

How Can We Find Out What Makes a Good Requirements Engineer in the Age of Digitalization?

<https://doi.org/10.3991/ijep.v7i3.7424>

Yvonne Sedelmaier^(✉), Dieter Landes
University of Applied Sciences and Arts, Coburg, Germany
yvonne.sedelmaier@hs-coburg.de

Abstract—Good requirements are commonly viewed as a key success factor for IT (and non-IT) projects, but still there seems to be insufficient insight into which competences requirements engineers need to have these days. Digitalization is likely to pose new challenges to requirements engineering. Chances are that digitalization will change the competences that are necessary for successful requirements engineering.

This paper proposes a research design that will be used for clarifying which competences requirements engineers need nowadays and how these competences change due to digitalization. To that end, qualitative and quantitative research methods will be combined for developing a comprehensive competence profile for requirements engineering on a scientific basis. The resulting competence profile constitutes a starting point for devising competence-oriented learning settings. Thus, our research contributes to a better understanding of competences for requirements engineering and improves education of future requirements engineers, in particular for coping with challenges posed by digitalization.

Keywords—requirements engineering; competence; competence profile; digitalization; education; didactics;

1 Motivation and Purpose

It is commonly accepted that proper requirements are key for the successful design and implementation of complex systems. Requirements engineering (RE) has a fairly longstanding tradition and provides methods and techniques for eliciting, analyzing, documenting, and validating requirements. Yet, while a lot of emphasis is put on the methodological aspects of RE, much less attention has been paid to the people that need to cope with and successfully apply these methods and techniques, namely the requirements engineers. Human aspects are considered with respect to stakeholders, but not with respect to requirements engineers. The latter, however, play a key role in the whole process of building systems: if they get things wrong, the system may not work properly or even be an utter failure.

In the last years, most parts of our professional, but also everyday life have been influenced by, or even based on, software. Current trends such as digitalization or

Industry 4.0 will arguably even intensify this phenomenon. There will be more software-controlled electronic devices around us in every instant of our lives, and these devices are increasingly connected in most intricate ways. To aggravate things, these devices also become more and more heterogeneous.

As a consequence, requirements that delineate what such complex systems are supposed to do become increasingly complex. This is due to various reasons: as the number of devices or components increases, so does the number and diversity of stakeholders, which may even be completely unknown when these systems are devised. Furthermore, the complexity of these systems (or even systems of systems) prevents a single person from getting the big picture. At the same time, the number and complexity of logical and physical interfaces between devices and components increases, thus augmenting the risk that important details may be overlooked.

RE is surprisingly difficult even for simpler systems, let alone for complex ones as in the age of digitalization. As of now, there are only few precise, significant, and valid indications of which competences good requirements engineers must have. It is even less clear which additional competences are required in the future to enable them to specify complex systems in the face of digitalization, Industry 4.0, or the Internet of Things. Also, there are neither established didactical concepts to convey these competences to requirements engineers-to-be, nor proper tools to assess competences of candidate requirements engineers.

The first major goal of this research is a systematic and methodologically sound identification and characterization of competences that requirements engineers need in order to be able to specify highly complex systems in a digitalized world. The research design rests on a solid scientific basis in human and social sciences and follows established principles such as, e.g., Grounded Theory. Results will necessarily combine technical knowledge and soft skills and require an interdisciplinary approach between informatics and pedagogy.

Based on the identified competences, this research aims, in a second step, at the systematic development of educational concepts and assessment instruments for RE both in higher as well as in continuing education. These concepts take technical and non-technical skills into account, will be based on sound pedagogical and didactical theories such as, e.g., constructivism and, thus, improve human resource (HR) management and university education.

All in all, in the long run our research will establish an indispensable basis for the education of future requirements engineers.

2 Requirements Engineering

Requirements engineering is a core issue in software engineering. If requirements do not adequately capture stakeholders' demands, the final software system will most likely not match their expectations. If requirements are not specified properly, the whole project may fail: A multitude of studies, e.g. [1], [2], indicate that issues related to requirements are a top risk factor for projects.

The importance of engineering requirements properly has been perceived for quite a while – one of the first textbooks on requirements engineering was published in the late 1980s [3]. Since requirements engineering first gained attention, it has established itself as an active area of research which is also reflected by the fact that the origins of the flagship conference in the field, the International Conference on Requirements Engineering (requirements-engineering.org), date back to the early 1990s.

Over the years, many research issues in RE have been addressed from many different points of view, mostly focusing on technical and methodological aspects of RE, namely methods and formalisms for eliciting, describing, or estimating requirements [4]–[7].

Although a solid basis for dealing with requirements has been established, things do not run smoothly all the time. This may be concluded, for instance, from an international initiative, NaPiRE (www.re-survey.org), which aims at “establishing an open and generalizable set of empirical findings about the status quo as well as practical problems and needs in RE”. Apparently, the initiators of this initiative still feel some pain in requirements engineering which needs to be better understood.

One of the studies of the NaPiRE network [8] characterizes the most critical problems in RE projects as follows:

- Moving targets (changing goals, business processes and / or requirements)
- Incomplete and / or hidden requirements
- Communication flaws within the project team
- Communication flaws between requirement engineers and customers

It is commonly accepted and verified that RE is a people process. Nearly 90 percent elicit requirements through interviews or workshops [8] in order to obtain requirements from various groups of stakeholders. Some research (see e.g. [9], [10]) has been devoted to issues of how stakeholders may be involved in that process most effectively.

