize the importance of well-defined requirements for the success of agile projects, highlighting the need for a common understanding among stakeholders and interdisciplinary collaboration [279, 281]. Rubin further elaborates on this by presenting a model for managing requirements' lifecycle in agile environments, stressing continuous integration and iterative development as central to agile methodologies [273]. Moreover, research on project management practices highlights that a clear definition of work streams improves project outcomes [282]. The role of review and reflection is highlighted as a critical element in agile development. Sutherland and Schwaber [283] emphasize the importance of regular review meetings, such as Sprint Reviews, for assessing progress and integrating evolving requirements. These reviews are evaluative and play a crucial role in adapting and refining requirements based on ongoing feedback. Leffingwell [284] extends this notion by advocating for a systematic review strategy to enhance requirements quality and maintain team engagement with evolving changes, reinforcing Schwaber and Beedle's focus on continuous improvement [279].

In summary, literature underscores the dynamic nature and lifecycle in agile development. Additionally, the importance of defining clear work streams and the need for regular reviews and reflections are emphasized. These insights inform our final propositions:

- *Proposition 6:* A strong lifecycle model of HF requirements positively impacts the integration of HF requirements in product development
- *Proposition 7:* A clear definition of a HF work stream in relation to other work streams positively impacts the integration of HF requirements in product development
- *Proposition 8:* A clear plan for review and reflection of HF requirements positively impacts the integration of HF requirements in product development

5.3 Methodology

This study adopts a qualitative research design, utilizing a priori coding to analyze semi-structured interview data. The approach was chosen to systematically categorize and interpret the experiences and insights of professionals in the automotive industry. Central to our method was the formulation and examination of propositions (Section 5.2.2) for each of our research questions, which guided our data collection and analysis process.

5.3.1 Data Collection

Semi-Structured interviews: Data were collected through 13 semi-structured interviews. This format was specifically chosen to probe the propositions related to each research question. Each interview lasted approximately 50-60 minutes. Given our intent to include companies from various global locations, virtual interviews were the most feasible approach. Consequently, 11 interviews were conducted using Microsoft Teams or Zoom video calls, while 2 interviews were conducted in person. The interviews were recorded with the participants' consent and later transcribed verbatim for analysis.

Prior to the interviews, we provided details about the current study objectives, mutual expectations and high-level questions. Each interview began with an introduction to the authors and an overview of the project, outlining the study's purpose and objectives. We then started with demographic questions, followed by more targeted questions about the integration of HF in automotive development, directly relating to our propositions. Interview questions were grouped in relation to the building blocks of Requirements Strategies, and in each block we asked for an assessment on how well the interviewee experiences their context to perform with respect to these aspects. The interview questions used can be found here. All participants have experience working within agile methodologies to varying degrees. While some have fully embraced agile practices, others have incorporated agile methodologies to a lesser extent.

Table 5.2: Interviewees' roles and work experience (Experience level: Low= 0–5 years, Medium=5–10 years, High= More than 10 years)

ID	Role	Experience	Supplier/ OEM	Agile Methodologies Usage
01	Senior Manager	High	OEM	Some projects adopt agile methodologies, while others follow traditional approaches
02	Technology Leader HF & Automation	High	Supplier & OEM	Some projects adopt agile methodologies, while others follow traditional approaches
03	Senior HF Specialist	High	Supplier & OEM	Using SAFe methodology
04	Engineering Manager	High	OEM	Using agile practices
05	HF expert	Medium	Supplier	Agile methodologies are employed for software aspects, while hardware components follow a more traditional approach
06	Chief Expert HMI	High	Supplier	Utilizes agile methodologies variably across projects and business units
07	HF Expert	High	Supplier	Using agile practices
08	Software Developer	High	OEM	Fully committed to SAFe methodology
09	Req. Eng. Researcher & Tester	High	OEM	Operates in a hybrid manner
10	HF Specialist & Automation and control engineer	Low	OEM	Embraces agile approaches in their tasks, although the organizational structure remains non-agile
11	AD Safety Specialist	High	Supplier	Engages in partial agility, incorporating agile practices, particularly in project execution, but not fully aligned with agile principles in strategic aspects
12	Senior Technical Specialist	High	OEM	Majority of projects are developed using agile methodologies
13	Global Agile Process Lead	High	Supplier	Adopted a framework based on SAFe principles

Sampling Strategy: Our selection criteria included respondents with expertise in HF who are employed in the automotive industry. Specifically, we focused on those experts who have a close working relationship with requirements or process design. Consequently, all our participants had knowledge of HF. Given our broader focus on investigating the integration of HF expertise in development processes in general, we did not delve into detailed discussions about specific types of HF. Instead, we relied on participants' overall understanding of HF and their involvement in key aspects such as requirements and design. This approach enabled us to capture a broad perspective on the role of HF across various domains within the automotive industry.

The sampling included individuals who could provide in-depth and specialized knowledge pertinent to our research objectives - based on the author contacts and by using LinkedIn to screen for potential candidates. We reached out to professionals from various automotive companies, selecting those who met our criteria and were willing to participate. This approach ensured rich, qualitative data from a diverse range of experts in the field.

In total, we conducted 13 interviews with professionals from 12 different companies, including industry leaders like Mercedes-Benz, and Volvo. Each participant was assigned a unique identifier (ID01 through ID13) to maintain confidentiality. In order to recruit these 13 participants, we contacted 39 suitable candidates, often with several reminders. We believe that the low response rate may indicate a particularly high workload among the population of suitable candidates. We would have preferred more participants to ensure that theoretical saturation was reached but had to stop our recruiting efforts. From the analysis of our results, we do not believe that additional interviews would significantly change our results (later interviews tend to bring up sentiments that were similarly mentioned before) and report our findings now to enable future research in this important field. An overview of these participants, highlighting their roles and levels of experience, can be found in Table 5.2.

5.3.2 Data Analysis

The data analysis was structured around our a priori coding framework, with three codes per proposition (supports, neutral, rejects). These codes were applied to the interview transcripts through line-by-line reading, assigning relevant codes to specific statements. The initial codes were identified before data collection and were refined as the analysis progressed. We adopted Saldana's guidelines for coding as outlined in [285]. The analysis process involved:

[a] Familiarization with the data through repeated reading of the transcripts

[b] Applying the a priori codes to the data

[c] Ensuring consistency across transcripts and refining the coding framework/guidelines as necessary

[d] Interpretation of the data in the light of our propositions and the broader study context

The use of a priori coding facilitated a structured and theory-driven analysis. This approach not only ensured a systematic examination aligned with the study's goals but also accommodated the emergence of new insights.

To ensure uniform understanding and application of the codes, two of the authors were involved in coding. While the first author coded all data, a second author coded a subset until we were satisfied with the inter-rater agreement. For this purpose, we relied on Cohen's Kappa [286] and iterative coding, evaluation, and improvement of coding guidelines until we reached substantial consensus (Kappa above 0.6).

5.3.3 Research Validity

To ensure the validity of the study, careful consideration was given to four key aspects: credibility, transferability, dependability, and confirmability.

Credibility, which pertains to the truthfulness and believability of the findings, faced potential threats due to the possibility of participant bias in semi-structured interviews. Participants might tailor their responses based on what they perceive the researcher wants to hear, which could lead to skewed data. To mitigate this threat, we ensured to have sufficient time to reach an understanding with interviewees and by asking similar questions from different perspectives.

Transferability, or the extent to which the study's findings can be applied to other contexts, was challenged by the unique and specific contexts of the professionals interviewed. Since these individuals might not represent the entire automotive industry, there was a risk of limited generalizability. To mitigate this threat, we collected demographics and additional information about each interviewee's context. We use this information in our interpretation of the results and discussion of their implications.

Dependability, which concerns the stability of the data over time, was threatened by the rapidly evolving nature of the automotive industry. As the industry changes, the study's findings might lose relevance. To ensure dependability, a detailed documentation of the research process and the decision-making in coding and theme development was maintained. This approach not only enhances transparency but also allows the research process to be replicated and critiqued by others. Additionally, consistency in data collection and analysis methods was rigorously maintained.

Finally, confirmability, the degree to which the findings are shaped by the respondents and not researcher bias, was a concern. The subjective interpretation of data by researchers could potentially lead to biased conclusions. This was addressed by maintaining a reflexive journal throughout the research process, documenting our thoughts, reflections, and potential biases. We also ensured reliability of coding, by leveraging inter-rater agreement for iterative improvement of coding guidelines. Furthermore, there is a threat that interviewees and researchers have different understandings of critical aspects, such as HF. We tried to mitigate this threat through careful sampling and by adopting a dialogue-based interview style rather than a strict question-answer style. This allowed us to identify potential misunderstandings early and to recover during the interview. These practices helped in maintaining objectivity and transparency in the research.

5.4 Findings

The summary of the interviewees' stances on our propositions, along with a representative quote, can be found here. Figure 5.1 offers a visual overview of the findings. In the figure, we derive the solution space from our findings below.

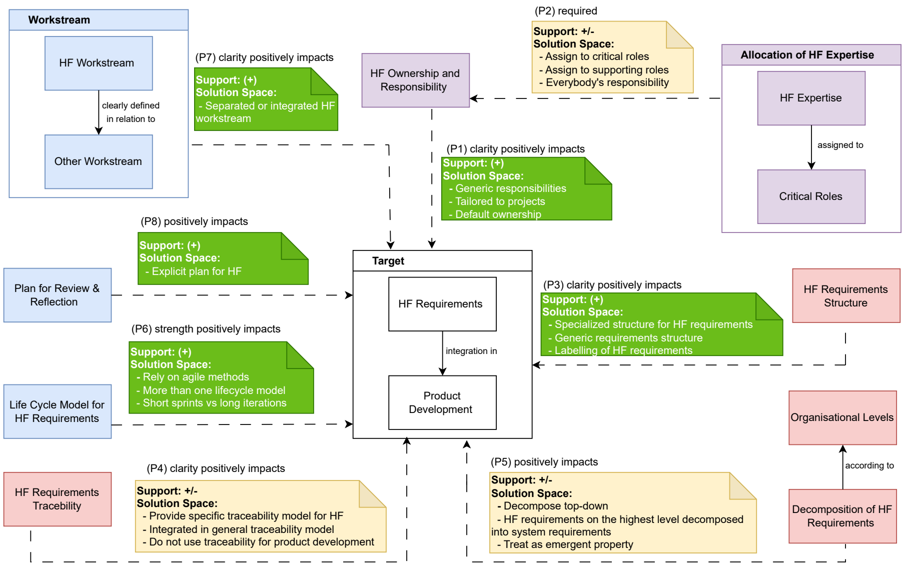


Figure 5.1: Overview of findings: Propositions P1 to P8 in relation to RQs (in different colours: RQ1 - purple, RQ2 - red, RQ3 - blue); propositions for which interviewees expressed a mix of positive and neutral sentiments have green summary boxes, while propositions that also received negative sentiments have yellow boxes.

5.4.1 RQ 1: How do ownership and responsibility for HF requirements impact the integration of HF requirements in product development?

The first research question addresses the organizational aspect of the Requirements Strategy and investigates propositions related to the impact of clearly defined ownership and responsibility on the integration of HF in AV development.

Proposition 1: Clear ownership and responsibility for HF requirements positively impact the integration of HF in product development: The findings **support the proposition**, with mostly positive, but also neutral sentiments among interviewees.

Many interviewees highlighted defined roles and responsibilities that positively impact the integration of HF in product development, such as the role of a product owner, specialized HF expert or HF team. These roles, as described by interviewees, foster clarity in communication, responsibility allocation, and overall integration of HF into the development cycle.

For instance, the beneficial influence of assigning HF responsibility to a product owner is illustrated by ID01's experience, where a product owner with UX expertise was able to effectively integrate HF into the development process:

> *"We hired a product owner who is also a UX person, so she is halftime product owner, half-time UX... So they're pulling it...the team is basically specifying those requirements."* — ID01

Similarly, ID04 described a situation where the product owner sets general priorities, while specialists like designers handle detailed aspects of HF. This clear division of responsibility and ownership is exemplified by direct communication between designers and engineers about specific details:

> *"So the priorities, the overall topics were decided by the product owner, but when it came to the nitty-gritty details, it was directly the designer coming to the engineers..."* — ID04

However, several interviewees maintained a neutral stance, discussing various aspects of HF integration and role responsibilities without explicitly supporting or rejecting the proposition. For example, ID05 commented on the integration of HF optimization into developers' responsibilities, indicating a more distributed approach to HF management.

> *"Optimization with respect to HF is a task that becomes integrated into the developer's responsibilities."* — ID05

Overall, the data supports Proposition 1, showing that clear ownership and responsibility for HF requirements positively influence the integration of HF in product development. However, the effectiveness of this integration may vary depending on the organizational structure and specific approaches adopted by different teams or departments.

Proposition 2: Clear ownership and responsibility for HF requirements requires to assign HF expertise to critical roles: Complementary to Proposition 1, we investigate here whether ownership and responsibility should be assigned to critical roles in the development lifecycle, by making sure that these roles possess HF expertise. In this context, 'critical roles' refer to key positions that significantly influence the direction and success of a project. These roles typically include positions like product owners, portfolio owners, or other decision-making positions.

The interview data provides a **nuanced view** on the proposition that assigning HF expertise to critical roles is pivotal for clear ownership and effective management of HF requirements in product development. Most interviewees are positive or neutral, but there are two interviewees that take a rejecting stance.

The support to this proposition is evidenced by numerous instances where integrating HF expertise into critical roles enhanced the management and implementation of HF requirements. The interviewees reveal a trend of merging HF expertise with critical roles such as product owners. For instance, ID13 emphasizes the importance of product owners possessing HF knowledge to effectively guide development teams:

> *"That product owner... should then also consider... human factors and the human behaviors."* — ID13

Similarly, ID08 discusses the responsibilities of portfolio owners:

> *"The portfolio owner, overseeing a range of products, plays a crucial role at a very high level. Their responsibility is not just to ensure that we meet customer requirements, but also to define and offer products that deliver value to customers."* — ID08

While designers and engineers can also be considered to be critical roles, we did not count such sentiments as clear support, since the need to have HF expertise covered so broadly shows that, depending on the context, it is more complicated than to just centralize it at selected critical positions. This interpretation is also supported by other responses that hint towards a distributed approach to HF responsibility, challenging the notion that clear ownership and responsibilities for HF requirements should be confined to specific critical roles. As for example, mentioned by ID12 and ID05, which contradicts the proposition's focus on assigning HF expertise to specific critical roles.

> *"The human factors team... they are responsible for all of those elements... including in some cases the design of the actual system itself."* — ID12

> *"You cannot really form a group of people, only doing this... Everybody could have some opinion on HF issues."* — ID05

In summary, our data presents a nuanced perspective. While there is substantial support for the proposition that assigning HF expertise to critical roles enhances the management of HF requirements, evidence also points to the effectiveness of a distributed or collaborative approach in certain contexts. This suggests that while assigning HF expertise to critical roles can be beneficial, it may not be the only effective approach for managing HF requirements across different organizational settings.

5.4.2 RQ 2: How does requirements structure and requirements information model impact the integration of HF requirements in product development?

The second research question examines how the structuring of HF requirements and the development of information models influence the integration of HF in AV development, assessing the structural part of the Requirements Strategy.

Proposition 3: Clear HF requirements structure positively impacts the integration of HF in product development: We find this proposition **rather supported**, with four positive and eight neutral sentiments among interviewees, but no rejections (one skipped this topic).

Several interviewees highlighted the beneficial impact of a well-defined requirements system on integrating HF in product development. The frequent mention of tools that are used to document requirements and tasks related to requirements for HF such as JIRA, Doors, and Code Beamer underscores their essential role in organizing HF requirements. These tools are key in establishing a structured environment conducive to HF integration. For example, ID04 emphasizes the effectiveness of a requirements tool for documenting HF aspects, indicating its importance in integrating these factors into product workflows. This highlights the crucial role of structured documentation systems in embedding HF considerations into product development processes.

> *"In this tool, we documented all the requirements [including design and HF]. We used this as a documentation tool. I'm not sure if that's good or bad, but that's how it was used... I think that the requirements structure worked really well."* — ID04

Similarly, ID07 described a structured documentation practice for HF studies, indicating a systematic approach that could enhance HF integration in development processes.

> *"We communicate through these channels, and then we maintain comprehensive documentation of all work that has transpired, including detailed accounts of how studies were conducted and their findings and results."* — ID07

ID12's shift towards a human-centered design philosophy, with centrally managed requirements, aligns with the proposition, indicating the benefits of a clear, organized structure.

> *"We have implemented a requirements management system for storing all requirements... [This system is part of] centralizing our design philosophy around being human-centered... everything now stems from this centralized approach..."* — ID12

In contrast to those who highlight explicit consideration of HF and human-centric considerations when designing the requirements documentation structure, other interviewees indicate that a generic requirements structure and tool environment can be sufficient or even preferred. This suggests that specific support for HF considerations is not needed or not beneficial in certain contexts.

> *"Partly they are part of the overall documentation environment...there should not be any specific other kind of documentation."* — ID06

Moreover, ID13 speaks to the inclusion of HF within a unified requirements database, lending support to the proposition through a comprehensive approach.

> *"I would say that it is a part of all other requirements in the requirement database."* — ID13

Conversely, some responses highlight challenges in the current documentation systems, suggesting that clear requirements structures are not universally effective or consistently utilized. ID09, for instance, points out a deficiency in HF documentation:

> *"Human factors requirements are typically not documented explicitly upfront."* — ID09

ID05 addresses challenges concerning the flexibility and practicality of detailed requirements structuring and documentation systems, emphasizing the extra workload for developers and the difficulty of adapting them in rapidly evolving, innovative settings.

> *"Documentation is seen as extra work for the developers... everyone deems it as an added task, leading to a general reluctance towards anything additional."* — ID05

The data presents a composite view. Evidence supports the proposition that a clear requirements structure facilitates the integration of HF into product development, as evidenced by the formal and centralized approaches of some interviewees. This is also indirectly supported by quotes that indicate difficulties when a clear requirements structure is missing. It is however unclear whether this structure must explicitly accommodate HF considerations or should be generic in nature. The diversity in practices among interviewees suggests that the impact of a clear requirements structure might be influenced by other variables, such as organizational culture or the unique nature of the project.

Proposition 4: Clear HF requirements traceability positively impacts the integration of HF in product development: The findings indicate a diverse range of practices regarding HF requirements traceability across various organizations. This variability reflects differences in both the implementation and the perception of the importance of traceability in integrating HF into product development.

For example, ID01 notes the use of traditional documentation and linking with testing activities, which implies a level of traceability that could support HF integration.

> *"That describes HF requirements and then from those documents there's some testing... So there's a process for that."* — ID01

ID06 describes a careful documentation process for high-complexity systems, indicating that in certain contexts, there is clear traceability of requirements, including HF aspects:

> *"If we are looking at a, let's say a part with the compnay's responsibility and higher complexity..... It [traceability] is extremely carefully taken on all steps of the requirements, it's in the setting...[this includes] clear documentation of possible issues, records of test outcomes, software version histories, and all these things."* — ID06

A notable number of interviewees remained neutral. Their responses ranged from acknowledging the use of tools like JIRA for managing requirements to expressing indifference towards the specific traceability of HF requirements. As, for example, ID08 reflects on their organizational structures, which, while not specific to HF, are applied to all requirements. This approach suggests an integration of HF requirements into existing models, but the effectiveness and clarity of these traceability methods are not specified:

> *"SAFe gives that structure in general, it's not specific for HF, but we applied the same structure for all requirements."* — ID08

Moreover, ID10 noted the absence of specific traceability models or processes for HF requirements. This lack of specialized traceability mechanisms could imply a gap in effectively integrating HF in product development.

> *" We don't have any traceability whatsoever."* — ID10

In contrast to the indicators above, some interviewees suggest that traceability does not directly impact the integration of HF in product development. For example, ID05 and ID07 argue that traceability is used for system level reporting and not directly useful for product developers, effectively rejecting our proposition.

> *"...traceability...more or less only for reporting, not useful for developing."* — ID05

> *"The levels, how you define this is more or less useless for the development."* — ID07

In summary, the mixed responses from the interviewees illustrate the **varied perspectives** on the role of traceability in HF requirements integration. While some see clear benefits in traceability for ensuring thorough documentation and process adherence, others view it as having limited practical value in the actual development process. This diversity in opinions suggests that the effectiveness of HF traceability may be highly contextual and dependent on the organizational practices and the nature of the product development processes.

Proposition 5: Clear decomposition of HF requirements with respect to organizational levels positively impacts the integration of HF in product development: The interview data presents a **mixed perspective on the Proposition**.

Several interviewees expressed support for the proposition, indicating that decomposing HF requirements across different organizational levels is crucial for their effective integration into product development.

For example, ID12's insights shed light on the decomposition of various requirements under a human-centered design philosophy, which reinforces this proposition. Their approach suggests that systematically addressing HF requirements substantially enhances their integration into product development.

> *"At the highest, we go quite deep into the human and understand the requirements of the human and whether they be physical or cognitive and then they are used to guide specific requirements which are then decomposed into the engineering requirements that are used to deliver the car....So the typical and hierarchical level approach"* — ID12

Similarly, ID13's emphasis on iterative design and user stories further aligns with this perspective, indicating that decomposing HF requirements into manageable parts facilitates improved understanding and implementation:

> *"Then we can quickly move forward and break that down into the different level of requirements. Implement, test, and validate and come back to you in two or three weeks."* — ID13

ID09 details a clear multi-level decomposition process, reinforcing the proposition:

> *"We treat our initial requirements as level one. These are then further decomposed into level two, or system requirements, which may differ from the conventional use of the term. Sometimes, there's even a third or fourth level for specific needs like FPGA. Generally, we operate with these three levels"* — ID09

However, ID05 and ID08 take a rejecting stance and noted challenges or alternative perspectives. ID05, for example, points out the complexities of decomposing HF requirements in innovative fields like autonomous driving:

> *"In at least in the new area, in the autonomous driving area... You cannot really decompose the requirement at the beginning because you don't know what the technique can do, what the sensor can see, what the system is capable of."* — ID05

ID08's observations suggest variability in the application of structured decomposition across different organizations:

> *"I would say we don't have the structure as you would probably want to see it...At the moment we don't have a standardized taxonomy or something like that."* — ID08

Some interviewees remained neutral, neither supporting nor rejecting the proposition outright. As, for example, ID07's comment implies a more adaptive rather than structured approach to HF integration.

> *"We sort of adapt to it and then we try to decompose according to the order"* — ID07

In summary, the mixed responses highlight the complexity and contextual nature of integrating HF requirements in product development. While a structured decomposition approach is favored by some, the diversity in organizational practices and the abstract nature of HF requirements pose challenges to its universal application. These findings suggest that while the proposition may hold value, its applicability and effectiveness are likely contingent on the specific context of the organization and the nature of the product development process.

