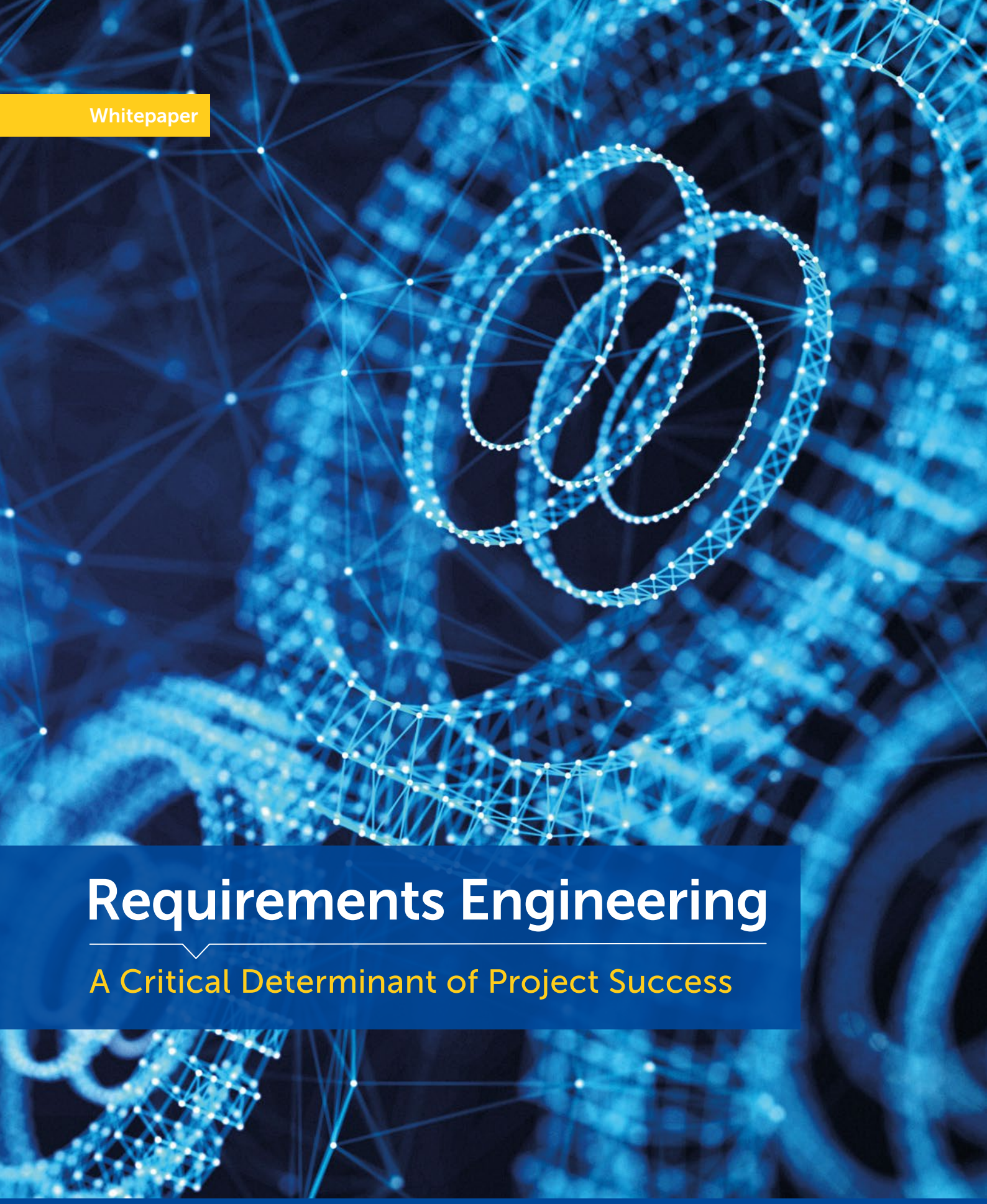


Whitepaper

Requirements Engineering

A Critical Determinant of Project Success

fortiss

Requirements Engineering
A Critical Determinant of Project Success

Authors

Anton Luckhardt

fortiss GmbH,
Guerickestr. 25
80805 Munich
luckhardt@fortiss.org

Prof. Dr. Daniel Mendez

fortiss GmbH,
Guerickestr. 25
80805 Munich
mendez@fortiss.org

Dr. Norman Schaffer

fortiss GmbH,
Guerickestr. 25
80805 Munich
schaffer@fortiss.org

Content

Abstract

Efficiently handling requirements - often labelled as "Requirements Engineering" - is a crucial task for successful software development projects. Requirements - such as functionalities that reflect business processes, requirements regarding the security, performance, usability and reliability of a software system - of all stakeholders have to be identified, processed into a structured and understandable and traceable format, and ultimately implemented to ensure the success of the project. Requirements Engineering ultimately deals with capturing the problem space as precisely as possible with structured approaches to requirements elicitation, analysis, documentation, and their validation and verification. In this whitepaper, we will give you a brief overview of the topic of Requirements Engineering and its immediate relevance for the success of software projects. Furthermore, we provide you with a brief insight into the ongoing practically relevant research. At fortiss, we engage in Requirements Engineering research with a strong focus on solving real world problems. Although Requirements Engineering might not seem as relevant as other disciplines at first sight and it might even seem to disappear under the umbrella of modern development approaches, such as "agile", we will demonstrate that efficiently handling requirements is - and should be - everyone's concern. Further, we deep-dive into two topics: the role humans take in Requirements Engineering and whether agile Requirements Engineering is the holy grail (it is not). We offer practitioners a lightweight and free-to-use tool to understand one's own practices and benchmark their organization against others. Lastly, we discuss ongoing challenges of practice-oriented research in Requirements Engineering, namely regulatory compliance in Requirements Engineering, data-driven Requirements Engineering and the role of automation, and Requirements Engineering for human-centered environments, building on creativity and methods like design thinking. This whitepaper is for practioners and researchers who want to better understand contemporary challenges in Requirements Engineering, including their own and who aim at learning about ongoing research endeavours to tackle those challenges.

Requirements Engineering – Quo Vadis

The success of any project aimed at developing software-intensive products and services - that is, any system where software makes a significant contribution to the design, development, and operation of the system[1] - ultimately depends on how well the final project deliverables reflect the diverse needs of the various stakeholders. This process is commonly referred to as **Requirements Engineering (short: RE)**. A basic RE process consists of requirements elicitation, documentation, validation, and management. However, these activities may be different across various existing approaches, such as the waterfall model and agile project management.

