Thesis for The Degree of Licentiate of Engineering

Managing Human Factors and Requirements in Agile Development of Automated Vehicles: An Exploration

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, 2022

Managing Human Factors and Requirements in Agile Development of Automated Vehicles: An Exploration

Amna Pir Muhammad

Technical Report No 5234
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 2022.

Abstract

Context: Automated Vehicle (AV) technology has evolved significantly in complexity and impact; it is expected to ultimately change urban transportation. However, research shows that vehicle automation can only live up to this expectation if it is defined with human capabilities and limitations in mind. Therefore, it is necessary to bring human factors knowledge to AV developers.

Objective: This thesis aims to empirically study how we can effectively bring the required human factors knowledge into large-scale agile AV development. The research goals are 1) to explore requirements engineering and human factors in agile AV development, 2) to investigate the problems of requirements engineering, human factors, and agile way of working in AV development, and 3) to demonstrate initial solutions to existing problems in agile AV development.

Method: We conducted this research in close collaboration with industry, using different empirical methodologies to collect data—including interviews, workshops, and document analysis. To gain in-depth insights, we did a qualitative exploratory study to investigate the problem and used a design science approach to develop initial solution in several iterations.

Findings and Conclusions: We found that applying human factors knowledge effectively is one of the key problem areas that need to be solved in agile development of artificial intelligence (AI)-intense systems. This motivated us to do an in-depth interview study on how to manage human factors knowledge during AV development. From our data, we derived a working definition of human factors for AV development, discovered the relevant properties of agile and human factors, and defined implications for agile ways of working, managing human factors knowledge, and managing requirements. The design science approach allowed us to identify challenges related to agile requirements engineering in three case companies in iterations. Based on these three case studies, we developed a solution strategy to resolve the RE challenges in agile AV development. Moreover, we derived building blocks and described guidelines for the creation of a requirements strategy, which should describe how requirements are structured, how work is organized, and how RE is integrated into the agile work and feature flow.

Future Outlook: In future work, I plan to define a concrete requirement strategy for human factors knowledge in large-scale agile AV development. It could help establishing clear communication channels and practices for incorporating explicit human factors knowledge into AI-based large-scale agile AV development.

Keywords

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

Acknowledgment

I am indebted to my supervisors. In particular, to Eric Knauss for his continuous guidance and encouragement. His support and patience helped me throughout the research. He was always there to share his technical knowledge and experience. To Jonas Bärgman and Alessia Knauss for their insightful advice and a lot of valuable feedback that has pushed me to sharpen my skills. Their collective encouragement and belief in me were a constant source of motivation.

I am thankful to our industry partner Zenseact and colleagues there for sharing their resources and experience on this topic. I am especially grateful to Jonas Ekmark and Christoffer Bäckvall for inspiring conversations and being positive, understanding, and helpful.

I want to thank my Interaction Design and Software Engineering Division friends, especially my office colleagues Hans-Martin Heyn and Khan Habib. They were incredibly supportive and reminded me that I was not alone in this journey.

My deepest gratitude is towards my beloved husband, Babar Farooq. Thank you for your sacrificial love and especially your patience and support, which always motivates me to stay on track. Even when the distance between us seemed unbearable, you gave me the strength to keep going and keep my path. Thanks also to our little girl Hafsa Farooq, who makes me happy every day with her big smile and kisses.

I am also thankful to my parents, who, despite being far away, supported me with calming words and reassurance when I felt out of place and nervous about my research. Last but not least, I am thankful to all my friends, especially Madiha Fatima and Wardah Mehmood, for their support whenever and wherever.

Finally, I would like to thank SHAPE-IT for funding my work and providing great opportunities. This project has 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] 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.

[B] Amna Pir Muhammad, Eric Knauss, Jonas Bargman "Human Factors in Developing Automated Vehicles: A Requirements Engineering Perspective"
In submission to Journal of Systems and Softwar, 2022. we have addressed peer review questions and it is under peer review again.

[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.

Other publications

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

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

Research Contribution

The included papers are published with the help of colleagues. I specifically made the following contributions to the papers included in this thesis. I joined the work on Paper A during the early planning stages of the problem exploration. I mainly contributed to describing the human factors part. I was responsible for data collection and analysis of Paper B. Moreover, I was also responsible for most of the writing and final publication.

Paper C is based on three case studies; my co-authors collected the initial data for two of the cases. I contributed to designing the overall study and took a leading role in data collection and analysis for the third case (incl. a workshop, document analysis, and observations). I also took the lead in executing and writing the final paper, including synthesizing the three studies.

Contents

Chapter 1

Introduction

By moving towards eliminating human-driver caused crashes, automated Vehicles (AVs) promise a number of benefits: fewer accidents, injuries, and deaths, as well as enabling drivers to engage in other activities while in the car [1, 2, 3]. Due to this promise, the automotive industry is currently competing to develop and market AVs with increasing levels of automation[1], ranging from specific automated functions in Advanced Driver Assistance Systems (ADAS) that can support the driver in the driving task, to fully autonomous vehicles that take over all driving tasks—at least under specific conditions (Operational Design Domains; ODDs; [5])—that do not require supervision. AVs are always software-intense and often rely on artificial intelligence (AI). Consequently, AVs are complex systems that require careful consideration in their design.

On the other hand, in spite of all their benefits, AVs also pose various challenges to humans, such as over-trust and over-reliance, extra workload on humans, or driver engagement and re-engagement [6, 7]. The challenges are not limited to drivers; other road users who interact with AVs will also be affected. Actually, unsafe and non-human-centered interactions between AVs and drivers and other road users can substantially reduce the benefits of AVs, affecting humans both inside and outside the vehicle.