5.4.3 RQ 3: How does defining a work and feature flow related to HF requirements influence their integration in product development?

This question evaluates the third aspect of requirement strategy, i.e., the work and feature flow perspective.

Proposition 6: A strong lifecycle model of HF requirements positively impact the integration of HF requirements in product development: Our findings **support Proposition 6** (7 positive, 5 neutral stances among interviewees), particularly highlighting the benefits of integrating agile practices. Our analysis reveals a trend towards the integration of HF requirements within agile and iterative development processes, reinforcing the notion that a robust lifecycle model of HF requirements is instrumental in enhancing product development. For example, ID01 provided insight into their organization's structured approach, illustrating the systematic adoption of agile methodologies:

> *"We're doing we have the set of meetings, there's the planning...understanding requirements, implementing, testing, and integration., get feedback and iterate based on that."* — ID01

This approach underscores the efficacy of agile methodologies as integral components of the lifecycle model, particularly in their role in facilitating the integration of HF requirements. The iterative nature of agile methodologies, as pointed out by our interviewees, enables the seamless incorporation of various specialties, including HF. ID13 elaborates on this process:

> *"In an agile way of working, you're working incrementally and iteratively. So we're doing one functional feature at a time, we use different modeling and simulation before we have the hardware available to us...you're doing some of the HF requirements depending on what you're developing next. [. . .] Well, it works very well with incrementally in an iterative approach."* — ID13

The emphasis on early decision-making and continuous feedback loops, as observed by ID12, further corroborates the value of a well-structured lifecycle model for HF integration:

> *"Lifecycle model involves early-stage requirements capture, evaluation, and iterative feedback... Separates design and delivery aspects....Significant architectural decisions, some of the big decisions that really have to be defined early on and not changed and made with always people being involved from this part of the process."* — ID12

However, some interviewees expressed neutral stances, indicating variability in the application of lifecycle models or the complexity of the project. For instance, ID10 noted:

> *"The process isn't as intuitive or straight- forward. It often requires a significant amount of repeated effort, depending on the complexity of the issue this iterative process involves a few steps for simpler problems but becomes more detailed when tackling complex issues."* — ID10

In summary, the data support the proposition that a robust lifecycle model for HF requirements positively influences their integration in product development, as evidenced by the majority of interviewees. However, the neutral responses indicate variability and complexity in the application of these models, suggesting that while a strong lifecycle model is generally beneficial, its implementation and effects may vary depending on specific project contexts and methodologies.

Proposition 7: A clear definition of a HF work stream in relation to other work streams positively impacts the integration of HF requirements in product development: Our findings provide **partial support for Proposition 7**, as three interviewees supported the proposition while eight others maintained a neutral stance.

Some interviewees suggest that when HF is clearly integrated into the workflow, particularly in agile environments, it is more effectively considered in product development. For instance, ID02 emphasizes that agile methodologies facilitate the integration of various specialists, including HF specialists. This is evident from the observation,

> *"Human factors specialist and HF requirements should be part of the regular work, together with everything else, we should not keep it separate because that's what we have today. Today we are in separate, we have this, these silos. We want to break silos with agile way of working."* — ID02

This statement supports the proposition, as it demonstrates that clear and integrated work streams, such as those found in agile environments, are conducive to incorporating HF requirements effectively. Similarly, ID13's perspective reinforces this view by advocating for a unified approach where HF is integrated along with other critical functions:

> *"There are no separate work streams for HF. The aspects are all the same, we don't have separate works streams for functional safety or for cyber security or anything else like that."* — ID13

However, some interviewees expressed a neutral stance, indicating a diverse implementation of HF integration in their processes. For instance, ID04's comment reveals a gap between intention and execution:

> *"The developers implementing the code always wanted to have the human factor requirements... the designers did not work with our PI planning."* — ID04

ID03's experience further illustrates the challenges of integrating HF into other workstreams, hinting at possible technical complexities:

> *"It's integrated, but sometimes I feel it's very technical."* — ID03

Moreover, we learned that the impact of clearly defined HF work streams on their integration appears to be context-dependent, varying across different organizational structures and project types. ID12's experience illustrates this variability:

> *"In some cases, separate teams for HF run their own sprints, typically delivering specifications because we don't need external resources. Alternatively, people work with joint teams to develop requirements that consider all aspects, including mechanical, electrical, and others."* — ID12

In summary, while the data provides evidence that clear definitions and structured work streams in agile methodologies can improve the integration of HF in product development, there is also an understanding that its effectiveness can vary based on project dynamics and organizational frameworks. Notably, none of the interviewees explicitly rejected the proposition.

Proposition 8: A clear plan for review and reflection of HF requirements positively impacts the integration of HF requirements in product development: Our findings **support Proposition 8**, with seven interviewees taking a supporting stance and five others taking neutral stances. Our interviewees highlighted practices and perspectives that align with the importance of a structured approach to reviewing and reflecting on HF requirements. This support is evidenced by the emphasis on agile flexibility, the importance of planning and communication, and the iterative improvement processes discussed by various interviewees. Iterative processes and continuous improvement, central to agile methodologies, are also highlighted as beneficial for integrating HF. For instance, ID12's emphasis on retrospectives underscores this point:

> *"We'll conduct retrospectives and discuss the process, people, and output. The main feedback, however, is whether what we've created works, is usable, and if people like it. Does it feel intuitive?... Continuous improvement is key. At the end of the day, it's about spending time, doing the work, reviewing, reflecting, and improving for the next time."* — ID12

Similarly, ID13's mention of sprint reviews and the role of retrospectives as a platform for stakeholder engagement to reflect on HF requirements indicates a structured approach:

> *"The Sprint review is the time and place for the team to invite stakeholders to review and reflect on the human factor requirements. As I said, this should preferably be done with the stakeholders. The team should work as closely as possible with them, regardless of who the stakeholders are."* — ID13

These insights illustrate the positive impact of regular review and reflection in aligning product development with HF requirements. ID07's comments further support this, illustrating a formalized review process:

> *"When you complete a study or project, you usually create an HF requirements report deliverable. It's reviewed by the authors or peer-reviewed, and then you receive feedback, addressing minor or major comments. We also typically have lessons learned, discussing improvements for future projects."* — ID07

Some interviewees did not explicitly support or reject the proposition. As for example, ID09 highlights the importance of the review and reflection processes but does not directly link these to the positive integration of HF requirements.

> *"We review and reflect overall for the workflow and testing."* — ID09

In summary, the data predominantly supports the proposition that a clear plan for review and reflection of HF requirements is beneficial for their integration in product development. The emphasis on iterative feedback, stakeholder involvement, and retrospective analyses by several interviewees underscores the value of reflective practices in effective product development. However, the presence of neutral stances suggests that the impact of such practices may vary depending on other factors, such as team dynamics, project nature, or organizational contexts.

5.5 Discussion

Our findings have been obtained from a qualitative analysis of semi-structured interview data based on the lens of Requirements Strategies. In this section, we discuss how these findings are contextualized within existing literature, drawing parallels and contrasts with strategies in related fields. Our findings indicate that while every company adopts agility to varying extents, they do so in distinct ways tailored to their specific contexts. Consequently, these results may have implications for organizations that go beyond 'traditional' processes at both system and organizational levels.

Impact of Ownership and Responsibility on HF Integration: The results indicate general support confirming that clear ownership and responsibility (Proposition 1) benefit HF integration in product development, resonating with Smith and Reinertsen's [126] observations on role clarity. However, the neutral stances suggest a nuanced application, dependent on organizational context and project nature. This aligns with the findings of Cockburn [271], who highlight the impact of organizational culture on the adoption of agile practices. For Proposition 2, the opinions on assigning HF expertise to critical roles were divergent. Hence, we conclude that this is not the sole effective approach. The diversity in responses implies that either a distributed approach to managing HF expertise or that a combined approach that distributes HF expertise generally over the development organization, and at the same time ensures that critical roles can utilize it in their decision making can be effective.

Impact of Requirement Structuring and Information Models on HF Integration in AV Development: The mixed perspectives on the impact of clear requirements structure on HF integration (Proposition 3) echo the broader discourse in software engineering [127], where structured approaches, while valuable, must be flexible enough to accommodate the dynamic nature of product development, as noted by Cooper et al. [287]. This also reflects the tension in software development expressed by the controversial agile value of working software over comprehensive documentation [112, 288]. Our findings contradict literature such as Hull et al. that describes the clarity and structure of requirements as pivotal for successful project outcomes [128], at least in the context of our investigation. We believe that due to the disruptive nature of AV technology, some of our interviewees are suspicious about a structure that may prevent them to explore emergent HF properties. The fact that not all areas of automotive development are equally disrupted may explain why our interviewees come to different assessments.

For Proposition 4, regarding traceability of HF requirements, the diversity of viewpoints underscores the practical challenges of maintaining traceability in various organizational contexts. This reveals that the effectiveness of traceability in HF integration may not be universally recognized, suggesting a gap between theoretical expectations [276, 277]and practical realizations. The findings on the decomposition of HF requirements in Proposition 5 show varied support. The iterative approaches to requirement decomposition discussed by interviewees reflect systems engineering best practices for breaking down complex requirements into manageable parts for efficient implementation. However, other interviewees felt that requirements decomposition does not work well in innovative domains, where traditional decomposition schemes may not fit well. Similar effects have also been reported in literature, e.g. by Hoda et al. [289].

Impact of Work and Feature Flow on HF Integration in AV Development: The findings regarding the lifecycle model of HF requirements (Proposition 6) support the positive impact of the integration of HF in product development. This aligns with agile methodologies which emphasize adaptability and iterative development for complex requirements, as discussed in Van Der Vyver's et al. [278] and Schwaber and Beedle's [279] work. Our study shows both positive and neutral sentiments among interviewees, which suggests diverse approaches to integrate HF into the development, influenced by factors like project size, nature, organizational culture, and scope [290].

The feedback on Proposition 7 highlight the debate on integrating versus separating HF activities in agile processes, echoing Cockburn and Highsmith's view on agile methodologies' adaptability across organizational contexts [131]. Integrating HF with other workstreams may improve recognition and integration, but might also limit its scope to specific areas. Conversely, external

HF expertise may have a broader reach across teams, but have less influence on decision-making. This balance between specialized knowledge and cross-functional collaboration is also discussed by Highsmith [281].

Lastly, the emphasis of Proposition 8 on the importance of a clear plan for review and reflection of HF requirements is supported by Sutherland and Schwaber's insights [283] into continuous improvement and iterative evaluation in agile development. The mix of positive and neutral responses indicates differences in how organizations implement these processes, ranging from specialized review approaches to relying on generic reviews schemes. This divergence likely stems from their distinct cultural foundations. For example, in firms where a human-centric design is deeply embedded, additional reporting may be viewed as redundant. However, we believe that in companies still developing this culture, detailed reports could be crucial for increasing awareness and driving cultural change.

Overall, the lens of Requirements Strategies allowed for meaningful discussions with experts, particularly benefiting companies transitioning away from purely process-driven approaches. The fact that we find a wide spectrum of different approaches for each proposition, and even a certain level of disagreements with the structural propositions, shows that a suitable Requirements Strategy would have to be defined for each specific context. For this reason, we derive different solutions from our interviews, to indicate the solution space for organizations that aim to integrate HF requirements into AV development. We believe that our findings will help organizations to explore this solution space and make good, consistent decisions. We are certain that this will help to systematically cover HF concerns in AV development.

5.6 Conclusion

In conclusion, this study contributes to the understanding of how HF requirements can be effectively integrated into AV development, examined through the lens of Requirements Strategies. Adopting agile methodologies has proven promising in addressing complex HF requirements through iterations and feedback loops, yet it becomes unclear how HF requirements can be systematically managed in agile scopes. This tension highlights the crucial role of a consistent Requirements Strategy in the successful integration of HF knowledge into the agile development of AVs. Our findings underscore the necessity for organizations to adapt their strategies to the unique demands of the development environments. In particular, they indicate that clear ownership and responsibility for HF requirements enhance their integration into product development, resulting in more streamlined and cohesive workflows. While foundational, the structuring and traceability of HF requirements also necessitate a balance between rigidity and flexibility. Both, organizational matters and structuring of requirements are fundamental to integrate activities related to HF requirements into agile workflows. Along these three components (organization, structure, workflow), our study identifies various solution spaces indicated in the findings, reflecting the diverse approaches employed within the industry to integrate HF requirements. We observed that while certain practices are effective, the effectiveness of others may vary, highlighting the need for future research and the development of tailored approaches that consider the unique organizational contexts.

Chapter 6

Paper E

Integrating Human Factors Expertise into Development of Automated Vehicles

Amna Pir Muhammad, Alessia Knauss, Eric Knauss, Jonas Bärgman

In submission to Empirical Software Engineering Journal.

Abstract

The contemporary development of automated vehicles (AVs) often strives to combine plan-driven systems development with agile approaches. Incorporating human factors knowledge is crucial for developing safety-critical systems like AVs. Continuous integration of this knowledge throughout the AV development process is essential for efficacy, safety, and acceptance of these advanced systems. However, agile development methodologies present challenges for this integration, including the absence of proper processes, lack of awareness, and a shortage of human factors expertise. Recent studies have proposed solutions to integrate human factors as requirements in agile development. This study goes a step further by exploring the role of human factors requirements at different levels of requirements abstraction, ranging from high-level goals to specific, detailed requirements. It also aims to identify the optimal placement of human factors experts within an organization to manage these requirements effectively and better integrate human factors knowledge in the development.

In our research, we employed a mixed-methods approach. Our data collection techniques included shadowing techniques, document analysis, informal interviews, workshops, and a survey. Our results indicate that human factors requirements cannot be confined to a specific high level of abstraction but instead must be managed on various levels of abstraction throughout.

However, considering the lack of human factors expertise, we need to strategically decide where to place human factors experts to efficiently manage these requirements at each level of abstraction and maximize their benefit on the product. We evaluated the placement of human factors experts from three perspectives: general ranking, effectiveness, and ease of implementation. Our results showed that *feature requirements* and *user experience teams* were the most preferred placements in the general ranking for maximizing product impact. These teams were also considered the most effective. In terms of ease of implementation, the *user experience teams* and *dedicated human factors teams* emerged as the top choices.

6.1 Introduction

Automated vehicles (AVs) are revolutionizing the automotive industry, and the role of human factors in their development has become increasingly crucial [58, 60]. Human factors knowledge encompasses various behavioral, physiological, and psychological aspects of humans, aiming to provide pleasure, satisfaction, and well-being[199]. By adhering to human factors principles, AVs can be designed to offer users a safe, intuitive, and enjoyable driving experience.

Traditionally, human factors considerations were integrated during the planning and pre-development phases in traditional development approaches [15, 16]. However, the need to deliver features more quickly to the market, embrace continuous development, and utilize data-driven decision making has created a push to combine plan-driven approaches with agile and lean methodologies. This shift emphasizes incremental and iterative work, de-emphasizes upfront pre-development, and introduces new organizational structures [125]. The need or ambition to combine a suitable combination of plan-driven and value-driven approach poses significant challenges for effectively embedding human factors knowledge within the workflows of large-scale agile development [22, 291]. This focus must be balanced with careful considerations so that human factors are not neglected. In particular, agile methodologies focus on short sprints and rapid delivery, often prioritizing technical aspects and building solutions over upfront analysis, introducing the risk of neglecting detailed human factors considerations [224]. We refer to such contemporary agile and hybrid development approaches [292] simply as agile, distinguishing them from the pure plan-driven approach to systems engineering.

Initial solutions have been proposed for including human factors knowledge as requirements in traditional development [15, 17] and in agile AV development [253, 291]. Despite these initial efforts, there is still a need to explore how human factors requirements can be systematically integrated at various levels of abstraction [293, 294]. Requirements at different levels address different aspects of the system, ranging from high-level strategic goals to detailed technical specifications. Understanding how to effectively incorporate human factors at each level ensures that these considerations are embedded throughout the development process, leading to a more cohesive and human-centered design.

Additionally, studies [253, 291] also note the lack of human factors experts in organizations, which poses a challenge to the consistent inclusion of such expertise in the teams. Therefore, it is imperative to identify the optimal placement of human factors within specific areas or teams in the AV development process where their integration can yield maximum impact on the product. However, the optimal placement of human factors experts within organizations to maximize their impact remains underexplored.

This study aims to fill these gaps by investigating how human factors requirements can be integrated at different levels of abstraction and identifying the optimal placement of human factors experts within organizations to enhance their impact on the product [1], particularly when resources are limited and

[1]The impact on the product is typically measured using methods such as user satisfaction surveys, usability testing, and performance metrics [295, 296, 297, 298]. These methods help assess how well human factors are integrated into AVs and their effect on user experience and safety. However, it is important to note that such measurements are not within the scope of this study.

having human factors experts in multiple areas is not feasible. The research goal is operationalized by addressing the following research questions (RQs):

RQ1: To what extent can human factors requirements be confined to specific levels of abstraction of requirements?

RQ2: Where should human factors experts be positioned within an organization?

To address this study goal, we employed a mixed-methods approach, in which we rely on shadowing techniques, document analysis, informal interviews, workshops, and surveys for data collection. For data analysis, we used both qualitative and quantitative methods.

Our findings indicate that human factors requirements are needed at all levels of abstraction, as demonstrated by various hypothetical examples in the paper. We also explored how human factors experts can be integrated into teams to contribute to various requirement levels, identifying eleven distinct strategic placements, each with its own advantages and potential challenges. These placements range from direct involvement in requirements and supporting teams to strategic positions in management.

Subsequently, we conducted a quantitative survey to determine the optimal placement of human factors within an organization among the eleven identified options. The survey results aim to help organizations decide where to place human factors expertise, particularly when resources are limited and it is not feasible to have human factors expertise in all areas. We assessed this from three perspectives: general ranking, effectiveness, and ease of implementation. For general ranking, participants ranked their preferences for placing human factors experts to maximize product impact. The results indicate that the most preferred options for placing human factors experts are (in this order) *feature requirements* and *user experience teams*. For effectiveness, participants rated how well placing human factors experts in various sections contributes to integrating human factors into the development process. The results show that *user experience teams*, *feature requirements teams*, and *person/team responsible for the overall system* are considered the most effective placements. For ease of implementation, participants evaluated how easily human factors expertise could be integrated into different options. The most preferred options for ease of implementation are *user experience teams*, *dedicated human factors teams*, and *Safety team*.

Note that in this study, our focus on human factors is centered on the product and not on their impact on developers or employees involved in the development process, as studied by Hidellaarachchi et al. [299].

The rest of the paper is divided as follows: Section 2 provides an overview of the background, and Section 3 presents into our research methodology. The results are outlined in Section 4 and 5, with Section 6 covering the discussion and potential threats to validity. Lastly, Section 7 serves as the conclusion of the paper.

6.2 Background and Related Work

This section provides a comprehensive overview of the background concepts and related work that form the foundation of our research. It is divided into two main subsections: "Background" and "Related Work." The "Background" subsection covers foundational theories and definitions relevant to our study, while the "Related Work" subsection discusses previous research that directly relates to our study focus.

6.2.1 Background

Abstraction Levels of Requirements: In developing complex systems, breaking down requirements is a widely used method [300]. System and software development includes various abstraction levels, ranging from broad feature requirements to complex design specifications [301, 302, 303].

When examining this topic from a requirements engineering perspective, it's essential to recognize that companies typically structure their requirements across various levels of abstraction, often ranging from one to five levels [293, 294]. Different companies and domains may use varying terminology for similar definitions. The number and definition of levels of requirements are company and domain-specific.

Within the automotive industry, for instance, Original Equipment Manufacturers (OEMs) and suppliers often operate with different levels of requirements and maintain a range of relationships among these levels [293, 294]. Note that these levels of abstraction are not intended to create a strict categorization or to serve as a scientific classification system. The main message is to recognize that, in practice, requirements encompass various levels of abstraction, and typically, the following three are found in relevant automotive standards such as ISO 26262[302] and ASPICE [303]: feature requirements, system requirements, and software requirements. Some companies add additional levels or sometimes call these three levels differently, but they usually serve as a foundation to comply with the standards.

Because all suppliers need to comply with ISO 26262 and ASPICE, it was essential to examine these standards to understand their requirements structures. Therefore, we mapped our structure to these standards and included descriptions for each abstraction level, indicating how they correspond to the levels in ASPICE and ISO 26262 standard to show alignment with these concepts.

- **Level 1: Feature Requirements (FR).** This highest level outlines the system's purpose, target users, and an abstract description of the features and their interactions with customers, the environment, and other actors. It provides a broad understanding of what the system should achieve. Establishing feature requirements is a prerequisite for breaking them down into system requirements. Depending on the standard or the company, these requirements may be referred to by different names. For example, in the ASPICE standard, they are called stakeholder requirements. In ISO 26262, these are part of the feature requirements. Despite the variation in terminology, the purpose of the highest level requirements remains consistent across different standards and organizations. Example: The vehicle must maintain its lane autonomously under various driving conditions.

- **Level 2: System Requirements (SR).** Mid-level requirements that define the overall system behavior and interactions. They bridge the gap between feature requirements and component requirements by specifying how the system will fulfill the feature requirements. At this level, a technical solution has already been chosen, detailing the approach the system will take to meet the feature requirements. In both our context and the ASPICE standard, this level is referred to as system requirements. In ISO 26262, this corresponds to the system-level safety requirements and technical safety concept. The purpose of these mid-level requirements is consistent across different standards and organizations. Example: The lane-keeping system will use cameras and sensors to detect lane markers and make real-time adjustments to steering.

- **Level 3: Software Requirements (SWR).** The lowest level requirements detail the specifications for individual components or subsystems within the overall system. These requirements are technical and focus on the internal workings and interactions of the components, ensuring that each part of the system functions correctly and integrates smoothly. In the ASPICE standard, these are known as software requirements, while in ISO 26262, they are addressed during the software design phases, which include detailed safety requirements for individual components. The purpose of these detailed technical requirements is consistent across different standards and organizations. Example: The steering control unit must adjust the steering angle within a response time of 200 milliseconds based on input from the lane detection algorithm.

Human Factors in AV Development: Human factors considerations are integral to the development of AVs, extending beyond mere technological advancements to shape user experiences, safety protocols, and regulatory frameworks. Several studies have widely recognized the pivotal role of human factors in the design and operation of AVs [7, 304].

Below are some examples related to human factors in AV development. While the human factors domain is extensive, this list offers a diverse representation of only a fraction of the considerations involved in AV design:

- Designing AVs that are predictable and safe for other road-users.
- Building trust among users in AVs to a reasonable degree.
- Ensuring transparency in AV capabilities to avoid over-reliance.
- Creating AVs that drivers find appealing.
- Developing Human-Machine Interfaces (HMI) for AV users, such as touch screens for adjusting settings, that prioritize safety, user-centricity, and alignment with company branding.
- Designing HMIs for other road users (external HMIs) to effectively communicate state and intent.
- Crafting AV motions that users find comfortable, including considerations of speed, acceleration, and interactions with other road users and infrastructure.