While different definitions for RE exist (e.g. IREB or ISO/IEC/IEEE 29148:2018 standard), we generally refer with "RE" to roles, activities, and outcomes that define goals and requirements for a software-intensive product or service, regardless of the surrounding process model and terminology used. In reality, we often observe that RE is subsumed under the umbrella of software process models or product management approaches, often without even using the term "Requirements Engineering". In this regard, we do not even try to distinguish between those various approaches but refer with RE to the systematic handling of requirements - from their inception to their specification and validation - which is in scope of any product development regardless of the chosen approach and terminology and regardless whether it is done explicitly or implicitly.

RE is crucial for a successful software and systems engineering

No matter how RE is ultimately carried out, it is a critical determinant for software quality. This applies especially to software and systems engineering in early stages, yet it accompanies a project over its entire life cycle. This is not surprising, considering that unambiguous and measurable requirements are a basis for various implementation activities, quality assurance, and project organization and management.

The RE process must therefore be carried out in a conscientous way. In reality, however, RE is often treated with little to no care or even neglected entirely, leading to a multitude of problems and additional effort along downstream activities, a quality decay in the product, or even a complete project failure. In a world pervaded by software and where most of our daily routines are supported – if not dominated – by software-intensive systems, excellence in RE has therefore become key.

The causes for these problems in RE often differ. Still, most often, it comes down to incorrect, missing, inconsistent, or ambiguous requirements. These phenomena occur irrespective of the whether companies develop software products on their own, whether they develop standard or custom software, or whether they have outsourced their development, thus, providing requirements to external suppliers.

> **"The serious problems that have happened with software have to do with requirements, not coding errors."**
>
> Nancy Leveson

1 IEEE Recommended Practice for Architectural Description for Software-Intensive Systems," in IEEE Std 1471-2000 , vol., no., pp.1-30, 9 Oct. 2000, doi: 10.1109/ IEE-ESTD.2000.91944.

Requirements Engineering - Quo Vadis

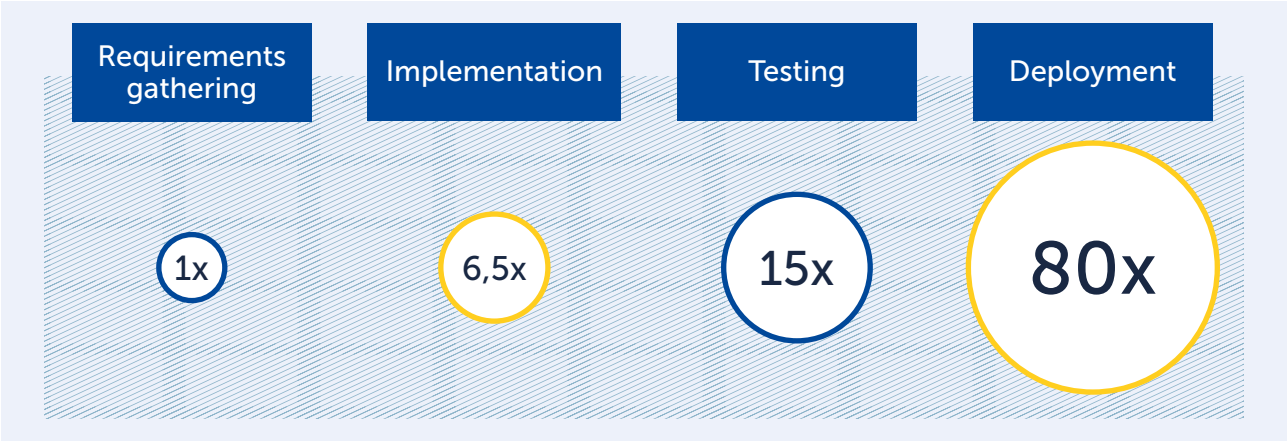


Figure 1: Increasing cost factor to fix RE errors at different stages of the product (Grady, Robert B. (1999). An Economic Release Decision Model : Insights into Software Project Management. ASMC, Software Quality Engineering, 227–239.)

An explicit RE is crucial

The consequence of an insufficient RE is often a time overrun, a budget overrun, dissatisfaction on the part of the customer, and/or the failure of the entire project. The increase in the cost of correcting incorrect or ambigous requirements increases the more advanced the project is when these requirements are discovered. An adequate and explicit RE with adequate quality assurance can significantly reduce the risks of project failure in advance.

RE is, however, not only complex, but it is also crucial for a successful development project. 33 %[2] of the errors in software development projects are rooted in an insufficient RE. Moreover, 36 %[3] of the errors that are encountered in RE, are known to lead to project failure. RE is therefore not only complicated, but is also critical.

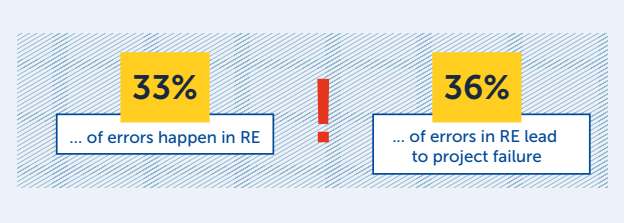


Figure 2: Relevance of an effective RE

No RE approach fits all situations

To approach the various challenges in RE, academia has developed a plethora of methodologies, templates, tools, and even fundamentally different strategies. However, which one to choose in which situation is mostly left to the individuals' judgment and expertise.

The challenge here remains: even if one approach has proven to be successful in one project, it might turn out entirely alien to the next project's characteristics and needs and even the individual preferences of the project participants. Many variables to consider can change from project to project and even from release to release. These can be internal influences such as the sector, company size, product type but also external such as legal or social influences and different technical and content knowledge levels among the stakeholders. The vast number of influencing variables involved makes the development of a universal one-size-fits-all solution for all stakeholders impossible. An agile RE is different to a plan-driven one, but is it a better fit for all project situations (and surrounding constraints, including regulatory ones)? The effectiveness of RE approaches largely depends on the practicalities of project environments. Yet, much of today's research in RE still relies on conventional and often purely academic wisdom and proposes universal one-size-fits-all approaches to practical problems and needs not well understood.

