Requirements Eng (2012) 17:57–78
DOI 10.1007/s00766-011-0144-x

REFSQ 2011

Industry needs and research directions in requirements engineering for embedded systems

Ernst Sikora · Bastian Tenbergen · Klaus Pohl

Received: 15 August 2011 / Accepted: 12 December 2011 / Published online: 29 December 2011

Abstract The industry has a strong demand for sophisticated requirements engineering (RE) methods in order to manage the high complexity of requirements specifications for software-intensive embedded systems and ensure a high requirements quality. RE methods and techniques proposed by research are only slowly adopted by the industry. An important step to improve the adoption of novel RE approaches is to gain a detailed understanding of the needs, expectations, and constraints that RE approaches must satisfy. We have conducted an industrial study to gain an in-depth understanding of practitioners' needs concerning RE research and method development. The study involved qualitative interviews as well as quantitative data collection by means of questionnaires. We report on the main results of our study related to five aspects of RE approaches: the use of requirements models, the support for high system complexity, quality assurance for requirements, the transition between RE and architecture design, and the interrelation of RE and safety engineering. Based on the results of the study, we draw conclusions for future RE research.

Keywords Industry needs · State of practice · Requirements models · Abstraction layers · Requirements engineering · Complexity management · Architectural design · Quality assurance · Safety engineering

This paper is a significantly revised and extended version of the contribution in [36].

E. Sikora
Automotive Safety Technologies GmbH, Sachsstrasse 16,
85080 Gaimersheim, Germany
e-mail: ernst.sikora@autosafety.de
URL: http://www.autosafety.de

B. Tenbergen (✉) · K. Pohl
paluno, The Ruhr Institute for Software Technology,
University of Duisburg-Essen, Gerlingstrasse 16,
45127 Essen, Germany
e-mail: bastian.tenbergen@paluno.uni-due.de
URL: http://www.paluno.de

K. Pohl
e-mail: klaus.pohl@paluno.uni-due.de

1 Introduction

The embedded systems domain encompasses branches such as automotive, avionics, and medical technology. The companies in the embedded systems domain face the challenge of bringing to market high-quality and innovative products in short time. This challenge is aggravated by the increasing system complexity caused, for instance, by the increasing number of functions and the increasing amount of interactions among different functions. For instance, an "automatic parking" function for passenger cars needs to interact with steering, drive train, braking, and sensor functions. High complexity increases the risk of undetected errors and deficiencies in the requirements specification that entail a significantly increased development effort in time and cost. Hence, an elaborated requirements engineering (RE) approach is crucial in order to meet time, cost, and quality goals in embedded systems development.

RE research has proposed sophisticated RE methods and techniques (in the following referred to as "RE approaches"), which aim to reduce development effort and improve quality, for instance, by means of model-based specification techniques. Yet, the adoption of such approaches in the industry is rather low. We partly attribute this to the insufficient consideration of industry needs and constraints during the development of the RE approaches, which makes these approaches difficult to integrate into industrial development processes. To improve the situation, RE researchers and method developers need a better understanding of practitioners' needs as well as of the constraints that RE approaches for embedded systems must account for.

Although a number of research contributions such as [1, 5, 33, 38] provide insights into the state of practice of RE, there is no recent, systematic attempt to reveal the essential needs and constraints that RE approaches should address. To reveal such essential needs and constraints, we have conducted an industrial study with the following research question:

> What are current industry needs concerning method support for requirements engineering in the embedded systems domain?

Since there is no common, clear-cut definition of which systems are considered as embedded systems, we outline some characteristic properties of the systems considered in our study:

- The purpose of the system is to control a physical process in real time. Therefore, the system needs to measure and control environment variables. The values of these variables must be acquired/updated at cycle times ranging between approximately 1 ms and 100 ms. The latency between an input event and the corresponding system reaction must usually be less than 100 ms.
- The system consists of one or several devices along with attached or built-in sensors and actuators. A control network interconnects the devices with one another as well as with other systems.
- The system needs to be realized through interdisciplinary collaboration of different technological domains such as analog electronics, digital electronics, mechanics, hydraulics, and software. Therefore, different departments and/or organizations are involved in the development. Furthermore, the development must be carried out simultaneously.
- Quality properties of the system such as availability, maintainability, safety, and security are vital to ensure the physical well-being of the system's users. In particular, functional safety requirements must be identified and enforced throughout the development process.

In order to gain meaningful insights for both, industry and research, we focused our study on aspects of RE approaches that are intensively discussed by researchers and practitioners:

- Use of natural language versus requirements models
- Support for high system complexity
- Quality assurance for requirements
- Transition between RE and architectural design
- Interrelation of RE and safety engineering

We have conducted the study with seven large, internationally operating companies in Germany from five different branches of the embedded systems domain (see Sect. 2.3). In order to gain a rich, in-depth understanding, we conducted interviews and collected additional data by means of questionnaires.

We consider the study results as particularly useful for the following audience:

- RE method developers who want to improve their approaches for easier integration into industrial development processes.
- RE tool developers who want to identify practitioners' needs and requirements for future RE tools.
- RE researchers who want to define research questions and set up hypotheses for an industrial study.

The paper is structured as follows: Sect. 2 outlines the design of our study. Section 3 presents the results. Section 4 draws conclusions for future RE research based on our findings. Section 5 discusses the validity of our results. Section 6 details selected aspects of the study based on experiences from an industrial development project. Section 7 relates our findings to the findings of previous studies and practice reports. Section 8 provides a summary and outlook.

2 Study design

Section 2.1 explains the focus of our investigation. Section 2.2 motivates the choice of the five investigation aspects stated in Sect. 1. Section 2.3 explains the study context in which the data were collected. Section 2.4 provides demographic data about the participating professionals. Section 2.5 outlines the investigative method and devices.

2.1 Investigative focus

Since most prior studies on the state of practice merely provide descriptive statistics of the techniques used in practice, their potential to inspire researchers to identify new research questions and conduct more in-depth studies is low. A major goal of our study was hence to identify new topics for future investigation. In other words, the purpose of this study was not to test hypotheses, but to explore what participants' opinions and attitudes are with regard to a carefully selected set of research questions (see Sect. 2.2). We, therefore, designed our study to be explorative in nature and chose a qualitative approach. Consequently, our study purposefully investigated a wider range of topics in order to provide a foundation for narrow-focused analytical investigations in the future. The results of our study will help researchers to purposefully identify research questions and define hypotheses to be tested by means of analytical studies.

2.2 Investigation aspects

The process of identifying and selecting the main investigation aspects of our study pursued two key goals:

- *Investigate topics that are highly relevant for RE research and practice alike*: To achieve this goal, we used literature research as well as discussions with RE practitioners to identify relevant investigation aspects.
- *Facilitate a high knowledge gain*: To achieve this goal, we selected aspects for which we perceived discrepancies between previous research results and industrial practice.

In the following, we outline the motivation for choosing each of the five investigation aspects stated in Sect. 1:

- *Natural language versus requirements models*: Many RE research endeavors focus on model-based RE approaches. In industry, engineers use model-based specification techniques during embedded systems design, for instance, to facilitate the simulation of control algorithms. Yet, as reported in various studies (e.g. [12, 23, 38]), requirements are specified predominantly using natural language. One goal of our study was hence to disclose the factors that presently inhibit a more intensive use of models in RE activities for embedded systems. Furthermore, the attitude of practitioners toward using models more intensively in RE is decisive for the future use of model-based RE approaches. As previous studies have so far neglected this aspect, we investigated practitioners' satisfaction with natural language requirements and their attitude toward an extended use of models in RE.
- *Support for high system complexity in RE*: Challenges in RE practice are often related to high system complexity (see e.g. [12, 19, 21, 38]). According to [12], the separation of different aspects by means of views is successfully applied in practice to deal with high system complexity. Among others, [38] suggest structuring the specification of a complex software-intensive system by means of a hierarchy of abstraction layers. Abstraction layers in the embedded systems domain facilitate, for instance, reasoning about system properties regardless of the details of specific implementation technologies such as electronics, mechanics, and software. Therefore, an important success factor of RE approaches is their support for defining requirements across different abstraction layers. We hence investigated RE practitioners' satisfaction and needs with regard to the support of RE approaches for abstraction layers.
- *Quality assurance for requirements*: The quality of requirements has a strong impact on the quality of the system. Since correcting errors introduced in the requirements specification later on leads to high cost and effort, it is a vital interest of the developing organization to detect requirements defects as early as possible. Previous work such as [5, 10, 12, 13] reports that major improvements are required to support the validation and verification of requirements, particularly for systems that must be highly dependable. Yet, to provide enhanced method support, method developers have to provide specialized approaches for different quality criteria. For instance, consistency checking demands different approaches than uncovering ambiguity or checking for testability. Researchers and method developers hence need to know which specific requirements quality properties demand an improved method support. To close this gap, we investigated which quality criteria for requirements are most challenging in embedded systems practice and hence require improved method support.
- *Transition between RE and architectural design*: Many RE textbooks demand a clear separation of requirements and architecture. However, in industry, this separation is not strictly adhered to. Some authors even report that a tight interrelation between RE and architectural design is inevitable in practice (see e.g. [1, 5]). This intertwining is particularly strong in embedded systems development, since, during system design and implementation, the initial requirements often turn out to be infeasible or too expensive to fulfill. Since previous studies have neglected practitioners' needs with regard to an improved support for tightly intertwined RE and design activities, we chose this topic as an investigation aspect of our study.
- *Interrelation of RE and safety engineering:*[1] As stated in Sect. 1, functional safety is a vital aspect of the development process of an embedded system. Defining safety goals at the system level and deriving safety requirements for subsystems and components is an essential part of the safety engineering process and is enforced by standards such as [15] and [17]. In contrast, research considers RE approaches and safety analysis mainly in isolation (see [3]). Our study hence

[1] This aspect was part of the original investigation reported on in [36]; however, it had not been included in the paper due to space limitations.

investigated in how far practitioners demand a stronger integration of RE and safety engineering approaches.

2.3 Study context

The study was conducted in 2009 with seven companies from five branches of the embedded systems domain: automation technology, automotive, avionics, medical technology, and energy technology. The companies were large, international equipment manufacturers as well as suppliers with branches in Germany. All companies operate on the international market. They produce both bespoke as well as market-driven products. Products developed by the participating companies are mainly safety-critical, software-intensive embedded systems designed for the world market. All seven companies are partners in the German Innovation Alliance SPES 2020 (Software Platform Embedded Systems, [16]).

2.4 Participants

The study participants were recruited from companies that participate in the SPES 2020 project, partly based on their company roles and partly by convenience sampling (see e.g. [4]). As the study required participants with a good overview of the needs related to RE and other development phases across different projects in their companies, only department leaders, research personnel, and internal project consultants were recruited to participate in the study. One inclusion criterion to participate was extensive hands-on experience with RE for embedded systems (see Sect. 2.5.1). In total, 17 company representatives participated in the study. The participants were not balanced across companies. However, when multiple representatives from the same company participated, the representatives were recruited from different departments or branches. Figure 1 shows the distribution of participants across all five branches. No effort was made to account for gender, age, or nationality in the study, as neither was deemed to have an impact on the participant's attitudes toward RE or embedded systems. However, it should be noted that by mere coincidence, only one participant was female.