- Ensuring that AV functions achieve their intended effects by considering the intentions and actions of users and surrounding road users.
- Facilitating auditory, visual, and haptic information exchange between AVs and users, including conveying information and warnings related to braking, active vehicle steering, and acceleration through actuators.
- Utilizing models of human behavior in virtual simulations to assess AV safety.

In addition to standalone human factors requirements, it is essential to consider their interplay with other functional and non-functional requirements within the AV system at large and its context. For instance, human factors requirements may intersect with cybersecurity protocols, regulatory compliance standards, and vehicle performance metrics [305]. Understanding these relationships enables holistic system design and ensures that human factors considerations are integrated into the broader AV development framework.

While the significance of human factors is widely acknowledged, the challenge lies in translating this awareness into concrete design and operational methodologies [291].

6.2.2 Related Work

Requirements Engineering and Human Factors: Incorporating human factors in the early phases of requirements engineering holds significant promise [17]. Requirements represent the foundation for subsequent design and implementation, demanding a holistic understanding of both technological and human aspects. Ahmad et al. [306] presents a framework for including human-centered aspects in RE when building AI-based software. The approach helps in achieving more responsible, unbiased, and inclusive AI-based solutions, emphasizing the importance of understanding human-centered needs early in the engineering process. While the paper highlights the significance of human-centered design, it falls short in providing empirical validation and a detailed exploration of specific challenges and does not extensively examine the role of human factors experts in organizations. In contrast, the current research goes further, focusing on human factors at different abstraction levels of requirements and the strategic integration of human factors expertise within organizations.

In our previous work [253, 291] have focused on the integration of human factors in AV development from a requirements engineering perspective. These studies emphasize the critical importance of integrating human factors into the requirements process. The findings indicate that integrating human factors expertise in teams is important. However, these studies also note the lack of human factors experts in the organizations, which poses a challenge to the consistent inclusion of such expertise in the teams.

Human Factors Experts Placement: Dul et al. recommend to incorporate human factors concepts at the highest level of strategic decision-making within an organization [307]. They recommend human factors professionals to foster relationships with key players (including system decision-makers and specialists). Furthermore, human factors play a vital role in mitigating risks (e.g., related to safety) [308] and require a dedicated investment of effort [59]. For example, having a 'Human Factor Champion' engaging with leadership to explain human factor concepts and ensure they align with corporate values and culture, emphasizing the importance of advanced communication and training. Furthermore, senior managers could play a pivotal role in bridging the gap between leadership and human factors, the need for ongoing learning, and the incorporation of feedback [308].

6.3 Research Methodology

In this research, we adopted Creswell's [80] mixed methodology approach to investigate the topic. Creswell's mixed methodology approach is a research framework that combines elements of both qualitative and quantitative research methods. This approach allowed us to gain a more comprehensive and nuanced understanding of our research topic by leveraging the strengths of both qualitative and quantitative data collection and analysis techniques.

6.3.1 Data Collection

To answer the research questions, data collection was conducted in three stages using a variety of methods. Our methods included shadowing techniques, document analysis, and informal interviews at an automotive company in Stage 1, complemented by a workshop involving both academic and industry professionals in Stage 2, and finally a survey with industry professionals in Stage 3. Figure 6.1 presents an overview of the data collection. Additionally, the principal questions used across different stages are accessible at this link.

6.3.1.1 Stage 1: Preparatory Study for RQ1

In Stage 1, we used shadowing techniques, document analysis, and informal interviews to gain a comprehensive understanding of requirements decomposition and human factors requirements across various levels of requirements abstraction. Data for this stage was collected from one company, which enabled an in-depth exploration of the topic.

The data collection strategy in this stage centered around key exploratory questions, guiding the research without being strictly bound to predefined questions. This flexible approach facilitated open-ended discussions and allowed us to gather detailed insights. While not every question was posed in each method, our overall focus remained consistent across these key questions.

Shadowing techniques: Two authors actively engaged in shadowing techniques at an automotive company during the research process. Their direct involvement in working with requirements at various levels provided insights into requirements decomposition and the human factors aspect of requirements across different levels. Shadowing techniques enhanced our knowledge of work processes in automotive companies, including how teams dealt with requirements.

Document analysis: Document analysis was conducted to examine requirements documents focusing on understanding the levels of requirements decomposition and associated examples. This analysis also involved a thorough

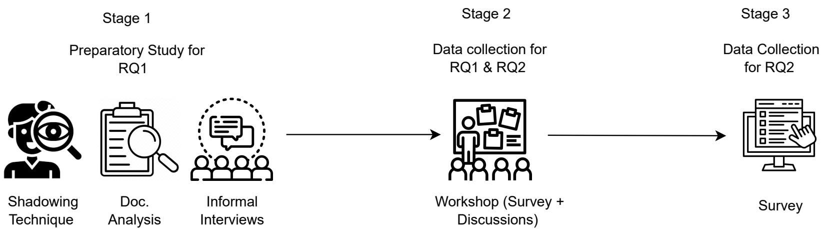


Figure 6.1: Overview of the three-stage data collection methodology.

exploration of documents to identify requirements related to human factors. We supplemented this approach with shadowing technique and informal interviewees to ensure a more comprehensive understanding and validation of our findings.

Informal interviews: Nine informal interviews were conducted with senior team members and feature owners from an automotive company. These interviews were informal, in that they were often ad hoc and focused on the particular expert and their field as opposed to focusing on a common interview guide. Notes were carefully taken, summarized, and coded for analysis.

6.3.1.2 Stage 2: Data collection for RQ1 & RQ2

To further explore and gain a broad view of the levels of abstraction at which human factors requirements are placed and to investigate the optimal placement of human factors experts within the organization, a workshop was conducted during a SHAPE-IT [309] project meeting, with approximately 30 expert participants from both the AV industry and academia. The participants were diverse, including industry professionals, senior researchers, and early-stage researchers with varying levels of experience in product development and engineering, all focusing on human factors and AV.

The workshop was structured into two interactive sessions.

The first session combined presentations with Mentimeter[2] survey that included both closed and open-ended questions. Initially, the presentation focused on the levels of requirements decomposition, illustrated with examples such as lane keeping. This was followed by a survey question addressing the integration of human factors requirements at each level of requirements decomposition, with 16 participants responding in this initial round.

After gathering responses, the presentation continued, introducing human factors requirements with specific examples for each level of decomposition. The same survey questions were then revisited, receiving responses from 20 participants in the second round. The survey then included open-ended questions.

The second session involved discussions, primarily focusing on the results gathered from the first session, with participants grouped into different rounds of approximately five members each round.

The use of open-ended questions fostered in-depth responses, enabling a comprehensive understanding of the subject matter. These discussions served

[2]https://www.mentimeter.com

as a platform for further exploration of the topic and importantly addressed the second research question, which focused on possible placement options for human factors experts within an organization. The workshop findings were summarized and discussed at the end of the sessions.

6.3.1.3 Stage 3: Data Collection for RQ2

After analyzing the data from Stages 1 and 2, we designed a survey. In this survey, we incorporated a mix of demographic, open-ended, and closed-ended questions. The open-ended questions were chosen to allow respondents the freedom to answer without the limitations of predefined options. The closed-ended questions comprised ranking and Likert scale options.

For survey data, we mainly relied on three key questions to explore the strategic integration of human factors expertise within an organization. We began by asking participants to rank a number of options for integrating human factors expertise into the organizational workflow according to their preference. Participants were allowed to rank the solution propositions, identified in Stage 2, between 1 and 10. Assigning a rank of 1 means that it is the best position for human factors expertise placement in the organization, and continues in descending order of priority up to 10. Initially, we intended to have each option ranked, assuming this would lead to a clear top-ranking option that would be considered a priority. Yet, prior to distributing the survey, we opted to refine our method. We modified the survey to allow participants the flexibility to rank only those options they deemed relevant, instead of ranking all.

Subsequent questions focused in on two critical dimensions of integration: effectiveness and implementational ease. "Effectiveness" here refers to how well or efficiently something works or achieves its intended goals or objectives. In this context, it pertains to how the strategic placement of human factors experts within different sections of the organization contributes to the overall success of integrating human factors considerations into the development process. Meanwhile, "Easiness" in this context, relates to the ease and feasibility of implementing the placement of human factors experts in various parts of the organization. It takes into account factors such as available resources (e.g., time, budget, and the number of human factors experts available) and assesses how easy it would be to put each placement option into practice. Together, these dimensions establish a comprehensive understanding of where human factors expertise can be most effectively and easily situated within an organization to drive human-centric development. The dataset collected from the survey is available here.

Before distribution, we conducted a pilot test of the survey with an expert in the field. We then primarily distributed the survey to human factors experts and developers working within the automated vehicle industry. The survey was sent to 92 individuals, from which we received 31 responses.

A majority of the study participants (51.9%) work in the Automotive OEMs sector, and 22.2% work for Automotive Suppliers. Most have expertise in human factors, with a few in engineering or both. Work experience is diverse, with equal numbers having over 10 years, between 5 to 10 years, and a minority with under 5 years experience.

6.3.2 Data Analysis

The initial phase of the research involved document analysis, interviews, and shadowing techniques within the company to better understand the level of abstraction of requirements in the context of AVs. By examining existing documentation, analyzing informal interviews, and observing work processes, insights were obtained regarding the requirements decomposition and exploration of human factors requirements.

For the data collected from Stage 1, we primarily focused on summarizing and identifying common themes. This analysis was used to elaborate the first part of RQ1 (examples of human factors at different levels of requirements abstraction) and served as a foundation for the subsequent workshop (Stage 2), where participants with relevant expertise (AV engineers and human factors experts) from industry and academia participated. The closed-ended part of the workshop survey was used to answer RQ1, where we examined at which abstraction level human factors should be involved. For this purpose, we used simple descriptive analysis.

For RQ2, we began by examining the written responses collected from open-ended questions in the first session of workshop. When examining the written responses from the open-ended survey questions, we noticed that participants often used different terms to convey the same concepts. For example, one participant mentioned placing human factors at a high level in the decomposition, while another referred to the same concept as feature requirements. Despite the different wording, both participants were referring to the same idea. We confirmed this during the discussion session, verifying that they meant the same concept, which allowed us to categorize them under one label. We grouped similar concepts and assigned them appropriate names, leading to the identification of strategies for the placement of human factors experts within an organization. We refer to these potential strategies as solution propositions. Similarly, we summarized the key points from the discussions, which contributed to defining associated advantages and challenges for solution propositions. Additionally, during the discussion, we collected other insights, which are described in the results section as additional findings.

The identified solution propositions suggest optimal placements for human factors within organizations. However, if companies lack human factors expertise and need to strategically decide where to place this expertise for maximum impact on the product, it becomes crucial to prioritize these placements. Therefore, after collecting solution propositions, we conducted a statistical analysis on the survey data collected in Stage 3. Our aim was to guide strategic decisions on effectively allocating human factors expertise, especially when resources are limited.

This study employs two analytical approaches—descriptive statistics and Bayesian analysis—to gain a comprehensive understanding of our dataset. The rationale for using both descriptive statistics and Bayesian analysis is to characterize observed phenomena and explain predictive patterns within the data. This combined approach allows us to make more reliable predictions and inform decision-making processes, even with limited data [310]. The details of how the Bayesian analysis model is developed are available in the replication package[3].

6.4 Findings RQ1: To what extent can human factors requirements be confined to specific levels of abstraction of requirements?

Based on our experience with production requirements at the automotive company(in stage 1), we made the conclusion that requirements cannot be confined to a specific requirement level of abstraction. Requirements could related to human factors at each level of abstraction, be it feature requirements, system requirements, or the detailed software requirements. Sometimes it is only obvious in the details how a system can support a driver in aspects that are related to human factors aspects.

To visualize our experience and abstract it away from the company IP, for the purpose of this paper, we came up with a fictive but realistic example of a "vehicle lane-keeping" feature. This feature visualizes an example where human factors requirements are present on all levels of requirements. We chose this example because it emphasizes the inclusion of human factors in automated features which include AI in their implementation.

The development of the lane keeping feature is broken down into different levels of requirements as shown in Figure 6.2. At the highest level (Feature Requirements (FR)), we have the "vehicle lane keeping" feature. This high-level requirement encompasses the overall functionality desired by stakeholders. An example for a FR is: "The vehicle shall have a lane-keeping feature that helps the driver stay within the lane." This broad concept is then broken down into more specific requirements on the System Requirements (SR) level. For instance, one such element could be "lane detection", which is crucial for lane keeping. An example for a SR is: "The lane detection shall detect lane markings on the road." Lane detection is further decomposed at the Software Requirements (SWR) level. They describe the behavior and functionalities of the software components needed to achieve the system requirements. This could include for instance requirements on vision processing and lane detection algorithms among others. An example for a SWR is: "The vision processing shall process camera images to identify lane markings".

6.4.1 Relating Human Factors to Requirements Levels

To gather broader input, we conducted a workshop (stage 2). In the first session of the workshop, we began by presenting and discussing all levels of requirements. We did not specify examples of human factors requirements at each level but included other types of requirements. We then asked the audience about the best level of abstraction for specifying human factors requirements (note that the text of the survey question was deliberately generic to support the exploratory nature of this study; from the context of the workshop, the answers clearly related to human factors requirements levels). The survey results showed that initially 75% (12) of respondents believed human factors

[3] https://doi.org/10.5281/zenodo.11652602

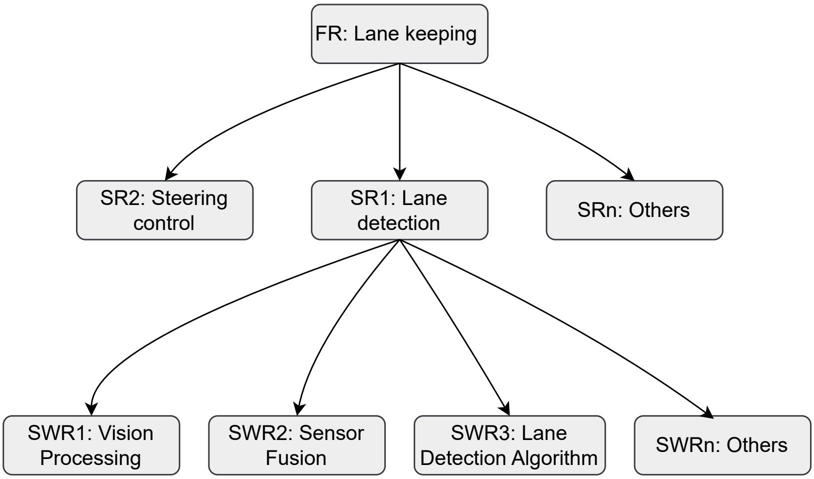


Figure 6.2: An example of levels of requirements abstraction.

requirements should be specified at level 1 (i.e., *feature requirements*), while 25% (4) indicated level 3 (i.e., *software requirements*) - see Figure 6.3, blue bars. Interestingly, no one voted for either level 2 (i.e.,*System Requirements*) or the option at each level of requirements.

We then proceeded to elaborate on human factors requirements for each level of requirements abstraction as described in section 6.4, using the hypothetical examples of human factors requirements for the lane-keeping feature. These examples have been compiled and are presented in Table 6.1. Note that we specifically looked into the complexities of requirements decomposition, with the primary objective of gaining insights from a requirements engineering perspective.

Following the explanation with examples, we asked the same question to the respondents again, and interestingly, the responses varied this time (Figure 6.3 - orange bars). 40% (8) of the participants stated that human factors are crucial at the high level, while 60% (12) showed that human factors should be considered at each level of requirements. This shift in responses highlights the evolving understanding of the importance of human factors with increased awareness. It also shows that even experts found it challenging to guide human factors requirements. However, this indicates a need for further research and the development of comprehensive guidelines.

6.5 Findings RQ2: Where should human factors experts be positioned within an organization?

To answer this research question, we first collected general recommendations on the optimal placement of human factors expertise. We identified eleven strategic placement options, described in Section 6.5.1. After collecting solution

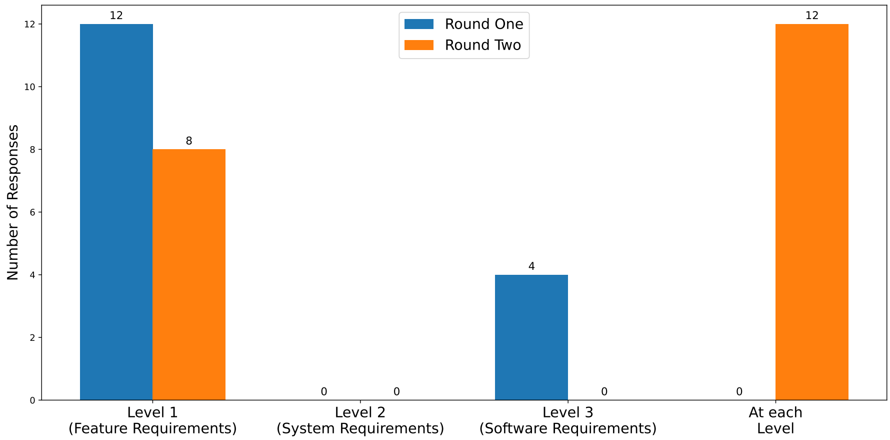


Figure 6.3: Survey response distribution for 'incorporation of human factors requirements across different levels of requirements abstraction'. This chart presents the percentage of participants choosing each option over two rounds. The first round shows choices made before presenting examples of human factors at each level, while the second round depicts selections made after presenting examples of human factors at each level of requirements abstraction.

propositions, we conducted a survey with a primary focus on a quantitative approach to determine the optimal placement of human factors within an organization among these eleven identified options, as detailed in Section 6.5.2.

6.5.1 Potential Strategies for Placing Human Factors Experts within an Organization

The results derived from the data collected during Stage 2 (the workshop) provide an overview of potential strategies for the placement of human factors experts within an organization. We refer to these potential strategies as *solution propositions*. Table 6.2 presents these solution propositions with IDs (S1 to S11) and their advantages and challenges, wherever these could be derived from the collected data.

The first solution proposition derived is to place the human factors expert at a higher level of requirements, with the *feature requirements (S1)* team. Engaging human factors expertise during Feature Requirements can ensure early consideration of relevant issues, but there's a concern that such considerations may be lost in the translation to more detailed requirements. The second solution proposition proposes to keep the human factors expertise *at system requirements (S2)*. The third proposition suggests placing the human factors expertise at the lowest level of requirements - *software requirements (S3)*. At the *software requirements level*, there can be more detailed and comprehensive consideration of human factors. This thorough attention at the *software requirements* stage is essential, but it might not be sufficient to address human factors issues that emerge later in the development. Also, aspects that only

Table 6.1: Human Factors Requirements at Different Levels of Requirements.

Levels	Hypothetical Examples
Feature Requirements	HF-R1: The lane-keeping feature shall ensure a smooth driving experience by minimizing abrupt and jerky motions to prioritize passenger comfort. HF-R2: The lane-keeping feature shall chose the appropriate vehicle lane position - either centralized or slightly inside the boundary, depending on driving scenarios. HF-R3: The lane-keeping feature shall ensure the driver feels safe by maintaining a minimum distance of X meters from vehicles that are larger than 4 meters in length, and a minimum distance of Y meters from other road participants.
System Requirements	HF-R1: Lane detection shall provide accurate lane information to maintain fluid motion and prevent jerky maneuvers. HF-R2: Steering control shall allow for dynamic lane positioning adjustments based on driver preferences, while adhering to safety protocols. HF-R3: Lane detection shall incorporate adaptive feedback mechanisms that adjust the intensity and type of alerts based on driver behavior and preferences.
Software Requirements	HF-R1: The vision processing unit must analyze video feeds to identify lane markers with a latency of less than 100ms - to ensure real-time updates to the driver and not cause confusion. HF-2: The lane detection algorithm shall minimize false positives and negatives to ensure driver trust and system reliability.

can be captured when having a more holistic view of the product may not be addressed, and the workload of the human factors expert may be low in some teams.

The fourth proposition is to have *one human factors expert in each team (S4)*, across all levels. Having a human factors expert in each team ensures comprehensive coverage and involvement from the start, but it is expensive and may be impractical in terms of staffing and workload. Alternatively, forming *a dedicated human factors team (S5)* allows for focused work on human factors but could potentially lead to communication gaps with other project teams. Another strategy involves assigning the responsibility for human factors to *a person/team responsible for the overall system (S6)*. This promotes a broader organizational culture of awareness but might distance experts from the everyday activities of development teams.

Having the human factors experts in *user experience teams (S7)* ensures a user-centered approach but may narrow the focus to user experience at the expense of other human factors considerations. Further, having human factors experts *at product owner (PO) level (S8)* can provide early and influential guidance. However, such a PO must also possess technical expertise and balance these with PO responsibilities. Being closer to the teams, the presence of such a PO may discourage teams from seeking out additional expertise when needed. Placing human factors *with non-functional requirements teams (S9)*, with *system or feature evaluation team (S10)*, and with *safety teams (S11)*, each presents its own set of benefits and challenges, such as a narrow focus or late integration in the development process.

This overview highlights the critical need for strategic placement of human factors expertise to achieve a balance between early involvement, specialized focus, and comprehensive coverage in product development.

6.5.2 Survey Results on Optimal Placement of Human Factors Experts

After collecting solution propositions, we conducted a survey with a primary focus on a quantitative approach to determine the optimal placement of human factors within an organization among the eleven identified options. This survey aims to guide organizations in deciding where to position human factors expertise, especially when resources are limited and it is not feasible to have experts in all areas.

We evaluated this from three perspectives: general ranking, effectiveness, and easiness. For the “ranking” perspective, participants were asked to rank their preferences for where human factors experts should be positioned to maximize their impact on the product. The “effectiveness” perspective aims to assess the most effective placement of human factors experts to contribute to the overall success of integrating human factors considerations into the development process. “Easiness” relates to the feasibility of positioning human factors experts in various parts of the organization, taking into account factors such as available resources and the practicality of each placement option.

Below, we present the survey results for all three main aspects using two analytical approaches: descriptive and statistical.

6.5.2.1 Descriptive Statistics

Ranking: One of the survey questions focused on the general ranking of various solution propositions. Figures 6.4 show the distribution of rankings for various solution propositions based on the percentage of votes they received. The results depicted in the figure reveal that *feature requirements* emerged as the most frequently ranked at the top position, followed by user experience teams, indicating these two as primary areas for placing human factors. On the other hand, *software requirements* and *non-functional requirements team* were the least favored options for the placement of human factors, suggesting these may not be ideal places for human factors integration.