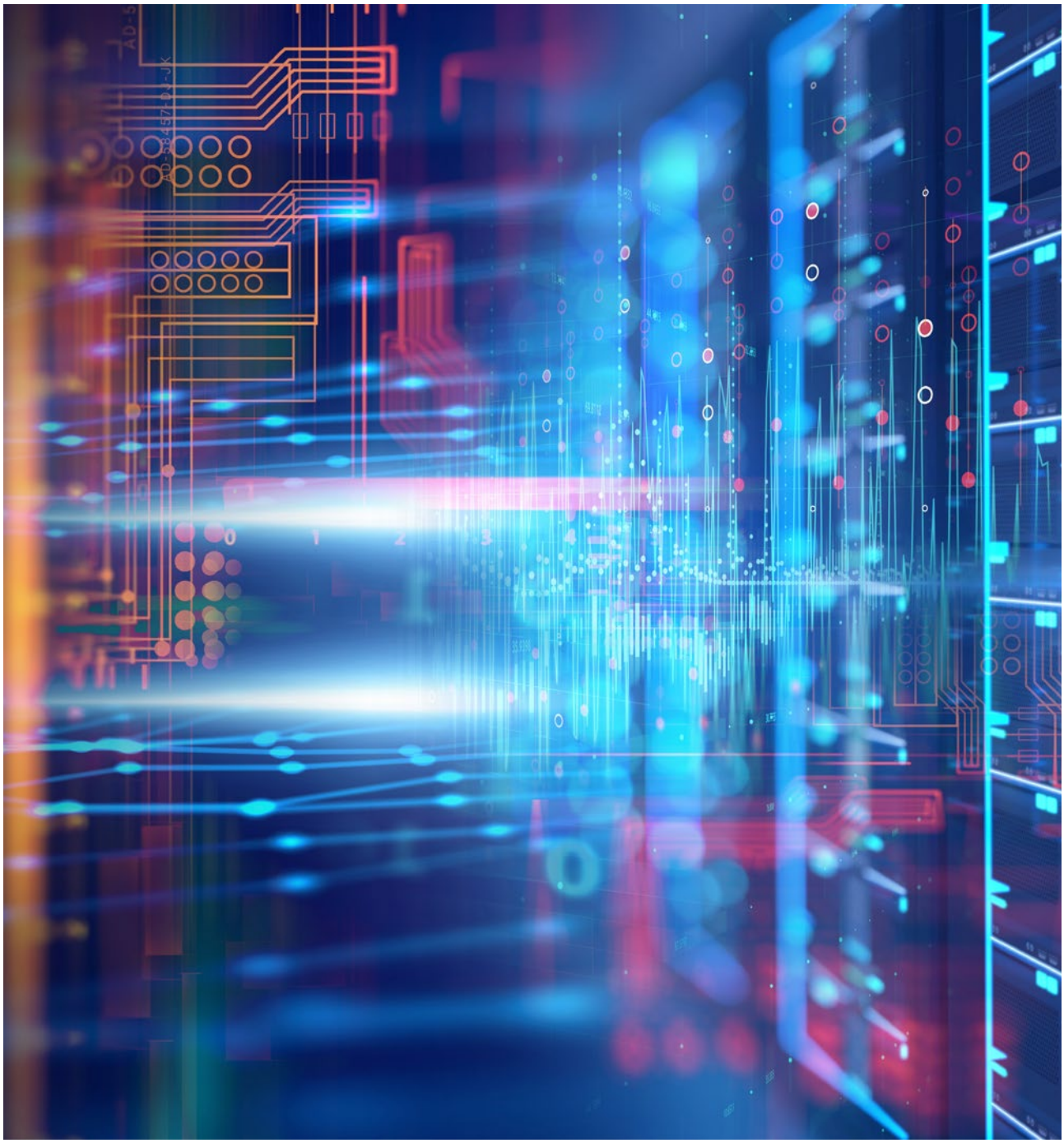
2 Hamill, M., & Goseva-Popstojanova, K. (07 2009). Common Trends in Software Fault and Failure Data. IEEE Trans. Software Eng., S. 484-496.

3 D. Mendez Fernandez, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetrò, T. Conte, M.-T. Christiansson, D. Greer, C. Laseni us, T. Männistö, M. Nayebi, M. Oivo, B. Penzenstadler, D. Pfahl,R. Prikladnicki, G. Ruhe, A. Schekelmann, S. Sen, R. Spinola, J.L. de la Vara, A. Tuzcu, R. Wieringa. Naming the Pain in Requirements Engineering: Contemporary Problems, Causes, and Effects in Practice. In: Empirical Software Engineering Journal, Springer, 2016

Everybody does RE

Although RE is undoubtedly an important task, our experience has shown that the general perception of its importance is comparatively low. The reasons for this can range from a general lack of interest to sheer unawareness of the goals and importance of RE. However, the task of RE is not the responsibility of the requirements engineer alone, but of all stakeholders involved.

The quality of the requirements is directly linked to the input from all stakeholders. The requirements engineer can facilitate this by improving awareness and using RE techniques to act as a catalyst. RE is therefore a task that everyone in the project can contribute to in order to increase the quality of the requirements, and at the same time, a well conducted RE process results in a benefit for everybody involved, not only for the software developers.



Deep dives

We, the RE team at fortiss, engage in research in RE with a strong focus on solving real world problems. With this whitepaper, we give you a short glance into ongoing practically relevant research in RE. In addition, to understand your own organisation's RE practices, methods used and challenges, and to understand how you fare compared to competitors or other industries, we have developed the lightweight **Requirements Engineering Quick Check** which we will present later.
In this section, we provide deep dives on nuanced topics, which, from our point of view, will be crucial in the upcoming years. We don't pretend to have and to provide universal wisdom, but rather give simple insights with high impact. With this whitepaper, we aim to raise awareness about the relevance of RE and the challenges it faces by providing an evidence-based perspective on the discipline. This perspective emerges from more than a decade of empirical research we have conducted under the "Naming the Pain in Requirements Engineering" (short: NaPiRE) initiative which constitute a global family of investigations of practices and problems in industry (see also **www.napire.org**).

Human factors are a major source for challenges in RE

Software is developed by people, for people. In other words, software is a by-product of myriad human activities incorporating problem-solving skills, social interactions, communication, and cognitive aspects. However, human nature is largely unpredictable and, thus, impossible to capture in models. It creates intricate dynamics in the software development process, which must be appropriately addressed by using competent skills.

Much change occurs while software is being developed, and agility by the project participants is required to adapt and respond to such changes. Recently, the discipline of software engineering has begun to adopt a multidisciplinary view and has embraced theories from more established disciplines, such as psychology, organizational research, and human-computer interaction. The RE phase of software development is characterized by intense communication activities involving a diverse range of people with varying levels of skills, knowledge, background, and status, where the overall goal is to achieve a shared and clear understanding of the problem between different people, which is further affected by the complexity, vastness, and volatility of the requirements itself. In this context, we refer to human factors as activities that are performed by humans and subsequently lead to problems in the RE process (an example could be miscommunications that result in incomplete requirements). However, the question arises why we should care about human factors at all. This can be answered simply by the fact that an analysis of software projects worldwide in the NaPiRE initiative showed that around 50% of all problems occurring in RE are due to human factors. The larger the projects become, the more complicated architectural or technical decisions become. Additionally, more people will be involved in carrying out these tasks. This creates the challenge of managing these projects properly.