2.5 Investigative devices and method

As the purpose of the study was to explore essential industry needs and thereby to identify novel research areas and challenges in the field of RE, the investigation was of qualitative nature and involved semi-structured interviews. However, we supplemented the interview data with quantitative data collected by means of questionnaires. Therefore, a combination of qualitative and quantitative data collection techniques was applied in our study. Three data acquisition devices were developed: a demographic questionnaire, an interview based on an interview guide, and a post-interview questionnaire. These devices are explained in more detail in the following subsections. All five investigation aspects outlined in Sect. 2.2 were addressed both in the interviews and in the questionnaires. The study was conducted in German; the questionnaire and interview guide items have been translated for the purpose of this paper such that the emphasis of the individual statements was preserved.

2.5.1 *Demographic questionnaire*

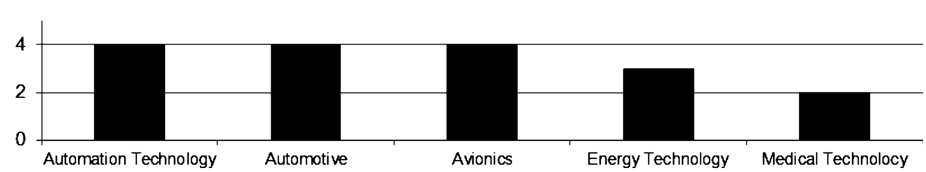
The demographic questionnaire consisted of 13 questions about the industrial context and the individual participants' professional backgrounds such as their experience with RE. This questionnaire was designed to verify that the participants meet the inclusion criteria of the study. 60% of the participants self-reported their experience with RE to be between 5 and 10 years, 20% reported more than 15 years of experience. Furthermore, 60% of the participants self-reported at least 5–10 years of experience with RE for embedded systems in their current or any previous job as shown in Fig. 2.

In addition, 90% of the participants self-reported their level of experience in RE as advanced or expert. Because our study aimed at revealing what practitioners' needs concerning RE research are, study participants were required to have long-lasting practical experience with RE in order to assess the current practice and trends therein. We have therefore chosen the participants' years of experience rather than "theoretical knowledge about RE" as the most suitable measure of the participants' qualification.

2.5.2 *Interview and interview guide*

The total time dedicated to each interview session was about 2 h. During the interviews, an interview guide was used to lead the interviewer and interviewees through the conversation. Strict adherence to the interview guide was not enforced, as the interviewer should be allowed to react flexibly to the participants' answers and investigate topics

Fig. 1 Participant distribution over the five application domains



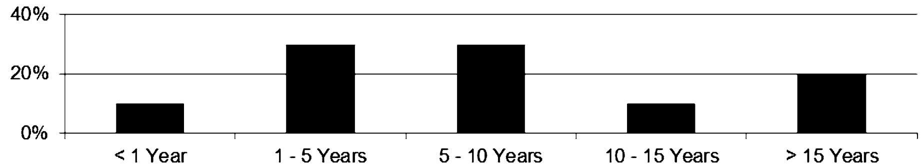


Fig. 2 Participant's years of experience with RE for embedded systems

emerging during the interview in more detail. Written protocols documented the participants' answers and were transcribed during the interviews. Each participant reviewed the protocol of his or her respective interview to ensure that the essence of the answers was gathered correctly and to clarify possible misunderstandings.

2.5.3 *Post-interview questionnaire*

The goal of the post-interview questionnaire was to gain quantitative data, supplementing the qualitative data from the interviews in order to provide additional evidence and support the interpretation of the qualitative data. The questionnaire contained 60 items. Forty-seven of these items contained predefined statements about RE practice and industrial needs to which the participants could express their approval or disapproval using five-point scales ("1: applies never/strongly disagree," "2: applies rarely/disagree," "3: applies sometimes/indifferent," "4: applies often/agree," and "5: applies always/strongly agree"; multiple answers were discouraged). The remaining items were questions to be answered according to predetermined answer choices (multiple answers were allowed). We deliberately used adverbs such as "predominantly" or "often" in our questions to make clear that we were interested in the typical case. We did not expect that an individual company representative can make a general statement that holds for the entire company. By using vague qualifiers, we aimed to avoid mediocrity in the data.

From the 17 participants, 10 answered the post-interview questionnaires.

3 Key results

This section presents the key findings of our study along with the corresponding evidence from interviews and questionnaires. For each investigation aspect, we first briefly list the key findings and then elaborate on the evidence for each finding. Section 4 draws conclusions for future research in RE resulting from our findings. Our findings are compared to findings of prior studies in Sect. 7.

3.1 Use of natural language versus requirements models

A significant number of RE approaches that have been developed focus on model-based RE. In contrast, requirements are documented in practice predominantly using natural language. Therefore, one investigation aspect of the study was in how far participants regard model-based RE approaches as beneficial (see Sect. 2.2).

3.1.1 *Key findings*

Finding 1.1: The fact that natural language is the dominant documentation format for requirements does not imply that natural language is considered a satisfactory specification technique. In fact, many embedded systems practitioners are dissatisfied with using natural language for requirements specification.

Finding 1.2: Although models are typically not included in the final requirements documents, intensive use of models in RE is considered beneficial and necessary.

Finding 1.3: Methodical uncertainties caused by a lack of method support inhibit more intensive use of models for documenting requirements. These uncertainties relate to how models can specify legally binding requirements, how requirements models can be distinguished from design models, and how safety standards can be satisfied if requirements are specified using models.

Finding 1.4: Domain-specific models and UML/SysML [27, 28] models are the most commonly used types of models in RE for embedded systems. In contrast, goal models are rarely used, which can be attributed to the fact that requirements engineers in the embedded systems domain are not familiar with goal-oriented RE approaches.

3.1.2 *Evidence*

Evidence for finding 1.1: The interviews and a questionnaire item stating "requirements are available to us predominantly as text documents" show that natural language is the dominant documentation format for requirements (see Fig. 3).

Regarding the satisfaction with natural language use, the interviews revealed that participants consider it tedious and error prone to deal with large bodies of natural language requirements. For example, interviewees frequently noted that checking the consistency of the natural language requirements specification must be done manually by means of inspections, which leads to an enormous effort.

One of the participants pointed out that, since embedded systems perform control tasks, the requirements specification must precisely define the physical process to be

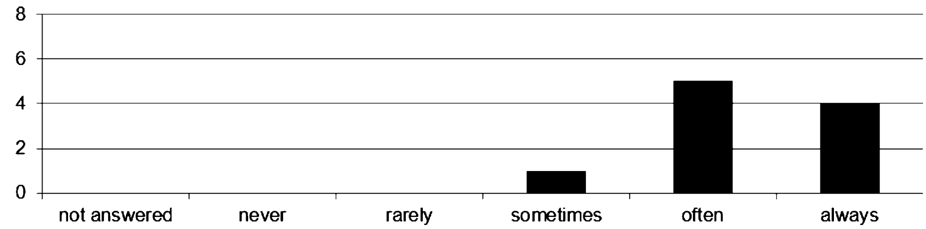


Fig. 3 Participants' answers for the statement "Requirements are available to us predominantly as text documents"

controlled as well as the requirements for the controller. According to the participant, there is a need for approaches that support the specification of controller requirements because controller requirements defined using natural language allow for too many interpretations whereas controller specifications written as pseudo-code preempt the controller design.

The participants' low satisfaction with the use of natural language for specifying requirements is also shown by a questionnaire item (see Fig. 4). Half of the participants agreed or strongly agreed with the statement that using natural language to document requirements is not satisfactory. Only one participant expressed disagreement.

Evidence for finding 1.2: Most participants expressed a rather low intensity of model use in typical requirements specifications as can be seen in Fig. 5 (black columns). However, the interviews revealed that models are frequently used as a supportive means during the RE process.

More than half of the participants consider that the use of requirements models is necessary for their projects (see Fig. 6). It should be noted that a negative formulation was used for the corresponding questionnaire item. The participants who express (strong) disagreement are thus in favor of model use. The participants stated different reasons why they consider requirements models necessary. For example, one participant mentioned that executable models that amend natural language documentation may ease early validation of requirements and allow for a simplified certification process. In addition, most participants agree that models help understand complex requirements better as confirmed by the data in Fig. 5 (light gray columns).

However, only three participants indicated in the questionnaire that models often or always help managing high system complexity as shown in Fig. 7.

Evidence for finding 1.3: Several participants stated that a major obstacle for using requirements models more intensively is the lack of appropriate method guidance and, in addition, noted the lack of (commercial) tool support that would allow integrating requirements models in a seamless development process. The dissatisfaction with method and tool support for model use in RE is confirmed by the questionnaire results (see Fig. 8).

The lack of method guidance and tool support leads to uncertainty about how models should be used in the RE process. For instance, the participants noted methodical uncertainties regarding how models can be used for documenting legally binding requirements. Due to these uncertainties, models are not included in the final requirements documents that are provided, for instance, to suppliers. According to the participants, a major objection against the use of models as a basis for contracts is that a model can be interpreted in different ways. Moreover, it was mentioned that it is difficult to extract the contractual requirements from a model. Furthermore, the participants expressed that uncertainties concerning the satisfaction of safety standards such as RTCA DO 178B [32] are obstacles for more intensive model use. In addition, several participants considered models as a result of a design activity and not of an RE activity (see Sect. 3.4 for more details). Only one participant was aware that functional requirements can be transferred into models without compounding requirements with design decisions.