Effectiveness: In another question, participants were asked to indicate their level of agreement concerning the effectiveness of integrating human factors experts for each of the placement options. Based on the survey results depicted

Table 6.2: Solution Propositions for placements of human factors experts.

ID	Solution Propositions	Advantages	Challenges
S1	At a higher level of requirements (Feature Requirements)	Allows to leverage human factors expertise in the early stages of product development	Human factors requirements may disappear in lower-levels of requirements
S2	At System Requirements		
S3	At the lowest level of requirements (Software Requirements)	More detailed and comprehensive consideration of human factors	May not address human factors issues that arise later in the project
S4	One human factors expert in each team	Ensures human factors involvement from the start and throughout the development; Comprehensive coverage	Costly to maintain human factors experts at every level; challenging, especially in terms of headcount and workload in some teams
S5	In a dedicated human factors team	Dedicated team can focus solely on human factors; If teams knew when to ask for help this would likely be a quite good solution	May create communication barriers between the human factors team and other project teams
S6	A person/team responsible for the overall system	Promotes an organizational culture of human factors awareness	May be too far from development teams
S7	In user experience teams (interaction design, HCI, UX)	Close collaboration with UX teams ensure user-centered design	Human factors may be limited to UX consid. and not cover broader aspects
S8	At the Product Owner (PO) Level	Early involvement; POs have a direct influence on project direction; POs look at the broader aspects and provide guidance to teams	Danger to rely only on PO and to not seek out additional expertise when needed.
S9	With non-functional requirements team	Focus on specific human factors aspects within a specific context	Might lead to a narrow focus on human factors
S10	With system/feature evaluation team	Allows for a thorough evaluation of human factors in the final stages of development of a feature or system	If only considered at this stage, earlier opportunities for integrating human factors might be missed, requiring changes late in the development process.

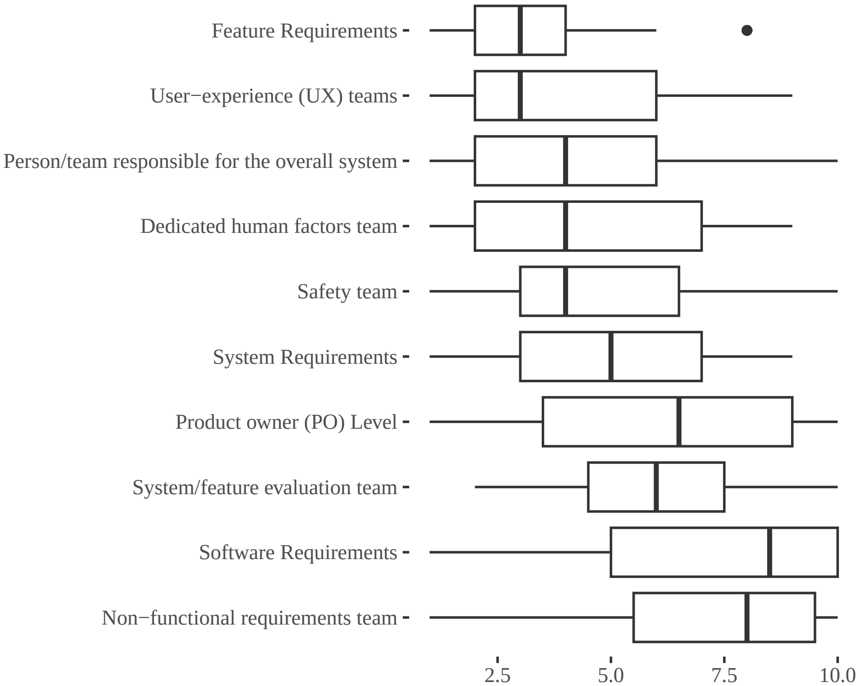


Figure 6.4: Ranking of most suitable human factors placement options. In this study, participants were asked to rank these teams from the most favorable (rank 1) to the least favorable (rank 10). Therefore, lower ranking values indicate a higher preference or perceived importance.

in Figure 6.5, participants expressed the highest level of strong agreement for the inclusion of human factors experts within User-experience teams (75% agreement). Notably, roles like *non-functional requirements teams* and *software requirements teams* drew more disagreeing opinions. Two categories- *system requirements teams* and *product owner (PO)* level seem to have a more varied distribution of responses. While they have a significant portion of participants agreed on their effectiveness, they also have a substantial amount of neutral and disagreeing responses.

Easiness: Figure 6.6 depicts participants' views on the ease of integrating human factors experts into different parts of the organization. The most favorable placements, for the easy implementation of human factors experts, as reflected by approximately 75% agreement, are within the *user-experience teams* and the *dedicated human factors team*. However, roles like *person/team responsible for the overall system* and *non-functional requirements team* drew more disagreeing opinions. Similar to effectiveness, two categories - *system requirements and product owner (PO)* level - seem to have more varied distributions of responses - ranging from agreement, as well as neutral, and disagreeing responses to similar amounts.

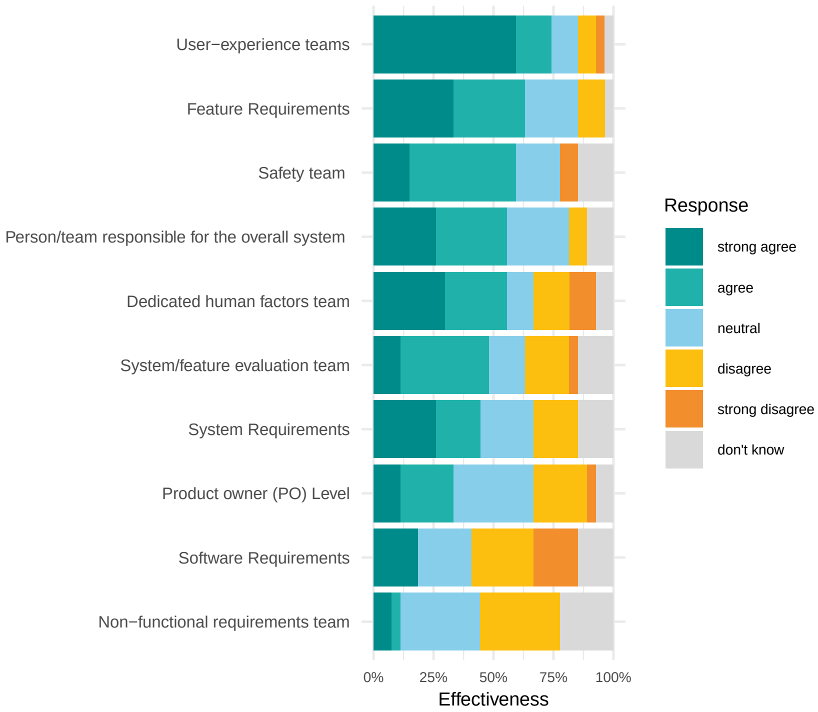


Figure 6.5: Distribution of responses indicating the perceived effectiveness of human factors experts at different placement options.

Human Factors vs. Engineering Responses: Here, we studied the opinions of study participants with different backgrounds - *engineering*, *human factors*, a combination of *engineering and human factors*, and participants with *other* backgrounds - to observe their distinct perceptions regarding the placement of human factors expertise. Figure 6.7, 6.8, 6.9 compares the survey responses across different respondent groups for effectiveness.

Human factors and engineering groups generally had similar positive responses regarding the effectiveness (see Figure 6.7) of different teams. Both the human factors and engineering groups aligned closely with these positive views. However, responses from the *other* group had more extreme answers - either agree or disagree.

For easiness (Figure 6.8), we observed a similar behavior across groups for all placement options, except for *system requirements* and *software requirements*, where the *engineering* group showed more disagreeing responses, while the *other* group showed more agreeing responses.

For ranking (Figure 6.9), we observed different behaviors among the respondent groups for a few placement options, while for other placement options, the responses among different groups were similar. For example, for *feature requirements*, the *other* group had more agreeing responses compared to other groups. For *non-functional requirements*, the *engineering* group had more agreeing responses. Conversely, for the *safety team*, *system requirements*, and

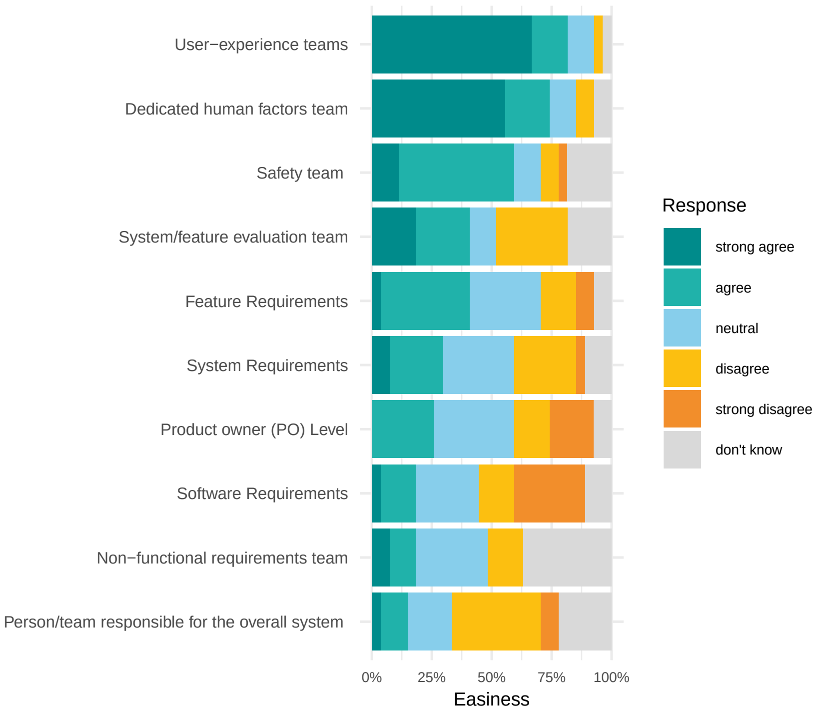


Figure 6.6: Distribution of responses indicating the perceived ease of placing human factors experts at the different placement options.

user-oriented teams, the *other* group had more disagreeing responses.

These results provide insights into the perceptions and preferences of different professional groups regarding the placement of human factors expertise in various teams. By comparing responses among diverse groups, we can identify specific areas where opinions converge or diverge, which is crucial for understanding potential challenges and opportunities in interdisciplinary collaboration. The similar responses from human factors and engineering groups suggest that both disciplines recognize and share common goals. However, the divergent responses from the *other* group may highlight the need for tailored communication and collaboration approaches when involving professionals from varied backgrounds. Their extreme responses indicate potential areas of misunderstanding or differing expectations.

6.5.3 Statistical Analysis

We further complement our descriptive analysis with Bayesian analysis [310].

Prior and Posterior Predictive Check: The Prior Predictive Check involves simulating data from the model using only the prior distributions, without incorporating any empirical data. This check allows us to examine the model's behavior under the prior assumptions and provides a baseline for

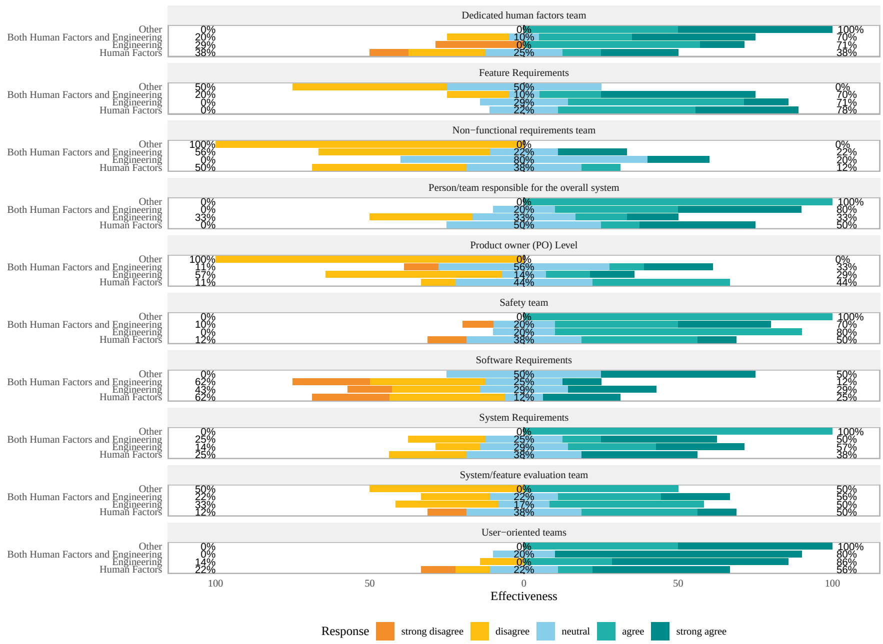


Figure 6.7: Distribution of survey responses regarding perceived ‘Effectiveness’ across different respondent groups.

comparison with the observed data. In our study, the Prior Predictive Check is used to assess the initial assumptions and spread of our priors across the outcome space. On the other hand, the Posterior Predictive Check evaluates the model’s predictions after incorporating empirical data. By comparing the simulated data from the posterior predictive distribution with the observed data, we can assess how well the model fits the empirical data and identify potential areas of model inadequacy.

The plots on the left in Figures 6.10, 6.11 and 6.12 represent the prior predictive check, where we sample only from the prior distributions without incorporating any empirical data. In contrast, the plot on the right shows the posterior predictive check, reflecting the model’s predictions after considering empirical data. In both plots, the dots represent the mean, and the lines show the 95% credible interval. The bars represent the levels of the effectiveness variable, ranging from State 1 to State 5 (note the different scales on the vertical axis). In the left plot, the uniform distribution of our priors across the outcome space suggests that our initial assumptions are spread evenly. However, after integrating empirical data in the right plot, we observe a well-fitted model. Here, the uncertainty around each mean is minimal, and the means align closely with the upper end of each effectiveness state bar.

Explanation of Bayesian Model Output Tables: Tables 6.3, 6.4, and 6.5 present the summary of population-level effects for the effectiveness, easiness, and ranking of placing human factors expertise at different places within an organization. The tables display estimates, estimation errors, and 95% credible

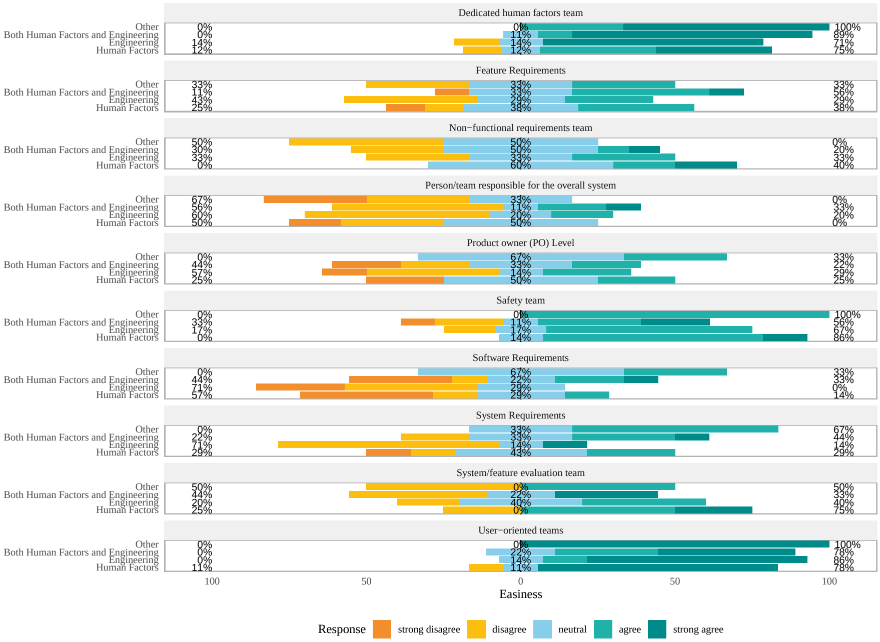


Figure 6.8: Distribution of survey responses regarding perceived 'Easiness' across different respondent groups.

intervals (CI) for each factor's impact on the effectiveness and easiness of placing human factors expertise. The estimated effects provide insights into the impact of each factor on the outcome variable. The estimates represent the average effect size of each factor on the outcome variable. Such as on perceived effectiveness and easiness, with positive values indicating an increase and negative values indicating a decrease. The "Est. Error" column shows the standard error of the estimate.

Additionally, the columns Rhat, Bulk_ESS, and Tail_ESS are included to provide diagnostic measures of the model's convergence and the reliability of the estimates. The Rhat column shows the potential scale reduction factor, which is a measure of convergence for the Bayesian Markov Chain Monte Carlo (MCMC) simulations. An Rhat value of 1 indicates perfect convergence. Therefore, all Rhat values being 1 in the table suggest that the model has converged well.

The Bulk_ESS (Effective Sample Size for bulk) and Tail_ESS (Effective Sample Size for tail) columns provide measures of the effective sample size. Bulk_ESS gives an indication of the sampling efficiency in the central part of the posterior distribution, whereas Tail_ESS assesses the efficiency in the tails of the posterior distribution. Higher values indicate more reliable estimates. These measures are crucial for understanding the precision of the estimates provided by the Bayesian model.

Table 6.3, 6.4, and 6.5 contains all the parameters (β's) of interest, with significant effects highlighted in bold, indicating that the 95% credible interval of an effect's distribution does not include zero. Confidence intervals (l-95% CI

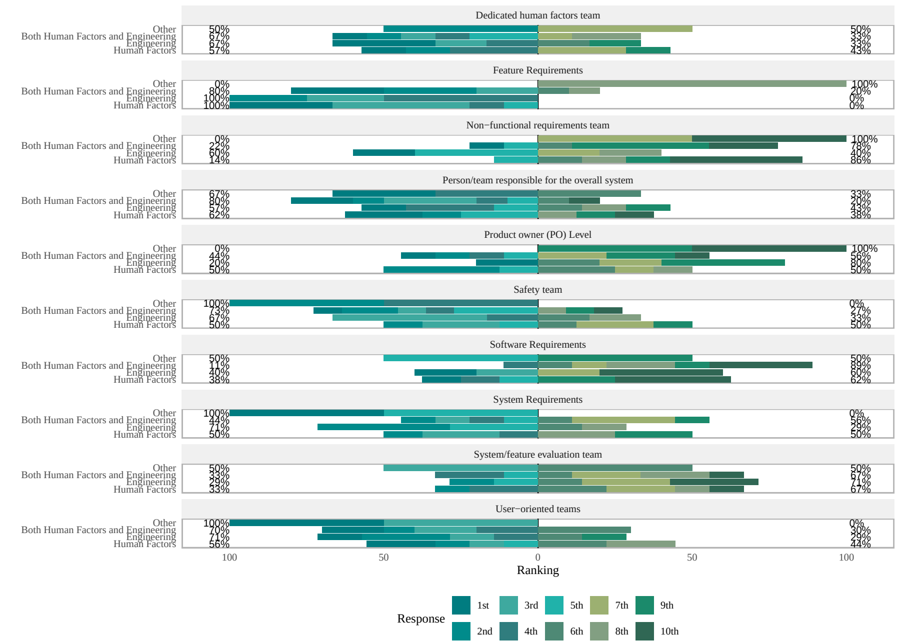


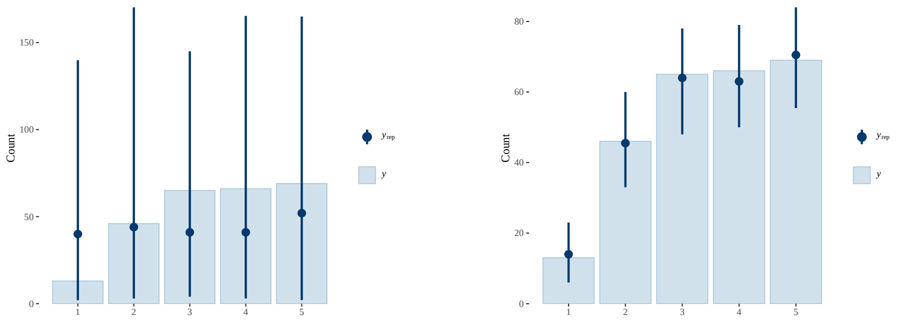
Figure 6.9: Distribution of survey responses regarding perceived ‘Ranking’ across different respondent groups.

and u-95% CI) provide a range of values within which we can be confident that the true parameter lies. For example, if the confidence interval for ’Feature Requirements’ in Table 6.3 is (0.19, 1.30), it means we are 95% confident that the true effect of placing experts in this position falls within this range.

The rows represent the effects of different factors on the outcome variable. The ‘Placement Option’ factors refer to different options for placement, while ‘Experience’, ‘Expertise’, and ‘AreaOfWork’ represent different levels of experience, expertise, and areas of work, respectively. The rows are the cutpoints that we use to estimate the deviation on each outcome. For example since we have 10 outcomes (placement options) we have 10-1 = 9 cutpoints. This is because one level of the placementOption variable is being treated as the reference level, and its coefficient is not explicitly displayed in the output. In Bayesian regression models, one category of each categorical variable is typically treated as the reference category, against which the other categories are compared.

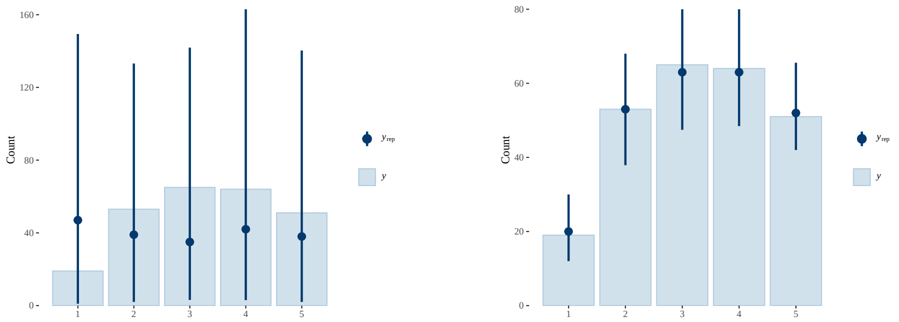
Effectiveness: Table 6.3 presents the summary of population-level effects for the effectiveness of placing human factors expertise at different places within an organization. Several placement options, such as *feature requirements*, *user-experience teams*, and *person/team responsible for the overall system*, demonstrate positive effects on effectiveness. Conversely, *software requirements* and *non-functional requirements team* exhibit negative effects. Notably, *user-experience teams* emerge as particularly influential, suggesting that allocating experts to these roles could significantly enhance product effectiveness.