To overcome the issues of managing human factors in vehicle automation (see, e.g., [8] for examples of automation failures) and achieve the full potential of automation, human factors researchers strongly advocate to consider knowledge about human factors when designing AVs [9, 10, 11]. This field, the study of human factors, involves researching human capabilities and limitations and other human characteristics and applying the findings to the design of systems to improve performance, safety and comfort [12].

In order to include human factors when designing automation, researchers recommend incorporating human factors knowledge into the early stages of development [13, 14, 15]. Traditionally such knowledge has been included in requirements of the system that have been specified up-front and form the foundation of subsequent design work [16]. The process of eliciting, analyzing, documenting, and validating the requirements during the engineering process is

[1] Definitions of levels of automation have been proposed by, for example SAE [4], but the interpretation of levels automation are still being discussed, and as this work is relevant for all levels.

called Requirements Engineering (RE) [17]. The recent trend in the automotive industry to adopt agile development methodologies however changes the role of RE significantly, however. Agile development methodologies are a collection of approaches based on incremental and iterative development in which self-organizing and cross-functional teams work together to generate requirements and functions [18]. Agile methods aim to deliver faster in less time to market, since, due to competition, developers want to deliver faster; they focus on technical details and often neglect others, such as those provided by human factors. Moreover, because agile methodologies do not focus on the processes, RE processes are not well integrated with agile methodologies and face different challenges [19].

Without a clear role for RE in agile development, there is a risk that introducing human factors knowledge as requirements will be difficult. This risk is increased by the lack of empirical research on how to include human factors knowledge in agile development; practitioners struggle with a lack of clear guidelines.

1.1 Research Goal and Questions

The overall goal of this PhD study is to investigate how we can effectively bring requirements based on human factors knowledge to AV developers in large-scale agile AV development. To reach this goal, we need to develop guidelines and methodologies. In this thesis, we embarked on a domain exploration and in-depth investigation of the problem to accomplish our ultimate goal. We then provided first solutions to the underlying problems to be used as a basis for defining the final solution of the overall PhD project. In order to realize the ultimate goal of our research, this thesis addresses the following goals: In order to realize the ultimate goal of our research, this thesis addresses the following goals:

Goal 1 (G1): Domain Exploration. To explore requirements engineering and human factors in agile AV development.

Goal 2 (G2): Problem space Investigation. To investigate the problems of requirements engineering, human factors, and agile way of working in AV development.

Goal 3 (G3): Solution space Investigation. To demonstrate initial solutions to existing problems in agile AV development.

Since this is multidisciplinary work, we need to carefully explore different aspects of the problem. From multiple perspectives, **G1** aims to establish a picture of the different problem areas and further explore these areas, namely RE and human factors in agile AV development, at different levels of detail. As our ultimate goal is to investigate how we can effectively bring human factors knowledge as a requirement to AV developers in large-scale Agile AV development from a requirements engineering perspective, **G2** investigates these problem areas more in depth. G2 discovers the relevant properties of agile and human factors, explaining the implications of agile ways of working, human factors, and RE. It also investigates how developers manifest RE challenges in agile development. The third goal, **G3**, aims to provide an initial solution by developing an approach to better integrate RE into agile development. In this Licentiate thesis we provided initial solution in the context of RE. In future work, we aim to build on these results to provide solutions relevant for human factors and AI.

Figure 1.1 illustrates how the research goals, the thesis research questions, and the included papers are related. It shows that each paper contributes to research goal G1: domain exploration. Paper A describes problem areas at a high level. Papers B and C are built on its findings and investigate the problem in detail, contributing to achieving goal G2: Problem space investigation. With Paper C, we make the first attempt into the solution space, which helps us to achieve goal G3: Solution space investigation. Figure 1.1 also shows that the depth of information is increasing with each paper.

Thesis Goals	Paper A – Problem areas of AI intense systems	Paper B – HF in agile	Paper C – RE in agile	Related RQ
G1: Domain Exploration	Problem areas of AI-Intense systems	HF definition	RE Challenges	RQ1.1, RQ1.2, RQ1.3
G2: Problem Investigation		Properties of HF & Agile Implications for HF, Agile and RE	How do challenges manifest at companies	RQ2.1, RQ2.2, RQ2.3
G3: Solution			Solution Strategy for RE challenge & Requirements Strategy	RQ3.1

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

To reach our goals, we formulated the following research questions:

G1: Domain Exploration:

To accomplish our goal, we find it essential to gain a good understanding of practitioners' problems and needs.

RQ1.1: From a requirements perspective, what are the critical research areas when building AI-intense systems?

RQ1.1 is related to the general exploration of problem areas related to requirements engineering in developing AI-intense systems.

RQ1.2: How do human factors experts and AV engineers characterize human factors in relation to agile AV development?

In order to explore the systematic capturing and management of human factors knowledge, it is important to share a common understanding of basic concepts. A wide range of definitions is available in different contexts. Therefore, RQ1.2 aims to define the term “Human Factors” in the context of AV development, for example based on agile methods.

RQ1.3: What are the critical challenges with requirements engineering in agile development?

Since we are exploring this topic in the context of requirements engineering and specifically targeting agile development, RQ1.3 aims to investigate the challenges of requirements engineering in agile development.

G2: Problem space Investigation:

With this goal we intend to lay the foundation for including the human factors knowledge in agile AV development.

RQ2.1: Which properties of human factors and agile ways of working impact AV development?

We first explore the properties of the agile way of working and human factors in relation to AV development. These properties raise important questions about the interplay of both disciplines.

RQ2.2: What are important implications when aiming to better integrate human factors into AV development?

RQ2.2 aims to enable future studies on how to effectively capture human factors' knowledge as requirements and ultimately bridge the knowledge gap between human factors and AV engineers.

