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


**Abstract** v

**Acknowledgement** vii

**List of Publications** xi

**Personal Contribution** xv

**Ethical Considerations** xvii

**1 Introduction** 1

1.1 Research Goal and Questions . . . 2
1.1.1 Research Goal . . . 2
1.1.2 Research Questions . . . 4
1.2 Background . . . 6
1.2.1 Requirements Engineering . . . 6
1.2.2 Agile Development . . . 7
1.2.3 Large-scale Agile Development . . . 8
1.2.4 Requirements Engineering in Agile . . . 9
1.2.5 Automated Vehicles . . . 10
1.2.6 Human Factors . . . 10
1.2.7 Human Factors and Automated Vehicles . . . 12
1.3 Research Approach . . . 13
1.3.1 Research Focus . . . 14
1.3.2 Research Method . . . 15
1.3.2.1 Qualitative Research . . . 15
1.3.2.2 Design Science Research . . . 16
1.3.2.3 Mixed-methods Research . . . 16
1.3.3 Data Collection Methods . . . 17
1.3.3.1 Interviews . . . 17
1.3.3.2 Questionnaire Survey . . . 18
1.3.3.3 Document Analysis . . . 18
1.3.3.4 Workshops . . . 18
1.3.3.5 Observations . . . 19
1.3.3.6 Shadowing Technique . . . 19
1.3.4 Ethical Considerations . . . 19
1.4 Threats to Validity . . . 20
1.4.1 Construct Validity . . . 20

1.4.2 Internal Validity . . . . . . . . . . . . . . . . . . . . 21
1.4.3 External Validity . . . . . . . . . . . . . . . . . . . . 21
1.4.4 Reliability . . . . . . . . . . . . . . . . . . . . . . . 22
1.5 Summaries of Studies . . . . . . . . . . . . . . . . . . . . . 23
1.5.1 Paper A . . . . . . . . . . . . . . . . . . . . . . . . 24
1.5.2 Paper B . . . . . . . . . . . . . . . . . . . . . . . . 26
1.5.3 Paper C . . . . . . . . . . . . . . . . . . . . . . . . 27
1.5.4 Paper D . . . . . . . . . . . . . . . . . . . . . . . . 29
1.5.5 Paper E . . . . . . . . . . . . . . . . . . . . . . . . 31
1.6 Answering the Thesis' Research Questions . . . . . . . . . . . 32
1.6.1 G1: Domain Exploration . . . . . . . . . . . . . . . . 32
1.6.2 G2: Solution Investigation . . . . . . . . . . . . . . . 37
1.7 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . 42
1.7.1 Implications . . . . . . . . . . . . . . . . . . . . . . 46
1.7.2 Future Work . . . . . . . . . . . . . . . . . . . . . . 47
1.8 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . 48

**2 Paper A 51**
2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . 52
2.2 Background and Related work . . . . . . . . . . . . . . . . . 54
2.2.1 Human Factors in Automated Vehicle Development . . 54
2.2.1.1 Human Factors and its Role in AV Development 55
2.2.1.2 What HF Issues Impact AV Development? . . 55
2.2.2 AV Development: Processes, Approaches, Recent Developments . . . . . . . . . . . . . . . . . . . . . . . 56
2.2.2.1 Requirements Engineering . . . . . . . . . . . 56
2.2.2.2 Development Practices . . . . . . . . . . . . . 57
2.2.2.3 Development Approaches at Scale . . . . . . . 58
2.2.3 Related work: Communicating Human Factors Knowledge and Requirements to AV Engineers . . . . . . . . . 58
2.3 Research Method . . . . . . . . . . . . . . . . . . . . . . . 59
2.3.1 Data Collection . . . . . . . . . . . . . . . . . . . . . 60
2.3.2 Data Analysis . . . . . . . . . . . . . . . . . . . . . . 62
2.4 Findings . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
2.4.1 Human Factors in Relation to AV Development (RQ1) . 63
2.4.2 Properties of Human Factors and Agile (RQ2) . . . . . 66
2.4.2.1 Properties of Agile. . . . . . . . . . . . . . . 66
2.4.2.2 Properties of Human Factors. . . . . . . . . . 70
2.4.3 Implications (RQ3) . . . . . . . . . . . . . . . . . . . 73
2.4.3.1 Implications for Agile . . . . . . . . . . . . . 73
2.4.3.2 Implications for HF . . . . . . . . . . . . . . 78
2.4.3.3 Implications for Requirements Engineering . . 80
2.5 Validation Study . . . . . . . . . . . . . . . . . . . . . . . 83
2.5.1 Participants' Demographics . . . . . . . . . . . . . . . 84
2.5.2 Evaluation of Properties of Agile and HF . . . . . . . . 84
2.5.3 Evaluation of Implications . . . . . . . . . . . . . . . . 86
2.6 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . 88
2.6.1 Implications for Practice . . . . . . . . . . . . . . . . 88
2.6.2 Implications for Research . . . . . . . . . . . . . . . . 89