Requirements engineers play a key role in that process. In particular, requirements engineers must be able to cope with a variety of challenges such as those mentioned above. To that end, they need various competences, most notably communication competences and skills to cooperate in a team (see e.g. [11]) and collaborate with stakeholders in a rapidly changing world. 17 % of companies mention that problems arise from missing skills and qualifications of requirements engineers.

So far, little attention has been paid to identify capabilities that requirements engineers need to have. Our research aims at closing this gap. In particular, we pursue the following research question: Which technical and non-technical capabilities does a requirements engineer need to have?

3 Related Work

As a first step in the pursuit of this research question, a literature review was done.

3.1 Necessary Capabilities for Requirements Engineering

Certification Schemes in Requirements Engineering. Various initiatives try to test and certify good requirements engineers, e.g. IEEE's Software Development Associate Engineer Certification or the Certified Requirements Engineer of the International Requirements Engineering Board (IREB).

The IEEE Computer Society launched its certification programmes in 2003 in order to "assess and confirm proficiency in established software development practices and to distinguish professionals that have demonstrated a commitment to lifelong learning" [12]. The Software Development Associate Engineer Certification scheme covers four knowledge areas including software requirements. Certification is provided after passing a three-hour online test focusing on issues based on the Software Engineering Body of Knowledge (SWEBOK) [13].

IREB (www.ireb.org) is a non-profit organization which was founded in 2006 and established a certification concept for RE professionals on a personal level.

Currently, IREB certification focuses on two levels, namely foundation and advanced level. Requirements engineers can be certified by a multiple-choice test on the foundation level and a multiple-choice test plus thesis on the advanced level. A third and even more advanced level, namely the expert level, is not yet published.

IREB focusses on testing knowledge, mainly through multiple choice questions.

Relationship of Requirements Engineering and Software Engineering. RE is part of software engineering. Due to this close relationship, it might be helpful to have a look at software engineering competences in order to draw some deductions from software engineering competences to requirements engineering competences.

In software engineering, some sources may be found that try to characterize the field and necessary capabilities.

A fairly popular source is the Software Engineering Body of Knowledge (SWE-BOK) [13]. SWEBOK is a collection of knowledge a software engineer should have after four years of working experience. In SWEBOK knowledge areas (KA) characterize the contents of software engineering. Each knowledge area is characterized by a brief definition and then broken down into subareas, topics, and sub-topics. For each topic or subtopic, a short description is given. One of the first SWEBOK chapters deals with software requirements and the knowledge a software engineer should have with respect to requirements engineering. As figure 1 shows, SWEBOK focuses on techniques and methods in requirements engineering, but does not pay attention to relevant skills or competences.

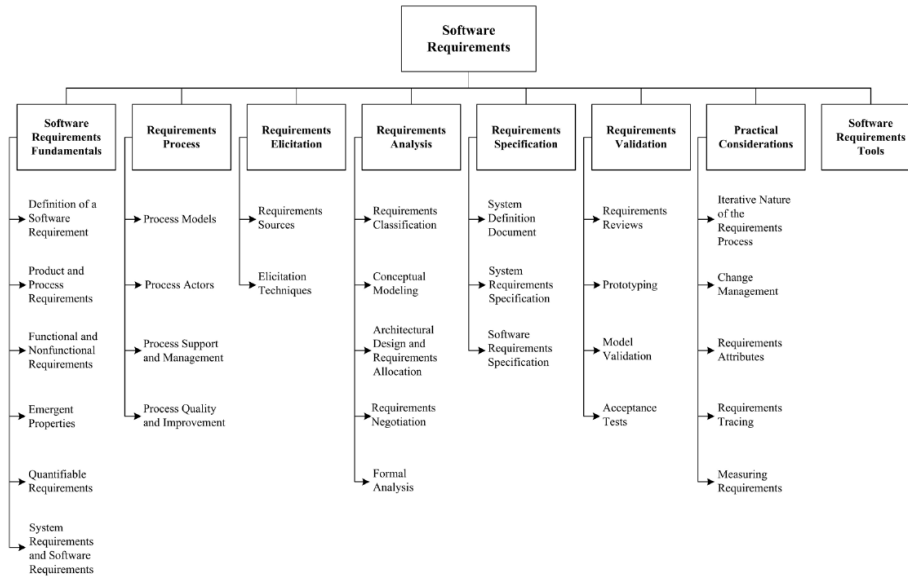


Fig. 1. SWEBOK Chapter 1: Breakdown of Topics for the Software Requirements KA

In 2014, IEEE published the Software Engineering Competency Model (SWECOM) [14] which is closely related to and substantially based on SWEBOK. In SWECOM, “Knowledge [...] is different from skill: knowledge is what one knows, while skill is what one can do”. This competency model specifies skill areas, skills within skill areas, and work activities for each skill. Activities are specified at five levels of increasing competency.

The Software Engineering Body of Skills (SWEBOS) [15] also provides a characterization of relevant competencies in software engineering, yet follows a different scientific approach. SWEBOS is based on sound qualitative research methods and covers competences in software engineering with a focus on non-technical skills. Competences in SWEBOS are described on three competence levels and also include competences for requirements engineering.

The Association for Computing Machinery (ACM) and the IEEE Computer Society launched curriculum guidelines for undergraduate degree programs in computer science [16]. These guidelines also mention software engineering, yet without definition of competences. The same applies to RE which is simply named.