RQ2.3: How do companies aim to address RE challenges?We need to define some solution strategies to address the challenges identified in RQ1.3. From our defined solution strategies we learn that in large-scale agile systems engineering, RE challenges can usually be categorized as aspects of organization, structure, or process.

G3: Solution space Investigation:

The solution strategies in RQ2.3 provide some common solution patterns for use in building a requirements strategy.

RQ3.1: Which potential building blocks should be considered for defining a requirements strategy?

Based on the common solution patterns, we aim to define a general *requirements strategy* which can be used as a template to define more concrete solutions to requirements challenges in agile development. The requirements strategy provides guidelines and building blocks to mitigate the identified challenges. Our ultimate goal is to work towards incorporating human factor knowledge in the form of requirements in agile AV development. Our ultimate goal is to work towards incorporating human factor knowledge in form of requirements in agile AV development.

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 our research methodology, while Section 1.4 explores the threats to validity. The summaries of the included papers are described in Section 1.5. In Section 1.6, we discuss our findings. Finally, we conclude the introduction chapter in Section 1.7 and Section 1.8 presents the future work.

1.2 Background

This section provides background information on the basic concepts used in this thesis. We introduce and clarify the terminologies such as agile development, requirements engineering, human factors, and automated vehicles.

1.2.1 Requirements Engineering

The International Requirements Engineering Board (IREB) defines *requirements* as a representation of the needs and desires of clients and users for developing new products or upgrading old products [20]. There are three different types of requirements:

- Functional requirement: a result or behavior that the product must provide
- Quality requirement: an issue not covered by functional requirements, such as performance, availability, security, or reliability
- Constraint: an additional restriction on acceptable solutions required to meet functional and/or quality requirements

Requirements engineering (RE) is the process of capturing these requirements and developing them into a set of accepted documents that serve as the basis for all subsequent activities [21]. According to IREB, RE is the process of defining and managing system requirements so that the resulting systems meet the wants and needs of their stakeholders [20].

A few decades ago, RE was viewed as a set of sequential activities, including requirements elicitation, analysis, specification, and validation [17].

In the first, *elicitation*, stakeholder requirements are gathered using various techniques such as storyboarding, questionnaires, and prototyping. In the second step, *requirements analysis and negotiation*, requirements are analyzed to see if anything is missing, and redundant conflicts are resolved through negotiation. The third step is *requirements specification*: the requirements are expressed formally or informally—for example, in mathematical form or as diagrams—and a requirements document is produced. The requirements are then *validated* in the final step for consistency and completeness [17].

However, traditional RE is not well integrated with the latest development methodologies and its implementation has become increasingly strained in the agile development environment. This tension has brought the realizations that it is challenging to produce products with changing user values using traditional requirements methods and that requirements are more likely to be developed with a focus on use together with the system rather than pre-specified [22].

1.2.2 Agile Development

Recently, agile methodology is becoming very popular in development companies due to being quite flexible and improving product success rates [23], over those

Table 1.1: Manifesto for Agile Software Development [26]

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

of traditional development methods. Agile methods encourage changes with low cost and deliver high-quality products in short iterations [24]. A key advantage of agile over traditional development is that it involves users throughout the development process, gets frequent feedback, and reflects customer values.

Agile methodology is often advised for small teams (six to eight developers) [24, 25]. According to the Agile Manifesto (Table 1.1) [26], the core principles of agile methodology are: focusing on people and interactions to create functional software in close collaboration with clients while de-emphasizing the use of processes, tools, plans, detailed documentation, and contract negotiation. Table 1.1 shows how agile manifesto is positioned compared to plan-driven approaches.

In agile development, the detailed requirements of plan-driven methods are abandoned in favor of continuous communication (with customers or product owners) [24]. Initially, an agile team will write user stories—short notes on little index cards—about the clients' requests. However, on the other hand, shortcomings of agile methods that have been mentioned include the discouragement of upfront planning and the limitation of engineering requirements to functional requirements specified by (ideal) scenarios [24].

1.2.3 Large-scale Agile Development

Agile techniques were initially applied by small development teams, but in recent years, they have also been widely adopted at large-scale companies [27]. The term *large-scale agile* refers to agile development in large teams and large multi-team projects [28]. According to the taxonomy of scale for agile software development, large-scale agile involves more than two teams [28]. Organizations with more than nine teams can be called *very large-scale*. However, Dikert et al. mentioned that large-scale agile development refers to agile development that includes more than six teams [29].

Several guidelines and frameworks have been developed for applying the agile paradigm in sectors other than software development (e.g., business strategy or operations) and in larger organizations. The most widely used framework for large-scale agile development, especially in the automotive industry, is the Scaled Agile Framework (SAFe [30]). SAFe organizes teams into larger groups (agile release trains) that develop and release their work regularly to offer end-user value [31]. Moreover, SAFe provides a requirements information model where multiple user stories are combined into one epic, which can represent mid- to long-term targets for groups of teams. The model also includes further constraints in the form of quality (non-functional) requirements [31].

1.2.4 RE in Agile

Agile RE or *RE for agile* agile development can be broadly defined as an agile way of performing RE, although no single definition has been agreed upon [32]. It has been determined that some traditional RE practices—such as face-to-face communication, customer involvement, requirements prioritization, review meetings, acceptance tests, and change management—can be applied to agile RE [33, 34].

However, RE in agile development presents a variety of challenges, such as ignoring non-functional requirements, client availability, team collaboration, knowledge sharing, insufficient documentation, and shared understanding of customer values [29, 34, 35].

The first attempts to address some of the challenges of RE in agile include works by Inayat et al. and Paetsch et al. [34, 36], who recommend combining traditional RE with agile RE and managing challenges like how much documentation is enough [37] to have a shared understanding of customer values. While it appears that many RE-related issues in agile can be tackled using RE methodologies, we need more studies to address challenges related to RE in agile.