2.6.3 Discussion of quality . . . 90
2.6.3.1 Credibility . . . 90
2.6.3.2 Resonance . . . 91
2.6.3.3 Usefulness . . . 91
2.6.3.4 Transferability . . . 91
2.7 Conclusion . . . 92

**3 Paper B 93**
3.1 Introduction . . . 94
3.2 Background and Related Work . . . 95
3.3 Methodology . . . 97
3.4 Findings . . . 98
3.5 Discussion . . . 105
3.6 Conclusion . . . 107

**4 Paper C 109**
4.1 Introduction . . . 110
4.2 Related Work . . . 111
4.3 Design Science Research Method . . . 111
4.3.1 Design Science Research . . . 112
4.3.2 Data Collection . . . 113
4.3.3 Data Analysis . . . 114
4.3.4 Threats to Validity . . . 115
4.4 Findings . . . 115
4.4.1 RQ1: Which challenges arise from an undefined requirements strategy? . . . 115
4.4.2 RQ2: How do companies aim to address these challenges? 118
4.4.3 RQ3: Which potential building blocks should be considered for defining a requirements strategy? . . . 120
4.5 Artifact: Guidelines for Defining a Requirements Strategy . . . 121
4.6 Discussion and Conclusion . . . 122

**5 Paper D 125**
5.1 Introduction . . . 126
5.2 Background . . . 127
5.2.1 Human Factors in Automotive Industry . . . 127
5.2.2 Requirements Strategy and Proposition Formulation . . 130
5.3 Methodology . . . 132
5.3.1 Data Collection . . . 132
5.3.2 Data Analysis . . . 134
5.3.3 Research Validity . . . 135
5.4 Findings . . . 136
5.4.1 RQ 1: How do ownership and responsibility for HF requirements impact the integration of HF requirements in product development? . . . 136
5.4.2 RQ 2: How does requirements structure and requirements information model impact the integration of HF requirements in product development? . . . 138

5.4.3 RQ 3: How does defining a work and feature flow related to HF requirements influence their integration in product development? . . . 142
5.5 Discussion . . . 145
5.6 Conclusion . . . 147

**6 Paper E 149**
6.1 Introduction . . . 150
6.2 Background and Related Work . . . 151
6.2.1 Background . . . 152
6.2.2 Related Work . . . 154
6.3 Research Methodology . . . 155
6.3.1 Data Collection . . . 155
6.3.1.1 Stage 1: Preparatory Study for RQ1 . . . 155
6.3.1.2 Stage 2: Data collection for RQ1 & RQ2 . . . 156
6.3.1.3 Stage 3: Data Collection for RQ2 . . . 157
6.3.2 Data Analysis . . . 158
6.4 Findings RQ1: To what extent can human factors requirements be confined to specific levels of abstraction of requirements? . 159
6.4.1 Relating Human Factors to Requirements Levels . . . 159
6.5 Findings RQ2: Where should human factors experts be positioned within an organization? . . . 160
6.5.1 Potential Strategies for Placing Human Factors Experts within an Organization . . . 161
6.5.2 Survey Results on Optimal Placement of Human Factors Experts . . . 163
6.5.2.1 Descriptive Statistics . . . 163
6.5.3 Statistical Analysis . . . 167
6.5.4 Open ended-Questions . . . 172
6.5.5 Additional Results: Raising Awareness and Cultivate a Culture of Human Factors . . . 174
6.6 Discussion . . . 176
6.6.1 Threats to Validity . . . 180
6.7 Conclusion . . . 181