We are currently aware of the relevance of human factors, but it is still not completely understood under which exact conditions they occur and, more importantly, how they can be prevented. Our field of research wants to answer these questions as we provide solutions that might increase the quality of the software by avoiding many problems in the first place.

Agile – The Holy Grail of RE?

At the time of writing this white paper and considering the software engineering literature, one might reasonably assume that "agile" has become the de-facto standard. Considering the distribution of the different working methods of the participants of the NaPiRE study in Figure 3, the same tendency can be observed. While only 25% of the participants stated that they work exclusively according to plandriven approaches, 75% already use at least a combination of both worlds, while 41% of all participants have already switched to exclusively agile working methods. The superordinate agile development principles include characteristics that can be somewhat counterintuitive for rather traditional RE where requirements are explicitly elicited, refined and classified, analyzed, documented, and validated. However, these agile principles promise to overcome the challenges associated with traditional RE by, among other things, being more open to change and incorporating the thoughts of the people working on the problems. However, is this really true?

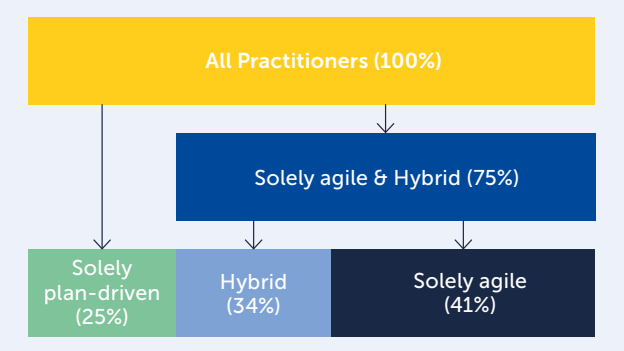


Figure 3: Self-assessment of practitioners interviewed as part of NaPiRE on which basic working methods they employ

	All participants (100%)	Solely Plan-driven process (25%)	Agile and Hybrid process (75%)	Solely Agile process (41%)
Top 1	Incomplete or hidden requirements	Incomplete or hidden requirements	Incomplete or hidden requirements	Incomplete or hidden requirements
Top 2	Communication flaws between the project team and customer	Communication flaws between the project team and customer	Time boxing	Time boxing
Top 3	Time boxing	Time boxing	Communication flaws between the project team and the customer	Communication flaws between the project team and the customer
Top 4	Communication flaws within the project team	Communication flaws within the project team	Moving targets	Communication flaws within the project team
Top 5	Moving targets	Underspecified requirements that are too abstract and allow for various interpretations	Communication flaws within the project team	Moving targets

The most outstanding property reflected in the results is that incomplete or hidden requirements are the most critical problem in each category. Neither agile nor plan-driven processes can overcome these problems. Considering agile practices welcome imcompleteness, this challenge might seem surprising and shows a possible conflict between the application of agile practices and the personal view on requirements of software engieneers.

In addition, we can see that in agile approaches the communication flaws with the customers and also within the project team itself are somewhat lower, but still belong to the most critical problems in the RE process. This is in clear contradiction to the advantages agile approaches claim for themselves.

The problem of insufficient time boxing is predominant in every working approach.

Another interesting observation is that the problem of moving targets occurs in the categories of exclusively agile and hybrid process approaches, but not in the top five problems in the category of exclusively plan-oriented approaches. One of the most frequently mentioned advantages for agile processes is that teams are resistant to changes, but the practitioners who use these practices still record that this is one of the top five problems in their projects.

Figure 4: The most critical problems that practitioners perceive in the comparison of different approaches to development

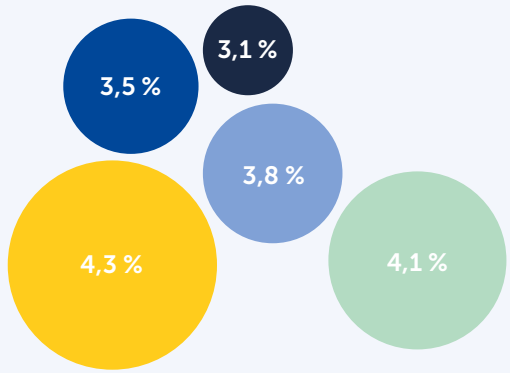
In *Figure 4* the most critical problems in RE are shown as they occur in practice. If we compare them with the different underlying working methods, an interesting picture emerges.

In the future, we will deepen our understanding about influences in the RE process, to understand which methodologies are adequate for different application areas. In order to be able to approach this challenge, we are dependent on your practical experience. If you are interested in working closely together such as in workshops or joint projects, please do not hesitate to contact us (see page 17).

Status Quo Requirements Engineering: Facts and Figures

Number of respondents: 488
(each respondent representing one team)

Top 5 causes for RE problems due to human factors



- Communication flaws between team and customer
- Lack of experience of RE team members
- Missing customer involvment
- Missing domain knowledge
- Customer does not know what he wants

Top five causes, problems and effects in practice

Top 5 causes for problems in RE

6,8% Lack of project management

5,5% Lack of time

4,3% Communication flaws between team and customer

4,1% Lack of experience of RE team members

3,8% Missing customer involvement

Top 5 problems in RE

11,3% Incomplete or hidden requests

9,7% Time boxing/Not enough time in general

9,6% Communication flaws between the project and customer

7,4% Communication flaws within the project team

7,0% Moving targets (changing goals, business processes and / or requirements)