1.2.5 Automated Vehicles

Automated vehicles allow the systems to perform all or some driving activities to support the driver. The driving activities range from adaptive cruise control, which controls the speed and headway of the longitudinal driving direction, to driving and interacting with other road users in various driving maneuvers. The Society of Automotive Engineers (SAE) has developed a six-level (0-5) taxonomy system to distinguish between different levels of automation (Figure 1.2) [4].

Starting at level 0, which is no automation, a vehicle is not considered an AV. This includes vehicles with assistance systems such as automated emergency breaking (AEB) and blind spot warnings. Levels 1 and 2 AVs are capable of limited driving activities (such as steering and/or speed adaptation). However, a human driver is still required to drive the car. At these lower levels, even when the vehicle performs certain activities, it is important to remember that the human is always considered the driver.

Level 3 is considered conditional automation; the vehicle can perform most dynamic tasks only within specific Operation Design Domains (ODDs). It can also ask a person to take over if needed. At Level 4 an AV can perform driving tasks without human intervention, but only within certain ODDs. In contrast, a Level 5 AV can perform all driving functions anywhere and anytime without any ODD limitations or human intervention.

These different levels of automation have different capabilities and limitations, which may have different impacts on the environment, traffic efficiency, and safety. AVs will affect not only their drivers, but also other humans who are involved in the driving system. For example, cyclists and pedestrians, who share the road with other users while traveling, will also be affected by the changes in the road traffic system resulting from the use of AVs.

To develop automotive systems, suppliers and the OEMs (Original Equipment Manufacturers) industry work together. According to Broy [38], there

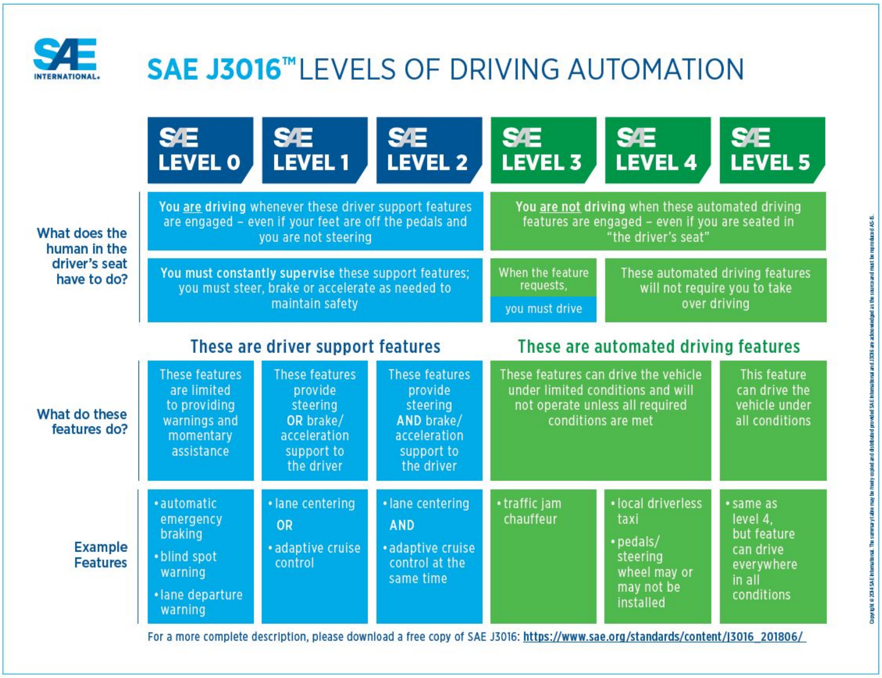

Figure 1.2: Description of levels of driving automation by SAE [4]

has been a trend toward outsourcing large parts of the implementation, which involves many fields, such as mechanics, electronics, and software. Since requirement specifications serve as a major channel of communication between suppliers and OEMs, requirements engineering is important to ensure collaboration between these fields. Since requirement specifications serve as a major channel of communication between suppliers and OEMs, requirements engineering is important.

Traditionally, automotive development has been characterized by long lead times [39] and consistent, sequential engineering procedures. The industry is currently transitioning from plan-driven, to more value-driven, continuous approaches (also known as agile techniques [24]) [40]. According to Gren and Lenberg, the fundamental motive for such a transition is the ability to respond to changing requirements [41]. However, automotive developers find it challenging to keep up with efficient methods for designing, documenting, and managing requirements for systems that are ever more complicated [30, 42]. As a result, we need to devise new methods for handling requirements in agile development for complex systems such as AVs [19].

A wide range of quality standards must be met when developing automotive systems. Examples include performance, usability, safety, and reliability. This matches very well with human factors knowledge. In addition, since automobiles are cyber-physical systems operating in a human environment, human factors are essential to the successful development and adoption of AVs. Therefore, to effectively build AVs, we must better incorporate knowledge of human capabilities and limitations. Moreover, RE is crucial in transferring human factors expertise to AV developers.

1.2.6 Human Factors

The Journal of the Human Factors and Ergonomics Society [12] defines human factors as the "scientific study of fundamental knowledge of human capabilities and limitations and the basic understanding of cognitive, physical, behavioral, physiological, social, developmental, affective, and motivational aspects of human performance, in order to yield design principles; enhance training, selection, and communication."

Further, the same journal describes human factors as "concerned with the application of what we know about people, their abilities, characteristics, and limitations to the design of equipment they use, environments in which they function, and jobs they perform."

According to a health and safety executive, "human factors are organizational, environmental, and job-related, as well as individual characteristics that affect the work environment and the quality of work" [43].