Discussion. Unfortunately, none of these approaches provides a thorough account of required competences of requirements engineers. Rather, they fall short of characterizing RE competences for several reasons.

“Wrong” Subject - Software Engineering instead of Requirements Engineering. SWEBOK is in essence a collection of software engineering content. Of course, this includes RE, but neglects the specific details of the field since RE is just one of 14 knowledge areas in SWEBOK. Consequently, capabilities that requirements engineers

need are not addressed in sufficient detail. Furthermore, SWEBOK has a strong bias towards technical or methodological content instead of describing competences.

Even though SWEBOS provides a competence profile for software engineering with a strong emphasis on non-technical skills, it focusses on software engineering rather than on requirements engineering, thus neglecting the specifics of the latter field.

Requirements engineering inevitably involves interaction with stakeholders of different disciplinary backgrounds and is first of all targeted to understanding the problem, rather than devising or building a solution. Therefore, competences in RE are expected to be different from competences in SE due to this higher degree of multidisciplinary, its primarily analytical and problem-oriented rather than constructive and solution-oriented focus, and specific emphasis of non-technical skills in the face of high initial uncertainty. These differences between software and requirements engineering most likely entail different competences of software and requirement engineers.

Lacking Scientific Basis. The scientific basis on which SWEBOK or SWECOM have been developed is unclear, if any. Evidently, SWEBOK and SWECOM express some form of informal consensus of the software community with a collection of issues compiled by experts that volunteered to characterize a particular area of individual expertise. Their draft was then subjected to a community discussion and formally approved, once substantial objections had been settled.

There is no precise description which method was used to arrive at exactly those knowledge areas or topics that are covered in SWEBOK and SWECOM. Such an approach would be completely inappropriate in social sciences. Social sciences presuppose a traceable research design in order to arrive at trustable, valid, and acceptable results. In social sciences, it is inevitable to discuss the way to derive research results to make them plausible. Only a structured, reliable, and valid research process can provide reliable data. And only reliable data are valuable data for understanding the field of research. Data in social sciences must reflect reality authentically. This is a core aim of social sciences. Thus, SWEBOK and SWECOM fail to meet the requirements of social sciences because it is not traceable where data came from and how they were derived. Consequently, some doubt will always remain if findings in SWEBOK or SWECOM depict the world correctly since there is no possibility of tracing and checking how they were developed. Therefore, SWEBOK and SECOM are only fairly weak guidelines since they suffer from a lack of sound scientific underpinning.

By and large, the same criticism can be expressed with regard to IREB which is also an effort of volunteers that are more or less representative for the requirements engineering community. In addition, IREB's certification scheme has an additional economic dimension since several companies offer trainings as a preparation for the certification examination, which might also be a source of bias.

Factual Knowledge and Tasks instead of Competences. IREB exhibits a very strong content orientation instead of really testing competences. This becomes obvious also in the fact that IREB often applies multiple-choice tests, on higher levels in combination with a thesis. This kind of tests does not lend itself to detect competences.

es. Effective training presupposes to know the required underlying competences. Learning something in theory – like IREB and others do – does not necessarily qualify for successful transfer of theoretically learned knowledge into practice. Factual knowledge alone does not suffice to apply RE methods and techniques; rather more complex competences are required. This is true for any type of competence: Watching someone riding a bike or listening to someone explaining how to ride a bike does not enable the observer to actually ride a bike by herself. The application of methods in RE requires not only generic soft skills such as presentation techniques, but rather context-sensitive non-technical skills [15] in tight combination with factual knowledge.

Furthermore, IREB explicitly focuses on continuing education and neglects vocational training as well as higher education. The different audience matters since participants in a training in continuing education usually already had some professional encounters with what may go wrong in requirements engineering, while students at universities initially often take requirements for granted.

Especially SWECOM concentrates on typical tasks in software engineering. Tasks are frequently used to identify competences on a scientific basis (see e.g. [17], [18]), i.e. tasks are collections of relevant activities which point to competences of a person who carries them out. But tasks may change, while competences remain (and may be used for many different tasks). Competences are “bigger” concepts than tasks [19]. Competences are multi-complex and even if someone could solve a single task, no automatism enables this person to transfer her recipe for a small task to more complex and variable situations [20]. In complex situations, tasks cannot be addressed independently, without taking their interrelationships into account. Thus, competences are more rewarding than tasks in order to find out what makes good requirements engineers.

By and large, the same criticism may be passed on SWEBOK with focuses on software engineering contents instead of the competences which are required for selecting, adapting, and applying appropriate action patterns in complex situations.

3.2 Conclusion

Current research neglects the requirements engineer as human individual who selects and applies these methods. RE techniques are implicitly assumed to work for all kinds of engineers, without taking the person which is meant to employ these methods and her competences into account.

Furthermore, all approaches share a basic problem: they were developed before digitalization gained momentum in terms of influencing our everyday lives. As a result, digitalization and its effects on requirements engineering are not considered in these approaches.

4 Digitalization

4.1 Trends in Digitalization

Digitalization is a broad term and affects our lives in various forms. The following characterization of the term might be appropriate to capture the underlying phenomenon: “Digitalization is the use of digital technologies to change a business model and provide new revenue and value-producing opportunities; it is the process of moving to a digital business” [21].