Top 5 effects of problems in RE

9,2% Poor product quality

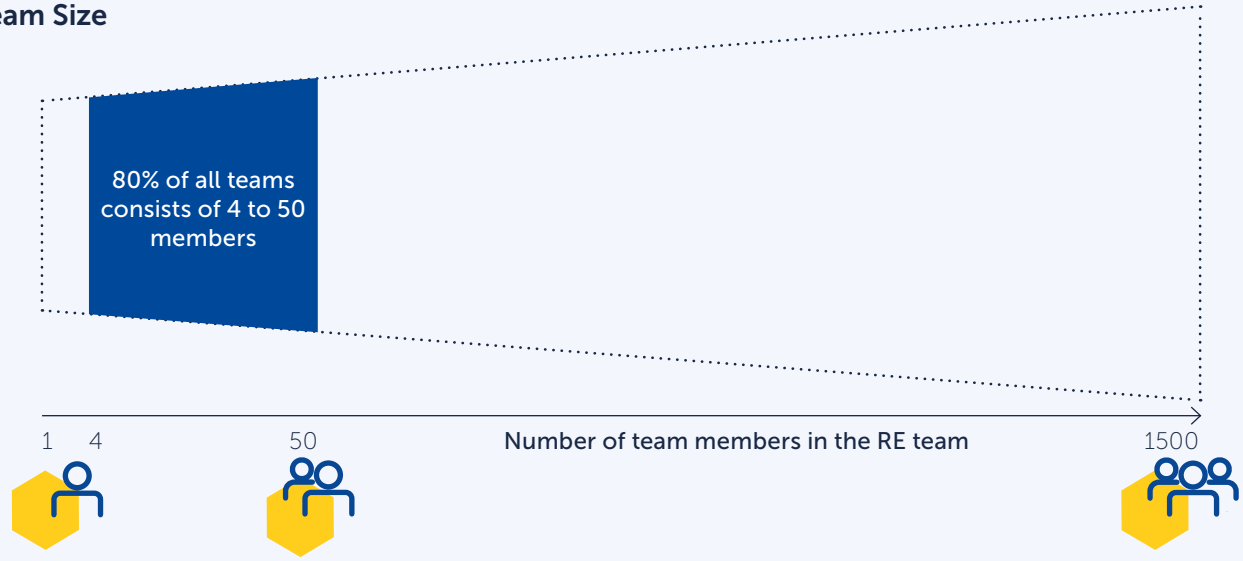
9,2% Inefficient development

9,1% Time overrun

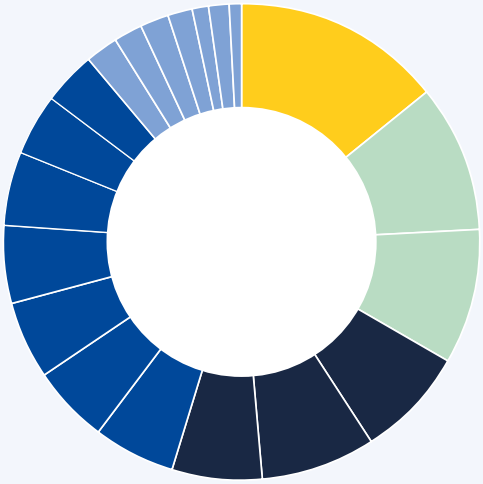
8,2% Difficulties in project management

7,7% Customer dissatisfaction

Team Size

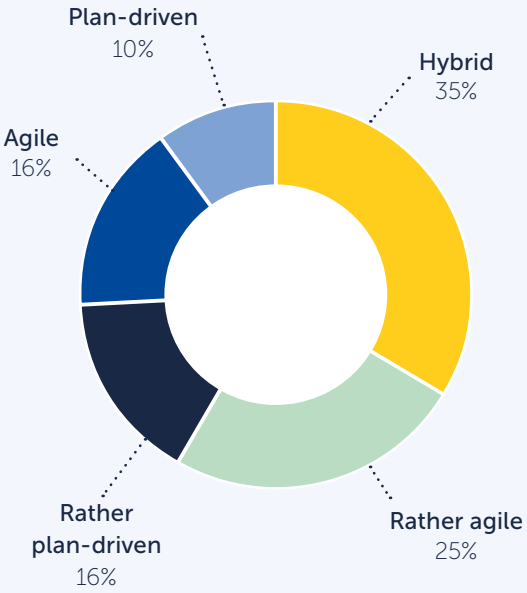


Distribution of sectors

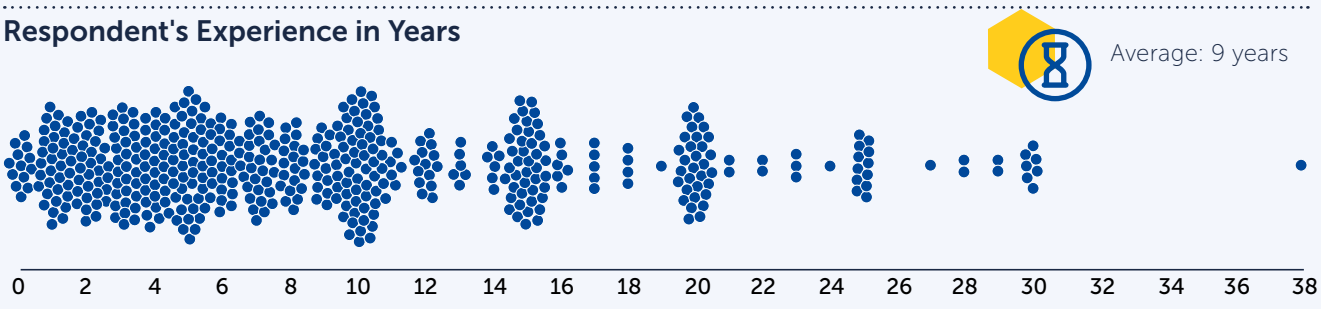


- **14,29%** Finance
- **10,3%** Public sector
- **9,02%** Healthcare
- **7,77%** e-Commerce
- **7,52%** Telecommunication
- **6,27%** Automotive
- **5,51%** Logistics
- **5,26%** Enterprise resource planning
- **5,26%** e-Government
- **5,26%** Manufacturing
- **5,01%** Energy
- **4,26%** Education
- **3,51%** Insurance
- **2,26%** Human resources
- **2,01%** Public transportation
- **2,01%** Security
- **1,50%** Railway
- **1,25%** Avionics
- **1,25%** Agriculture
- **0,75%** Games engineering

Development process



Respondent's Experience in Years



Elicitation Techniques

Interviews (69%)

Analysis of existing documents (60%)

Workshops and focus groups (57%)

Prototyping (48%)

Observations (35%)

Design Thinking / Lean Startup (20%)