Given that there are many different definitions of human factors, individuals may have different perspectives on what constitutes human factors, and their perspectives may influence how they approach human factors in their profession. Communicating needs and information can be challenging when people are not using the same definitions [12, 44].

It is important to determine the design and development experts' actual perceptions of human factors. This may be particularly relevant when investigating the role of human factors in a multidisciplinary study (as in the present), where requirements engineers, human factors specialists, and a range of other engineers are involved. As a result, it is important to define a specific definition for a specific topic (here, AV design).

Considering the different perspectives on human factors, we can see that it presents the basic values that are considered important when developing software and road transport [10, 45]. 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.

In AV development, human factors belong to both software development and hardware AV design. Software aspects of human factors include how the vehicle stays in the lane [46, 47], how external road users are to be communicated with [48, 49], how and when software-based HMIs (human-machine interfaces) display information [50], and how humans and AVs communicate [51]. Physical factors include everything from seating ergonomics (how AV capabilities affect automotive interiors [52]) to the physical design and location of HMIs.

These examples show how much knowledge of human factors is required for good engineering. Some other researchers also highlight why it is important to include human factors in development [9, 10, 11]. However, human factors is not keeping pace with AV development, and it is unclear how aware engineers are of human factors in their design decisions. Thus, we must find effective ways to maintain momentum and incorporate human factors into development. Early research suggests that human factors should be included in the early

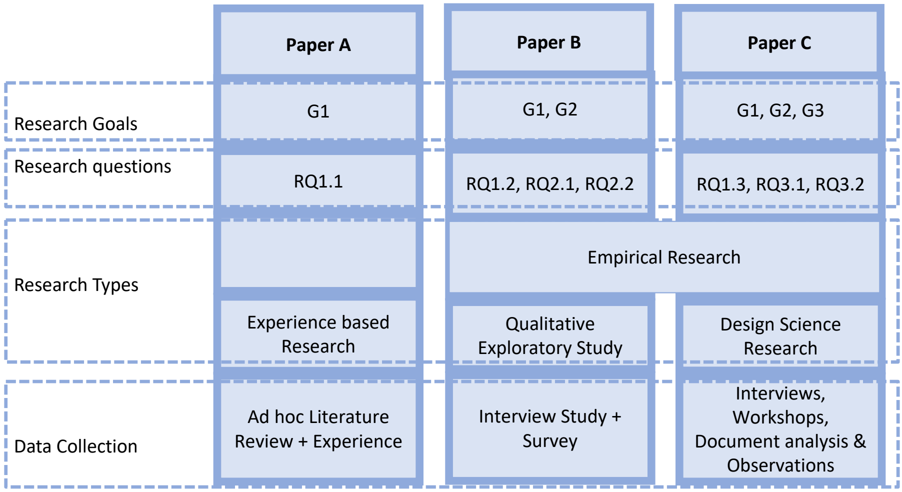

Figure 1.3: Research methodology with research goals, research questions and references to included papers

stages of development [14, 15]. However, it is not clear how to effectively incorporate human factors knowledge in the early stages of AV development when adopting an agile methodology.

1.3 Research Methodology

This research investigates multidisciplinary areas, including AI, HF, RE, and agile in AV development. We need to explore each area in our specific context that involves considering human factors when designing automotive systems or software. The research methodology used in this thesis mainly follows the empirical paradigm with a focus on investigating problems in the real world. Empirical studies focusing on a practical problem are excellent for exploring emerging topics of relevance to the industry [53]. These studies can expand our understanding of the current status and lead us to potential solutions for RE and human factors in developing agile AV. Empirical studies can be either qualitative or quantitative [54]. We primarily relied on qualitative research methodologies to support our exploratory research goals.

This section describes the methods and how they were used in the included studies. To lay the foundation of the empirical work in papers B and C, in Paper A we first explored the problem in connection to AI, RE, and human factors. That is, Paper A is a problem statement. The problem statement was then followed by empirical work to obtain a deep understanding of the issues facing practitioners in the industry and their needs (Papers B-C). We used different research strategies for the two empirical studies. In paper A data was collected using interview study and validated the outcomes of the data analysis with a survey. In Paper B, we used a mix-methods approach to collect data. Figure 1.3 shows the research methodologies we used across the three papers and the data collection methods and references to papers used to compile this thesis.

1.3.1 Qualitative Exploratory 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 which 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 [55]. 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 be used for different purposes, for example explanatory (for example to explain results from a previous quantitative study) or exploratory (for example to explore a research area and to propose hypothesis or potential causal relationships for future in-depth studies). Such exploratory studies are in particular useful when a researcher is unsure about the key factors to examine due to the complexity of a particular domain. Further, this type of approach may be necessary when the topic is novel, has never been studied with a specific sample or group of people, and existing theories do not apply to the particular sample or group [55].

The study in Paper B is exploratory in nature, so emergent coding-based qualitative research methods are the most suitable. In our desire to map the human factors landscape in relation to AV development, we were particularly interested in the personal opinions of experts in the field; therefore, we conducted a qualitative exploratory interview study.

An interview study is one of the most widely used methods for qualitative research studies. In Paper B, we used open-ended, semi-structured interviews, from which common themes using an iterative coding strategy were extracted [56]. After completing our interview study, we validated the findings during a workshop, in which we presented our results to experts and used a survey to collect their opinion about our findings.

1.3.2 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 [57], 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 [58].

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 [59]. 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, we adopted the design science research method in Paper C. Our research aims to create appropriate methods for creating requirements strategies (our design artifact) for businesses using large-scale agile development. Such requirements strategies should be able to meet real-world needs, incorporate state-of-the-art information, and be empirically evaluated in real-world settings. Hevner et al., [58] state that it is crucial that the underlying issue be pertinent 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.