Digitalization happens on two levels of granularity. On the micro level, technological progress allows to make traditional devices more “intelligent” at a reasonable cost. For instance, telephones turned into smartphones, i.e. devices that provide many features on top of being able to place a phone call. Likewise, smartwatches can do a lot more than just indicating what time it is. Sensors can be placed almost everywhere in order to capture data and transmit these data to other devices or some form of repository. Put differently, almost every “thing” will sooner or later be in a position to communicate via internet – giving rise to the Internet of Things (IoT).

Transmission of data among devices is a key feature of digitalization which establishes a second level of granularity, namely the macro level. Machine-to-machine (M2M) communication paves the way to establish (heterogeneous) systems of “intelligent” devices that monitor their environment by exchanging data with their peers, e.g. in the context of autonomous driving or industrial production. In particular, in the latter setting, components may exchange data with the machines that work on them, but also machines in a production line communicate heavily among each other, e.g. in order to ensure the correctness and sanity of the production process. In order to highlight the fact that computerization may lead to fairly autonomous production lines, thus leading to major changes in industrial production, the term “Industry 4.0” has been coined.

Another aspect at the macro level is commonly called “big data”– large volumes of potentially inaccurate data from heterogeneous sources in diverse formats are combined in real-time to identify valuable patterns in the data that become visible only in a global rather than local context.

Yet, cooperation does not stop at the level of individual devices. Instead, even complex systems may be linked with each other, thus giving rise to something that may be called “systems of systems”. In such a setting, systems work largely autonomously, but still need to work together with all types of other systems to warrant their proper operation. Currently, no established ways to ensure proper cooperation between complex systems are available (see, e.g., DYMASOS project (www.dymasos.eu)).

4.2 Methodological Challenges

Technological progress and the complexity of the resulting systems may well cause traditional development approaches to fail. In particular, early development phases might well require novel capabilities. More specifically, a particular burden may be

imposed on requirements engineers: On the one hand, technological advances may open up new and hitherto unforeseen opportunities – as a consequence, requirements engineers need to be fairly familiar with the technical details in order to be able to recognize the potential of these technologies for novel applications.

On the other hand, components that use novel technology are just a small ingredient in a much larger system, yet with an important role for the correct operation of the overall system. As a consequence, requirements engineers also need to have the capability to see the big picture, irrespective of all the details.

By and large, the balancing act between thorough knowledge of the technical details and their potential on the one hand and the vision how individual components might be combined into larger ensembles might get more and more difficult to handle, in particular for requirements engineers.

Digitalization also tends to exert influence on aspects of everyday life that previously were not affected by IT. As a consequence, requirements engineers need to handle large groups of heterogeneous stakeholders and establish some consensus among them with respect to the system in question. Also, some stakeholder groups never came that close to IT systems beforehand – consequently, they have no idea what requirements are and why it is important to capture them precisely. With other words, requirements engineers have to come to terms with stakeholders that had never before been exposed to requirements engineering techniques. There are indications (e.g. in [22]) that RE will change due to digitalization, and so will the required competences. Chances are that “classical” requirements engineering will not work anymore in digitalization.

5 Research Interest and Research Goals

Our research aims at gaining a better understanding of competences that requirements engineers need to be able to do their job appropriately, in particular in the age of digitalisation. To that end, a competence profile shall be developed that serves as a basis for analysing individual competences in RE, e.g., in HR management, and allows to develop specific RE trainings beyond the factual level. A precise characterization of competences needed in RE will also lead to systematic educational concepts which support higher education as well as HR management in companies, on a sound pedagogical basis. This presupposes, however, that the research process is based on sound pedagogical principles in order to obtain reliable and comprehensive results.

Our earlier work led to first sound characterization of required competences in the area of software engineering. Yet, software engineering and RE are likely to differ in terms of competence. Consequently, conclusions drawn with respect to didactical decisions will also be different.

The first goal of our research is a systematic and methodologically sound identification and characterization of competences that requirements engineers need in order to be able to specify highly complex systems in a digitized world. The research design rests on a solid scientific basis in human and social sciences and follows established principles such as, e.g., Grounded Theory [23]. Results will necessarily combine

technical knowledge and soft skills and require an interdisciplinary approach between informatics and pedagogy.

Based on the identified competences, the second step of our research aims at the systematic development of educational concepts and assessment instruments for RE, both in higher as well as in continuing education. These concepts take technical and non-technical skills into account and are based on pedagogical and didactical theories such as, e.g., constructivism [24]. Thus, they bear the potential to improve human resource (HR) management and university education, e.g. by developing competence assessment tools which may decide on employees' future careers in RE.

In summary, our research will contribute to establish an indispensable basis for the education of future requirements engineers. Our research is targeted to obtain a better understanding of which factors need to be taken into account and how they affect learning and teaching of particular topics in requirements engineering, under particular consideration of the effects of digitalization on these competences. Furthermore, we want to develop new educational approaches that help learning and assessing competences in RE. Both parts of the research rest upon sound pedagogical and didactical principles.

The following section outlines the theoretical background and defines important terms. Then, the research goals are precisely defined before the planned research design is described.

6 Competences

A core characteristic of our research lies in its focus on competences instead of tasks or factual knowledge. Competences are more powerful concepts than tasks - a person needs some competences if she has to carry out a task. Competences are more than factual or procedural knowledge, but rather also encompass skills or, more generally, non-technical aspects [15]. A comprehensive, scientifically sound competence profile for requirements engineers must encompass both technical and non-technical competences, just like a competence profile for software engineering does. The latter competence profile distinguishes general soft skills, context-sensitive soft skills and factual expertise with particular focus on context-sensitive competences intertwined with factual expertise.