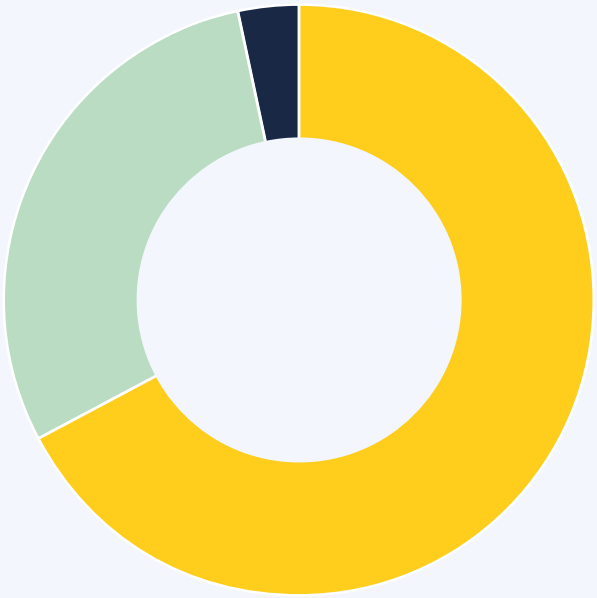
Risk analysis (20%)

(Requirements) Reuse database and guidelines (17%)

External experts (17%)

We do not elicit requirements (ourselves) (4%)

How do you elicit requirements



- **67%** We elicit and / or refine requirements in several iterations
- **30%** We elicit and / or refine requirements in a specifically dedicated project phase
- **3%** Other

Usage of Documentation

They are the basis for the implementation (82%)

They are source for tests (58%)

They are used customer acceptance (50%)

They are a reminder for further discussions with the customer, product owner (41%)

They are part of the contract (26%)

Requirements Alignment with testing

88% align their tests with requirements

Testers participate in requirements reviews.

Yes	No
44%	56%

We define acceptance criteria and tests for requirements.

Yes	No
56%	44%

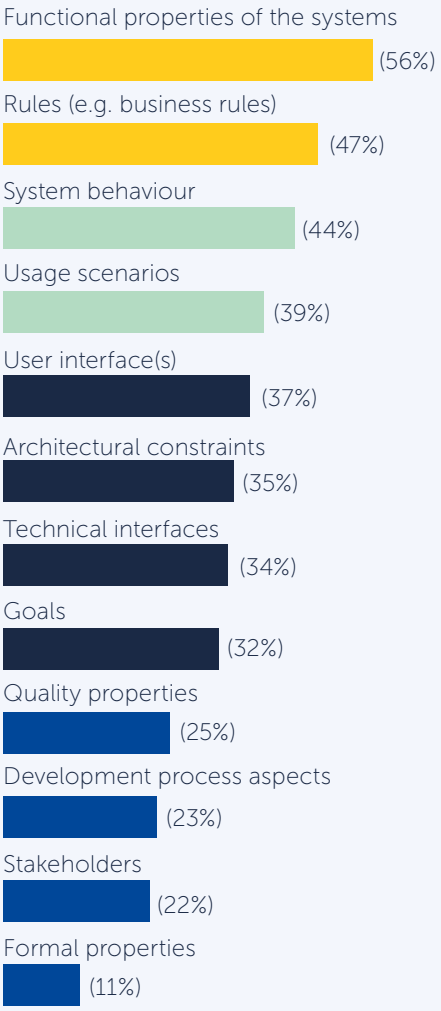
We derive tests from system models.

Yes	No
17%	83%

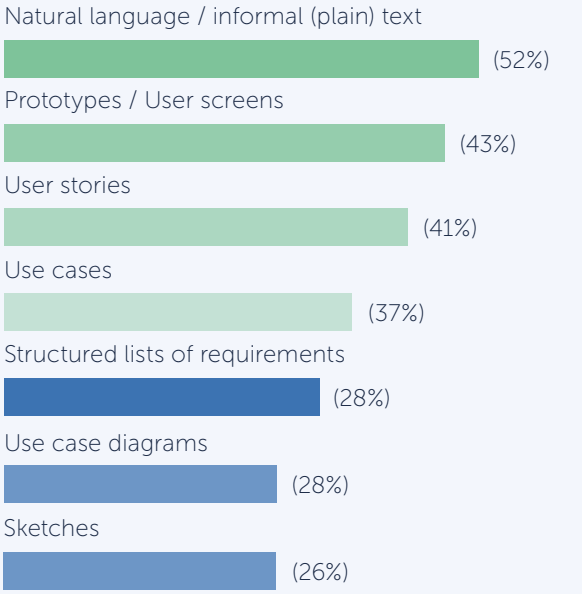
We check the coverage of requirements with tests.