The impact of experience levels, both low and medium, on effectiveness



(a) Prior- Effectiveness (b) Posterior -Effectiveness

Figure 6.10: Prior and posterior predictive checks for Effectiveness



(a) Prior - Easiness (b) Posterior - Easiness

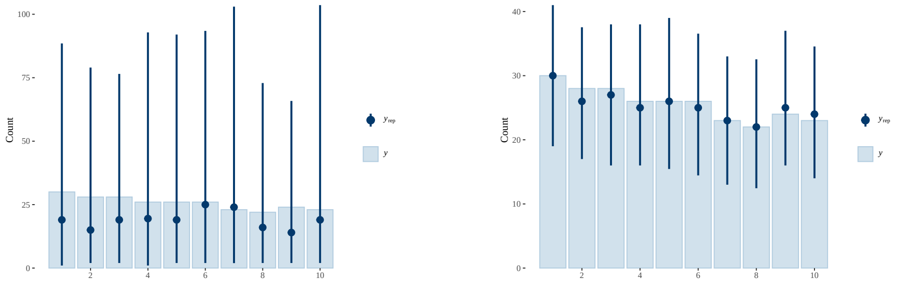
Figure 6.11: Prior and posterior predictive checks for Easiness

ratings appears to be minimal. The confidence intervals for these estimates are wide and cross zero, reinforcing the lack of substantial difference. Similarly, when examining expertise, the findings suggest that its effects vary and are not substantial due to wide confidence intervals. When considering areas of work, the Research Institute shows a positive effect, but with uncertainty due to broad confidence intervals. The non-automotive, OEM, and supplier sectors show negligible or slightly negative effects without substantial evidence.

Figure 6.13 visually represents the posterior probability distribution of each parameter, providing additional insight compared to Table 6.3 alone.

Easiness: Table 6.4 focuses on the perceived easiness of placing human factors expertise within an organization. Notable factors such as the presence of *dedicated human factors teams*, *user-experience teams*, *safety teams* and *System/feature evaluation team* show substantial positive effects on easiness. Conversely, factors like *software requirements* exhibit negative effects, although to a lesser extent.

The impact of experience levels, both low and medium, and areas of expertise, including engineering and human factors, were minimal and did not show substantial evidence of influence, as indicated by credible intervals that



(a) Prior - Ranking

(b) Posterior - Ranking

Figure 6.12: Prior and posterior predictive checks for Ranking

include zero. This suggests that these factors have a lesser influence on the perceived easines. Similarly, the area of work, whether it be non-automotive, OEM, research institute, or supplier, showed no significant effects on easiness, with wide credible intervals and estimates close to zero.

Figure 6.14 visually represents the posterior probability distribution of each parameter, providing additional insight compared to Table 6.4 alone.

Ranking: When interpreting the results of Table 6.5, it's crucial to note that the ranking of options was inverted during analysis. In the original study, participants were instructed to rank the most favorable option as one and the least favorable as ten. However, in the Bayesian model presented here, the values are calculated in reverse order, with higher values indicating more favorable options.

The negative estimates indicate a beneficial effect on the outcome, while positive estimates suggest a detrimental effect. For example, *feature requirements* has a negative estimate of -1.03, indicating that it is associated with higher rankings or more favorable outcomes. Conversely, *software requirements* has a positive estimate of 0.61, suggesting a lower ranking or less favorable outcome associated with this feature.

In terms of experience levels (low and medium), areas of expertise (engineering, human factors, other), and areas of work (non-automotive, OEM, research institute, supplier) do not exhibit significant effects on ranking, as their estimates are close to zero and their credible intervals are wide.

Figure 6.15 visually represents the posterior probability distribution of each parameter for ranking pf placement options.

6.5.4 Open ended-Questions

Inclusion of human factors expertise in senior management: It also emphasized the importance of having human factors expertise included at the senior management level. This suggests a belief that strategic decisions and company culture regarding human factors can be significantly influenced by leadership roles.

> *"Importantly, human factors and safety need to be included in a senior management role within the organization."* —

Table 6.3: Summary of population-level effects on effectiveness.

	Estimate	Est. Error	l-95% CI	u-95% CI	Rhat	Bulk_ ESS	Tail_ ESS
Placement Options							
Feature Requirements	**1.25**	**0.48**	**0.32**	**2.20**	**1.00**	**1563**	**2581**
System Requirements	0.80	0.51	-0.19	1.79	1.00	1688	2378
Software Requirements	-0.74	0.53	-1.77	0.28	1.00	1844	2855
Dedicated human factors team	0.83	0.51	-0.17	1.81	1.00	1613	2729
User-experience teams	**2.12**	**0.53**	**1.08**	**3.17**	**1.00**	**1742**	**2195**
Non-functional requirements team	-0.53	0.50	-1.50	0.46	1.00	1846	2472
System/feature evaluation team	0.42	0.50	-0.57	1.40	1.00	1722	2339
Safety team	0.93	0.48	-0.03	1.88	1.00	1525	2297
Person/team responsible for the overall system	**1.14**	**0.50**	**0.16**	**2.14**	**1.00**	**1628**	**2548**
Experience Levels							
ExperienceLevelLow	0.09	0.61	-1.08	1.33	1.00	2552	2572
ExperienceLevelMedium	-0.06	0.33	-0.72	0.58	1.00	2695	2457
Expertise							
ExpertiseEngineering	-0.60	0.45	-1.46	0.24	1.00	1900	1892
ExpertiseHumanFactors	-0.68	0.40	-1.53	0.09	1.00	2056	2066
ExpertiseOther	-0.58	0.62	-1.85	0.66	1.00	2564	2320
Areas of Work							
AreaOfWorkNonAutomotive	-0.11	0.89	-1.95	1.58	1.00	2208	2264
AreaOfWorkOEM	0.23	0.46	-0.67	1.17	1.00	1499	1886
AreaOfWorkResearchInstitute	0.94	0.64	-0.28	2.24	1.00	1909	2186
AreaOfWorkSupplier	-0.28	0.52	-1.32	0.80	1.00	1879	1866

Organizational structure: Respondents point out that there are missing options for advancement or involvement of human factors experts at higher levels within the company. This suggests that the current organizational structure may not adequately integrate human factors expertise in senior roles or decision-making processes.

Integration across several teams: Some experts should be integrated across various placement options (from early target setting to practical engineering) rather than being placed into one group.

> *"Ranking this is difficult, as there should be a human factors person in many of these places simultaneously, from my experience."* —

Integration with other teams: Some respondents believe that human factors experts should be integrated within various teams rather than being isolated in a dedicated team. This indicates a preference for a collaborative approach where human factors expertise is embedded directly in the teams that require it, suggesting that such integration could lead to more immediate and impactful contributions to the product.

> *"Human factors experts should not be separated and be in a dedicated team. Their work and support will be most valued and reach the desired impact when they are part of the teams that address the issues directly. That is the only way they can support, act fast, and also disseminate knowledge."* —

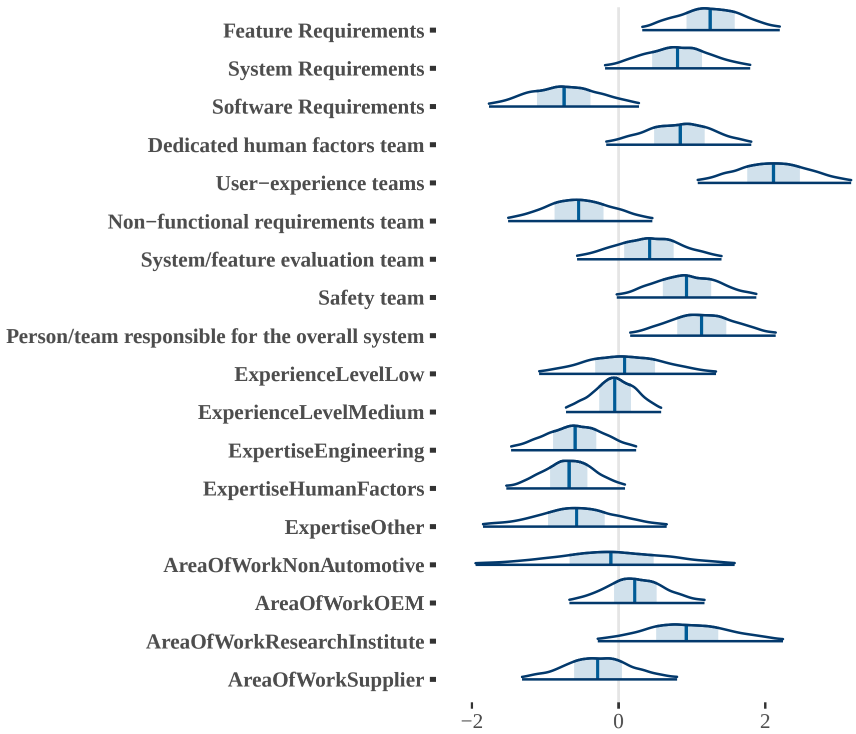


Figure 6.13: Density plot of all population-level effects on effectiveness

Human factors vs UX/UI Design: There is a clear expression of concern that the perception of human factors has been "contaminated" by UX research and UI design. This could mean that the unique and specialized contributions of human factors are being overshadowed by the more widely recognized fields of UX and UI, potentially leading to a misallocation of human factors resources or the overlooking of human factors considerations in the product development process.

> *"Nowadays especially, when talking human factors, many think of a Designer or a UX researcher (which often is assumed is also a designer). Most companies do not see the value of having someone trained in cognitive and ergonomics engineering and do not see that those people often come with deep technical knowledge that can support the development process in many ways, beyond UI evaluation. If the perception of human factors does not change, and currently it is thoroughly contaminated by UX research and UI design."* —

6.5.5 Additional Results: Raising Awareness and Cultivate a Culture of Human Factors

During the qualitative data analysis from Stage 2, it emerged that several approaches effectively educate and inspire teams to better integrate human factors knowledge. These approaches aim to raise awareness and cultivate a culture of human factors integration within the organization, thus complementing the placement of human factors in relation to RQ1 and RQ2. Below, we

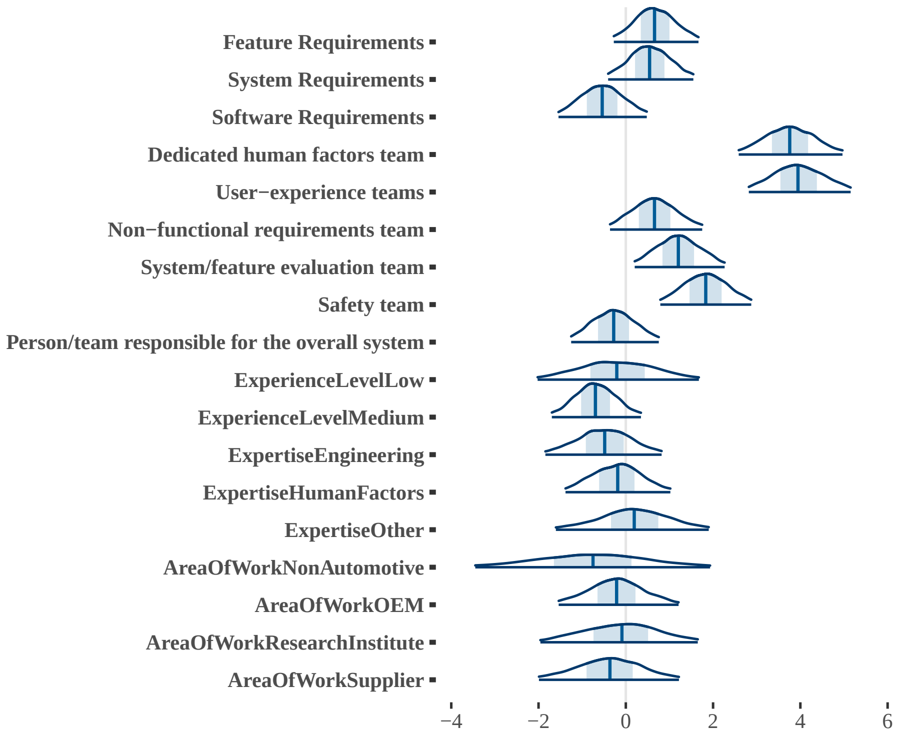


Figure 6.14: Density plot of all population-level effects on easiness

present some of the most widely discussed motivational guidelines in order to provide a more comprehensive report of our results.

Showcasing Tangible Results: Our findings indicate that one essential strategy is to emphasize the benefits by showcasing tangible results from previous projects where human factors considerations were successfully implemented. Presenting real-world examples that highlight the positive impact of human factors in achieving project goals e.g. in the form of *success stories* can effectively demonstrate its value to the teams.

Learning from Past Neglect: Educating teams about the risks associated with overlooking human factors, especially in *edge cases*, is essential. Such education should not only highlight potential pitfalls but also actively engage team members in analyzing these *edge cases*. This deeper exploration can reveal the extensive consequences that may result from neglecting human factors.

Creating a Collaborative Environment: Furthermore, we learned that fostering a collaborative environment is crucial in motivating teams to embrace human factors. For instance, regular brainstorming sessions can be organized, allowing team members to actively participate in generating ideas and solutions while considering human factors throughout the entire development process.

Engaging with Human Factors Experts: To enrich the above mentioned brainstorming sessions further, inviting human factors experts or individuals with human factors expertise to share insights and engage in discussions with the teams can be beneficial. These interactions can provide teams with

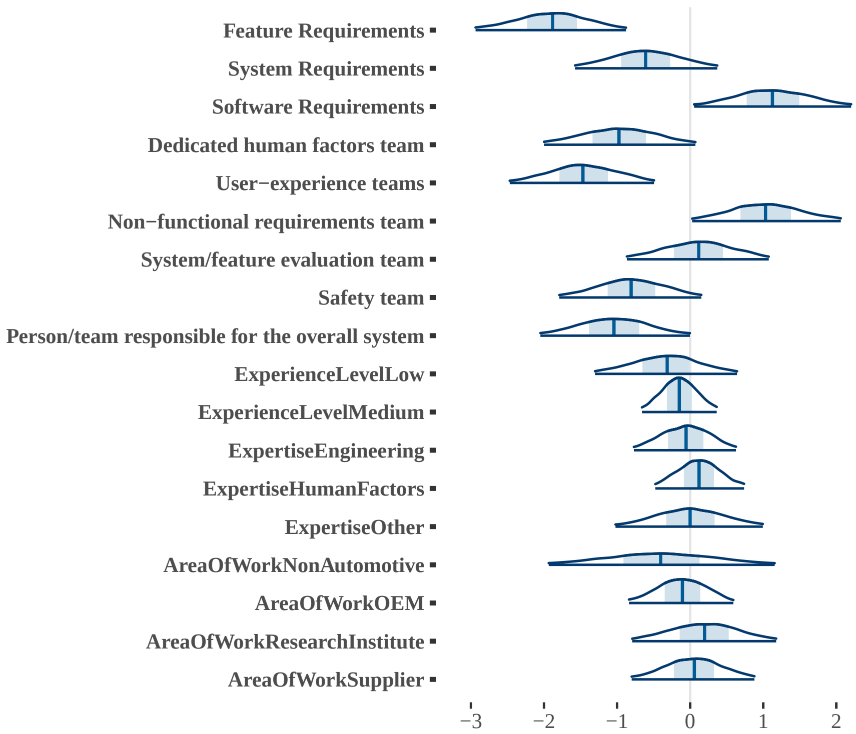


Figure 6.15: Density plot of all population-level effects on ranking

valuable knowledge and guidance, helping them develop a deeper understanding of human factors principles and their practical application within their project context.

6.6 Discussion

Our work aims to facilitate the integration of human factors requirements and expertise in contemporary development of AVs.

We began the in-depth investigation by exploring the extent to which human factors requirements can be confined to specific levels of requirements abstraction. Initially, at Stage 1, we examined how human factors requirements could be placed at each level of requirements abstraction within one company. In Stage 2, we broadened our scope and collected responses from a wider audience in a workshop setting, including participants from various organizations. This provided us with a comprehensive view of different companies' practices.

The results from the first session of the workshop showed a preference for specifying human factors requirements at the highest level of abstraction, i.e. *feature requirements*. This suggests a recognition of the importance of considering human factors from the very beginning of the design process [17]. However, after showing examples to the workshop participants that show how human factors can provide input at each level of requirements abstraction (in

Table 6.4: Summary of population-level effects on easiness.

	Estimate	Est. Error	l-95% CI	u-95% CI	Rhat	Bulk_ ESS	Tail_ ESS
Placement Options							
Feature Requirements	0.68	0.49	-0.28	1.67	1.00	2053	2921
System Requirements	0.55	0.50	-0.41	1.55	1.00	1958	2724
Software Requirements	-0.54	0.52	-1.54	0.48	1.00	2134	2640
Dedicated human factors team	**3.76**	**0.60**	**2.59**	**4.97**	**1.00**	**2112**	**2626**
User-experience teams	**3.96**	**0.61**	**2.82**	**5.15**	**1.00**	**2049**	**2477**
Non-functional requirements team	0.66	0.54	-0.36	1.75	1.00	2128	2665
System/feature evaluation team	**1.21**	**0.53**	**0.20**	**2.26**	**1.00**	**1951**	**2854**
Safety team	**1.83**	**0.54**	**0.79**	**2.87**	**1.00**	**1947**	**2388**
Person/team responsible for the overall system	-0.27	0.52	-1.25	0.76	1.00	2151	3137
Experience Levels							
ExperienceLevelLow	-0.15	0.53	-1.15	0.91	1.00	2340	2776
ExperienceLevelMedium	-0.40	0.29	-0.96	0.14	1.00	1807	2345
Expertise							
ExpertiseEngineering	-0.27	0.38	-1.02	0.48	1.00	1661	2381
ExpertiseHumanFactors	-0.12	0.34	-0.77	0.56	1.00	1855	2300
ExpertiseOther	0.15	0.48	-0.79	1.09	1.00	1822	2374
Area of Work							
AreaOfWorkNonAutomotive	-0.40	0.82	-2.09	1.24	1.00	2035	2305
AreaOfWorkOEM	-0.14	0.39	-0.91	0.63	1.00	1565	2094
AreaOfWorkResearchInstitute	0.03	0.53	-1.05	1.06	1.00	1688	2416
AreaOfWorkSupplier	-0.19	0.47	-1.15	0.75	1.00	1532	2059

the second session of workshop), there was a notable shift in the mindset - with most participants indicating that human factors should be considered at all levels. This shift highlights that even experts can be significantly influenced by practical examples.

It also highlights that the experts may not have a very good understanding of requirements levels, possibly because they are not accustomed to reflecting on different ways of working. It emphasizes the importance of education and awareness of human factors requirements, although more research is needed to provide guidance from empirical investigations. The effect of examples on the opinion of participating experts echoes Pikaar's findings on the influence of examples [311], although Pikaar's work primarily focuses on case studies.

We conclude with two insights from the results of RQ1:

[a] Even Experts require guidelines on how to integrate human factors requirements and expertise in development.

[b] Based on expert opinions, there is an indication that human factors requirements are relevant on all levels of abstraction.

However, resources may be limited to have human factors experts at all levels, so we continued to investigate strategic placement in RQ2.

Table 6.5: Summary of population-level effects on ranking.

	Estimate	Est. Error	l-95% CI	u-95% CI	Rhat	Bulk_ ESS	Tail_ ESS
Placement Options							
Feature Requirements	**-1.89**	**0.52**	**-2.94**	**-0.88**	**1.00**	**2029**	**2629**
System Requirements	-0.61	0.50	-1.58	0.37	1.00	2072	2328
Software Requirements	1.13	0.54	0.05	2.20	1.00	2267	2365
Dedicated human factors team	-0.97	0.53	-2.00	0.07	1.00	2340	2702
User-experience teams	**-1.46**	**0.50**	**-2.47**	**-0.49**	**1.00**	**1961**	**2236**
Non-functional requirements team	1.04	0.52	0.03	2.06	1.00	2240	2316
System/feature evaluation team	0.11	0.50	-0.87	1.08	1.00	1964	2540
Safety team	-0.81	0.49	-1.79	0.15	1.00	2091	2403
Person/team responsible for the overall system	-1.04	0.52	-2.05	-0.00	1.00	2215	2935
Experience Levels							
ExperienceLevelLow	-0.33	0.50	-1.30	0.64	1.00	4937	3254
ExperienceLevelMedium	-0.15	0.26	-0.66	0.36	1.00	5316	2906
Expertise							
ExpertiseEngineering	-0.06	0.36	-0.77	0.63	1.00	4292	3043
ExpertiseHumanFactors	0.12	0.31	-0.48	0.74	1.00	4429	3233
ExpertiseOther	0.00	0.50	-1.02	0.99	1.00	4491	2869
Area of Work							
AreaOfWorkNon-Automotive	-0.40	0.78	-1.94	1.16	1.00	4710	3573
AreaOfWorkOEM	-0.11	0.36	-0.84	0.59	1.00	3304	2773
AreaOfWorkResearchInstitute	0.19	0.50	-0.79	1.18	1.00	3661	2668
AreaOfWorkSupplier	0.05	0.42	-0.80	0.88	1.00	3507	3129

Our findings suggest various strategic placement options (S1-S11) for human factors experts, each with its own advantages and challenges. Many of these suggestions are also supported by Dul et al.'s strategy [307], recommendations from the Oil Companies International Marine Forum (OCIMF) [308], and Norman's principles of incorporating human factors concepts at different organizational levels [312]. However, the focus of their studies is different from ours. For example, Dul et al.'s work is focused on developing the discipline and profession of human factors and ergonomics [307], while OCIMF focuses on integrating human factors mainly in management systems, to help companies and leadership address the conditions and systems that influence human factors and decisions, promoting safety and excellence [308].

In order to provide further guidance to experts, we used these strategic placements as input to the third stage of our research by conducting a survey to collect broader experts opinions on how these placements should be ranked.

Considering the quantitative data analysis from Stage 3, in terms of all three main evaluative perspectives: general ranking, effectiveness, and the easiness of implementation of each solution proposition, placement within the *user-experience teams* consistently ranked high among the participants. This underscores the pivotal role of user-experience teams in strategically placing human factors expertise. Human factors, crucial for UX design, focus on optimizing products and systems for human use. Human-centered design principles, as highlighted by Norman [59] and the International Ergonomics

Association (IEA) [133], stress understanding users' needs to create user-friendly designs. Thus, the prominence of user-experience teams aligns with human factors' inherent value in UX.

While placement within the *feature requirements* ranks high in effectiveness and overall general ranking, results show that its implementation is not seen as easy. This suggests that while human factors expertise is vital for *feature requirements*, placing experts at the Feature Requirements level may be challenging. Similarly, the results indicate that placing human factors expertise with the *person/team responsible for the overall system* is effective. However, implementing this option is also not that easy. Our explanation for this observation is that while human factors expertise is beneficial for someone with an overview of the system, finding such an individual with the required technical knowledge and organizational coordination skills could be challenging.