7 Research Design

Competences for RE need specific attention as they are arguably not simply a subset of software engineering competences. In particular, RE is characterized by an increased amount of fuzziness, creativity, and multi-disciplinarity. Good software engineers are not necessarily also good requirements engineers.

7.1 First Research Phase: Investigation of the RE Competence Profile Applying Grounded Theory

The first phase is directed towards developing a competence profile for RE on a sound scientific basis. This phase aims at gaining a deep understanding of the required technical and context-sensitive non-technical competences of requirements engineers based on qualitative research methods. To that end, the research design employs a data-driven and traceable approach following the principles of Grounded Theory [23], based on data from real practice.

To achieve reliable results, a mixture of research methods, combining qualitative with quantitative methods, is employed [25].

Qualitative research basically relies on linguistic data, e.g. in the form of texts, while quantitative methods use numeric data [26]. Qualitative research methods aim at a relationship of trust between interviewer and interviewee [27], [28]. Detailed insights into the field of requirements engineering in the age of digitalization are inevitable for our research. In particular, a clear picture of these issues is a prerequisite for understanding requirements engineering processes with their structural and process variables.

In our qualitative analysis, we primarily rely on interviews and observations. In order to uncover relevant aspects and understand RE, in particular in the context of digitalization, we conduct guided interviews face-to-face or on the phone with requirements engineers. Guided interviews are semi-structured, loosely following a prepared interview guideline. They allow an open view on previously unknown factors. The interview guideline contains a spectrum of potential questions and focuses on the research themes of interest. It also ensures comparability of the collected data, at least to some extent. Interviews are recorded, transcribed, and interpreted. Open questions are employed instead of closed ones to get new information on, e.g., factors influencing RE processes in a digitalization context. Qualitative research results are relevant for building hypotheses at a later point of time.

This step of our research is based on Grounded Theory [23] which allows to discover basic concepts and processes in a social system. We use this methodology to develop hypotheses and theories to better understand RE concepts, processes, and competences and how they change due to digitalization. Grounded Theory assumes that there is no predefined research agenda. Planning a research endeavor precisely before beginning would require enough knowledge and clear-cut hypotheses to decide which data need to be collected from whom and which research method shall be used. Yet, such an approach is not suitable for the sort of completely new and unexpected phenomena which we are looking for. Thus, we use Grounded Theory as a methodology with continuous planning. In spite of the name, it is a research strategy without strict rules rather than a theory. Grounded Theory is characterized by several features: A research project does not start with theoretically deduced hypotheses about the research theme, but with assumptions and initial knowledge about the research field that has to be structured. Through continuous target-directed data collection, preliminary concepts can be developed and specified in an interplay of deduction and induction. We also do theoretical sampling [29], [30], i.e. we analyse qualitative data while

collecting, decide which data are needed next while evaluating data, and we stop collecting data once it becomes clear that there will be no further new information.

In addition, we use several different research methods. Qualitative analysis builds upon two basic principles, namely the principle of openness [28] and the principle of communication. This means that investigations take place in an open and unconstrained fashion and look for previously unknown aspects of learning. In most cases, the required data are obtained by communicating and interacting with individuals such as requirement engineers and stakeholders that are involved in some part of the process.

Qualitative research methods are employed to build hypotheses and generalize and structure the field of study. In this way, we develop a theory about the required competences in RE in a digitalization context.

Qualitative methods are used in combination with quantitative ones through triangulation. Triangulation means that quantitative data are collected in order to complement and confirm qualitative data, thus leading to a more comprehensive view on the area of research [25].

All in all, in its first phase, our research will identify target competences in RE and collect a broad range of data. This is targeted on getting a deep understanding of required competences of requirements engineers in digitalization and how competences change due to digitalization.

The main focus lies on understanding RE thoroughly. The resulting competence profile contains precise and semantically rich “thick descriptions” [31] of RE competences. Thus, skills are not just characterized in a broad and general manner, but rather specifically adapted to RE in digitalization. Then, detected RE competences can be analyzed according to questions such as: Are some competences more “important” than others? Are there changes in required competences due to digitalization? Thus, the resulting competence profile reflects the real world and paves the way to deduce specific competences, develop educational designs, and assess competences.

7.2 Second Research Phase: Education in Requirements Engineering

The second phase puts a focus on teaching and learning RE. Now that the competences that make good requirements engineers are known, the question arises how these competences can be systematically developed in requirements engineers-to-be. Thus, our research establishes the basis for developing and assessing goal- and competence-oriented educational concepts in RE and evaluating context-sensitive competences as well as educational concepts in RE.

Related Work. Quite a few publications deal with requirements engineering education in some way or the other. A series of international workshops, namely the International Workshop on Requirements Engineering Education and Training (REE&T), is devoted to the topic. A recent overview of activities in the field can be found in [32], up-to-date work is reported by, e.g., [33], [34]. Many papers in requirements engineering education focus on describing learning approaches in more or less anecdotal format, but lack a sound didactical underpinning. At best, a group of papers refers back to curricula such as SWEBOK or the IEEE/ACM Computer Sci-

ence Curriculum (see sec. 3.1). Likewise, a systematic evaluation is missing in a large portion of the reported work [32]. A curriculum with a didactical underpinning requires a clear goal [35] which is described in intended learning outcomes and is necessarily deduced from required competences in RE, namely the competence profile. Yet, there seems to be little work that mentions required competences explicitly. Usually, the focus of existing work is on methods that were used in RE classes, yet without referring to competences that would benefit from the proposed methods. Often, an underlying overall didactical concept or scientific foundation is completely missing.