1.4 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 [60]. 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 previously planned comparisons, strategies, or statistical analyses to improve validity [56]. Even though it is difficult to validate qualitative studies, some methodologies propose to do just that [61], aiming to evaluate and mitigate threats to validity. We consider the four perspectives of validity threats as presented in Runeson and Host [62] and in Easterbrook et al. [53].

1.4.1 Construct Validity

Construct validity relates to how well the operational measures considered match the primary concerns of the researchers and the interview guide [62]. In particular, threats to construct validity arise from using concepts with different interpretations. For example, during the interviews, misunderstanding might occur while using the terms agile, human factors, strategy, etc. To mitigate this threat, we ensured that the interview guide and its questions would be correctly interpreted by the interviewer and the interviewees (with respect to the purpose of the study). While collecting data, we carefully established a common understanding of terms at the beginning of interviews and workshops.

Moreover, we relied on the complementary knowledge and experience of the coauthors, 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 B, 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 of the interviews, no interview was conducted by only one of the authors. We were able to resolve these difficulties during the interviews and revise the interview guide in order to make sure that all interviewees interpreted the questions similarly. In the same way, we improved the interview guides and data over the multiple iterations of the design science study in Paper C.

1.4.2 Internal Validity

Internal validity considers the design of a study and whether the findings are derived from the collected data [53] and investigates if external factors could impact the findings.

To minimize this threat, we carefully collected data about the topics and their contexts and provided detailed descriptions of our findings [53]. In both studies that relied on interviews for data collection (Papers B and C), we ensured that interview transcripts were accurate—either through member checking or audio recording. Additionally, analyzing the data over several iterations and reporting it in the paper also helped reduce internal reliability threats.

Collecting data from multiple sources (with a variety of roles in different contexts) facilitated the triangulation of the findings. To collect data representing different perspectives and roles, we carefully selected participants by first understanding the companies they came from. We chose different roles from different companies to avoid overly narrowing our findings. However, given that the interviews were selected through our industry contacts, there may still be a selection bias. To mitigate this threat, the results of our studies were discussed in workshops (Papers B and C) and validated through a survey (Paper B) to collect their opinion about our findings.

1.4.3 External Validity

External validity refers to the ability to generalize results to different contexts. Our research aims to understand the phenomena under study and to represent specific contexts rather than to establish statistical generalizations. Easterbrook [53] 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.

For example, in Paper C, we identified the building blocks for defining the requirements strategy (RQ3.1) 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 concerning RQ2.1 and RQ2.2 with several participants from different companies. We also included participants from different domains in our studies, increasing external validity through triangulation. However, only experts in the automotive industry were considered when developing properties and implications (RQ2.1 and 2.2) for the agile way of working, human factors, and RE (Paper B). As a result, we were able to provide more specific results that were directly relevant and applicable to this area. We expect the results to be applicable to other disciplines, but more research is required to prove this.

1.4.4 Reliability

According to Easterbrook [53], the extent to which other researchers can reproduce one's work is called reliability. In qualitative research, researcher biases and reactions may threaten the reliability of the results due to interactions with participants or interviewers.

To reduce these threats, we involved other researchers during our studies. For example, during the interviews, more than one researcher was present. We also carefully described our methods to make them as replicable as possible. Additionally, we have prepared interview and analysis guides so that others can use them to reproduce our research methods in different contexts. Finally, by laying out our analysis in detail when describing the results, we allow others to recover,in cases where their results differ, e.g., by identifying reasons.

1.5 Summaries of Studies

This section briefly outlines the three papers on which this thesis is built. Full papers can be found in Chapters 2-4. Table 1.2 provides an overview of the papers and related research questions with each paper's main contribution to each research question. Note that RQ1.1 was not explicitly part of the paper's research question, but emerged as relevant during the data analysis of this thesis. Concretely, we translated the objective of paper A as a research question.

1.5.1 Paper A

Artificial intelligence (AI) advances are enabling a new generation of complex, AI-intensive systems and applications which promise exciting improvements for societal growth. However, developing these applications and systems pose new challenges. The goal of this paper is to determine the challenges related to the requirements of AI-intense systems.

We present these challenges in four areas: (i) contextual definitions and requirements, (ii) data attributes and requirements, (iii) performance definition and monitoring, and (iv) human factors. Moreover, we derive these challenges from three complex AI-intense domains: i) industrial automation, ii) home automation, and iii) transportation.

The first important problem is the definition of the AI-intense system's context and requirements. It is necessary to train the machine learning model for the newly defined context, because the expected system behavior (and particularly, safety features) cannot be assured when a trained machine learning model is applied to a different context [63, 64].

The second problem area defines the necessity for appropriate definitions of quality attributes and requirements on data that are used in the AI-intensive

Table 1.2: Overview of papers with research questions and contributions

Paper	Research questions in corresponding papers	Contributions/ main findings	Thesis RQs
A	**RQ1**: What are problem areas related to requirements engineering for AI-intense system development?* *This research question was not explicitly mentioned in the paper, but characterizes the content of the paper	**I**dentification of RE-related problem areas for AI-intense systems	RQ1.1
B	**RQ1**: How do human factors experts and AV engineers characterize human factors in relation to AV development?	Defintion of "Human factors" in context of AV development	RQ1.2
	RQ2: Which properties of human factors and agile ways of working impact AV development?	Properties of agile and human factors in the context of AV development	RQ2.1
	RQ3: What are important implications when aiming to better integrate human factors into AV development?	Implication for agile, human factors and requirements engineering in agile Av development	RQ2.2
C	**RQ1**: What are the critical challenges with requirements engineering in agile development?	Challenges related to requirements engineering in agile development	RQ1.3
	RQ2: How do companies aim to address these challenges?	Solution strategies for challenges.	RQ2.3
	RQ3: Which potential building blocks should be considered for defining a requirements strategy?	Identification of building blocks for defining a requirements strategy	RQ3.2

system. This study argues that the quality of data is the most crucial component of AI systems and that system quality (such as safety and robustness) cannot be guaranteed without guaranteeing such requirements on the data that are used in the system [65, 66, 67].