Establishing a *dedicated human factors team* emerged as one of the easiest solutions to implement, indicating that organizations see the value and feasibility of dedicating resources to human factors expertise. A dedicated team can focus on human factors considerations, ensuring that user needs and usability are prioritized in design and development processes [313]. This approach also aligns well with human-centered design principles, emphasizing specialized teams that focus on user needs [314]. However, it is important to note that this option didn't rank high in terms of effectiveness, most likely because development teams are unlikely to regularly request input from experts in this constellation.

Our results show that while integrating human factors into *safety teams* is beneficial for safety considerations, it is limited in scope. This approach should be part of a broader human factors strategy, for example in compliance with ISO 9241 standards [134]. Throughout all results, human factors placement on the levels of *software requirements* and within *non-functional teams* consistently ranks low. This indicates that participants generally do not consider these aspects as the most suitable options for integrating human factors expertise.

The guidelines for creating a culture of human factors presented in Section 6.5.5 align with more general findings in existing literature. For example, the principle of *showcasing tangible results* resonates with the findings of Pikaar [311], highlighting the effectiveness of presenting outcomes. Similarly, the concept of *learning from past neglect* finds support in Rhaiem's and Amara's research [315], emphasizing the value of reflecting on previous oversights. Our research also highlighted the significant impact that specific examples can have on shaping participants' views. In our study, when participants in the workshop were shown specific examples of requirements from Table 6.1, their perspectives on incorporating human factors requirements at different levels of requirements abstraction shifted significantly, emphasizing awareness of the importance of human factors requirements. Furthermore, the idea of *creating a collaborative environment* aligns with the insights from O'Daniel and Rosenstein [316]. While their research primarily addresses the human factors related to individuals, it also underscores the importance and benefits of fostering a collaborative atmosphere. These guidelines, while not directly tied to human factors expertise, provide valuable insights into effective team motivation strategies.

Notably, the strategy of assigning *one human factors expert in each team* was not considered in our analysis. While this approach is arguably ideal, offering optimal integration of human factors expertise, our study was oriented toward finding the most suitable solutions within the constraints of limited resources. This decision was informed by previous research [253, 291] which indicated a general shortfall in human factors expertise across teams. Consequently, our focus was not on the apparently best solution but rather on identifying practical starting points for organizations seeking to enhance their human factors integration under resource constraints.

While our study has illuminated several key strategic placements for human factors expertise, however, it is important to recognize that this exploration may not have encompassed all potential placement options. Interestingly, placing human factors experts within development teams was not mentioned as a possible solution during the workshop. We could speculate that two main aspects play into this: a) it appears to be infeasible to place fully trained human factors experts in every team; b) it appears to be infeasible that engineers without extensive human factors expertise can manage human factors requirements. These are possible reasons that our experts did not mention this option, but more research would be needed to confirm our speculation.

6.6.1 Threats to Validity

In our mixed method research, we mitigate validity threats by conducting a thorough examination of issues inherent to mixed method methodologies.

In this study, several threats to ***internal validity*** were recognized. Selection bias, potentially distorting findings due to non-representative participant selection, was addressed through meticulous attention to diversity in participant inclusion across all stages of data collection. While Stage 1 predominantly involved shadowing techniques and interviews within a single automotive company, efforts were made to include multiple participants, ensuring varied perspectives. Similarly, Stage 2, centered around the SHAPE-IT workshop, gathered participants from diverse backgrounds and organizations, minimizing the risk of bias. Furthermore, Stage 3 expanded the participant pool to include professionals from various companies around the globe, contributing to a more comprehensive representation.

External validity concerns the extent to which research findings can be generalized to diverse contexts. In this mixed method study, we made deliberate efforts to mitigate potential limitations by including a diverse set of participants from various companies and backgrounds in Stage 2 and Stage 3 of the data collection.

Construct validity refers to the suitability of our measurement tools within the context of our research topic. It can become compromised when different interpretations of the terms and concepts used in the study emerge, resulting in varied interpretations during data collection. For example, the term “human factors” was subject to various interpretations. To reduce potential confusion, we provided clear definitions of basic terms and offered concrete examples to illustrate the concepts.

Additionally, our staged research approach significantly enhanced construct validity by progressively refining our understanding and alignment of key terms and concepts. Stage 1’s contextual and dialogue-based methods allowed us to explore and refine our understanding of key terms. In Stage 2, the workshop with a broader audience facilitated further clarification and alignment of terms through interactive sessions. Stage 3's survey incorporated these insights, ensuring consistent interpretation and reducing potential ambiguities. This iterative process of refinement across stages enabled us to clarify potential risks for misunderstanding and ensure uniformity in the use of constructs across all methods employed.

Reliability focuses on the consistency of our research method and whether other researchers conducting the same study would reach similar conclusions. To enhance reliability and recoverability, we strived for transparency in our research methodology and the derivation of our findings. We achieved this by presenting our data collection instruments and data replication package, which can be found here.

6.7 Conclusion

This paper sheds light on the consideration of human factors at different levels of requirements abstraction for AV systems and provides solution propositions for integrating human factors expertise within an organization. Our qualitative analysis identified potential strategic placements, while our quantitative analysis aimed to identify the optimal starting point for human factors expertise placement within an organization. The results show that the *user-experience teams* are the most effective choice for integrating human factors experts, with *feature requirements* teams being a close second, and *person team responsible for the overall system* comes at third place. In terms of ease of integration, *user-experience teams* again lead, followed closely by *dedicated human factors team*, and then *safety teams*. When ranking overall, *feature requirements* ranked highest, with *user experience teams* as a close second.

We hope that our results help organizations to find good strategic placements for human factors expertise to work efficiently with human factors requirements, despite the potential lack of experts.

Acknowledgements

The authors express their sincere gratitude to Prof. Richard Torkar for his invaluable guidance, support in the analysis, and constructive feedback. They also extend their thanks to the participating automotive company, all members actively involved in this study, workshop participants, and survey participants for their valuable contributions, input, and constructive feedback. This research was made possible by the support of the Marie Skłodowska-Curie grant, agreement number 860410, as a part of the European Union's Horizon 2020 research and innovation program.

Data Availability

The replication package, which includes, questionnaire, the original quantitative data, analysis scripts, figures, and results, is available on Zenodo at https://doi.org/10.5281/zenodo.11652602. However, the qualitative responses are not provided due to their sensitive nature, which could inadvertently disclose identifiable information.

Conflict of Interest

The authors declare that they have no conflict of interest.

Bibliography

[1] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015.

[2] W. D. Montgomery, "Public and Private Benefits of Autonomous Vehicles," Report, Autonomous Vehicle Workforce Study, 2018. [Online]. Available: https://avworkforce.secureenergy.org/wp-content/uploads/2018/06/W.-David-Montgomery-Report-June-2018.pdf

[3] SAE International, "SAE levels of driving automation™ refined for clarity and international audience," 2021, accessed on 23 August 2021. [Online]. Available: https://www.sae.org/blog/sae-j3016-update

[4] National Highway Traffic Safety Administration, "Automated Driving Systems 2.0: A Vision for Safety," US Department of Transportation, Tech. Rep. DOT HS 812 442, 2017. [Online]. Available: https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf

[5] A. Morando, P. Gershon, B. Mehler, and B. Reimer, "A model for naturalistic glance behavior around Tesla Autopilot disengagements," *Accident Analysis & Prevention*, vol. 161, p. 106348, 2021.

[6] J. D. Lee, C. D. Wickens, Y. Liu, and L. N. Boyle, *Designing for People: An Introduction to Human Factors Engineering*, 3rd ed. Scotts Valley, CA: CreateSpace Independent Publishing Platform, 2017.

[7] P. A. Hancock, "Automation: how much is too much?" *Ergonomics*, vol. 57, no. 3, pp. 449–454, 2014.

[8] P. A. Hancock, "Imposing limits on autonomous systems," *Ergonomics*, vol. 60, no. 2, pp. 284–291, 2017.

[9] C. D. Wickens, S. E. Gordon, Y. Liu, and J. Lee, *An introduction to human factors engineering.* Pearson Prentice Hall Upper Saddle River, NJ, 2004, vol. 2.

[10] J. Navarro, "A state of science on highly automated driving," *Theoretical Issues in Ergonomics Science*, vol. 20, no. 3, pp. 366–396, 2019.

[11] Human Factors and Ergonomics Society, "Definitions of human factors and ergonomics," Federated Society of the International Ergonomics Association, 2021, accessed on February 17, 2021. [Online]. Available: https://www.hfes.org/About-HFES/What-is-Human-Factors-and-Ergonomics

[12] United Nations Economic Commission for Europe (UNECE), "R157: Automated Lane Keeping Systems (ALKS)," Online resource, 2023, available at: https://unece.org/sites/default/files/2023-12/R157e.pdf, accessed on 11 December 2023.

[13] P. A. Hancock, "Some pitfalls in the promises of automated and autonomous vehicles," *Ergonomics :1*, 2019.

[14] W. Biever, L. Angell, and S. Seaman, "Automated driving system collisions: Early lessons," *Human factors*, vol. 62, no. 2, pp. 249–259, 2020.

[15] M. H. Calp and M. A. Akcayol, "The importance of human computer interaction in the development process of software projects," *arXiv preprint arXiv:1902.02757*, 2019.

[16] Z. K. Chua and K. M. Feigh, "Integrating human factors principles into systems engineering," in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference.* IEEE, 2011, pp. 6A1–1.

[17] E. Håkansson and E. Bjarnason, "Including human factors and ergonomics in requirements engineering for digital work environments," in *2020 IEEE First International Workshop on Requirements Engineering for Well-Being, Aging, and Health (REWBAH).* IEEE, 2020, pp. 57–66.

[18] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries *et al.*, "Manifesto for agile software development," 2001. [Online]. Available: https://agilemanifesto.org/

[19] T. Dingsøyr and N. B. Moe, "Towards principles of large-scale agile development," in *International Conference on Agile Software Development.* Springer, 2014, pp. 1–8.

[20] E.-M. Schön, J. Thomaschewski, and M. J. Escalona, "Agile requirements engineering: A systematic literature review," *Computer standards & interfaces*, vol. 49, pp. 79–91, 2017.

[21] R. Kasauli, E. Knauss, J. Horkoff, G. Liebel, and F. G. de Oliveira Neto, "Requirements engineering challenges and practices in large-scale agile system development," *Journal of Systems and Software*, vol. 172, p. 110851, 2021.

[22] R. Steinberg and N. Grumman, "Human factors at the speed of relevance for agile engineering," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 66, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2022, pp. 1897–1901.

[23] D. Leffingwell and D. Widrig, *Managing software requirements: a unified approach.* Addison-Wesley Professional, 2000.

[24] International Requirements Engineering Board, "IREB certified professional for requirements engineering: Foundation level," IREB e.V., Tech. Rep. Version 3.0.1, 2020. [Online]. Available: https://www.ireb.org/en/downloads/#cpre-foundation-level-syllabus-3-0

[25] G. Kotonya and I. Sommerville, *Requirements engineering: processes and techniques.* Wiley Publishing, 1998.

[26] B. Boehm, "Some future trends and implications for systems and software engineering processes," *Systems Engineering*, vol. 9, no. 1, pp. 1–19, 2006.

[27] P. Serrador and J. K. Pinto, "Does Agile work? A quantitative analysis of agile project success," *International Journal of Project Management*, vol. 33, no. 5, pp. 1040–1051, 2015.

[28] B. Meyer, "The ugly, the hype and the good: an assessment of the agile approach," in *Agile!* Springer, 2014, pp. 149–154.

[29] K. Beck, *Extreme Programming Explained: Embrace Change.* Addison-Wesley, 1999.

[30] A. Jarzebowicz and P. Weichbroth, "A systematic literature review on implementing non-functional requirements in agile software development: Issues and facilitating practices," in *Lean and Agile Software Development: 5th International Conference, LASD 2021, Virtual Event, January 23, 2021, Proceedings 5.* Springer, 2021, pp. 91–110.

[31] B. Ramesh, L. Cao, and R. Baskerville, "Agile requirements engineering practices and challenges: an empirical study," *Information Systems Journal*, vol. 20, no. 5, pp. 449–480, 2010.

[32] W. Alsaqaf, M. Daneva, and R. Wieringa, "Quality requirements in large-scale distributed agile projects–a systematic literature review," in *Requirements Engineering: Foundation for Software Quality: 23rd International Working Conference, REFSQ 2017, Essen, Germany, February 27–March 2, 2017, Proceedings 23.* Springer, 2017, pp. 219–234.

[33] M. Saleh, F. Baharom, S. F. P. Mohamed, and M. Ahmad, "A systematic literature review of challenges and critical success factors in agile requirement engineering," in *Proceedings of the Knowledge Management International Conference (KMICe) 2018*, 2018, pp. 248–254.

[34] E. Sherif, W. Helmy, and G. H. Galal-Edeen, "Managing non-functional requirements in agile software development," in *International Conference on Computational Science and Its Applications.* Springer, 2022, pp. 205–216.

[35] C. Larman, *Practices for scaling lean & Agile development: large, multisite, and offshore product development with large-scale scrum.* Pearson Education, 2010.

[36] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp. 87–108, 2016.

[37] R. Kasauli, E. Knauss, J. Nakatumba-Nabende, and B. Kanagwa, "Agile islands in a waterfall environment: Requirements engineering challenges and strategies in automotive," in *Proc. of International Conference on Evaluation and Assessment in Software Engineering (EASE)*, Trondheim, Norway, 2020, pp. 31–40.

[38] D. Leffingwell, *Agile software requirements: lean requirements practices for teams, programs, and the enterprise.* Addison-Wesley Professional, 2010.

[39] P. d. O. Santos and M. M. de Carvalho, "Exploring the challenges and benefits for scaling agile project management to large projects: a review," *Requirements engineering*, vol. 27, no. 1, pp. 117–134, 2022.

[40] A. Putta, M. Paasivaara, and C. Lassenius, "Benefits and challenges of adopting the scaled agile framework (safe): preliminary results from a multivocal literature review," in *Product-Focused Software Process Improvement: 19th International Conference, PROFES 2018, Wolfsburg, Germany, November 28–30, 2018, Proceedings 19.* Springer, 2018, pp. 334–351.

[41] V. T. Heikkilä, D. Damian, C. Lassenius, and M. Paasivaara, "A mapping study on requirements engineering in agile software development," in *2015 41st Euromicro conference on software engineering and advanced applications.* IEEE, 2015, pp. 199–207.

[42] E. Bjarnason, K. Wnuk, and B. Regnell, "A case study on benefits and side-effects of agile practices in large-scale requirements engineering," in *proceedings of the 1st workshop on agile requirements engineering*, 2011, pp. 1–5.

[43] R. Kasauli, G. Liebel, E. Knauss, S. Gopakumar, and B. Kanagwa, "Requirements engineering challenges in large-scale agile system development," in *International Requirements Engineering Conference (RE)*, 2017, pp. 352–361.

[44] F. Paetsch, A. Eberlein, and F. Maurer, "Requirements Engineering and Agile Software Development," in *International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises.* IEEE, 2003, pp. 308–313.

[45] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Computers in human behavior*, vol. 51, pp. 915–929, 2015.

[46] Z. Hoy and M. Xu, "Agile software requirements engineering challenges-solutions—a conceptual framework from systematic literature review," *Information*, vol. 14, no. 6, p. 322, 2023.

[47] R. Hoda, J. Noble, and S. Marshall, "How much is just enough? Some documentation patterns on Agile projects," in *Proceedings of the 15th European Conference on Pattern Languages of Programs*, 2010, pp. 1–13.

[48] Requirements Engineering Magazine, "Inputs to requirements engineering in agile projects," 2021, accessed: 2024-09-05. [Online]. Available: https://re-magazine.ireb.org/articles/inputs-to-requirements-engineering-in-agile-projects

[49] C. Berger and U. Eklund, "Expectations and challenges from scaling agile in mechatronics-driven companies–a comparative case study," in *International Conference on Agile Software Development.* Springer, 2015, pp. 15–26.

[50] U. Eklund, H. Holmström Olsson, and N. J. Strøm, "Industrial challenges of scaling agile in mass-produced embedded systems," in *International Conference on Agile Software Development.* Springer, 2014, pp. 30–42.

[51] L. Gren and P. Lenberg, "Agility is responsiveness to change: An essential definition," in *Proc. of the Evaluation and Assessment in Software Engineering*, 2020, pp. 348–353. [Online]. Available: https://doi.org/10.1145/3383219.3383265

[52] G. Liebel, M. Tichy, and E. Knauss, "Use, potential, and showstoppers of models in automotive requirements engineering," *Software & Systems Modeling*, vol. 18, no. 4, pp. 2587–2607, 2019.

[53] SAE, "SAE J3016:201806 - SURFACE VEHICLE RECOMMENDED PRACTICE - Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," 2018.

[54] Health and Safety Executive (HSE), "Introduction to Human Factors," Online resource, 2022, accessed on 14 November 2022, available at: https://www.hse.gov.uk/humanfactors/introduction.htm.

[55] D. Licht, D. Polzella, and K. Boff, "Human factors, ergonomics, and human factors engineering: Analysis of definitions (tech. rep. 89-01)," *Wright-Patterson Air Force Base OH: CSERIAC Program Office*, 1991.

[56] L. Pirzadeh, "Human factors in software development: a systematic literature review," Master's thesis, Dept. of Computer Science and Engineering, Chalmers | University of Gothenburg, 2010.

[57] C. D. Wickens, S. E. Gordon-Becker, Y. Liu, and J. D. Lee, *An Introduction to Human Factors Engineering*, 2nd ed. Pearson Prentice Hall, 2004.

[58] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human factors*, vol. 37, no. 1, pp. 32–64, 1995.

[59] D. A. Norman, *The Design of Everyday Things: Revised and Expanded Edition.* Basic Books (US) and MIT Press (UK edition), 2013.

[60] J. D. Lee and K. A. See, "Trust in automation: Designing for appropriate reliance," *Human factors*, vol. 46, no. 1, pp. 50–80, 2004.

[61] N. Stanton, "Advances in human factors of transportation, proceedings of the ahfe 2019 international conference on human factors in transportation," *Advances in Intelligent Systems and Computing*, 2019.

[62] Y. Xu, B. Chen, X. Shan, W. Jia, Z. Lu, and G. Xu, "Model predictive control for lane keeping system in autonomous vehicle," in *2017 7th International Conference on Power Electronics Systems and Applications-Smart Mobility, Power Transfer & Security (PESA)*. IEEE, 2017, pp. 1–5.

[63] E. E. Miller and L. N. Boyle, "Behavioral adaptations to lane keeping systems: Effects of exposure and withdrawal," *Human factors*, vol. 61, no. 1, pp. 152–164, 2019.

[64] C. Ackermann, M. Beggiato, S. Schubert, and J. F. Krems, "An experimental study to investigate design and assessment criteria: What is important for communication between pedestrians and automated vehicles?" *Applied ergonomics*, vol. 75, pp. 272–282, 2019.

[65] S. M. Faas, L.-A. Mathis, and M. Baumann, "External HMI for self-driving vehicles: which information shall be displayed?" *Transportation research part F: traffic psychology and behaviour*, vol. 68, pp. 171–186, 2020.

[66] O. Carsten and M. H. Martens, "How can humans understand their automated cars? HMI principles, problems and solutions," *Cognition, Technology & Work*, vol. 21, no. 1, pp. 3–20, 2019.

[67] S. Salter, C. Diels, P. Herriotts, S. Kanarachos, and D. Thake, "Motion sickness in automated vehicles with forward and rearward facing seating orientations," *Applied ergonomics*, vol. 78, pp. 54–61, 2019.

[68] N. Merat, A. H. Jamson, F. C. Lai, M. Daly, and O. M. Carsten, "Transition to manual: Driver behaviour when resuming control from a highly automated vehicle," *Transportation research part F: traffic psychology and behaviour*, vol. 27, pp. 274–282, 2014.

[69] H. Azevedo-Sa, S. K. Jayaraman, C. T. Esterwood, X. J. Yang, L. P. Robert Jr, and D. M. Tilbury, "Real-time estimation of drivers' trust in automated driving systems," *International Journal of Social Robotics*, vol. 13, no. 8, pp. 1911–1927, 2021.

[70] Z. K. Chua and K. M. Feigh, "Integrating human factors principles into systems engineering," in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*. IEEE, 2011, pp. 6A1–1.

[71] O. Sohaib and K. Khan, "Integrating usability engineering and agile software development: A literature review," in *2010 international conference on Computer design and applications*, vol. 2. IEEE, 2010, pp. V2–32.

[72] J. Ferreira, J. Noble, and R. Biddle, "Agile development iterations and ui design," in *Agile 2007 (AGILE 2007)*. IEEE, 2007, pp. 50–58.

[73] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting empirical methods for software engineering research," in *Guide to advanced empirical software engineering*. Springer, 2008, pp. 285–311.

[74] C. Wohlin, M. Höst, and K. Henningsson, "Empirical research methods in software engineering," in *Empirical methods and studies in software engineering.* Springer, 2003, pp. 7–23.

[75] J. W. Creswell and J. D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches.* Sage publications, 2017.

[76] H. L. Stuckey, "The second step in data analysis: Coding qualitative research data," *Journal of Social Health and Diabetes*, vol. 3, no. 01, pp. 007–010, 2015.

[77] R. J. Wieringa, "Design science as nested problem solving," in *4th Intl. Conf. on Design Science Research in Inf. Sys. and Techn.*, Philadelphia, 2009, pp. 1–12.

[78] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[79] E. Knauss, "Constructive master's thesis work in industry: Guidelines for applying design science research," in *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 2021, pp. 110–121.

[80] J. W. Creswell and J. D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches.* Sage publications, 2017.

[81] E.-M. Ihantola and L.-A. Kihn, "Threats to validity and reliability in mixed methods accounting research," *Qualitative Research in Accounting & Management*, vol. 8, no. 1, pp. 39–58, 2011.

[82] A. Elhami and B. Khoshnevisan, "Conducting an interview in qualitative research: The modus operandi." *Mextesol Journal*, vol. 46, no. 1, p. n1, 2022.

[83] J. A. Smith, *Qualitative psychology: A practical guide to research methods.* Sage Publications Ltd, 2024.

[84] F. J. Fowler Jr, *Survey research methods.* Sage publications, 2013.

[85] M. S. Matell and J. Jacoby, "Is there an optimal number of alternatives for likert-scale items? effects of testing time and scale properties." *Journal of applied psychology*, vol. 56, no. 6, p. 506, 1972.

[86] G. A. Bowen, "Document analysis as a qualitative research method," *Qualitative research journal*, vol. 9, no. 2, pp. 27–40, 2009.

[87] R. Ørngreen and K. T. Levinsen, "Workshops as a research methodology," *Electronic Journal of E-learning*, vol. 15, no. 1, pp. 70–81, 2017.

[88] M. Angrosino, *Doing ethnographic and observational research.* Sage Publications Ltd, 2007.