Systematic Development of Educational Concepts and Assessment Instruments for RE. The second research phase deals with transforming the theoretical competence profile into practical RE education. IT companies as well as educational institutions benefit from the RE competence profile which serves as a compass in educational processes since intended learning outcomes and goals of educational processes can be precisely defined. As a result, lecturers and HR managers can design better goal-oriented trainings and learning processes. Nevertheless, trainings and learning processes need to be evaluated systematically. A better understanding of the characteristics and challenges in training and learning RE is a precondition for a) systematically designing and offering goal-oriented trainings and learning processes, b) evaluating the effectiveness of learning processes, and c) identifying goal- and competence-oriented didactical mechanisms to lay the basis for a subject-matter didactics in RE. The latter is the theoretical basis for selecting, qualifying, and training future requirements engineers systematically.

Subject-matter didactics aim at establishing environments that promote learning in a specific field of study. A prerequisite for any such subject-matter didactics is a sufficient understanding of learning mechanisms. A clarification of the previous knowledge and the mental concepts that students bring into a course at its outset is the basis to adapt the course to students' real needs. In case of a mismatch between students' needs and the didactical concept, learning remains on a very simple level and will not foster relevant competences. Course design encompasses various issues which have to be decided upon and need to be coordinated, such as the following questions:

- Which contents should be learned and why?
- How can instructors promote learning?
- Which methods support students' learning?

Didactics facilitate these decisions and help creating good learning environments by systematically analyzing the specific learning situation and putting decisions on a sound theoretical basis.

In our previous work we developed a subject-matter didactics for software engineering which rests upon a sound pedagogical basis. This didactics for software engineering follows competence-oriented principles which are in line with constructivism.

According to constructivist didactics, teachers act as coaches and can only give learners room for their individual learning experience. Constructivist theory also argues that learning depends on the individual world and on whatever a person learned before. Understanding arises from the interaction between the learner and the envi-

ronment [36]. Savery and Duffy [36] conclude that “cognitive conflict or puzzlement is the stimulus for learning and determines the organization and nature of what is learned. [...] Knowledge evolves through social negotiation and through the evaluation of the viability of individual understandings.” It is necessary that the learner ties up his already existing knowledge and expertise to further develop it in her own way. Therefore, students learn individual things according to their subjective previous understandings, skills, and knowledge even if they experience the same learning situation.

Learning takes place when learners consider the topics as relevant for their purposes [37], [38]. Then, they are interested in the issues and, consequently, motivation for learning grows. Instead of teaching solutions for problems which students cannot even imagine, it is far more advantageous to make them see and understand the problems right at the beginning. Only after recognizing the problem, students may effectively learn possible solutions and apply their new knowledge (learning by doing). Educational psychology recognizes these principles as main factors for successful learning [38].

As [39] shows, a core problem in teaching and learning software engineering is to make learners understand the necessity for software engineering at all and generate problem awareness for the subject. Furthermore, software engineering is based on competences - simply learning methods and approaches by heart does not help to master a challenge in real life. Due to the fact that software engineering and RE are heavily interwoven, it should be evaluated if problems in teaching and learning RE are similar to problems in software engineering.

In this research, two fields of RE education can be distinguished, namely RE education at universities and continuing education. Arguably, the practical application of subject-matter didactics will differ in these two different cases.

RE in higher education: Since students often have no working experience it is difficult for them to understand the necessity of RE. There is a wide-spread belief among students that requirements come on a silver tray and that it is no problem to elicit them. For this reason, it is a precondition for successful teaching and learning RE to arise problem awareness before addressing factual knowledge and context-sensitive competences.

RE in continuing education: In continuing education for RE, the problems in learning and teaching may be a little different. Here, participants of trainings and workshop usually know the pain in RE and in most cases even experienced this pain personally in their work environment. Therefore, problem awareness is already existent and the focus may shift towards training and fostering required competences that are needed to overcome problematic situations.

In both cases, our research will lead to a better understanding for teaching and learning RE, e.g. by a competence-oriented evaluation of didactical approaches. Furthermore, it is expected that approaches for certifying and evaluating competences in RE are developed, e.g. as a supplement for or extension to content-oriented approaches such as IREB’s certification scheme.

In our previous research, we developed an assessment tool, SECAT, for evaluating software engineering competences in order to refine didactical approaches [40], [41].

A similar approach can be taken for requirements engineering in such a way that SECAT may be adapted to RE. In particular, this would include an adaptation of the underlying model, namely the criteria, concretions and, of course, the items. This may result in e.g. questionnaires that assess RE competences in order to select employees, offer individual trainings or evaluate the effectiveness of didactical approaches, in the way that our earlier work did for software engineering. The refinement of didactical concepts and corresponding evaluation instruments needs to be continued as an ongoing process, in particular since digitalization is still developing and changing rapidly.