**Bibliography 185**

# 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 Experimentat ion | 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

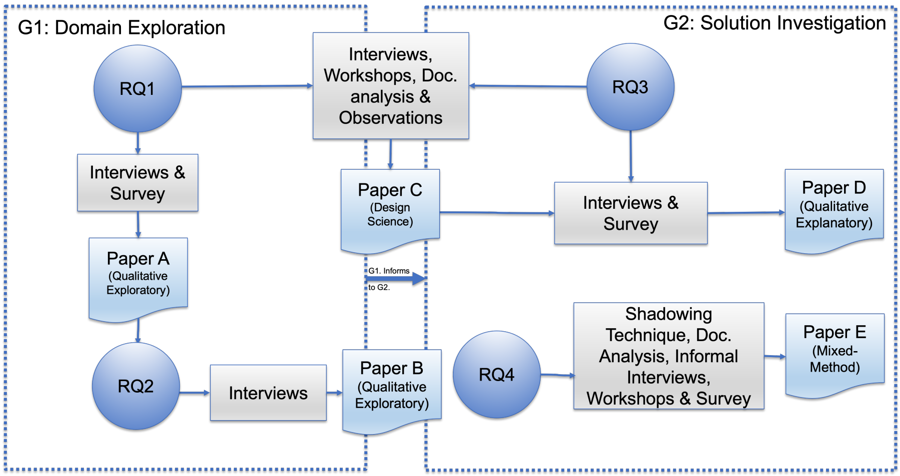


Figure 1.2: Overview of the relationships between the research goals (G1 and G2), research questions (RQ1–RQ4), and the corresponding papers (Papers A–E).

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

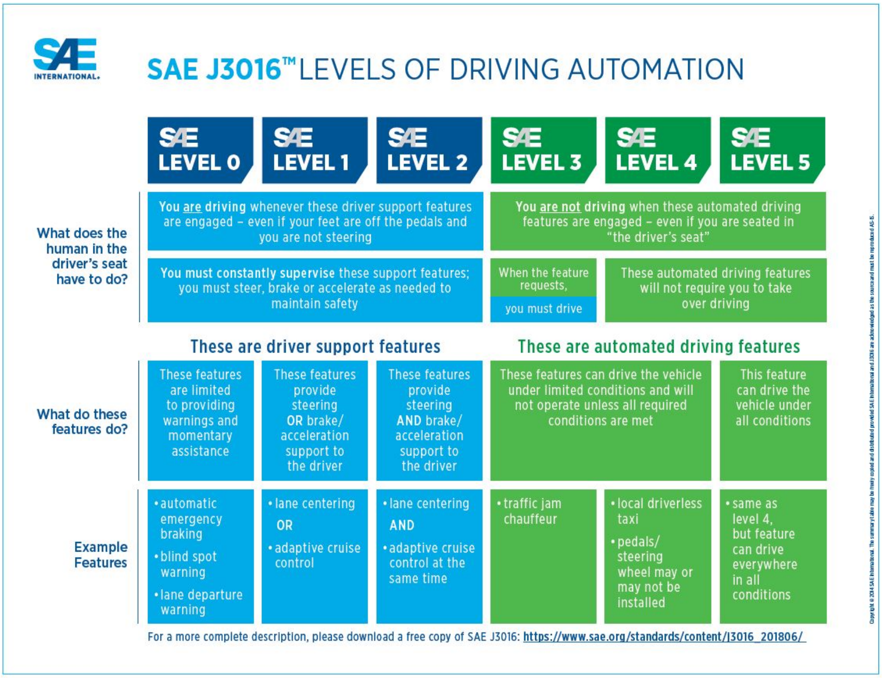


Figure 1.3: Description of levels of driving automation by SAE [53]

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.

| RE | Shared Understanding: *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]<br>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]<br>m) Req. open for comments[3] | n) No time for stakeholder involvement[2] |
| Documentation | o) Customer value description lost between systems[1]<br>p) Lack of knowledge about writing requirements[1,2,3]<br>q) No dedicated time for requirements[1,2,3] | r) Too much/not enough document.[1,2]<br>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]

| | **Support for shared understanding** | | |
|---|---|---|---|
| **Perspective** | **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.