The third problem area is related to performance definition and monitoring. After context and data requirements for an AI-intensive system were established, the issue of how to track fulfillment of these requirements and the system's performance arose. However, before addressing how anything can be monitored, it is critical to first comprehend what needs to be monitored [68, 69].

The fourth problem area concerns human factors knowledge, which is particularly important for this thesis. To improve vehicles' safety as well as people's trust in (and acceptance of) automated vehicles, we need to concentrate on integrating human factor requirements into modern large-scale AV development processes [70, 71, 72].

Each of the problem areas invites a number of research questions, which are listed in Paper A along with a preliminary road map describing how to approach them. We conclude that requirements engineering is pivotal when considering the safety, trust, and acceptance of AI-based automated systems (such as AVs) and human factors play a role in each of the identified problem areas for defining and guaranteeing the behavioral and quality characteristics of AI-based systems.

This paper motivated us to look into the detail of human factors and requirements perspective.

1.5.2 Paper B

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.

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 present the analyzed themes and come to the following conclusions: 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; however, our main contribution is adding the *design cycle* part of the definition. It is paramount 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 reflects 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 [73]. 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 [74] for development teams.

We believe that requirements engineering 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.

Paper B answers some of our thesis research questions: contributing to our understanding of human factors (RQ1.2) and exploring properties and implications of human factors, agile, and requirements (RQ2.2 and RQ2.3).

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 Requirements Engineering 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 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 workflows and concrete challenges. 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. 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 process 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 lens of Fricker and Glinz's shared understanding [75]. 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 an 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. 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 workflow and the feature flow. We need to map the structural and organizational perspectives to the work on 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 valuable, improving the requirements quality and keeping reviewers informed about recent changes.

Paper C contributes to answering RQ1.3: What are the critical challenges with requirements engineering in agile development? This paper is also related to the thesis research question RQ2.3 (Solutions or strategies to counteract

RE challenges). Finally, it defines a requirements strategies model for agile development (RQ3.1). It is, however, less focused on human factors because it will serve as a foundation for future research to develop a requirements strategy for human factors knowledge.

1.6 Discussion

The thesis research goals have been addressed by our answers to the research questions, as follows.

1.6.1 G1: Domain Exploration

To achieve research Goal 1, we addressed the three questions RQ1.1, RQ1.2, and RQ1.3.

RQ1.1: From a requirements perspective, what are the critical research areas when building AI-based systems?

RQ1.1 is related to the general exploration of problem areas within RE in developing AI-intense systems. We found that there are four key areas of requirements engineering: (i) contextual definitions and requirements, (ii) data attributes and requirements, (iii) performance definition and monitoring, and (iv) human factors knowledge (in order to guarantee system behavior and quality attributes and establish good system process support). As the scope of this thesis is human factors requirements, in this section we will only discuss challenges related to human factors.

Building complex, automated systems and products requires a focus on the technical aspects of the system along with human factors. AI-intensive systems present new types of challenges to humans (e.g., activating and deactivating automated features). When a feature such as adaptive cruise control (helping the driver keep the speed and the distance to the vehicle in front) is activated relieves a mental burden; however, in some difficult situations, e.g., when humans need to take over control, it can create mental stress. Knowledge of human factors can help identify such scenarios during system development and thus guide it according to human capabilities and limitations—ultimately providing a safe, acceptable, and reliable system.

Therefore, understanding and incorporating human factors knowledge during the development of automated systems is key to successfully deploying automated vehicles. Human factors must be considered earlier in the development phases, right when the concepts are developed: that is, in the requirements engineering phase. However, it is challenging to include the human factors knowledge in the agile development of complex AI-intense systems such as AVs. One of the reasons is that to compete with the market, agile focuses on short delivery cycles. Hence, development teams are more focused on technical knowledge and often neglect the details of human factors. This raises important questions:

- In what way must human factors be considered in order to understand and ensure the appropriate behavior of AI-intense systems?
- In what way must human factors be considered in order to understand and ensure the appropriate quality attributes of AI-intense systems?

- How can human factors knowledge be effectively used in modern system development approaches?

We need to guarantee system behavior because it is a safety-critical system; we need to guarantee system quality attributes to make sure that these qualities are in there, and we actually also need to make sure that we can develop it using a modern approach (i.e., based on agile or continuous development methods).

The answers to these questions will help to guarantee certain behaviors and quality attributes of the overall system and its applications. In particular, requirements of various types (quality/functional/constraints) should cover relevant human factors knowledge.

In this research question we investigate which problems we need to solve in the context of developing AI-intense systems or designs. We identified a few important perspectives including human factors. Note that RQ1.1 only highlights the questions and does not answer them because it is not an empirical study. Instead, it explores the problem of how human factors challenges fit into the landscape of other challenges to AI intensive systems. RQ1.1 is used as a motivation for achieving our thesis goals. This question spotlights the importance of human factors and related problems in developing AI-intense automated systems. However, it focuses less on the perspective of large-scale agile. This paper emphasizes the fact that human factors can contribute to understanding and ensuring system behavior from a qualitative perspective and thus help improve system acceptance, trust, and safety.

Answering RQ1.1 led to looking into the details of human factors (RQ1.2) and further exploring the challenges related to requirements in agile development (RQ1.3).

In the following, we elaborate on and answer RQ1.2.

RQ1.2: How do human factors experts and AV engineers characterize human factors in relation to agile AV development?