8 Summary and Outlook

It is commonly accepted that good requirements are a key success factor for IT (but also many non-IT) projects. As a consequence, requirements engineering aims at the effective and efficient gathering of requirements and turned into one of the core areas of software engineering. Requirements engineering is “people business” [8], but still there seems to be insufficient insight into which competences requirements engineers need to have these days.

Digitalization is likely to pose additional challenges to requirements engineering. On the one hand, requirements engineers need to have some understanding of a multitude of technologies in order to be able to assess their potential in a particular situation. On the other hand, requirements engineers need to be able to see the big picture in increasingly complex systems (of systems). Furthermore, digitalization causes complex IT systems to propagate into areas of everyday life that so far had not been affected by such systems. In turn, this means that requirements engineers have to cope with new groups of stakeholders that may not be accessible with traditional RE methods and techniques. Chances are that digitalization will change the competences that are necessary for successful requirements engineering.

This paper presents a research design for clarifying the competences that requirements engineers will need in the age of digitalization. To that end, qualitative and quantitative research methods will be combined for developing a comprehensive competence profile for requirements engineering. The resulting competence profile will be beneficial for higher education as well as for industry since this profile may serve as a compass when devising didactical settings in university, but also continuing education in requirements engineering.

An important aspect in devising competence-oriented learning settings is their evaluation. The competence profile for requirements engineering will also be the starting point to adapt an existing competence evaluation tool, SECAT, to the requirements engineering domain.

All in all, this research will make an important contribution to better understand requirements engineering and improve education of future requirements engineers, in particular for coping with challenges posed by digitalization.

9 Acknowledgment

The work is part of the EVELIN project and funded by the German Ministry of Education and Research (Bundesministerium für Bildung und Forschung) under grants 01PL12022A and 01PL17022A.

10 References

- [1] K. El Emam and A. G. Koru, ‘A Replicated Survey of IT Software Project Failures’, *IEEE Softw.*, vol. 25, no. 5, pp. 84–90, 2008. <https://doi.org/10.1109/MS.2008.107>
- [2] The Standish Group International, ‘CHAOS: A Recipe for Success’. 1999.
- [3] D. C. Gause and G. M. Weinberg, *Exploring requirements: Quality before design*. New York: Dorset House Pub., 1989.
- [4] K. Pohl, *Requirements engineering: Fundamentals, principles, and techniques*. Heidelberg, New York: Springer, 2010. <https://doi.org/10.1007/978-3-642-12578-2>
- [5] S. Robertson and J. Robertson, *Mastering the requirements process: Getting requirements right*, 3rd ed. Upper Saddle River, N.J: Addison-Wesley, 2013.
- [6] C. Rupp, *Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil*, 6th ed. München: Hanser, 2014.
- [7] K. Wiegers and J. Beatty, *Software requirements*, 3rd ed. Redmond, Wash.: Microsoft Press, 2013.
- [8] D. Méndez and S. Wagner, ‘Naming the Pain in Requirements Engineering (NaPiRE)’. 2014.
- [9] M. Bano and D. Zowghi, ‘Users’ Involvement in Requirements Engineering and System Success’, in *3rd International Workshop on Empirical Requirements Engineering (EmpiRE)*, 2013, pp. 24–31.
- [10] J.-L. de la Vara, L. Hoyos, E. Collado, and M. Sabetzadeh, ‘Towards Customer-Based Requirements Engineering Practices’, in *2nd International Workshop on Empirical Requirements Engineering (EmpiRE)*, 2012, pp. 37–40.
- [11] I. Inayat, S. Marczak, and S. S. Salim, ‘Studying Relevant Socio-technical Aspects of Requirements-Driven Collaboration in Agile Teams’, in *3rd International Workshop on Empirical Requirements Engineering (EmpiRE)*, 2013, pp. 32–35.
- [12] IEEE Computer Society, Ed., ‘IEEE Computer Society Certification and Credential Program’.
- [13] P. Bourque and R. E. Fairley, Eds., ‘Guide to the Software Engineering Body of Knowledge Version 3.0 - SWEBOK’. 2014.
- [14] IEEE Computer Society, ‘Software Engineering Competency Model Version 1.0 (SWECOM)’.
- [15] Y. Sedelmaier and D. Landes, ‘SWEBOS - The Software Engineering Body of Skills’, in *International Journal of Engineering Pedagogy (iJEP)*, 2014, vol. 5, no. 1, pp. 12–19.
- [16] Association for Computing Machinery and IEEE Computer Society, ‘Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science’. ACM Press, 2013.
- [17] F. Rauner, *Messen beruflicher Kompetenzen*, vol. 24. Münster: Lit, 2011.
- [18] F. Schott and S. Azizi Ghanbari, *Kompetenzdiagnostik, Kompetenzmodelle, kompetenzorientierter Unterricht*. Münster: Waxmann, 2008.