[89] S. McDonald, "Studying actions in context: a qualitative shadowing method for organizational research," *Qualitative research*, vol. 5, no. 4, pp. 455–473, 2005.

[90] Swedish Ethical Review Authority, "Swedish Ethical Review Authority," https://etikprovningsmyndigheten.se, accessed: 2024-12-10.

[91] P. Hayashi Jr, G. Abib, and N. Hoppen, "Validity in qualitative research: A processual approach," *The Qualitative Report*, vol. 24, no. 1, pp. 98–112, 2019.

[92] J. A. Maxwell, "Qualitative research design: An interactive approach," *Sage publications*, 2012.

[93] R. Whittemore, S. K. Chase, and C. L. Mandle, "Validity in qualitative research," *Qualitative health research*, pp. 522–537, 2001.

[94] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical software engineering*, vol. 14, no. 2, pp. 131–164, 2009.

[95] J. C. De Winter, R. Happee, M. H. Martens, and N. A. Stanton, "Effects of adaptive cruise control and highly automated driving on workload and situation awareness: A review of the empirical evidence," *Transportation research part F: traffic psychology and behaviour*, vol. 27, pp. 196–217, 2014.

[96] J. D. Lee, "Humans and Automation: Use, Misuse, Disuse, Abuse," *Human Factors*, vol. 50, no. 3, pp. 404–410, 2008.

[97] R. Van Der Valk, P. Pelliccione, P. Lago, R. Heldal, E. Knauss, and J. Juul, "Transparency and contracts: continuous integration and delivery in the automotive ecosystem," in *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*. IEEE, 2018, pp. 23–32.

[98] S. J. Czaja and S. N. Nair, "Human factors engineering and systems design," *Handbook of human factors and ergonomics*, vol. 3, pp. 32–53, 2006.

[99] M. Glinz and S. A. Fricker, "On shared understanding in software engineering: an essay," *Computer Science-Research and Development*, vol. 30, no. 3, pp. 363–376, 2015.

[100] L.-O. Bligård, *Predicting mismatches in user–artefact interaction. Development of an analytical methodology to support design work.* Chalmers Tekniska Hogskola (Sweden), 2012.

[101] G. Schermann, J. Cito, and P. Leitner, "Continuous experimentation: challenges, implementation techniques, and current research," *Ieee Software*, vol. 35, no. 2, pp. 26–31, 2018.

[102] F. Dumitriu, G. Meșniță, and L.-D. Radu, "Challenges and solutions of applying large-scale agile at organizational level," *Informatica Economica*, vol. 23, no. 3, pp. 61–71, 2019.

[103] R. Kasauli, E. Knauss, B. Kanagwa, A. Nilsson, and G. Calikli, "Safety-critical systems and agile development: A mapping study," in *Proc. of Euromicro SEAA*, Prague, Czech Republic, 2018, pp. 470–477.

[104] TryQA, "What Is Test Strategy?" 2022, accessed on 27 October 2022. [Online]. Available: https://tryqa.com/what-is-test-strategy-types-of-strategies-with-examples/

[105] A. D. Cross, "Exploring software testing strategies used on software applications in the government," Ph.D. dissertation, Walden University, 2020.

[106] N. E. Haskouri, "Requirement strategy in large-scale agile development: A design science research," Master's thesis, Dept. of Computer Science and Engineering, Chalmers | University of Gothenburg, 2021, https://gupea.ub.gu.se/bitstream/2077/69096/1/gupea_2077_69096_1.pdf.

[107] E. M. Méndez, M. A. Pérez, and L. E. Mendoza, "Improving Software Test Strategy with a Method to Specify Test Cases (MSTC)," in *ICEIS (1)*, 2008, pp. 159–164.

[108] J. A. Cannon-Bowers and E. Salas, "Reflections on shared cognition," *Journal of Organizational Behavior: The International Journal of Industrial, Occupational and Organizational Psychology and Behavior*, vol. 22, no. 2, pp. 195–202, 2001.

[109] X. Zhang, G. Auriol, H. Eres, and C. Baron, "A prescriptive approach to qualify and quantify customer value for value-based requirements engineering," *International Journal of Computer Integrated Manufacturing*, vol. 26, no. 4, pp. 327–345, 2013.

[110] A. Cockburn, *Writing effective use cases.* Pearson Education India, 2001.

[111] A. P. Muhammad, E. Knauss, O. Batsaikhan, N. E. Haskouri, Y.-C. Lin, and A. Knauss, "Defining requirements strategies in agile: A design science research study," in *International Conference on Product-Focused Software Process Improvement.* Springer, 2022, pp. 73–89.

[112] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries *et al.*, "Manifesto for Agile Software Development," Online resource, 2001, available at: https://agilemanifesto.org/.

[113] J. D. Lee and B. D. Seppelt, "Human factors in automation design," *Springer handbook of automation*, pp. 417–436, 2009.

[114] M. Martens and A. van den Beukel, "The road to automated driving: Dual mode and human factors considerations," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 2262–2267.

[115] W. Karwowski, "Ergonomics and human factors: the paradigms for science, engineering, design, technology and management of human-compatible systems," *Ergonomics*, vol. 48, no. 5, pp. 436–463, 2005.

[116] M. Saghafian, T. A. Sitompul, K. Laumann, K. Sundnes, and R. Lindell, "Application of human factors in the development process of immersive visual technologies: challenges and future improvements," *Frontiers in psychology*, vol. 12, p. 634352, 2021.

[117] F. Fagerholm, A. S. Guinea, H. Mäenpää, and J. Münch, "The right model for continuous experimentation," *Journal of Systems and Software*, vol. 123, pp. 292–305, 2017.

[118] G. Salvendy, *Handbook of human factors and ergonomics.* John Wiley & Sons, 2012.

[119] P. A. Hancock, "Some pitfalls in the promises of automated and autonomous vehicles," *Ergonomics*, vol. 62, no. 4, pp. 479–495, 2019.

[120] N. J. Goodall, "Machine ethics and automated vehicles," *Road vehicle automation*, pp. 93–102, 2014.

[121] I. G. Cohen, R. Amarasingham, A. Shah, B. Xie, and B. Lo, "The legal and ethical concerns that arise from using complex predictive analytics in health care," *Health affairs*, vol. 33, no. 7, pp. 1139–1147, 2014.

[122] P. Schnöll, "Human-centered test setups for the evaluation of human-technology interaction in cockpits of highly-automated vehicles," in *Human Interaction, Emerging Technologies and Future Applications IV: Proceedings of the 4th International Conference on Human Interaction and Emerging Technologies: Future Applications (IHIET–AI 2021), April 28-30, 2021, Strasbourg, France 4.* Springer, 2021, pp. 555–562.

[123] M. Benmimoun, A. Pütz, A. Zlocki, and L. Eckstein, "eurofot: Field operational test and impact assessment of advanced driver assistance systems: Final results," in *Proceedings of the FISITA 2012 World Automotive Congress: Volume 9: Automotive Safety Technology.* Springer, 2013, pp. 537–547.

[124] M. Penttinen, P. Rämä, M. Dotzauer, D. Hibberd, S. Innamaa, T. Louw, T. Streubel, B. Metz, J. Wörle, R. Brouwer *et al.*, "Experimental procedure: Deliverable d3. 2 of l3pilot," 2019.

[125] N. B. Moe, T. Dingsøyr, and T. Dybå, "A teamwork model for understanding an agile team: A case study of a scrum project," *Information and software technology*, vol. 52, no. 5, pp. 480–491, 2010.

[126] P. G. Smith and D. G. Reinertsen, *Developing products in half the time: new rules, new tools.* John Wiley & Sons, 1997.

[127] I. Sommerville, "Software engineering (ed.)," *America: Pearson Education Inc*, 2011.

[128] E. Hull, K. Jackson, and J. Dick, *Requirements engineering in the solution domain.* Springer, 2005.

[129] J. Cleland-Huang, "Traceability in agile projects," in *Software and Systems Traceability.* Springer, 2011, pp. 265–275.

[130] B. Boehm and R. Turner, "Management challenges to implementing agile processes in traditional development organizations," *IEEE Software*, vol. 22, no. 5, pp. 30–39, 2005.

[131] A. Cockburn and J. Highsmith, "Agile software development, the people factor," *Computer*, vol. 34, no. 11, pp. 131–133, 2001.

[132] M. Lycett, R. D. Macredie, C. Patel, and R. J. Paul, "Migrating agile methods to standardized development practice," *Computer*, vol. 36, no. 6, pp. 79–85, 2003.

[133] International Ergonomics & Human Factors Association, 2024, online (assessed: 2024-Apr-16). [Online]. Available: https://iea.cc/about/what-is-ergonomics/

[134] International Organization for Standardization, "ISO 9241-210:2019 - Ergonomics of Human-System Interaction – Part 210: Human-Centred Design for Interactive Systems," International Organization for Standardization, 2019, ISO 9241-210:2019(en), Available at: https://www.iso.org/standard/77520.html.

[135] Y. Erdal, "Increasing the safety and reliability of autonomous vehicles," Integrated Transport Research Lab, KTH, Tech. Rep., 2018, online (accessed: 2020-Oct-2). [Online]. Available: https://www.itrl.kth.se/polopoly_fs/1.863811.1600689663!/YasinErdal_IRPs.pdf

[136] H. Yu, H. E. Tseng, and R. Langari, "A human-like game theory-based controller for automatic lane changing," *Transportation Research Part C: Emerging Technologies*, vol. 88, pp. 140–158, 2018.

[137] T.-F. Wu, P.-S. Tsai, N.-T. Hu, J.-Y. Chen, and J.-J. Jheng, "Implementation of auto parking system based on ultrasonic sensors," *Journal of Internet Technology*, vol. 20, no. 2, pp. 471–482, 2019.

[138] M. Schwall, T. Daniel, T. Victor, F. Favaro, and H. Hohnhold, "Waymo public road safety performance data," *arXiv preprint arXiv:2011.00038*, 2020.

[139] M. Anderson, "The road ahead for self-driving cars: The av industry has had to reset expectations, as it shifts its focus to level 4 autonomy-[news]," *IEEE Spectrum*, vol. 57, no. 5, pp. 8–9, 2020.

[140] T. Litman, "Autonomous vehicle implementation predictions implications for transport planning," Victoria Transport Policy Institute, Tech. Rep., 2021. [Online]. Available: https://www.vtpi.org/avip.pdf

[141] V. A. Banks, K. L. Plant, and N. A. Stanton, “Driver error or designer error: Using the perceptual cycle model to explore the circumstances surrounding the fatal tesla crash on 7th may 2016,” *Safety science*, vol. 108, pp. 278–285, 2018.

[142] Tesla Deaths, “Tesla Deaths: A Record of Tesla Accidents Involving Driver, Occupant, Cyclist, Motorcyclist, or Pedestrian Fatalities,” Tech. Rep., 2020, accessed on 17 February 2021, available at: https://www.tesladeaths.com.

[143] National Transportation Safety Board, “Preliminary Report Highway HWY19FH008,” Published online, 2019, available at: https://www.ntsb.gov/investigations/Pages/HWY19FH008.aspx, accessed on 2021-10-11.

[144] P. Kohli and A. Chadha, “Enabling Pedestrian Safety Using Computer Vision Techniques: A Case Study of the 2018 Uber Inc. Self-driving Car Crash,” in *Future of Information and Communication Conference*. Springer, 2019, pp. 261–279.

[145] T. Inagaki and M. Itoh, “Human's overtrust in and overreliance on advanced driver assistance systems: a theoretical framework,” *International journal of vehicular technology*, vol. 2013, 2013.

[146] Euronews, “Tesla Autopilot Crash Investigation Expands as Ford, BMW and 10 Other Carmakers Asked for Data,” 2021, published online on 2021-09-16, accessed on 2021-10-11.

[147] Los Angeles Times, “A Tesla Mystery: Why Didn't Auto-Braking Stop These Crashes?” Published online, 2021, published on 2021-10-07, available at: https://www.latimes.com/business/story/2021-10-07/why-arent-automatic-braking-systems-stopping-deadly-tesla-crashes, accessed on 2021-10-11.

[148] P. A. Hancock, “Automation: how much is too much?” *Ergonomics*, vol. 57, no. 3, pp. 449–454, 2014.

[149] J. D. Lee, “Review of a pivotal human factors article: “humans and automation: use, misuse, disuse, abuse”,” *Human Factors*, vol. 50, no. 3, pp. 404–410, 2008.

[150] B. Meyer, *Agile!: The Good, the Hype and the Ugly.* Springer, 2014.

[151] T. Kahkonen, “Agile methods for large organizations-building communities of practice,” in *Agile development conference.* IEEE, 2004, pp. 2–10.

[152] L. Lagerberg, T. Skude, P. Emanuelsson, K. Sandahl, and D. Ståhl, “The impact of agile principles and practices on large-scale software development projects: A multiple-case study of two projects at ericsson,” in *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement.* IEEE, 2013, pp. 348–356.

[153] O. Salo and P. Abrahamsson, “Agile methods in european embedded software development organisations: a survey on the actual use and usefulness of extreme programming and scrum,” *IET software*, vol. 2, no. 1, pp. 58–64, 2008.

[154] J. Pernstål, A. Magazinius, and T. Gorschek, “A study investigating challenges in the interface between product development and manufacturing in the development of software-intensive automotive systems,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 22, no. 07, pp. 965–1004, 2012.

[155] R. Knaster and D. Leffingwell, *SAFe 4.0 distilled: applying the Scaled Agile Framework for lean software and systems engineering.* Addison-Wesley Professional, 2017.

[156] K. Wiegers and J. Beatty, *Software requirements.* Pearson Education, 2013.

[157] R. Kasauli, E. Knauss, J. Horkoff, G. Liebel, and F. G. de Oliveira Netoa, “Requirements engineering challenges and practices in large-scale agile system development,” *Systems and Software*, 2020. [Online]. Available: https://doi.org/10.1016/j.jss.2020.110851

[158] G. Beauchamp, “Technology in the dystopian novel,” *Modern fiction studies*, vol. 32, no. 1, pp. 53–63, 1986.

[159] C. Gold, F. Naujoks, J. Radlmayr, H. Bellem, and O. Jarosch, “Testing scenarios for human factors research in level 3 automated vehicles,” in *International conference on applied human factors and ergonomics.* Springer, 2017, pp. 551–559.

[160] I. J. Reagan, D. G. Kidd, and J. B. Cicchino, “Driver acceptance of adaptive cruise control and active lane keeping in five production vehicles,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 61. SAGE Publications Sage CA: Los Angeles, CA, 2017, pp. 1949–1953.

[161] A. Morando, T. Victor, and M. Dozza, “Drivers anticipate lead-vehicle conflicts during automated longitudinal control: Sensory cues capture driver attention and promote appropriate and timely responses,” *Accident Analysis & Prevention*, vol. 97, pp. 206–219, 2016.

[162] G. Abe, K. Sato, and M. Itoh, “Driver trust in automated driving systems: The case of overtaking and passing,” *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 1, pp. 85–94, 2017.

[163] J. Kovaceva, G. Nero, J. Bärgman, and M. Dozza, “Drivers overtaking cyclists in the real-world: Evidence from a naturalistic driving study,” *Safety science*, vol. 119, pp. 199–206, 2019.

[164] C. Ackermann, M. Beggiato, S. Schubert, and J. F. Krems, “An experimental study to investigate design and assessment criteria: What is important for communication between pedestrians and automated vehicles?” *Applied ergonomics*, vol. 75, pp. 272–282, 2019.

[165] A. G. Mirnig, P. Wintersberger, C. Sutter, and J. Ziegler, "A framework for analyzing and calibrating trust in automated vehicles," in *Adjunct proceedings of the 8th international conference on automotive user interfaces and interactive vehicular applications*, 2016, pp. 33–38.

[166] J. Kraus, D. Scholz, D. Stiegemeier, and M. Baumann, "The more you know: trust dynamics and calibration in highly automated driving and the effects of take-overs, system malfunction, and system transparency," *Human factors*, vol. 62, no. 5, pp. 718–736, 2020.

[167] M. Kyriakidis, J. C. de Winter, N. Stanton, T. Bellet, B. van Arem, K. Brookhuis, M. H. Martens, K. Bengler, J. Andersson, N. Merat *et al.*, "A human factors perspective on automated driving," *Theoretical issues in ergonomics science*, vol. 20, no. 3, pp. 223–249, 2019.

[168] I. Y. Noy, D. Shinar, and W. J. Horrey, "Automated driving: Safety blind spots," *Safety science*, vol. 102, pp. 68–78, 2018.

[169] CARTRE, "Position paper on human factors," Coordination of Automated Road Transport Deployment for Europe, Tech. Rep., 2018. [Online]. Available: https://knowledge-base.connectedautomateddriving.eu/wp-content/uploads/2019/08/CARTRE-Human-Factors-Theme-PositionPaper_180904_FINAL.pdf

[170] M. Saffarian, J. C. De Winter, and R. Happee, "Automated driving: human-factors issues and design solutions," in *Proceedings of the human factors and ergonomics society annual meeting*, vol. 56. Sage Publications Sage CA: Los Angeles, CA, 2012, pp. 2296–2300.

[171] J. Y. Chen, S. G. Lakhmani, K. Stowers, A. R. Selkowitz, J. L. Wright, and M. Barnes, "Situation awareness-based agent transparency and human-autonomy teaming effectiveness," *Theoretical issues in ergonomics science*, vol. 19, no. 3, pp. 259–282, 2018.

[172] IEEE, "IEEE Standard Glossary of Software Engineering Terminology," *IEEE Std 610.12-1990*, pp. 1–84, 1990.

[173] International Requirements Engineering Board, "IREB certified professional for requirements engineering: Foundation level," IREB e.V., Tech. Rep. Version 3.0.1, 2020. [Online]. Available: https://www.ireb.org/en/downloads/#cpre-foundation-level-syllabus-3-0

[174] B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap," in *Proceedings of the Conference on the Future of Software Engineering*, 2000, pp. 35–46.

[175] V. T. Heikkilä, M. Paasivaara, C. Lasssenius, D. Damian, and C. Engblom, "Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson," *Empirical Software Engineering*, pp. 1–45, 2017.

[176] A. Knauss, D. Damian, X. Franch, A. Rook, H. A. Müller, and A. Thomo, "Acon: A learning-based approach to deal with

uncertainty in contextual requirements at runtime," *Information and Software Technology*, vol. 70, pp. 85 – 99, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584915001676

[177] F. Fagerholm, A. S. Guinea, H. Mäenpää, and J. Münch, "The right model for continuous experimentation," *Systems and Software*, vol. 123, pp. 292–305, 2017.

[178] M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen, "Large-scale agile transformation at Ericsson: a case study," *Empirical Software Engineering*, vol. 23, no. 5, pp. 2550–2596, 2018.

[179] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*. Prentice Hall PTR, 2001.

[180] P. Hohl, J. Münch, K. Schneider, and M. Stupperich, "Forces that prevent agile adoption in the automotive domain," in *International Conference on Product-Focused Software Process Improvement*. Springer, 2016, pp. 468–476.

[181] M. Lindvall, D. Muthig, A. Dagnino, C. Wallin, M. Stupperich, D. Kiefer, J. May, and T. Kahkonen, "Agile software development in large organizations," *Computer*, vol. 37, no. 12, pp. 26–34, Dec 2004.

[182] R. Wohlrab, E. Knauss, and P. Pelliccione, "Why and how to balance alignment and diversity of requirements engineering practices in automotive," *Systems and Software*, vol. 162, 2019. [Online]. Available: https://doi.org/10.1016/j.jss.2019.110516

[183] C. J. Vincent, Y. Li, and A. Blandford, "Integration of human factors and ergonomics during medical device design and development: It's all about communication," *Applied ergonomics*, vol. 45, no. 3, pp. 413–419, 2014.

[184] S. L. Star and J. R. Griesemer, "Institutional ecology,translations' and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39," *Social studies of science*, vol. 19, no. 3, pp. 387–420, 1989.

[185] A. Bruseberg, "Human views for modaf as a bridge between human factors integration and systems engineering," *Journal of Cognitive Engineering and Decision Making*, vol. 2, no. 3, pp. 220–248, 2008.

[186] A. Bodenhamer, "Adaptations in the US Army MANPRINT process to utilize HSI-inclusive system architectures," *Procedia Computer Science*, vol. 8, pp. 249–254, 2012.

[187] A. L. Ramos, J. V. Ferreira, and J. Barceló, "Lithe: an agile methodology for human-centric model-based systems engineering," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 3, pp. 504–521, 2012.

[188] D. W. Orellana and A. M. Madni, "Human system integration ontology: enhancing model based systems engineering to evaluate human-system performance," *Procedia Computer Science*, vol. 28, pp. 19–25, 2014.

[189] M. E. Watson, C. F. Rusnock, J. M. Colombi, and M. E. Miller, "Human-centered design using system modeling language," *Journal of Cognitive Engineering and Decision Making*, vol. 11, no. 3, pp. 252–269, 2017.

[190] P.-P. van Maanen, J. Lindenberg, and M. A. Neerincx, "Integrating human factors and artificial intelligence in the development of human-machine cooperation," in *Proc. of the 2005 international conference on artificial intelligence (ICAI'05)*, 2005.

[191] K. Revell, P. Langdon, M. Bradley, I. Politis, J. Brown, and N. Stanton, "User Centered Ecological Interface Design (UCEID):A Novel Method Applied to the Problem of Safe and User-Friendly Interaction Between Drivers and Autonomous Vehicles," *Intelligent Human Systems Integration, Advances in Intelligent Systems and Computing*, 2018.

[192] N. Merat and J. D. Lee, "Preface to the special section on human factors and automation in vehicles: Designing highly automated vehicles with the driver in mind," *Human factors*, vol. 54, no. 5, pp. 681–686, 2012.

[193] M. Kyriakidis, J. C. de Winter, N. Stanton, T. Bellet, B. van Arem, K. Brookhuis, M. H. Martens, K. Bengler, J. Andersson, N. Merat *et al.*, "A human factors perspective on automated driving," *Theoretical issues in ergonomics science*, vol. 20, no. 3, pp. 223–249, 2019.

[194] H. Mehrfard, H. Pirzadeh, and A. Hamou-Lhadj, "Investigating the capability of agile processes to support life-science regulations: the case of XP and FDA regulations with a focus on human factor requirements," in *Software Engineering Research, Management and Applications 2010*. Springer, 2010, pp. 241–255.

[195] P. H. Kashfi, *Integrating User eXperience Principles and Practices into Software Development Organizations: An Empirical Investigation*. Chalmers Tekniska Hogskola (Sweden), 2018.