Several definitions of human factors are available [12, 44], even on the homepages of significant journals in the field, depending on the specific research context (e.g., [12]). Clearly, communicating requirements and knowledge could be challenging when people use different definitions [76].

Even when the definitions seem straightforward, different people may have different opinions about what human factors involves [44], 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 [71].

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, creating a specific definition related to a specific topic (here, AV design) is warranted. Thus, RQ1.2 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 [12], making it more precise about the relationship between human factors and AV. Based on the definition by The Journal of the Human Factors and Ergonomics

Society [12] we define the human factors field as *"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****."*

We identified the crucial qualities that a practical definition should have by answering this question (see Section 3.4.1). We began with a general definition of human factors (derived from [12]), and connected it to the development cycle. We added the *design cycle* to the definition because linking human factors to AV design cycle, is essential in the context of our research. We observed that the engineering design cycle part is not that visible in many existing definitions, but it is important and needs to be connected to a strong working definition. As a result, we have developed a shared vocabulary for human factors in AV development.

RQ1.3: What are the critical challenges with requirements engineering in agile 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 [75]. According to Fricker and Glinz [75], 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 1.3 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, 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 [29, 77, 78]. 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.

Table 1.3: Overview of Challenges. Indices ([1],[2],[3]) show in which case study a challenge was encountered.

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

1.6.2 G2: Problem space Investigation

While studying Goal 1, we discovered several connections between human factors, the agile way of working, and requirements in AV development. The discovery motivated us to study these associations and further investigate the problem (G2). Thus, we studied RQ2.1, RQ2.2, and RQ2.3 to reach Goal 2.

RQ2.1: Which properties of human factors and agile ways of working impact AV development?

This research question is explicitly addressed by Paper B, in which we analyzed 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 [24]. 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 3.4.2).

We observed that human factors knowledge is closely related to agile development. For example, agile supports iterative incremental work, and

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 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 typically implements fast, iterative increments, which do not usually allow time for the rigorous experiments that HF 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) [24]. In contrast, human factors emphasize having extensive knowledge and detailed system evaluation before release.

We conclude that the properties of agile and human factors complement each other in principle. Thus, the inclusion of human factors in agile methods 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.

RQ2.2: What are the important implications when aiming to better integrate human factors into AV development?

Paper B reveals several implications in three themes, i.e., agile work, human factors, and requirements engineering. 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 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 HF 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 [79]. 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 new roles for human factors and RE in agile AV development. The role of human factors knowledge and RE becomes less clear in the agile setting. 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 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.

In summary, 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.

Furthermore, we believe that our implications provide beneficial knowledge to those who are responsible for developing design procedures and tools—as well as to HF 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 HF studies and their findings.

RQ2.3: How do companies aim to address RE challenges?
In the following, we propose several solution strategies that address the challenges related to the needs identified in RQ 1.3 (see Table 1.3).

[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 solutions 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 (see RQ3.1).

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 in RQ3.2. 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.

1.6.3 G3: Solution space Investigation

RQ3.1: Which potential building blocks should be considered for defining a requirements strategy?

There are many challenges related to RE that can be solved through RE approaches. In RQ3.2 of this thesis, we introduce the concept of a "requirements strategy" as a method to define requirements engineering practices to tackle challenges related to requirements engineering in agile.

To define *requirements strategy*, our inspiration comes from *test strategy* [80, 81], 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 [82, 83]. 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-and-feature flow perspective. Table 1.5 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 [84] among these perspectives, particularly in terms of developing a common language (i.e., the functional perspective in Table 1.3) and facilitating the flow of information (i.e., evaluating the building and approach in Table 1.3).

A requirements strategy should be created and systematically documented to ensure all objectives are properly addressed and understood by all stake-

Table 1.5: Building Blocks of a Requirements Strategy: How shared understanding impacts decisions in the workflow [87]

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

holders. 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. [85, 86].

Our guidelines for requirements strategies aim to support organizations 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 workflows. 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.

1.7 Conclusion

This PhD project aimed to investigate how human factors knowledge can be efficiently communicated to AV developers in large-scale agile development. We have approached the solution from a requirements engineering perspective. We conducted studies to understand current challenges and interrelations between the agile way of working, human factors, and requirements engineering.

In Paper A, we studied a number of problem areas for requirements engineering for AI-intense systems. We identified that to guarantee the desired system behavior, to guarantee system attributes, and to establish process support in an organization, there are four major problem areas that need to be addressed: defining contextual descriptions and requirements, setting data attributes and requirements, establishing performance definition and monitoring, and managing human factors knowledge efficiently.

In Paper B, we derived a working definition of human factors for AV development. Then we discovered properties of the agile way of working and human factors in the context of AV development. Moreover, we also defined relevant implications for agile development, human factors, and requirement engineering.

Lastly, in Paper C, we identified challenges to exploring the domain further and provided several solution strategies to overcome those challenges. To support practitioners with formulating a strategy on how to generally address the challenges in a specific context, we provided a requirements strategy template.

1.8 Future Work

Although we find the requirements strategy concept very promising, the guidelines developed as a template warrant further improvement. In the future, we would like to define and develop a more concrete requirements strategy to be applied in different contexts (for example, AI or human factors). As a first step, we aim to provide guidelines and concrete suggestions for focusing on managing human factors knowledge in a specific case study.

Moreover, in one of our studies, we learned that experiments are integral parts of human factors. Therefore, an interesting area for future work might be to learn how we can support the integration of human factors experiments into large-scale agile development. This integration may include protocols and methods where the process can support an environment suitable for iterative human factors experiments. These protocols will eventually allow developers to accumulate highly reliable information and knowledge. Therefore, future research should provide a theoretical framework to support AV engineers and human factors experts as they conduct iterative experiments to learn more about human factors.