Yes	No
46%	54%

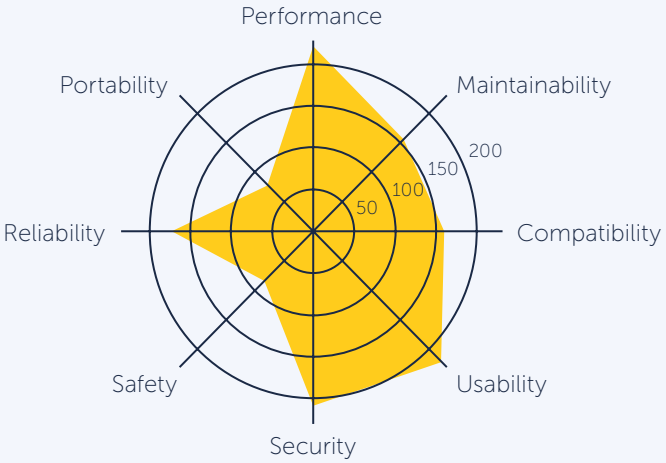
Contents of the documentation



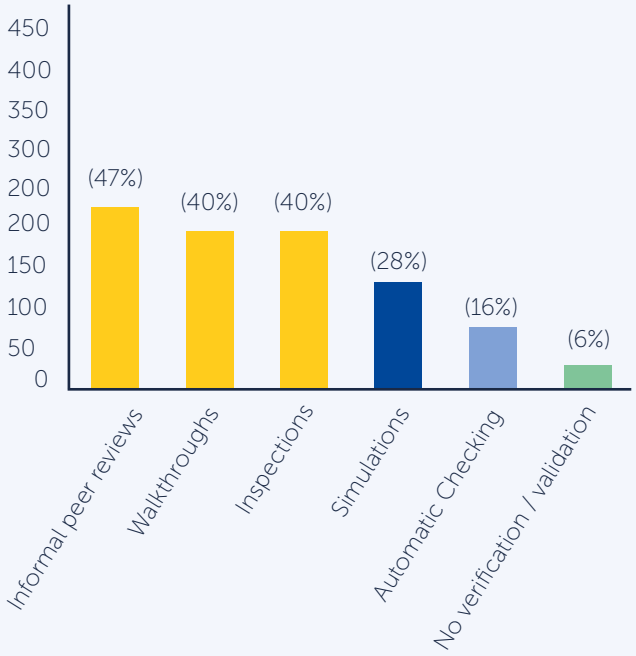
Documentation Techniques

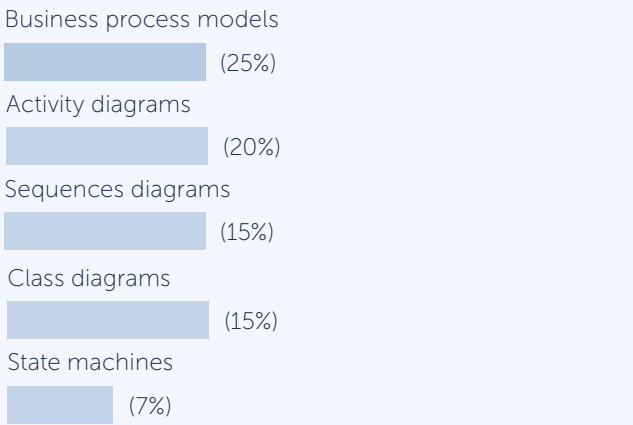


Relevance of quality requirements



Requirements validation techniques





Requirements Engineering Quick Check

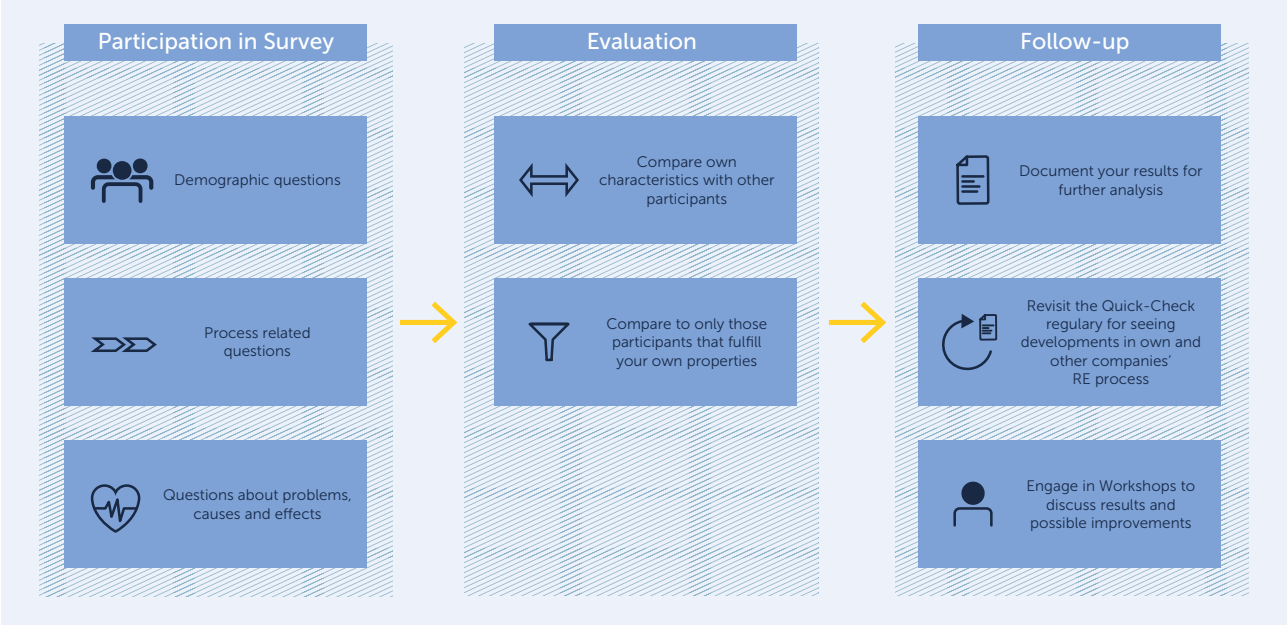


Figure 5: Overview of Requirements Engineering Quick Check

One question might now arise: "What can be done to improve Requirements Engineering in my environment?". This leads us to our Quick Check for Requirements Engineering, introduced next.

Currently, we can see in practice the application of many normative approaches that attempt to provide optimization plans by applying maturity models (e.g. Capability Maturity Model). The application of these models is associated with lengthy processes and a high resource investment. This is a particular challenge for organisations, as investing in these activities requires a lot of time and financial resources. In addition, in research, the established approaches showed difficulties adapting to company-specific circumstances.

To address these challenges, we are committed to providing a lightweight, easy-to-use, versatile tool that RE practitioners can use to obtain results quickly. With the **Requirements Engineering Quick Check** for practitioners can assess the status quo of their RE practices in their software development projects by benchmarking with other organisations. While the Quick Check will not render existing approaches irrelevant, it is rather intended to provide another light-weight and evidence-based alternative into the existing toolkit for RE practitioners.

The Quick Check consists of a ten-to-fifteen-minute survey that enquires about the following topics:

- **Demographic/contextual information** such as the industrial sector, team size, nature, and scope of the projects
- **Process related questions** that enquire about how requirements are gathered, elicited, documented, and applied
- **Critical problems, causes, and effects** you encounter while carrying out the projects

After completing the survey, you will be redirected to an evaluation page. You can compare your provided information, reflecting your status quo of RE and benchmarking your results to other participants.

Afterwards, it is immediately possible to analyse and raise questions such as:

- **"Do other practitioners face the same challenges as we do?"**
- **"How do others elicit requirements compared to us?"**
- **"Which quality attributes for requirements are of relevance?"**

A general view and the comparison with practitioners from the various environments can already offer some interesting insights into the subject. However, due to the already mentioned non-standardisability of RE, a comparison of this kind is possibly too vague, depending upon the area you are involved in. A software system developed for safety-critical systems or software for medical devices needs to include different properties compared to a software system for day-to-day business operations.

To enable meaningful benchmarking with other organizations based on categories that are relevant to you, and thus better tailored to your organization's unique characteristics and your current challenges or interests, we provide the possibility to specify the results through a selection of filters. For example, you might only be compared with organisations that operate in a similar sector, are of a similar size and also have a strong focus on safety requirements.

In this way, we aim to enable the comparability for your area of application to be tailored to your needs, if required. With these features we would like to fulfil our ambition to provide you with the best possible basis for assessing possible problems within your RE process in order to derive optimisation potentials.

If you wish to participate in the Quick Check,
please feel free to visit the website:
https://cce.fortiss.org/quick-checks/survey/requickcheck

Challenges in RE research and practice

We have seen that project success depends on how well the final results reflect the different needs of the various stakeholders. There are different approaches to software engineering in general and in RE in particular, and their suitability depends strongly on the context. With these practices employed, various challenges in RE exist in research and practice alike.