Evidence for finding 1.4: It became obvious during the interviews that, among different model types, domain-specific models are of particular interest in the RE process of an embedded system. These models typically stem from disciplines such as mechanic engineering, electrical engineering, or control engineering. They are used to describe structural aspects of the system and/or its environment such as a power plant, a pumping station, a vehicle, or an airplane. Furthermore, the interviewees from the avionics domain reported that engineers in their companies are

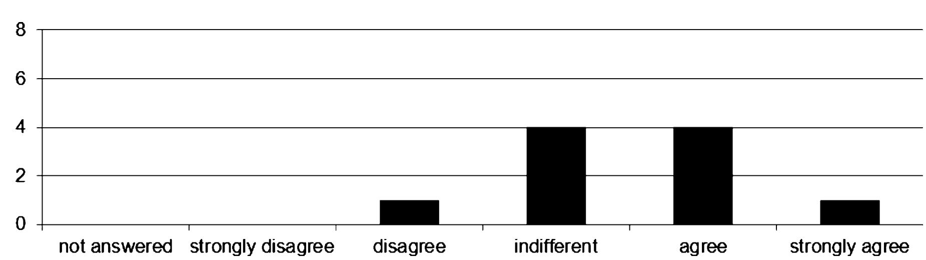


Fig. 4 Participants' answers for the statement "The specification of requirements by means of natural language is often not satisfying"

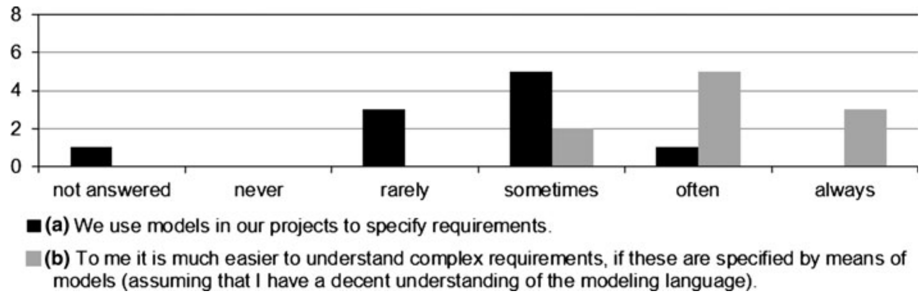


Fig. 5 Participants' answers for two questionnaire items concerning the benefits of model use

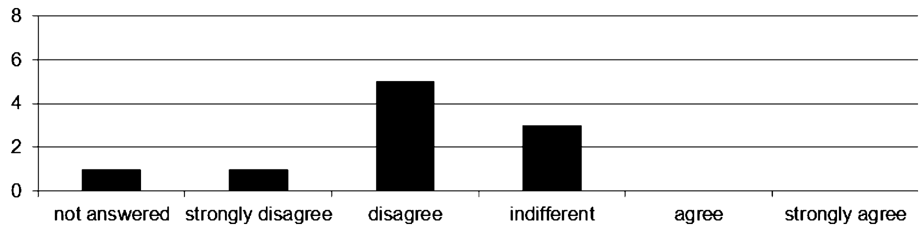


Fig. 6 Participants' answers for the statement "Intensively using models in our project is not necessary"

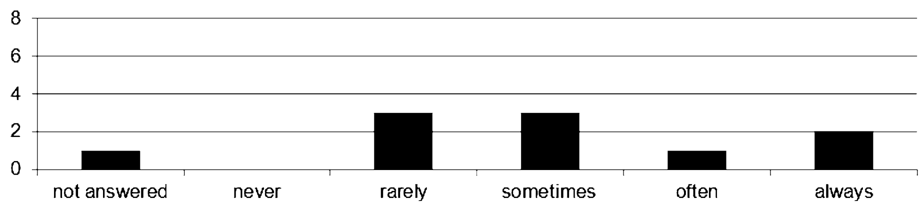


Fig. 7 Participants' answers for the statement "Without using models to specify system requirements, the complexity of the systems that are being developed cannot be handled"

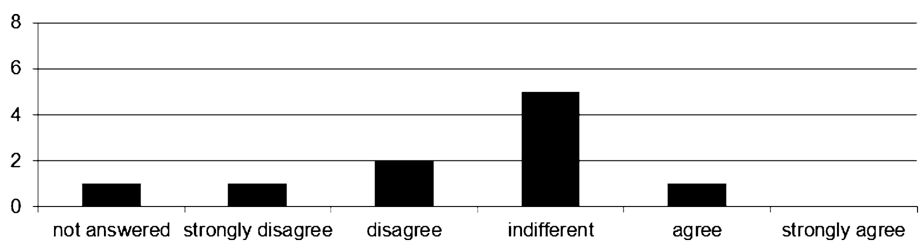


Fig. 8 Participants' answers for the statement "The support we have in our company/in our department to specify models in RE is completely satisfactory"

trained in using UML/SysML [30] as the use of a standardized modeling language is seen as an important advantage for systems engineering.

In the questionnaire, the participants stated which types of models they regard as beneficial for RE (see Fig. 9, black columns) and which modeling languages the participants have used at least once before to specify requirements (see Fig. 9, light gray columns).

Interestingly, while goal-oriented RE plays an important role in current research [37] and appears to be a sound foundation for a continuous RE process [34], goal modeling is only of marginal importance in current practice. This can partially be explained by the fact that practitioners are largely unaware of goal-oriented RE approaches (see Fig. 10), and consequently, goal models are rarely used as well. In addition, participants reported during the interviews that even when a model type has a potential benefit for the development project, it tends to be used less frequently, if its use is not mandated, e.g. by project constraints or intra-organizational regulations.

3.2 Support for high system complexity

High system complexity is one of the greatest challenges for system development. Sophisticated approaches are needed to manage this complexity (see Sect. 2.2). RE approaches proposed by research do not always account for complex systems. A goal of our study was hence to elicit the actual needs of RE practitioners related to the support for managing high system complexity in the embedded systems domain.

3.2.1 Key findings

Finding 2.1: Existing RE methods are insufficient for handling requirements for complex embedded systems. The lack of appropriate method support leads to an

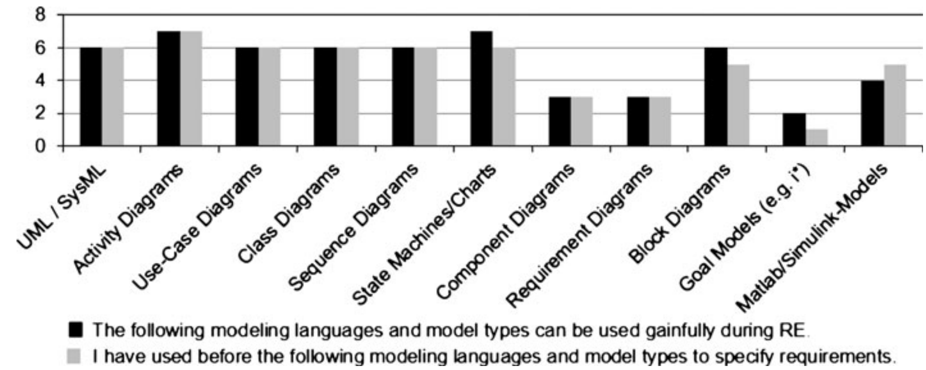


Fig. 9 Participants' answers for two questionnaire items investigating the use of concrete modeling languages

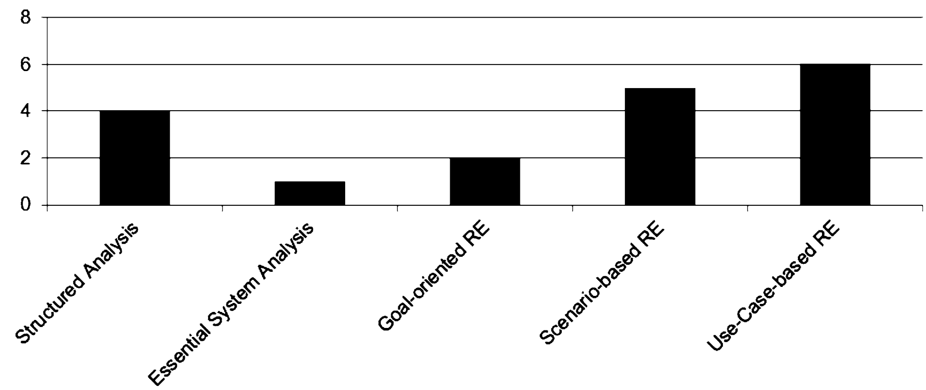


Fig. 10 Participants' answers for the statement "I am familiar with the following RE approaches"

enormous effort for ensuring requirements consistency across different subsystems and engineering domains such as mechanics, electronics, and software.

Finding 2.2: Abstraction layers are widely used in the embedded systems domain in order to manage high system complexity. Therein, the specific abstraction layers used are different for every branch of the embedded systems domain, company-specific, and in some cases tailored to the project or system.

Finding 2.3: Since the current use of abstraction layers in RE is often inconsistent, improved method guidance for specifying requirements across different abstraction layers of an embedded system is urgently needed.

3.2.2 Evidence

Evidence for finding 2.1: In the questionnaire, the study participants expressed a low satisfaction with the support of existing RE methods for handling complexity, as depicted in Fig. 11.

During the interviews, most participants revealed that this dissatisfaction is partly caused by the difficulties that arise when managing large amounts of interrelated requirements. A complex embedded system typically realizes a large number of interdependent functions and is structured into several interrelated subsystems. Technically, each subsystem consists of mechanic components, electronic components, communication networks, and software components. Requirements need to be specified for the system itself, for its functions and subsystems, as well as for each component. Moreover, the requirements need to be consistent with each other. Over the course of the development, additionally elicited requirements may require changing already existing requirements which in turn may affect a number of subsystems and components. A participant from the automation domain stated that carrying out these requirements changes consistently across the different technical domains of a plant—e.g. across electrics, electronics, mechanics, and software—is, in fact, the key challenge.

During the interviews, the example of a steel-mill (automation technology branch) was considered to illustrate this challenge: if a requirement for the plant changes to incorporate a 5% increase in production speed, requirements concerning all components may change as well. For example, higher production speeds may increase the load on mechanical components, increase power demands for the conveyer belt motor, and necessitate changes in the software that controls the assembly.

Evidence for finding 2.2: In the questionnaire, six participants reported that their projects often or always involve the specification of requirements at different abstraction layers (see Fig. 12).

The use of abstraction layers was investigated in more detail in the interviews. Table 1 provides examples of abstraction layers in three branches based on the interview results.

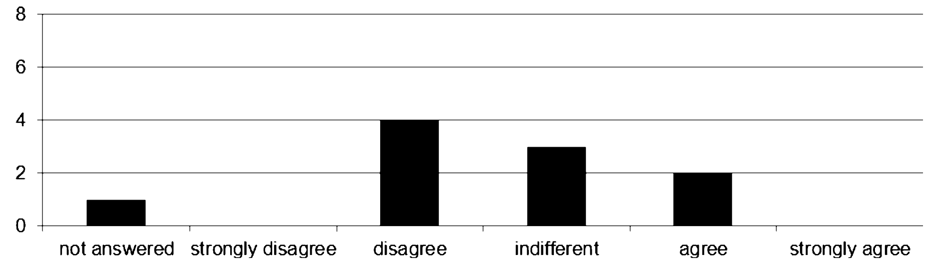


Fig. 11 Participants' answers for the statement "Existing requirements engineering methods are capable of dealing with the complexity of the systems we develop"

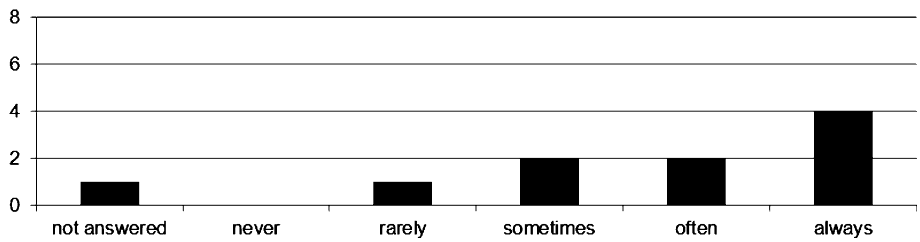


Fig. 12 Participants' answers for the statement "In our company/in our department, requirements are strictly managed on different abstraction layers (e.g. market/product requirements, system requirements, subsystem requirements, or hardware/software requirements)"