[196] L. A. Palinkas, S. M. Horwitz, C. A. Green, J. P. Wisdom, N. Duan, and K. Hoagwood, "Purposeful sampling for qualitative data collection and analysis in mixed method implementation research," *Administration and Policy in Mental Health and Mental Health Services Research*, vol. 42, no. 5, pp. 533–544, 2015.

[197] H. Noble and J. Smith, "Qualitative data analysis: a practical example," *Evidence-Based Nursing*, vol. 17, no. 1, pp. 2–3, 2014.

[198] M. Birks, Y. Chapman, and K. Francis, "Memoing in qualitative research: Probing data and processes," *Journal of research in nursing*, vol. 13, no. 1, pp. 68–75, 2008.

[199] The Journal of the Human Factors and Ergonomics Society, 2020, online (assessed: 2020-Nov-16). [Online]. Available: https://journals.sagepub.com/description/hfs

[200] I. Jacobson, P.-W. Ng, P. E. McMahon, I. Spence, and S. Lidman, "The essence of software engineering: the SEMAT kernel," *Communications of the ACM*, vol. 55, no. 12, pp. 42–49, 2012.

[201] S. Freeman and N. Pryce, *Growing object-oriented software, guided by tests.* Pearson Education, 2009.

[202] G. Liebel, M. Tichy, E. Knauss, O. Ljungkrantz, and G. Stieglbauer, "Organisation and communication problems in automotive requirements engineering," *Requirements Engineering*, vol. 23, no. 1, pp. 145–167, 2018.

[203] A. M. Nascimento, L. F. Vismari, C. B. S. T. Molina, P. S. Cugnasca, J. B. Camargo, J. R. de Almeida, R. Inam, E. Fersman, M. V. Marquezini, and A. Y. Hata, "A systematic literature review about the impact of artificial intelligence on autonomous vehicle safety," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 4928–4946, 2019.

[204] J. Bosch, H. H. Olsson, and I. Crnkovic, "It takes three to tango: Requirement, outcome/data, and AI driven development," in *SiBW 2018, Software-intensive Business: Start-ups, Ecosystems and Platforms, Espoo, Finland, December 3, 2018.* CEUR-WS. org, 2018, pp. 177–192.

[205] P. Carayon and P. Hoonakker, "Human factors and usability for health information technology: old and new challenges," *Yearbook of medical informatics*, vol. 28, no. 01, pp. 071–077, 2019.

[206] J. Dul and W. P. Neumann, "Ergonomics contributions to company strategies," *Applied ergonomics*, vol. 40, no. 4, pp. 745–752, 2009.

[207] F. Flemisch, A. Schieben, J. Kelsch, and C. Löper, "Automation spectrum, inner/outer compatibility and other potentially useful human factors concepts for assistance and automation," *Human Factors for assistance and automation*, 2008.

[208] A. Fabijan, P. Dmitriev, H. H. Olsson, and J. Bosch, "The evolution of continuous experimentation in software product development: from data to a data-driven organization at scale," in *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE).* IEEE, 2017, pp. 770–780.

[209] M. N. Meyer, "Two cheers for corporate experimentation: The a/b illusion and the virtues of data-driven innovation," *Colo. Tech. LJ*, vol. 13, p. 273, 2015.

[210] R. Kohavi, R. Longbotham, D. Sommerfield, and R. M. Henne, "Controlled experiments on the web: survey and practical guide," *Data mining and knowledge discovery*, vol. 18, no. 1, pp. 140–181, 2009.

[211] K. Kevic, B. Murphy, L. Williams, and J. Beckmann, "Characterizing experimentation in continuous deployment: a case study on bing," in *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP).* IEEE, 2017, pp. 123–132.

[212] L. Leung, "Validity, reliability, and generalizability in qualitative research," *Family Med Prim Care*, vol. 4, pp. 324–327, 2015.

[213] R. Wieringa and M. Daneva, "Six strategies for generalizing software engineering theories," *Science of Computer Programming*, vol. 101, pp. 136–152, 2015.

[214] P. C. Verhoef, T. Broekhuizen, Y. Bart, A. Bhattacharya, J. Q. Dong, N. Fabian, and M. Haenlein, "Digital transformation: A multidisciplinary reflection and research agenda," *Journal of business research*, vol. 122, pp. 889–901, 2021.

[215] H. H. Olsson, H. Alahyari, and J. Bosch, "Climbing the "Stairway to Heaven"–A Mulitiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software," in *2012 38th Euromicro Conference on Software Engineering and Advanced Applications*. IEEE, 2012, pp. 392–399.

[216] S. G. Yaman, M. Munezero, J. Münch, F. Fagerholm, O. Syd, M. Aaltola, C. Palmu, and T. Männistö, "Introducing continuous experimentation in large software-intensive product and service organisations," *Journal of Systems and Software*, vol. 133, pp. 195–211, 2017.

[217] L. M. Maruping and S. Matook, "The evolution of software development orchestration: current state and an agenda for future research," *European Journal of Information Systems*, vol. 29, no. 5, pp. 443–457, 2020.

[218] G. A. Boy, "Human-centered design of complex systems: An experience-based approach," *Design Science*, vol. 3, p. e8, 2017.

[219] A. D. Franklin, "What makes a 'good' experiment?" *The British Journal for the Philosophy of Science*, vol. 32, no. 4, pp. 367–374, 1981.

[220] D. G. Feitelson, E. Frachtenberg, and K. L. Beck, "Development and Deployment at Facebook," *IEEE Internet Computing*, vol. 17, no. 4, pp. 8–17, 2013.

[221] K. Kevic, B. Murphy, L. Williams, and J. Beckmann, "Characterizing Experimentation in Continuous Deployment: A Case Study on Bing," in *39th International Conference on Software Engineering:(ICSE-SEIP)*. IEEE, 2017.

[222] A. P. Muhammad, E. Knauss, J. Bärgman, and A. Knauss, "Towards challenges and practices with managing human factors in automated vehicle development," in *31st IEEE International Requirements Engineering Conference, (RE'23)*. IEEE, 2023.

[223] A. P. Muhammad, E. Knauss, and J. Bärgman, "Human factors in developing automated vehicles: A requirements engineering perspective," *Journal of Systems and Software*, p. 111810, 2023.

[224] G. B. Sætren, S. Hogenboom, and K. Laumann, "A study of a technological development process: Human factors—the forgotten factors?" *Cognition, Technology & Work*, vol. 18, no. 3, pp. 595–611, 2016.

[225] S. Gandevia, "A human factor in 'good'experiments," *The British Journal for the Philosophy of Science*, vol. 37, no. 4, pp. 463–466, 1986.

[226] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods: Review and analysis," *arXiv preprint arXiv:1709.08439*, 2017.

[227] R. Kohavi, R. M. Henne, and D. Sommerfield, "Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO," in *Proceedings of the 13th ACM SIGKDD*, 2007, pp. 959–967.

[228] A. Fabijan, P. Dmitriev, H. H. Olsson, and J. Bosch, "The evolution of continuous experimentation in software product development: from data to a data-driven organization at scale," in *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*. IEEE, 2017, pp. 770–780.

[229] R. Kohavi, T. Crook, R. Longbotham, B. Frasca, R. Henne, J. L. Ferres, and T. Melamed, "Online Experimentation at Microsoft," *Data Mining Case Studies*, vol. 11, no. 2009, p. 39, 2009.

[230] E. Lindgren and J. Münch, "Software development as an experiment system: A qualitative survey on the state of the practice," in *Agile Processes in Software Engineering and Extreme Programming: 16th International Conference, XP*. Springer, 2015.

[231] O. Rissanen and J. Münch, "Continuous experimentation in the B2B domain: a case study," in *2015 IEEE/ACM 2nd International Workshop on Rapid Continuous Software Engineering*. IEEE, 2015, pp. 12–18.

[232] B. Shneiderman, "Human Factors Experiments in Designing Interactive Systems," *Computer*, vol. 12, no. 12, pp. 9–19, 1979.

[233] P. Hancock and J. K. Caird, "Experimental evaluation of a model of mental workload," *Human factors*, vol. 35, no. 3, pp. 413–429, 1993.

[234] M. Royer, K. Houser, D. Durmus, T. Esposito, and M. Wei, "Recommended methods for conducting human factors experiments on the subjective evaluation of colour rendition," *Lighting Research & Technology*, vol. 54, no. 3, pp. 199–236, 2022.

[235] K. W. Williams, "Impact of aviation highway-in-the-sky displays on pilot situation awareness," *Human Factors*, vol. 44, no. 1, pp. 18–27, 2002.

[236] Y. Nakao, M. Moriguchi, and H. Noda, "Using agile software development methods to support human-centered design," *NEC Technical Journal*, vol. 8, no. 3, pp. 37–40, 2014.

[237] S. G. Yaman, "Initiating the transition towards continuous experimentation: empirical studies with software development teams and practitioners," *Series of Publications A*, 2019.

[238] J. A. Maxwell, *Qualitative research design: An interactive approach.* Sage publications, 2012.

[239] V. Clarke, V. Braun, and N. Hayfield, "Thematic analysis," *Qualitative psychology: A practical guide to research methods*, vol. 3, pp. 222–248, 2015.

[240] A. M. Madni, "Integrating humans with and within complex systems," *CrossTalk*, vol. 5, 2011.

[241] S. G. Charlton and T. G. O'Brien, *Handbook of human factors testing and evaluation.* CRC Press, 2019.

[242] L. Lagerberg, T. Skude, P. Emanuelsson, K. Sandahl, and D. Ståhl, "The Impact of Agile Principles and Practices on Large-Scale Software Development Projects: A Multiple-Case Study of Two Projects at Ericsson," in *International Symposium on Empirical Software Engineering and Measurement*, 2013, pp. 348–356.

[243] M. Jorgensen, "Relationships between project size, agile practices, and successful software development: results and analysis," *IEEE Software*, vol. 36, no. 2, pp. 39–43, 2019.

[244] K. Elghariani and N. Kama, "Review on agile requirements engineering challenges," in *Int. Conf. on Computer and Information Sciences*, 2016, pp. 507–512.

[245] V. Vaishnavi and W. Kuechler, *Design Science Research Methods and Patterns: Innovating Information and Communication Technology.* Taylor & Francis, 2007.

[246] O. Batsaikhan and Y.-C. Lin, "Building a shared understanding of customer value in a large-scale agile organization: A case study," Master's thesis, Dept. of Computer Science and Engineering, Chalmers | University of Gothenburg, 2018, https://hdl.handle.net/20.500.12380/304465.

[247] J. Saldaña, *The Coding Manual for Qualitative Researchers*, 3rd ed. Sage, 2015.

[248] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical software engineering*, vol. 14, no. 2, pp. 131–164, 2009.

[249] J. Maxwell, "Understanding and validity in qualitative research," *Harvard educational review*, vol. 62, no. 3, pp. 279–301, 1992.

[250] B. Henderson-Sellers and J. Ralyté, "Situational method engineering: state-of-the-art review," *Journal of Universal Computer Science*, 2010.

[251] K. R. Ohnemus, "Incorporating human factors in the system development life cycle: marketing and management approaches," in *IPCC 96: Communication on the Fast Track. IPCC 96 Proceedings.* IEEE, 1996, pp. 46–53.

[252] D. Norman, *Things that make us smart: Defending human attributes in the age of the machine.* Diversion Books, 2014.

[253] A. P. Muhammad, E. Knauss, J. Bärgman, and A. Knauss, "Managing human factors in automated vehicle development: Towards challenges and practices," in *2023 IEEE 31st International Requirements Engineering Conference (RE).* IEEE, 2023, pp. 347–352.

[254] S. V. Zykov and A. Singh, *Agile Enterprise Engineering: Smart Application of Human Factors.* Springer, 2020, vol. 175.

[255] F. Walker, J. Steinke, M. Martens, and W. Verwey, "Do engineer perceptions about automated vehicles match user trust? consequences for design," *Transportation research interdisciplinary perspectives*, vol. 8, p. 100251, 2020.

[256] M. R. Siddiqi, S. Milani, R. N. Jazar, and H. Marzbani, "Ergonomic path planning for autonomous vehicles-an investigation on the effect of transition curves on motion sickness," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 7, pp. 7258–7269, 2021.

[257] S. Kolekar, B. Petermeijer, E. Boer, J. de Winter, and D. Abbink, "A risk field-based metric correlates with driver's perceived risk in manual and automated driving: A test-track study," *Transportation research part C: emerging technologies*, vol. 133, p. 103428, 2021.

[258] I. J. Reagan, D. G. Kidd, and J. B. Cicchino, "Driver acceptance of adaptive cruise control and active lane keeping in five production vehicles," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 61, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2017, pp. 1949–1953.

[259] A. S. Tomar, M. S. Gull, S. R. Penmetsa, and F. Tillema, "Towards a human-centric design solution for automated systems to enhance driver's comfort and acceptance," in *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME).* IEEE, 2021, pp. 1–6.

[260] S. Jokhio, P. Olleja, J. Bärgman, F. Yan, and M. Baumann, "Analysis of time-to-lane-change-initiation using realistic driving data," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[261] U. Sander, "Opportunities and limitations for intersection collision intervention—a study of real world 'left turn across path'accidents," *Accident Analysis & Prevention*, vol. 99, pp. 342–355, 2017.

[262] J. Bärgman, K. Smith, and J. Werneke, "Quantifying drivers' comfort-zone and dread-zone boundaries in left turn across path/opposite direction (LTAP/OD) scenarios," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 35, pp. 170–184, 2015.

[263] C. A. Hobbs and P. J. McDonough, "Development of the european new car assessment programme (euro ncap)," *Regulation*, vol. 44, no. 3, pp. 2439–2453, 1998.

[264] R. Cirino Goncalves, C. M. Goodridge, J. Kuo, M. G. Lenne, and N. Merat, "Using driver monitoring to estimate readiness in automation: A conceptual model based on simulator experimental data," *Available at SSRN 4640135*.

[265] M. Kyriakidis, J. C. de Winter, N. Stanton, T. Bellet, B. van Arem, K. Brookhuis, M. H. Martens, K. Bengler, J. Andersson, N. Merat *et al.*, "A human factors perspective on automated driving," *Theoretical issues in ergonomics science*, vol. 20, no. 3, pp. 223–249, 2019.

[266] A. S. Mueller, I. J. Reagan, and J. B. Cicchino, "Addressing driver disengagement and proper system use: human factors recommendations for level 2 driving automation design," *Journal of cognitive engineering and decision making*, vol. 15, no. 1, pp. 3–27, 2021.

[267] F. Ekman, M. Johansson, and J. Sochor, "Creating appropriate trust in automated vehicle systems: A framework for hmi design," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 1, pp. 95–101, 2017.

[268] M. R. Endsley, "Designing for situation awareness in complex systems," in *Proceedings of the Second International Workshop on symbiosis of humans, artifacts and environment*, 2001, pp. 1–14.

[269] K. E. Wiegers and J. Beatty, *Software requirements.* Pearson Education, 2013.

[270] M. Cagan, *Inspired: How to create products customers love.* SVPG Press Sunnyvale, CA, USA, 2008.

[271] A. Cockburn, *Agile software development: the cooperative game.* Pearson Education, 2006.

[272] M. Paasivaara and C. Lassenius, "Collaboration practices in global inter-organizational software development projects," *Software Process: Improvement and Practice*, vol. 8, no. 4, pp. 183–199, 2003.

[273] K. S. Rubin, *Essential Scrum: A practical guide to the most popular Agile process.* Addison-Wesley, 2012.

[274] A. Aurum and C. Wohlin, "The fundamental nature of requirements engineering activities as a decision-making process," *Information and Software Technology*, vol. 45, no. 14, pp. 945–954, 2003.

[275] A. M. Davis, *Software requirements: objects, functions, and states.* Prentice-Hall, Inc., 1993.

[276] O. C. Gotel and C. Finkelstein, "An analysis of the Requirements Traceability Problem," in *Proceedings of IEEE international conference on requirements engineering.* IEEE, 1994, pp. 94–101.

[277] C. Rupp, M. Simon, and F. Hocker, "Requirements engineering und management," *HMD Praxis der Wirtschaftsinformatik*, vol. 46, no. 3, pp. 94–103, 2009.

[278] G. Van Der Vyver, A. Koronios, and M. S. Lane, "Agile methodologies and the emergence of the agile organisation: a software development approach waiting for its time?" in *Proceedings of the 7th Pacific Asia Conference on Information Systems (PACIS 2003).* University of South Australia, 2003, pp. 1344–1358.

[279] K. Schwaber and M. Beedle, *Agile software development with Scrum.* Prentice Hall PTR, 2001.

[280] M. Cohn and D. Ford, "Introducing an agile process to an organization [software development]," *Computer*, vol. 36, no. 6, pp. 74–78, 2003.

[281] J. A. Highsmith, *Agile software development ecosystems.* Addison-Wesley Professional, 2002.

[282] P. Guide, *A guide to the project management body of knowledge*, 2008.

[283] J. Sutherland and K. Schwaber, "The scrum guide," *The definitive guide to scrum: The rules of the game. Scrum. org*, vol. 268, p. 19, 2013.

[284] D. Leffingwell, *Agile software requirements: lean requirements practices for teams, programs, and the enterprise.* Addison-Wesley Professional, 2010.

[285] J. Saldaña, *The Coding Manual for Qualitative Researchers.* Sage, 2021.

[286] S. Kılıç, "Kappa testi," *Journal of mood disorders*, vol. 5, no. 3, pp. 142–144, 2015.

[287] A. Cooper, R. Reimann, D. Cronin, and C. Noessel, *About face: the essentials of interaction design.* John Wiley & Sons, 2014.

[288] A. Davis, *Just enough requirements management: where software development meets marketing.* Addison-Wesley, 2013.

[289] R. Hoda, N. Salleh, and J. Grundy, "The Rise and Evolution of Agile Software Development," *IEEE Software*, vol. 35, no. 5, pp. 58–63, 2018.

[290] J. Sutherland and K. Schwaber, "The scrum papers," *Nuts, bolts and origins of an Agile process*, 2007.

[291] A. P. Muhammad, E. Knauss, and J. Bärgman, "Human factors in developing automated vehicles: A requirements engineering perspective," *Journal of Systems and Software*, p. 111810, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121223002054

[292] M. Kuhrmann, P. Diebold, J. Munch, P. Tell, K. Trektere, F. McCaffery, V. Garousi, M. Felderer, O. Linssen, E. Hanser *et al.*, "Hybrid software development approaches in practice: a European perspective," *IEEE Software*, vol. 36, no. 4, pp. 20–31, 2018.

[293] S. Buhne, G. Halmans, K. Pohl, M. Weber, H. Kleinwechter, and T. Wierczoch, "Defining requirements at different levels of abstraction," in *Proceedings. 12th IEEE International Requirements Engineering Conference, 2004.* IEEE, 2004, pp. 346–347.

[294] T. Gorschek, P. Garre, S. B. Larsson, and C. Wohlin, "Industry evaluation of the requirements abstraction model," *Requirements engineering*, vol. 12, pp. 163–190, 2007.

[295] P. Brauner and M. Ziefle, *Human factors in production systems.* Springer International Publishing Cham, Germany, 2015, vol. 30.

[296] J. Choi, K. Kahyun Lee, and J. Choi, "Determinants of User Satisfaction with Mobile VR Headsets: The Human Factors Approach by the User Reviews Analysis and Product Lab Testing," *International Journal of Contents*, vol. 15, no. 1, 2019.

[297] J. Nielsen, *Usability engineering.* Morgan Kaufmann, 1994.

[298] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research," in *Advances in psychology.* Elsevier, 1988, vol. 52, pp. 139–183.

[299] D. Hidellaarachchi, J. Grundy, R. Hoda, and I. Mueller, "The influence of human aspects on requirements engineering-related activities: Software practitioners' perspective," *ACM Transactions on Software Engineering and Methodology*, 2022.

[300] M. Sun, Z. Chen, G. Bakirtzis, H. Jafarzadeh, and C. Fleming, "A formal process of hierarchical functional requirements development for set-based design," *arXiv preprint arXiv:2210.14434*, 2022.

[301] SystemWeaver, "Requirements in SystemWeaver," 2023, accessed on 17 April 2023, available at: https://support.systemweaver.se/en/support/solutions/articles/31000142945-requirements-meta-model#Requirements-in-SystemWeaver.

[302] International Organization for Standardization (ISO), "ISO 26262-1:2018 – Road Vehicles: Functional Safety," International Standard, 2018, available at: https://www.iso.org/standard/68383.html.

[303] Automotive SPICE, "Process Reference Model and Process Assessment Model, Version 4.0," Standard document, 2023, available at: https://www.automotivespice.com/.

[304] R. Parasuraman and V. Riley, "Humans and automation: Use, misuse, disuse, abuse," *Human factors*, vol. 39, no. 2, pp. 230–253, 1997.

[305] C. I. Schlenoff, Z. Kootbally, P. Rachakonda, S. Lightman, A. Vassilev, D. A. Wollman, and E. Griffor, "Standards and performance metrics for on-road automated vehicles," 2024.

[306] K. Ahmad, M. Abdelrazek, C. Arora, A. A. Baniya, M. Bano, and J. Grundy, "Requirements engineering framework for human-centered artificial intelligence software systems," *Applied Soft Computing*, vol. 143, p. 110455, 2023.

[307] J. Dul, R. Bruder, P. Buckle, P. Carayon, P. Falzon, W. S. Marras, J. R. Wilson, and B. van der Doelen, "A strategy for human factors/ergonomics: developing the discipline and profession," *Ergonomics*, vol. 55, no. 4, pp. 377–395, 2012.

[308] OCIMF, 2021, online (assessed: 2023-Aug-16). [Online]. Available: https://www.ocimf.org/publications/information-papers/human-factors-management-and-self-assessment

[309] "SHAPE-IT: Supporting the Interaction of Humans and Automated Vehicles: Preparing for the Environment of Tomorrow," Online resource, available at: https://www.shape-it.eu, 2023, accessed on 15 November 2023.

[310] R. McElreath, *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC, 2018.

[311] R. N. Pikaar, "Case studies underrated–or the value of project cases," in *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018) Volume V: Human Simulation and Virtual Environments, Work With Computing Systems (WWCS), Process Control 20*. Springer, 2019, pp. 711–720.

[312] D. Norman, "User centered system design," *New perspectives on human-computer interaction*, 1986.

[313] C. D. Wickens, "Multiple resources and performance prediction," *Theoretical issues in ergonomics science*, vol. 3, no. 2, pp. 159–177, 2002.

[314] A. Dix, *Human-computer interaction*. Pearson Education, 2003.

[315] K. Rhaiem and N. Amara, "Learning from innovation failures: a systematic review of the literature and research agenda," *Review of Managerial Science*, vol. 15, 2021.

[316] M. O'Daniel and A. H. Rosenstein, "Professional Communication and Team Collaboration," *Patient Safety and Quality: An Evidence-Based Handbook for Nurses*, 2008.