At the Requirements Engineering field of competence at fortiss, our research currently focuses on three topics around the plethora of challenges: Regulatory RE, Data-driven RE, and RE in human-centered environments. With an evidence-based, problem-oriented approach, we try to tackle contemporary challenges within research consortia as well as industry collaborations.

Regulatory Requirements Engineering

refers to an effective translation of regulatory norms and standards (such as GDPR or Medical Device Regulations) into requirements. The role of Regulatory RE and its relevance in today's software-intensive products and services is reflected in the need to comply with an ever-increasing plethora of regulations that range from preclude technical standards to ambiguous regulations on personal data processing or AI. This is especially challenging because regulatory requirements have a pervasive impact on all of the software development life cycle (SDLC).

This makes regulatory RE crucial to enable seamless and consistent compliance throughout the SDLC. We take a "practical approach" to regulatory RE. Our regulatory RE toolbox can help software engineers to conduct regulatory RE by compensating for some of the required legal knowledge. And in complex compliance settings in which extensive legal expertise is required, this toolbox can facilitate the engineering-legal interaction. The regulatory RE toolbox we offer is flexible enough to address different regulations and support different regulatory RE tasks in a comprehensive way for better compliance and engineering results.

Data-driven Requirements Engineering

reflects a rather new development within the software engineering domain. With an increased availability of data as well as increased use of software and systems labeld as "AI", the tasks associated to RE change. Complex RE situations demand the elicitation of requirements from different types of requirements sources. In particular, in this research stream we aim at understanding how to elicit, classify, prioritize, and document requirements for AI-centric (and ML-enabled) software systems. What are exactly non-functional requirements for such systems? How can we specify them in an unambiguous manner and how can we assure their quality? How can we test them? These questions are already difficult to answer for more traditional software-intensive systems and pose new challenges for this new family of systems. Of particular interest to us is to guide engineers with multi-disciplinary backgrounds in handling such heterogenous requirements for such systems efficiently and in a seamless manner. We do this by elaborating a holistic artefact model to be used as a tailorable reference model.

Requirements Engineering in human centric environments.

Like no other discipline in Software and Systems engineering, RE is highly dependent on human factors. These include, if not limited to, expectations and perceptions when communicating requirements, skills, expertise, and individual preferences influencing the choice of models and description techniques when specifying requirements. Humans also tend to mix up thinking about the problem and thinking about the solution and therefore focus on a particular solution already in the design process without considering all options. Those human factors render how RE can and shall be carried out as something highly unique to individual projects, as described in the deep dives. In our competence field, we look into different facets of these human factors to grasp the heterogenous challenges in-depth and develop possible mitigation strategies. Strongly related to these challenges – sometimes even part of the mitigation strategies – are methods for requirements engineering in human-centered environments. These mainly refelect methods for value-driven or value-oriented RE. In our case, we focus on the use and integration of creativity methods into the RE process as well as concepts and methods to understand problems from a user-driven perspective, such as by using Design Thinking methods, and their integration into model-based engineering methods. We elaborated an integrated view on Design Thinking for RE to promote empathy and creativity with the tools and concepts provided by Design Thinking.

If you are interested in more details on our research, or see similar challenges in your organization, come and talk to us. We are not only interested in your perspective - such as on these three topics - in general and also in their role in your organisation, but are also explicitly open to working on joint projects related to them.

Contacts

Prof. Dr. Daniel Mendez

mendez@fortiss.org
+49 (89) 3603522 168

Head of competence field & Professor at BTH
Senior Scientist and Head of the Competence Field Requirements Engineering at fortiss and Full Professor at the Software Engineering Research Lab of the Blekinge Institute of Technology, Sweden.

Parisa Elahidoost

elahidoost@fortiss.org
+49 (89) 3603522 428

Researcher & PhD Student
Parisa's research focus is the development of a toolsupported approach for the automatic extraction and compliance checking of regulatory requirements from legal texts. She is a research associate at fortiss GmbH and currently enrolled at the BTH's graduate program.

Anton Luckhardt

luckhardt@fortiss.org
+49 (89) 3603522 213

Deputy head of competence field & Researcher
Anton investigates human factors and problems in software development with particular focus on cultural dimensions. He is further leading the tool development at the lab with a particular focus on establish an automated platform for assessing and visualising industrial RE practices and problems.

Oleksandr Kosenkov

kosenkov@fortiss.org
+49 (89) 3603522 195

Researcher & PhD Student
Oleksandr is investigating the area of regulatory Requirements Engineering with a particular focus on establishing an artefact model to guide the elaboration of requirements from legal contexts. A domain of interest is the one of public service media platforms. He is a research associate at fortiss GmbH and currently enrolled at the BTH's graduate program.

Prof. Dr. Tony Gorschek

gorschek@fortiss.org
+49 (89) 3603522 251

Researcher & Professor at BTH
Senior Scientist at the research division Requirements Engineering at fortiss GmbH and Full Professor at the Software Engineering Research Lab of the Blekinge Institute of Technology, Sweden.

Dr. Jannik Fischbach

fischbach@fortiss.org
+49 (89) 3603522 455

Postdoctoral Researcher
Jannik works as a postdoctoral researcher at the research division Requirements Engineering at fortiss GmbH and as a consultant at Netlight GmbH. His research focuses on applying LP methods to support developers in implementing software intensive systems.

Imprint

Publisher
fortiss GmbH
Guerickestraße 25
80805 Munich

Layout
fortiss GmbH

Print
viaprinto GmbH & Co. KG

ISSN Print
2699-1217

ISSN Online
2700-2977

1st Issue:
April 2023

Photo credits:
Title: AdobeStock © Dmitry
Page 4: AdobeStock © nongkran_ch
Page 7: AdobeStock © monsitj
Page 18: © fortiss

Find here more
fortiss Whitepaper

fortiss is the Free State of Bavaria research institute for software-intensive systems based in Munich. The institute collaborates on research, development and transfer projects together with universities and technology companies in Bavaria and other parts of Germany, as well as across Europe. The research activities focus on state-of-the-art methods, techniques and tools used in Software & Systems-, AI- and IoT-Engineering and their application with cognitive cyber-physical systems.

fortiss is legally structured as a non-profit limited liability company (GmbH). The shareholders are the Free State of Bavaria (majority shareholder) and the Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.

fortiss GmbH
Guerickestraße 25
80805 München
Deutschland
www.fortiss.org
Tel: +49 89 3603522 0
E-Mail: info@fortiss.org

fortiss