Table 1 Examples of abstraction layers in three different branches of the embedded systems domain as reported by the study participants

Branch of embedded systems domain	Example of abstraction layers	Explanation of abstraction layers
Avionics	Top level aircraft requirements	Requirements for the entire aircraft
	Top level cabin/cockpit requirements	Requirements pertaining to the cabin, cockpit, cargo hold, etc.
	Multi-system requirements	Requirements for multiple systems belonging to the same functional group, such as flight control or climate control
	System requirements	Requirements for individual systems such as a flap control computer
Automation technology	Plant level	Requirements for the entire plant, e.g. a roller mill facility. At this level, embedded systems are considered as "systems of systems"
	Integrated systems level	Requirements for parts of the plant installation such as assembly cells or production lines
	Machine level	Requirements pertaining to systems within assembly cells, such as production robots, CNC-machines, or rolling stands
	Automation system level	Requirements for individual systems that belong to a machine, such as programmable logic controllers, actuators, sensors, or other devices
Energy technology	Energy grid level	Requirements pertaining to the entire energy grid, consisting of power plants, consumers, and routing facilities
	Power plant level	Requirements for power production facilities
	Producer/consumer device level	Requirements for devices installed in producer and consumer households

In some cases, abstraction layers are formally imposed. For instance, standards such as [2, 32] define several abstraction layers for requirements. However, the interviews revealed that, to a large extent, the abstraction layers used are not standardized and highly influenced by the specific branch, e.g., automation, automotive, avionics. For instance, different abstraction layers are used for specifying a steel-mill than for an aircraft. Even within a specific branch, each company uses different abstraction layers. For instance, a car manufacturer uses different abstraction layers than a supplier. Furthermore, some participants reported that the number of abstraction layers is not fixed across different projects but is influenced by the system complexity, i.e. to account for higher system complexity, additional abstraction layers may be introduced in the project.

Evidence for finding 2.3: In the questionnaire, nearly all participants expressed a strong need for method support for refining requirements across abstraction layers, as depicted in Fig. 13:

- As the black columns in Fig. 13 indicate, most participants agree that a systematic approach for refining, e.g. system requirements into component requirements would significantly ease development.
- As the light gray columns in Fig. 13 indicate, 80% of the participants agreed that a methodical approach regarding the transition between abstraction layers would significantly reduce development effort.

In the interviews, the participants reported substantial confusion regarding the proper use of abstraction layers in RE. According to the interviewees, the available method guidance does not provide sufficient guidelines regarding, for instance:

- What kind of requirements should be defined at each abstraction layer?
- Which level of detail is appropriate at which abstraction layer?
- How can a hierarchy of abstraction layers be tailored for a system or project?
- How can requirement consistency be maintained across abstraction layers?

The questionnaire results are consistent with the interview results: the majority of participants consider that maintaining requirements consistency across multiple abstraction layers is difficult to accomplish as shown in Fig. 14 (note the negative formulation of the questionnaire item). The need for improved method support for assuring requirements quality, in general (i.e. not specifically related to abstraction layers), is further discussed in Sect. 3.3.

The interviews further revealed that, as a consequence of the insufficient guidance, engineers mostly decide how to use abstraction layers during requirements specification based on personal intuition and experience. This leads to an inconsistent overall use of abstraction layers in the projects and an inconsistent level of detail within each abstraction layer.

3.3 Quality assurance for requirements

Research puts a strong emphasis on the use of formal methods for quality assurance. In practice, the specialized training and the high effort needed for applying formal methods limit their application in RE. A goal of our study was therefore to elicit how practitioners perform quality assurance during RE and what their needs are for systematic quality assurance approaches.

3.3.1 *Key findings*

Finding 3.1: Reviews are the dominant technique in embedded systems practice for checking requirements against predefined quality criteria. To ensure that the right requirements are defined, simulation and prototyping are applied.

Finding 3.2: The industry has a strong need for improved approaches for requirements quality assurance, particularly concerning requirements consistency, traceability, and testability. The use of requirements models is seen as a promising way to achieve the desired improvements.

Finding 3.3: Requirements quality assurance for embedded systems is strongly influenced by safety regulations.

3.3.2 *Evidence*

Evidence for finding 3.1: In the interviews, the participants reported reviews to be the dominant technique for checking requirements documents against predefined quality criteria. Furthermore, the participants reported that a significant effort is made to define the right requirements early in the project. Therefore, prototyping and simulation activities are performed.

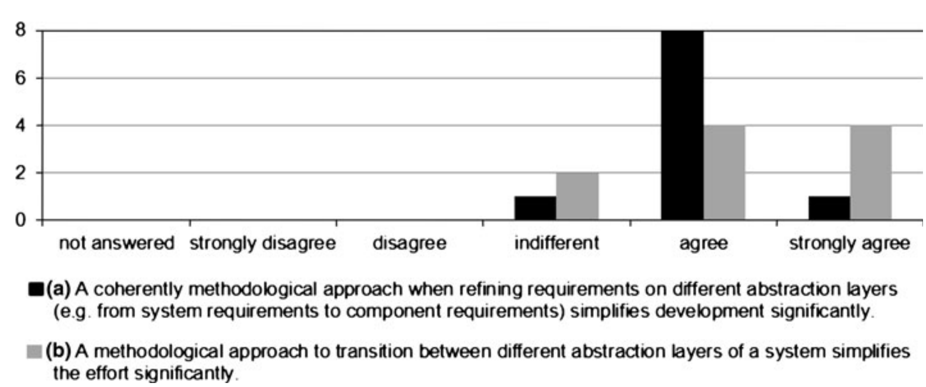


Fig. 13 Participants' answers for two questionnaire statements concerning the use of abstraction layers during RE

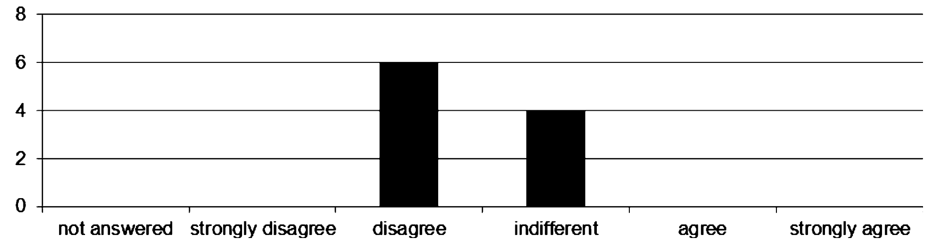


Fig. 14 Participants' answers for the statement "Maintaining consistency between requirements on different abstraction layers is not a major challenge in our projects"

In the questionnaire, nine out of ten participants reported that reviews or inspections are regularly conducted in their respective company for checking requirements quality (see Fig. 15). In contrast, only one participant reported the use of automated consistency checking of requirements. As further shown in Fig. 15, four participants reported that formal verification is used in their projects. However, as the participants' conception of formal verification was not in the focus of the investigation, further studies are needed to investigate, for instance, which specific verification techniques are applied.

Evidence for finding 3.2: As our goal was to determine whether RE research is needed to provide improved RE approaches, we asked the participants about their satisfaction with existing quality assurance techniques for requirements. The participants judged existing methodical support as only partly satisfactory as shown in Fig. 16.

Several participants stated in the interviews that part of the reason for the poor satisfaction is the high effort that must be spent for ensuring, for instance, requirements traceability and consistency using manual techniques such as reviews. The participants expressed a strong desire for automated validation and verification, however, also indicated that current RE activities and documentation practice do not meet the prerequisites for applying automated techniques (see Sect. 3.1). In the questionnaire, the participants expressed a particularly strong need for improved method support for the quality criteria consistency, testability, and traceability. These three criteria were stated by 80% to 100% of the participants. The results for all requirements quality criteria considered in the study are shown in Fig. 17.

The questionnaire data further indicate that the desire to use requirements models more extensively (see Sect. 3.1) is partly motivated by the prospect that model-based RE will ease requirements validation and verification. 70% of the participants agree that using models intensively during RE would improve validation and verification of requirements tremendously, only 20% distinctively disagree as shown in Fig. 18.

Evidence for finding 3.3: As revealed by the interviews, the participants regard high requirements quality as particularly important in the case of safety-critical systems or functions since, for these, the level of quality assurance to be achieved and, in part, the techniques to be applied are regulated by safety standards (e.g. ISO 26262 [17], RTCA DO 178B [32]). An example was given by a participant from the avionics domain: Satisfying the safety regulations for safety-critical flight software means that each element in the code must be justified by a low-level requirement, and each low-level requirement must be justified by a high-level requirement. A more in-depth discussion of the findings of the study concerning safety is given in Sect. 3.5.

3.4 Transition between RE and architecture design

Many existing methods assume a clear separation between RE and architecture design, i.e. that RE activities and artifacts are clearly separated from architecture design activities and artifacts. However, as sketched in Sect. 2.2, observations indicate a less clear separation between RE and architecture design or even a close intertwining of RE and architecture design in practice (see [25, 26]). One goal of our study was hence to obtain a better understanding of the actual relationship between RE and architecture design in practice.

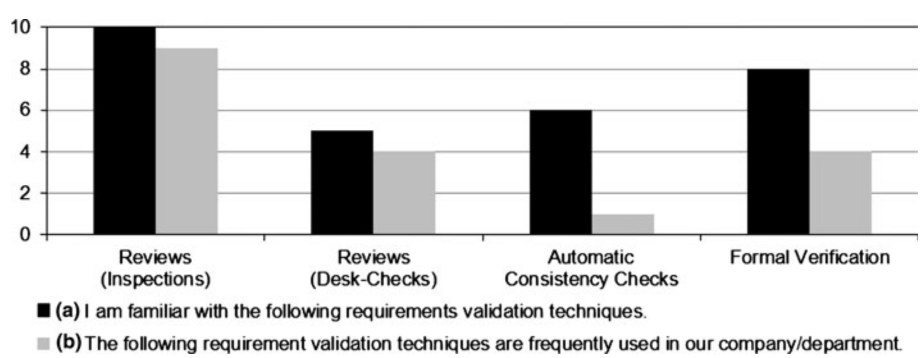


Fig. 15 Participants' answers for two questionnaire statements concerning the familiarity and use of requirements validation techniques

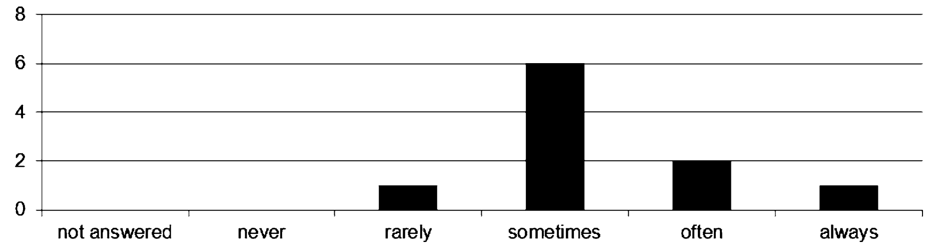


Fig. 16 Participants' answers for the statement "The method support we have for validation and verification of requirements is completely sufficient"

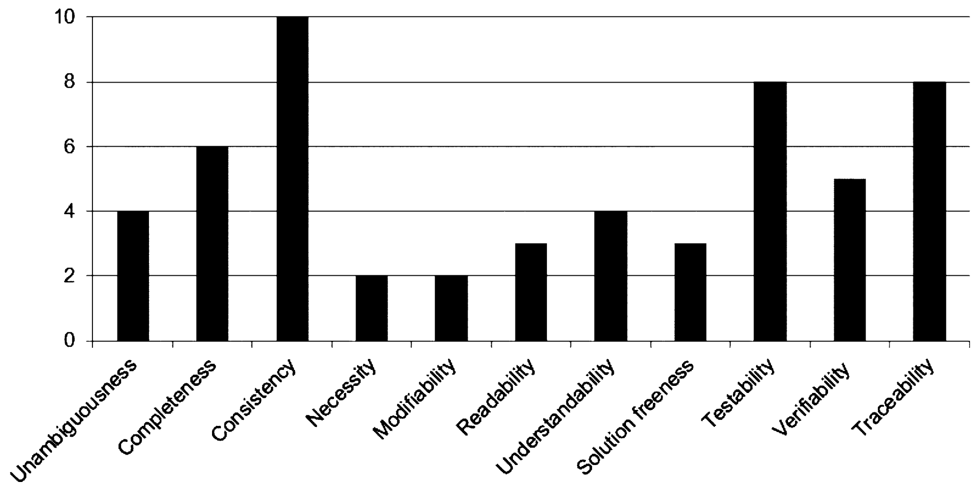


Fig. 17 Participants' answers for the question "For which requirement quality properties do you expect a significantly improved methodical/tool support?" Multiple answers per participant were possible

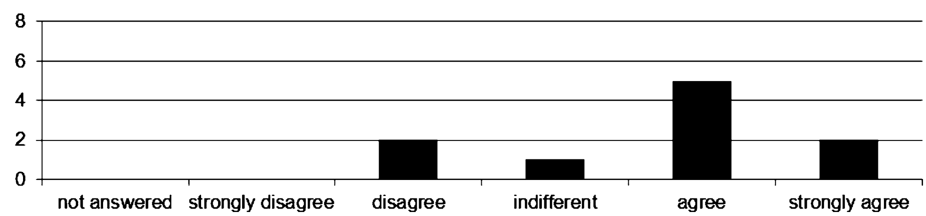


Fig. 18 Participants' answers for the statement "Using models during requirements engineering would significantly improve requirement validation and verification"

3.4.1 Key findings

Finding 4.1: While process models indicate a clear separation between RE and architecture design, for embedded systems, there is no such clear separation in actual development practice.

Finding 4.2: Distinguishing between requirements and design becomes more difficult, when requirements are specified using models.

Finding 4.3: Among the different possible approaches for supporting the transition between requirements and design, practitioners mostly ask for approaches supporting traceability between requirements and design. Approaches automating the transition from requirements to design could not be identified as an essential need.

3.4.2 Evidence

Evidence for finding 4.1: The participants' responses during the interviews indicate that the process models and guidelines used in the companies assume a clear distinction between RE and architecture design activities and demand requirements and design artifacts to be clearly separated from each other. However, when asked about the actual development practice, the interviewees indicated strong interactions between RE and architecture design and hence provided evidence for the tight intertwining of RE and architecture design activities. The post-interview questionnaire confirmed this result. Only two participants stated that a clear separation between RE and design is maintained often or always in their projects (see Fig. 19).

Evidence for finding 4.2: Nearly all interviewees expressed a major confusion regarding how requirements models can be separated from design models. Furthermore, since partly the same modeling languages are used for requirements models as for design models (see Sect. 3.1), the confusion for practitioners is further increased.

We further investigated in the interviews how the participants differentiate between requirements and design. All interviewees described the separation between requirements and design either as the distinction between "what?" and "how?" or as the distinction between problem definition and solution finding. The participants were not aware that different roles in the development process

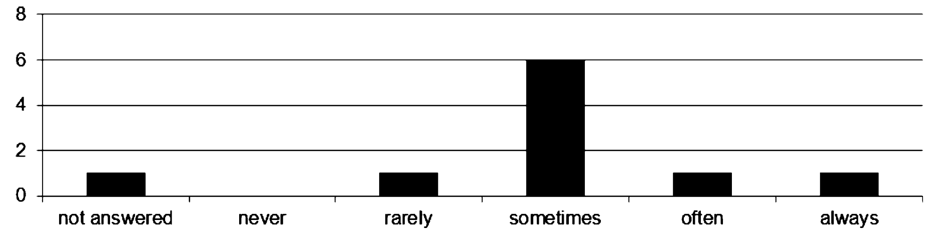


Fig. 19 Participants' answers for the statement "We clearly distinguish between RE and architecture design in our projects"

have different views regarding what is the problem, i.e. the "what?", and what is the solution, i.e. the "how?" (see [30, pp. 24–26]).

Evidence for finding 4.3: In the questionnaire, we asked what kind of support the participants need regarding the transition from requirements to design. We differentiated between the following three kinds of support representing the main types of approaches described in recent RE literature:

- Traceability between requirements and design
- A systematic approach for the transition between requirements and design
- An automated transition from requirements to design

The strongest agreement was expressed by the participants with regard to the systematic support for traceability between requirements and design (see the black columns in Fig. 20). Neither strict agreement nor strict disagreement could be determined from the results with regard to improved systematic approaches for the transition between requirements and design (see the dark gray columns in Fig. 20). Also, neither distinctive agreement nor disagreement could be observed with regard to the automation of the transition between requirements and design (see the light gray columns in Fig. 20).

3.5 Interrelation of RE and safety engineering

A successful interplay between safety engineering and requirements engineering is crucial for certification and standard compliance. For example, ISO 26262 [17] requires for automotive systems that safety requirements are defined at the system level and refined into detailed software and hardware requirements. In order to comply with such standards, the development process must meet the requirements specified therein. However, typically, RE approaches do not provide explicit guidelines whether they comply with specific safety integrity levels or how the approach should be tailored to achieve compliance. Our study thus investigated participants' needs regarding the interplay of RE and safety engineering.

3.5.1 Key findings

Finding 5.1: RE approaches for safety-critical embedded systems need to provide a significantly improved account of safety engineering concerns.

Finding 5.2: RE approaches need to provide a dedicated support for the traceability between requirements and safety analyses.

3.5.2 Evidence

Evidence for finding 5.1: During the interviews, it became apparent that dedicated departments are in charge of safety engineering in development projects in many companies that develop safety-critical systems. Safety engineers typically review the requirements documents in early development stages in order to perform safety analyses. Such reviews are periodically repeated throughout the entire development process in order to align the safety analyses with requirements changes. As a major result of the safety analyses, safety engineers define safety requirements. During RE, the safety requirements are further refined into more detailed requirements for the system and its components.

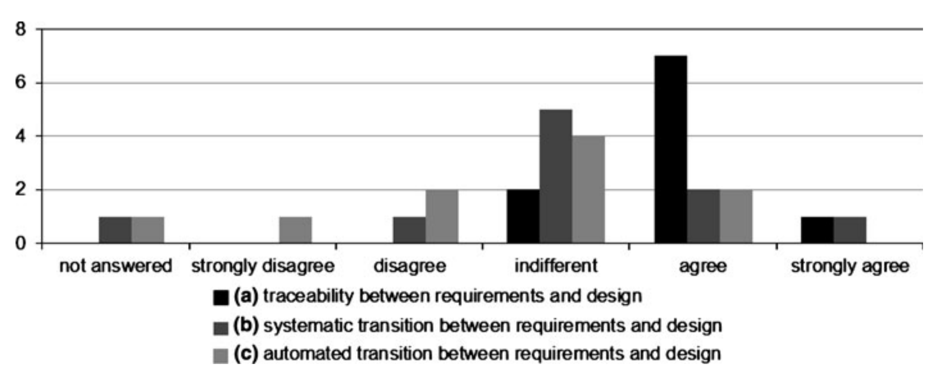


Fig. 20 Participants' answers for three questionnaire statements concerning the support for the transition between requirements and design

The interview findings indicate that the following aspects should be considered by RE approaches:

- Which information must be provided by requirements engineering for safety engineering such that safety analyses can be executed correctly?
- How can requirements specifications be devised such that it becomes obvious which safety requirements originate from which safety considerations?
- Under which conditions must a safety analysis be performed anew after requirements have been changed?
- How can functional requirements and quality requirements be derived systematically in order to satisfy the safety requirements?

The interview findings are in line with findings from two separate questionnaire items, as depicted in Fig. 21. 70% agree or strongly agree with the statement that it is necessary to account for safety engineering concerns during RE when developing safety-critical systems (see Fig. 21, black columns). Only one participant disagreed with that statement. Half of the participants agree that an improved integration of safety engineering and requirements engineering leads to simplified development, while only two participants disagree (see Fig. 21, light gray columns).

Evidence for finding 5.2: The participants noted during the interviews that RE approaches must allow for documenting how the results of safety analyses affect requirements, and how requirement changes affect safety-related concerns. This result is supported by a questionnaire item. As depicted in Fig. 22, the minority of participants see current support for traceability between requirements and safety models as sufficient.

4 Conclusions for RE research

In this section, we draw some conclusions from our study results (see Sect. 3) for future RE research. Potential threats to the validity of our results are discussed in Sect. 5. In Sect. 7, we relate our findings and conclusions to the results of previous studies.

4.1 Support for model-based RE

As indicated in Sect. 3.1, the study participants expressed a strong interest in using requirements models more intensively, which was partly attributed to the higher level of automation of RE tasks that models facilitate. Yet, we also identified substantial issues that inhibit the use of models as the representation format for requirements:

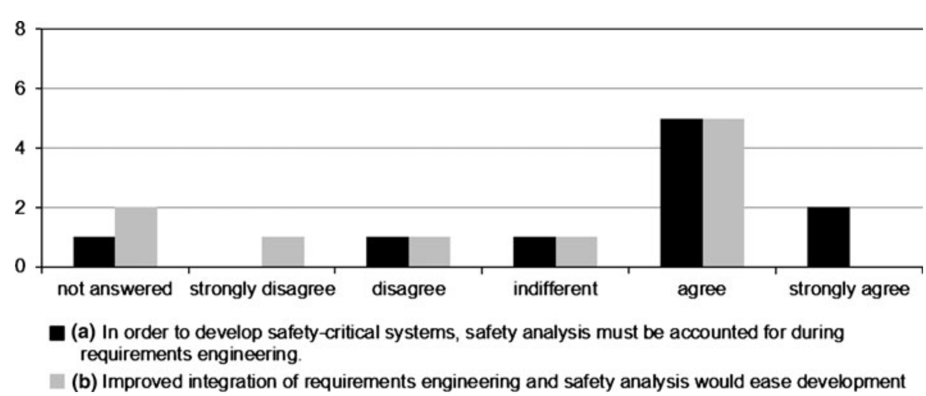
- It is not clear how safety standards for embedded systems such as [17] or [32] can be satisfied when using model-based requirements instead of natural language requirements.
- Practitioners feel that the risk of specifying the design instead of the requirements is increased by model-based RE (see Sects. 3.1 and 3.4)
- Requirements are often used as a contract between different parties. Requirements models are not considered suitable to negotiate and establish such contracts.

We conclude from these issues that presently, model-based approaches cannot replace natural language requirements. Hence, a possible way of resolving part of the current issues regarding the use of natural language requirements is to investigate techniques that allow for a tight integration of requirements models and textual requirements, while reducing the effort for keeping the two representations consistent, and hence allow switching more easily between textual and model-based requirements as appropriate for the specific RE task at hand.

As pointed out in Sect. 3.1, domain-specific models, e.g. from control engineering or mechanical engineering play an essential role in RE practice. Furthermore, a successful interdisciplinary collaboration among different technical domains such as electronics, mechanics, and software is a key challenge for RE in the embedded systems domain.

We conclude that RE research needs to investigate how the models from the these domains with their different

Fig. 21 Participants' answers for two questionnaire items concerning the interrelation of requirements engineering and safety analysis



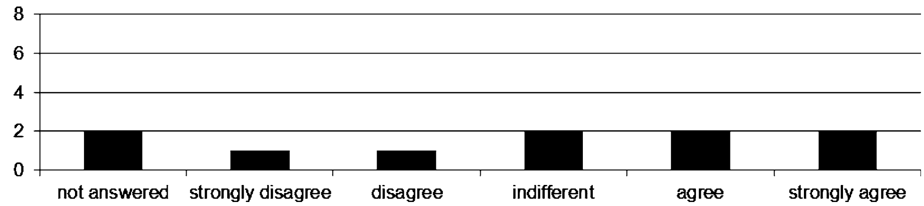


Fig. 22 Participants' answers for the statement "Support for traceability between requirements and safety models must be improved"

characteristic properties can be incorporated in current RE approaches such as goal-oriented or scenario-based RE in order to provide industrial strength methods for the embedded systems domain. Therein, also the mutual interdependencies between the domains need to be accounted for.

4.2 Support for abstraction layers in RE

As indicated in Sect. 3.2, in practice, requirements for embedded systems are specified at different abstraction layers. In a typical case, requirements need to be defined for the product (e.g. an airplane or a power plant), each individual system of this product, each system function, and each hardware and software component needed to implement the system functions. Yet, the use of abstraction layers in RE is still immature, which was attributed to the insufficient guidance provided by existing RE approaches (see Sect. 3.2).

To overcome these issues, research must answer the question of how abstraction layers can be defined in a structured manner, respecting individual domain and project properties. To provide practitioners with workable guidelines for developing requirements at different abstraction layers, the concept of abstraction layers itself needs further clarification. A possible way of clarifying the methodical issues is to provide reference models for specific classes of embedded systems with clearly defined abstraction layers, rules stating what should and what should not be specified at each abstraction layer, and guidelines how requirements at different abstraction layers should relate to each other. A major challenge for defining abstraction layers for embedded systems is to facilitate a technology-independent specification at higher abstraction layers and to account for detailed technology-specific requirements at lower abstraction layers such as electronics requirements, mechanics requirements, and software requirements.

We further conclude from our study results that current RE methods need to be extended and improved to support the development of requirements across different abstraction layers such as those outlined in Table 1. This method support should at least include refinement, traceability, and consistency checking across the different abstraction layers. Furthermore, the method support must account for the fact that, for embedded systems, it is vital to support the specification of performance requirements and safety requirements across the different abstraction layers.

4.3 Support for quality assurance in RE

As stated in Sect. 3.3, existing quality assurance techniques for requirements are only partly satisfactory. We conclude from the findings described in Sect. 3.3 that the existing review techniques should be refined to provide workable guidelines for practitioners to systematically deal with specific requirements quality criteria such as consistency, traceability, or testability. As outlined in Sects. 4.1 and 4.2, these guidelines need to account for checking requirements across different abstraction layers. The guidelines need to respect the different characteristics of requirements at different abstraction layers (see e.g. Table 1) and different types of components such as mechanics, electronics, and software.

As requirements validation in the embedded systems domain is largely based on simulation and prototyping (see Sect. 3.3), RE research should investigate how the modeling for simulation and prototyping in the embedded systems domain can benefit from model-based RE approaches and vice versa. For this purpose, a detailed understanding of the commonalities and differences between requirements models and simulation models are needed.

As indicated in Sect. 3.3, in the opinion of practitioners, a very strong motivation for model-based RE is its potential to significantly improve the quality assurance process in RE. To provide stronger evidence that model-based RE can realize this potential, researchers need to thoroughly investigate which quality criteria are positively influenced by the use of which requirements models in an industrial context.

Furthermore, as validation and verification techniques consume a substantial amount of project resources, RE research should investigate methods for estimating the effort needed for requirements validation and verification and provide workable guidelines for choosing the appropriate techniques based on projects goals.

4.4 Support for the transition between RE and design

Regarding the transition between RE and design, we conclude from the findings presented in Sect. 3.4 that the conditions under which an automated transition from requirements to design is desirable and feasible in the industry should be examined carefully.

Furthermore, as stated in Sect. 3.4, industry has a strong need for systematic, workable approaches for traceability between requirements and design. We conclude that research should investigate the alignment of RE and design methods with a particular focus on traceability between requirements and design and a high level of automation concerning the creation and maintenance of traceability links. Therein, the essential use cases for traceability that occur in embedded systems development such as traceability imposed by safety standards need to be identified and taken into account to provide RE methods with industrial strength traceability support.

Furthermore, our study has provided additional evidence that RE and design activities are tightly intertwined in development practice. Research needs to take these interactions into account, for instance, by investigating process models and development methods that support these interactions. The natural links between requirements and architecture in the embedded systems domain are system and component interfaces which are typically described in terms of the signals exchanged between systems or components. Functional requirements in the embedded systems domain are often statements about the relationship between input and output signal values. At the same time, the signal interfaces are also essential elements of embedded system architectures. This key role of interfaces must be accounted for by approaches supporting the intertwining of RE and design.

With regard to tool support, the RE and design tools for embedded systems should be adapted in order to allow engineers to flexibly switch between the requirements and the design while performing a development activity.

4.5 Support for the interrelation of RE and safety

The results presented in Sect. 3.5 indicate that a tighter integration of safety engineering concerns into the requirements engineering process is desired by industry.

We conclude from the results outlined in Sect. 3.5 that RE researchers and method developers should focus on developing and augmenting approaches that foster safety-standard compliance. For example, existing RE approaches should be amended in a way that applicators can easily identify which safety standard, if any, the approach satisfies, or which part of a standard is met. Furthermore, RE research should focus on developing or altering RE approaches in such a way that they allow for easy configuration of the approach in order to comply with the regulations for the safety integrity levels that are relevant for the project. For example, RE approaches could feature activities and provide guidelines on how to tailor these activities to meet the safety integrity levels of ISO 26262 [17].

Furthermore, for model-based RE approaches such as [29], a promising avenue to integrate requirements engineering and safety engineering more tightly is to incorporate dedicated safety models and to interrelate the requirements and safety models.

5 Critical evaluation

Despite the extensive efforts we made to ensure the validity of the results, not all investigative issues could be completely resolved within the given constraints of the study. This section discusses potential threats to validity.

5.1 Vagueness in questionnaires

It could be objected that the use of vague quantifiers in questionnaire items (e.g. through the use of adverbs such as "predominantly") may have led to distortions in the measured data due to different interpretations of the questions by individual participants. However, this strategy was deliberately pursued. Using the questionnaires, we sought to support interview findings by assessing participants' attitudes toward the statements in the respective questionnaire items. The purpose of the questionnaires was hence not to test a hypothesis. Using vague quantifiers contributed to avoid possible mediocrity in the data, which was likely to be the case using strict quantifiers (e.g. "always") or numeric quantifiers (e.g. "75% of the time") instead. By introducing controlled vagueness, we were able to capture the trends in typical everyday cases in development projects.

5.2 Researcher bias

Since our research focus is on model-based RE, researcher bias might have influenced the results concerning the participants' attitude with regard to using models in RE during the interviews. To reduce this threat to validity, we motivated the participants to express their true opinion and paid special attention to adequately honor any objections against the use of models in RE. In addition, the participants' expertise made them less susceptible to be influenced by the researchers' opinion. Furthermore, we specifically designed the questionnaires to counteract possible researcher bias by means of the Flip-Flop Technique [4] and counter-balanced question wording. In addition, as mentioned in Sect. 2.5.2, every participant reviewed the result protocol from the interview he or she participated in order to ensure that the responses have been transcribed correctly and without researcher bias.

Lastly, the joint application of interview data and questionnaires helped reducing the risk of invalid results by cross-checking interview and survey data. The questionnaires were designed to complement the qualitative findings from the interviews (see Sect. 2.5). It should be noted that the purpose of the questionnaires was not to allow statistical testing of hypotheses as this study is qualitative in nature.

5.3 Representativity

The participant population might not be representative for all companies in the embedded systems domain. Section 2.4 shows how this issue was reduced by involving companies from five different branches of the embedded systems domain with different roles in product development and by focusing on personnel with a good overview of RE-related issues. The participants' (albeit self-reported) many years of industrial experience further increases the trustworthiness of the obtained results.

Furthermore, due to the qualitative nature of our study, the individual participants' reports are regarded as valuable themselves even if statistical significance was not established. During recruitment, an effort was made to ensure that the participants have a good overview over the different roles and activities in embedded systems development and have detailed knowledge of the interrelation of requirements engineering with other development activities. The selected participants were hence able to deliver an accurate assessment of the status quo in their respective application domains. With respect to the study goals, widening the selection criteria in order to increase the number of participants would not have necessarily increased the quality of the results.

Considering the selection of participants, it can be assumed that repeating the study in the same application domains, but with different companies and in a broader choice of countries, will yield similar results. Yet, we expect a high likelihood of additional findings if companies are recruited that make use of very specialized RE approaches, such as formal specification.

6 Practice view: insights from a large automotive project

In this section, we provide insights gained in a large automotive development project after the original study was conducted. The evidence complements the results of the study within two of the five investigation aspects and thereby contributes to the credibility of the study results as well as the conclusions drawn in Sect. 4.

6.1 Obstacles and opportunities for requirements models

The observations made in the considered project confirm that requirements are specified predominantly using natural language. To reduce ambiguity and increase precision of requirements, the guidelines provided to requirements authors recommend, for instance, requirements patterns.

The original study indicated insufficient tool support as a major reason for low usage of requirements models. This suggests that better tool support would lead to an increased use of requirements models. Practical experience partly supports this finding, since, with existing tools, it is difficult to identify individual requirements and track the status of individual requirements in a requirements model. Due to these limitations, models can presently only supplement the textual requirements specifications.

In addition, the project indicated several other factors—not investigated in the original study—that need to be taken into account when designing a model-based RE approach:

- Creative tasks such as inventing requirements and interactive tasks such as negotiating requirements receive more attention and are devoted more time and effort than documentation tasks.
- Requirements documents are rather seen as contracts between different parties than a way of preserving the requirements engineers' expert knowledge for the future.
- There is little previous experience with requirements models to build on in contrast to the many years of experience with natural language requirements.
- There is no strong, quantitative evidence that model-based requirements engineering is beneficial for the industry in terms of cost reduction. A requirements engineer, who decides to invest time into creating requirements models, is hence at risk of wasting effort.

As long as no stronger evidence is available that shows the cost efficiency of model-based RE, the following measures can be taken to reduce the risk of wasting effort:

- *Use requirements modeling as a first step toward test specification*: When creating requirements models, a requirements specification language and a requirements tool should be used that can be reused in testing for creating requirements-based test cases. In this way, rather than spawning additional effort, the effort is merely reallocated from testing to RE.
- *Build on proven tool chains*: It is industry's attitude that extending the tool chain by integrating an additional requirements modeling tool increases the risk for the project. Therefore, as far as possible, model-based RE should build on modeling tools that are already part of

the tool chain that is common in industrial practice. However, it is necessary to clarify the difference between requirements models and other development models, e.g. models used for code generation.
- *Provide sample cases of successful use of requirements models*: Engineers perceive the risk of adopting a new technology such as model-based RE to be lower, if successful examples of model usage in RE are available. Therefore, it is helpful to provide such examples, for instance, by initially applying model-based RE to a small portion of the system functionality.

6.2 Interrelation of requirements, architecture, and abstraction levels

The original study indicated that requirements are specified at different abstraction levels and that requirements and architecture are interwoven. In the following, we characterize the interrelations between requirements, architecture, and abstraction levels based on insights gained in the project:

- *Requirements*: Requirements engineering is performed for a specific object to be developed. There are different types of objects such as vehicle functions, system functions, vehicle systems, control units, sensors, and actuators.
- *Architecture*: An object in a system has interfaces to other objects in the system or its environment. The objects, interfaces, and their interconnections constitute the architecture of the system. As there are different types of objects in the system, several architecture views exist such as functional architecture and physical architecture. Additional architecture views are needed to represent the mapping or deployment, for instance, of functions to control units.
- *Abstraction levels*: The type of the object to be specified and its location in the architecture affect the abstraction level of the requirements for this object. For instance, requirements for a vehicle function are defined at a higher abstraction level than the requirements for a system or subsystem function.

Due the relationships outlined above, requirements, architecture, and abstraction levels affect each other and cannot be considered independently. For instance, a change in the interface of a system function affects the requirements for this system function as well as the functional architecture. Furthermore, changes in the functional architecture must be mapped to the physical architecture that implements the system functions.

As pointed out in Sect. 3.2, participants expressed the need for improved tools and methods that support the specification of requirements across several abstraction layers as well as the intertwining of requirements and architecture. This desire can be attributed to the fact that engineers spend a large amount of their effort on keeping requirements and architecture artifacts consistent across the different abstraction layers. In many cases, this effort is maintenance effort that does not bring about new and innovative functionality. Based on these insights, the conclusion can be drawn that methods and tools should free requirements engineers and architects from this maintenance effort by automating the maintenance tasks as far as possible. A tightly integrated tool and method support that allows specifying and linking natural language requirements, model-based requirements, and architectural artifacts seems appropriate for achieving this goal. Moreover, a tool and method support that allows organizing the specification in a structured manner into several abstraction layers that are aligned with the different architectural views appears beneficial.

7 Related work

In this section, we relate our findings presented in Sect. 3 to findings of previously published work and explicitly show the additional findings that are contributed by our study.

In the past, several studies have already investigated the state of practice in RE. We differentiate these studies into the following two categories:

- *Empirical studies* of the state of practice in RE are contributions that employ an empirical methodology to collect and analyze the data, for instance, by questionnaires. Examples of contributions in this category include [10, 19, 21–24, 39].
- *Industrial case reports* describe the state of practice from the perspective of individual practitioners. Usually, such reports do not disseminate quantitatively collected data. Industrial case reports can be found in [1, 6, 9, 12, 31, 38].

In the following, we shortly summarize the related work with regard to the five investigation aspects outlined in Sect. 2.2 and relate prior findings to our results from Sect. 3.

7.1 Natural language versus requirements models

Neill, Laplante, and Jacobs report in [20, 23] that more than half of the participants in their study indicated that natural language is the foremost format of requirements documentation. Similarly, [12, 18, 31] report that natural language is the most common documentation format of requirements, and acknowledge problems resulting from large quantities of textual requirements, such as maintaining requirements traceability and consistency. Furthermore, several studies have investigated the use of models in requirements practice. In [12], the authors state that UML-like diagrams not adhering to strict syntactical and semantic rules are used, albeit not in a strict, methodical manner and only whenever it appears practical. In [23], 51% of the participants are reported to use informal models for product specification, while 7% use formal languages. In addition, it can be seen from [38] that models are rarely integrated into the final requirements specification, even though models are used for elicitation, communication, and validation of requirements.

Several approaches are aimed at alleviating some of the issues that arise from using large bodies of textual requirements, e.g. by means of requirements patterns [8] or ontologies [11]. However, prior studies do not indicate practitioners' satisfaction with processing natural language requirements, attitude toward using it or whether supplementing textual requirements with requirements models is a viable option. In addition, the studies neither state whether a more intensive use of models is desired nor which factors inhibit a more intensive model use.

Our study partly closes this gap. The results indicate that, for practitioners, the main advantage of natural language is its suitability for requirements negotiation especially when the negotiating parties conclude a contract based on the agreed requirements (see Sect. 3.1). Apart from this, the study shows a strong dissatisfaction of practitioners with natural language use and a positive attitude toward model-based RE. A strong motivation for using requirements models is their potential of easing quality assurance (see Sect. 3.3). It can hence be assumed that participants do not wish to replace natural language requirements with requirements models, but to amend them instead. In other words, our study has shown a need for RE research to provide methods that allow for the joint use of natural language requirements and requirements models. According to our findings, domain-specific models such as models of power plants or airplanes are considered particularly useful for RE. Furthermore, standardized modeling languages such as UML and SysML are valued. Our results further indicate that a chief inhibiting factor for a more extensive model use is the insufficient method guidance (see Sect. 3.1) concerning how to perform common RE activities using requirements models instead of natural language.

7.2 Support for high system complexity in RE

Karlsson et al. report in [19] that the challenges due to high requirements complexity are aggravated by monolithic requirement specifications. This problem is alleviated by separating requirements according to predefined criteria during elicitation and documentation. Similarly, Graaf et al. report in [12] that in order to manage high system complexity during development, different aspects of the system are separated by adopting different views on the requirements, each view containing a smaller or larger set of details. In support of this claim, Weber and Weisbrod state in [38] that structuring the requirements and design artifacts of a complex software-intensive system using a hierarchy of abstraction layers is crucial to support the understanding of the specification. In contrast, typical requirements documents in the automotive domain are reported to focus on the detailed technical requirements, i.e. mainly the lowest abstraction layer of the abstraction hierarchy [38]. Regarding the related work, information about how abstraction layers are used, what type of abstraction layers are common in the industry, and what challenges, if any, arise with the use of abstraction layers is largely missing.

Our results support the prior findings that using abstraction layers is a common principle that is applied in order to manage high system complexity. Furthermore, our results show challenges due to a lack of RE method support for abstraction layers. We identified a number of questions concerning abstraction layer use that are unanswered by existing RE approaches. In addition, our study gives an account on typical examples of abstraction layers used in the different branches of the embedded system domain (see Sect. 3.2).

Insufficient tool support for managing high system complexity is reported by [21]: the authors show that commonly used tools are unable to handle large system specifications and/or do not encourage developers to structure requirements in a way that allows for complexity reduction, e.g. by means of different abstraction layers. In addition, [38] mentions that current requirements modeling tools and approaches do not offer the proper method support with regard to modeling activities, and therefore do not support the requirements engineer well enough. Using models as a means for complexity management has been investigated in [7]. Davies et al. note that model use increases in large companies handling mainly large and complex projects. However, information in what way tool and model usage can help in managing complexity has thus far not been subject of investigation.

Our study aimed at investigating the idea of using models in particular as a means to manage complexity in RE. However, our results show no conclusive evidence that models aid in complexity management. One possible explanation is the confusion among engineers regarding the use of models as requirements specifications (see Sect. 3.2) and the challenges related to managing complex models.

7.3 Requirements quality assurance

Prior research, such as [10, 13], and particularly [12], reports a need for better integration of RE and validation and verification. Several studies such as [5, 12] report that major improvements are necessary to support the validation and verification of requirements. Some authors relate to specific quality criteria and report on challenges in achieving these criteria such as the high effort that is required to ensure proper requirements traceability and consistency, particularly in large specification documents (see e.g. [18]). Yet, prior studies neither provide evidence regarding the relative importance of supporting specific requirements quality criteria nor do they indicate what improvement goals are most important.

In our study, we were able to assign priorities to the different quality criteria, showing a particularly strong need for approaches that support checking requirements consistency, testability, and traceability. According to our study, the main improvement goal is to reduce the enormous effort that is required for checking these quality criteria in large requirements specifications by means of reviews.

Furthermore, prior research does not provide conclusive evidence in how far practitioners regard requirements models as beneficial for resolving the challenges in requirements quality assurance. Our study shows that disburdened quality assurance is, in fact, a key motivation for model-based RE from the viewpoint of RE practitioners (see Sect. 3.3).

7.4 Transition between RE and architectural design

The literature reveals conflicting views concerning the intertwining of RE and architecture design. Several contributions report an insufficient integration of RE with architectural design (see e.g. [12]). Anderson and Felici report in [1] that in the avionics domain, functional as well as architectural models are specified together and remain tightly interrelated over the course of development. Similarly, [5] reports that integration of RE and architecture design is done by involving architecture experts in the development teams. Yet, in contrast to what Anderson and Felici report, an insufficient integration of requirements specifications and architecture is revealed in [5]. In addition, so far, no study has investigated what practitioners needs are with regard to a clear separation or a tight integration of requirements and architectural design.

Our study gives clearer insight into practitioners' attitudes and needs with regard to this topic. Our results indicate, for instance, that practitioners strive for requirements specifications that are properly integrated into the overall system architecture but are solution-free with regard to the specified function or component itself. The existing method support is judged as insufficient for achieving the proper balance between architectural integration and solution-freeness of requirements.

7.5 Transition between RE and safety engineering

The related work outlines the necessity to consider safety-related concerns early on during the requirements engineering process. For example, Grimm reports in [13] that a tighter integration of safety analysis in RE processes is required. Some studies such as [12, 14] argue that while formal specification appears to be beneficial for safety-critical systems, projects rarely apply formal methods. The authors theorize that this may be due to the difficulties that arise in industrial settings, as developers require special skills to deal with formal methods. Juristo et al. note in [18] that there is little guidance on how to make use of the best available technology in safety-critical software. This is supported by [33]: the authors report that companies producing safety-critical software do not always follow standard RE process models.

In conclusion, prior research shows that the integration of RE and safety engineering needs is unsatisfactory, yet the investigated reports state neither reasons nor alleviations, and existing work does not address practitioners' needs regarding how this improvement might be achieved.

The results of our study show that practitioners do in fact desire tighter integration of safety concerns with RE, particularly with regard to safety standards. Specifically, RE approaches lack information on what safety standard they comply with, if any, and how approaches can be tailored to comply with standards. Furthermore, RE approaches must be devised such that results from safety engineering activities and RE activities are related to each other in order to facilitate assessing the impact of requirements changes on safety concerns more easily and more reliably.

8 Summary and outlook

This paper reports on the results of an industrial survey in the embedded systems domain. Our main contribution is to reveal major needs of industry practitioners in the field of RE and thereby to indicate directions for future RE research as well as method and tool development. The study focused on five aspects of RE: use of natural language versus requirements models, support for high system complexity, quality assurance of requirements, the transition between RE and design, and the interrelation of requirements engineering and safety engineering. These five aspects have been selected based on their relevance for research and practice as well as expected knowledge gain, for instance, because mismatches exist between RE research and observed or reported RE practice.

Despite a number of industry surveys have reported on the state of practice in systems engineering and RE, no previous study has had such a focus, to systematically investigate practitioners' needs and constraints with regard to future RE approaches.

The study was conducted in 2009 with seven large companies from five branches of the embedded systems domain (automation technology, automotive, avionics, medical technology, and energy technology). The data were gathered by means of interviews as well questionnaires in order to increase the confidence in the results. In total, 17 company representatives participated in the interviews, 10 of the representatives additionally completed questionnaires.

From these results, we have derived conclusions indicating important threads of future research in RE (Sect. 4). An important result of the study is that practitioners advocate a more intensive use of models in RE, yet the use of models is currently impaired by uncertainties regarding the use of requirements models in legally binding requirements documents, particularly for safety-critical systems. Furthermore, method support for abstraction layers is of critical importance for the adoption of RE methods in industry. A strong need for workable solutions for requirements quality assurance with regard to the quality criteria consistency, traceability, and testability has been revealed whereas an automated transition from requirements to design could not be identified as a prevalent need. Furthermore, our study has pointed out the need for an improved consideration of safety engineering concerns in RE approaches.

An account on the generalizability of these results was given in the paper, and the threats to validity were discussed. To further increase the credibility of our results, we cross-checked key results of study with insights gained by hands-on observations in a large automotive project after the study.

In our future work, we will exploit the study results to extend and enhance an existing RE method for embedded systems that was developed in our group (see e.g. Chapter 35 in [30] and [35]). The aim is to tailor our research in order to address industrial needs and constraints to a larger extent and thereby ease the integration of new RE methods into industrial RE processes. We believe that the results conveyed in this paper will also help other researchers as well as method and tool developers to provide RE methods that can be transferred more easily into embedded systems practice.

Acknowledgments This paper was funded in part by the German Federal Ministry of Education and Research (BMBF) in the innovation alliance SPES 2020 (grant number 01 IS 08 045). We thank Dr. Kim Lauenroth for his support in conducting the study, Marian Daun, Sebastian Gabrisch, and Heiko Stallbaum for their help in evaluating the data, as well as our industry partners for participating in the study.

References

1. Anderson S, Felici M (2000) Controlling requirements evolution—an avionics case study. In: Proceedings of international conference on computer safety, reliability and security (SAFECOMP 2000), pp 361–370. doi:10.1007/3-540-40891-6_31
2. Automotive SIG (2010) Automotive SPICE®—process assessment model (PAM) v2.5. http://www.automotivespice.com. Accessed Oct 2010
3. Biegert, U (2002) Ganzheitliche modellbasierte Sicherheitsanalyse von Prozessautomatisierungssystemen. Dissertation, University of Stuttgart
4. Corbin J, Strauss A (2008) Qualitative research, 3rd edn. Sage Publications, California
5. Curtis B, Krasner H, Iscoe N (1988) A field study of the software design process for large systems. Commun ACM 31(11):1268–1287. doi:10.1145/50087.50089
6. Cysneiros JM (2002) Requirements engineering in the health care domain. In: Proceedings of the IEEE joint international conference on requirements engineering (RE 2002), pp 350–356. doi:10.1109/ICRE.2002.1048548
7. Davies I, Green P, Rosemann M, Idulska M, Gallo S (2006) How do practitioners use conceptual modeling in practice? Data Knowl Eng 58:358–380. doi:10.1016/j.datak.2005.07.007
8. Denger C, Berry D, Kamsties E (2003) Higher quality requirements specifications through natural language patterns. In: Proceedings of the IEEE international conference on software—science, technology and engineering (SwSTE 2003), pp 80–90. doi:10.1109/SWSTE.2003.1245428
9. Denger C, Feldmann R, Höst M, Lindholm C, Shull F (2007) A snapshot of the state of practice in software development for medical devices. In: Proceedings of the international symposium on empirical software engineering and measurement (ESEM 2007), pp 485–487. doi:10.1109/ESEM.2007.54
10. Emam K, Madhavji N (1995) A field study of requirements engineering practices in information systems development. In: Proceedings of the 2nd IEEE international symposium on requirements engineering (RE 1995), pp 68–80. doi:10.1109/ISRE.1995.512547
11. Farfelder S, Moser, T, Krall A, Ståalhane T, Omoronyia I, Zojer H (2011) Ontology-driven guidance for requirements elicitation. In: Proceedings of the 8th extended semantic web conference (ESWC 2011) pp 212–226. doi:10.1007/978-3-642-21064-8_15
12. Graaf B, Lormans M, Toetenel H (2003) Embedded software engineering: the state of the practice. IEEE Softw (20)6:61–69. doi:10.1109/MS.2003.1241368
13. Grimm K (2003) Software technology in an automotive company—major challenges. In: Proceedings of the international conference on software engineering (ICSE 2003), pp 498–503
14. Hsia P, Davis A, Kung D (1993) Status report: requirements engineering. IEEE Softw 10(6):75–79. doi:10.1109/52.241974
15. IEC 62304 (2006) International standard medical device software—software life cycle process, 1st edn
16. Innovation Alliance Software Platform Embedded Systems 2020 research project. http://www.spes2020.de. Accessed Oct 2010
17. ISO 26262 (2010) International standard road vehicles—functional safety, 1st edn, under publication
18. Juristo N, Moreno A, Silva A (2002) Is the European industry moving toward solving RE problems? IEEE Softw 19(6):70–77. doi:10.1109/MS.2002.1049395

19. Karlsson L, Dahlstedt A, Dag J (2002) Challenges in market-driven requirements engineering—an industrial review study. In: Proceedings of the 8th international workshop on requirements engineering: foundations for software quality (REFSQ 2002), pp 37–49
20. Laplante P, Neill C, Jacobs, C (2002) Software requirements practices: some real data. In: Proceedings of the 27th annual NASA Goddard/IEEE software engineering workshop (SEW 2002), pp 121–128. doi:10.1109/SEW.2002.1199458
21. Lubars M, Potts C, Richter C (1993) A review of the state of the practice in requirements modeling. In: Proceedings of the IEEE symposium on requirements engineering (RE 1993), pp 2–14. doi:10.1109/ISRE.1993.324842
22. McPhee C, Eberlein A (2002) Requirements engineering for time-to-market projects. In: Proceedings of the 9th annual IEEE international conference and workshop on the engineering of computer-based systems (ECBS 2002), pp 17–24. doi: 10.1109/ECBS.2002.999818
23. Neill C, Laplante P (2003) Requirements engineering: the state of the practice. IEEE Softw 20(6):40–45. doi:10.1109/MS.2003.1241365
24. Nikula U, Sajaniemi J, Kälviäinen H (2000) A state-of-the-practice survey on requirements engineering in small- and medium-sized enterprises. Technical report, Lappeenrante University of Technology
25. Nuseibeh B (2001) Weaving together requirements and architecture. IEEE Comput 34(3):115–119. doi:10.1109/2.910904
26. Nuseibeh B, Kramer J, Finkelstein A (1994) A framework for expressing the relationship between multiple views in requirement specification. Trans Softw Eng 20(10):760–773. doi:10.1109/32.328995
27. Object Management Group (2010) OMG systems modeling language (OMG SysML) language specification v1.2. OMG document number: formal/2010-06-02
28. Object Management Group (2010) UML superstructure v2.3, OMG document number: formal/10-05-05
29. Pohl K, Sikora E (2007) COSMOD-RE: supporting the co-design of requirements and architectural artifacts. In: Proceedings of the 15th IEEE international requirements engineering conference (RE 2007), pp 258–261. doi:10.1109/RE.2007.25
30. Pohl K (2010) Requirements engineering—fundamentals, principles, techniques. Springer, Germany
31. Pretschner A, Broy M, Krüger I, Stauner Th (2007) Software engineering for automotive systems: a roadmap. In: Proceedings of future of software engineering (FOSE 2007), pp 55–71. doi: 10.1109/FOSE.2007.22
32. Radio Technical Commission for Aeronautics (1999) International standard DO-178b—software considerations in airborne systems and equipment certification, 2nd edn
33. Sadraei E, Aurum A, Beydoun G, Paech B (2007) A field study of the requirements engineering practice in Australian software industry. Requir Eng 12(3):145–162. doi:10.1007/s00766-007-0042-4
34. Sikora E, Pohl K (2010) Evaluation eines modellbasierten Requirements-Engineering-Ansatzes für den Einsatz in der Motorsteuerungs-Domäne. In: Proceedings of the workshop on the future of software-intensive embedded systems (ENVISON2020 2010)
35. Sikora E, Daun M, Pohl K (2010) Supporting the consistent specification of scenarios across multiple abstraction levels. In: Proceedings of 16th international working conference REFSQ, LNCS 6182, pp 45–59
36. Sikora E, Tenbergen B, Pohl K (2011) Requirements engineering for embedded systems—an investigation of industry needs. In: Proceedings of requirements engineering foundations for software quality (REFSQ 2011), pp 151–165. doi:10.1007/978-3-642-19858-8_16
37. Van Lamsweerde A (2009) Requirements engineering: from system goals to UML models to software specifications. Wiley, West Sussex
38. Weber M, Weisbrod J (2003) Requirements engineering in automotive development—experiences and challenges. IEEE Softw 20(1):16–24. doi:10.1109/MS.2003.1159025
39. Weidenhaupt K, Pohl K, Jarke M, Haumer P (1998) Scenarios in system development: current practice. IEEE Softw 15(2):34–45. doi:10.1109/52.663783