- [19] F. E. Weinert, 'Concepts of Competence: Definition and Selection of Competencies: Theoretical and Conceptual Foundations (DeSeCo)', 1999.
- [20] J. Erpenbeck and W. Sauter, *So werden wir lernen! Kompetenzentwicklung in einer Welt fühlender Computer, kluger Wolken und sinnsuchender Netze*. Berlin, Heidelberg: Springer Gabler, 2013.
- [21] Gartner Inc., Ed., 'IT-Glossary'. 2016.
- [22] M. Koch, D. P. Magin, and C. Tamanini, 'RE auf dem Land - alles andere als Standard? Anforderungserhebung in ländlichen Regionen', *Softwaretechnik-Trends*, p. to appear, 2016.
- [23] B. G. Glaser and A. L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chicago: Aldine Transaction, 2009.
- [24] H. Siebert, *Didaktisches Handeln in der Erwachsenenbildung: Didaktik aus konstruktivistischer Sicht*, 2nd ed. Neuwied: Luchterhand, 1997.
- [25] U. Flick, *Triangulation: Eine Einführung*, 3rd ed., vol. 12. Wiesbaden: VS Verlag für Sozialwissenschaften, 2011. <https://doi.org/10.1007/978-3-531-92864-7>
- [26] U. Flick, *Qualitative Forschung: Theorie, Methoden, Anwendung in Psychologie und Sozialwissenschaften*, Orig.-Ausg. Reinbek bei Hamburg: Rowohlt, 1999.
- [27] H. Hermanns, 'Interviewen als Tätigkeit', in *Qualitative Forschung*, E. von Kardorff, I. Steinke, and U. Flick, Eds. Reinbek bei Hamburg: Rowohlt, 2000, pp. 360–368.
- [28] P. Mayring, *Einführung in die qualitative Sozialforschung: Eine Anleitung zu qualitativem Denken*, 5th ed. Weinheim: Beltz, 2002.
- [29] N. Emmel, *Sampling and choosing cases in qualitative research : a realist approach*. .
- [30] H. Merkmens, 'Auswahlverfahren, Sampling, Fallkonstruktion', in *Qualitative Forschung*, E. von Kardorff, I. Steinke, and U. Flick, Eds. Reinbek bei Hamburg: Rowohlt, 2000, pp. 286–299.
- [31] C. Geertz, *Dichte Beschreibung: Beiträge zum Verstehen kultureller Systeme*, 12th ed. Frankfurt a.M: Suhrkamp, 1987.
- [32] S. Ouhbi, A. Idri, J. L. Fernández-Alemán, and A. Toval, 'Requirements engineering education: A systematic mapping study', *Requir. Eng.*, vol. 20, no. 2, pp. 119–138, 2015. <https://doi.org/10.1007/s00766-013-0192-5>
- [33] D. Landes and Y. Sedelmaier, 'A Software Modelling Course at the Age of Three', in *2nd European Conference Software Engineering Education (ECSEE)*, 2016, pp. 131–142.
- [34] R. L. Quintanilla Portugal, P. Engiel, J. Pivatelli, and J. C. S. do Prado Leite, 'Facing the challenges of teaching requirements engineering', in *ICSE 2016*, 2016, pp. 461–470.
- [35] W. Klafki, 'Didactic analysis as the core of preparation of instruction (Didaktische Analyse als Kern der Unterrichtsvorbereitung)', *J. Curric. Stud.*, vol. 27, no. 1, pp. 13–30, 1995. <https://doi.org/10.1080/0022027950270103>
- [36] J. R. Savery and T. M. Duffy, 'Problem Based Learning: An Instructional Model and Its Constructivist Framework', in *Constructivist learning environments: case studies in instructional design*, B. G. Wilson, Ed. Englewood Cliffs N.J: Educational Technology Publications, 1998, pp. 135–148.
- [37] K. Holzkamp, 'Wider den Lehr-Lern-Kurzschluß', in *Lebendiges Lernen*, R. Arnold, Ed. Baltmannsweiler: Schneider, 1996, pp. 21–30.
- [38] C. R. Rogers, *Freedom to learn: A view of what education might become*. Columbus, Ohio: C.E. Merrill Pub. Co., 1969.
- [39] Y. Sedelmaier, *Interdisziplinäre Fachdidaktik für Software Engineering -- Forschungsba-sierte Entwicklung und Evaluation eines anwendungsbezogenen didaktischen Ansatzes*. Bamberg: opus, 2016.

- [40] Y. Sedelmaier and D. Landes, 'A multi-perspective framework for evaluating software engineering education by assessing students' competencies: SECAT — A software engineering competency assessment tool', in *Frontiers in Education Conference (FIE), 2014 IEEE*, 2014.
- [41] Y. Sedelmaier and D. Landes, 'Evaluating Didactical Approaches Based upon Students' Competences', in *Global Engineering Education Conference (EDUCON)*, 2016, pp. 527–536. <https://doi.org/10.1109/EDUCON.2016.7474603>

11 Authors

Yvonne Sedelmaier holds a diploma and a PhD in pedagogy with a major focus on adult learning and continuing education at the University of Bamberg, Germany. After ten years in the educational sector and in quality management she is now senior researcher in the project "Experimental improvement of learning software engineering" (EVELIN) and investigates students and their learning processes. Her research interests are teaching and learning software engineering at universities and software engineering didactics. She is with Faculty of Electrical Engineering and Informatics, University of Applied Sciences and Arts, 96450 Coburg, Germany.

Dieter Landes holds a diploma in informatics from the University of Erlangen-Nuremberg, Germany, and a PhD in economics from the University of Karlsruhe, Germany. After several years in industry, e.g. with Daimler Research, he became a full professor of software engineering and database systems at the University of Applied Sciences, Coburg, Germany. His research interests are in requirements engineering, software engineering education, and data mining. He (co-)authored around 50 papers in books, journals, and conferences in these areas. Since 2012 he heads the research project EVELIN. He is with Faculty of Electrical Engineering and Informatics, University of Applied Sciences and Arts, 96450 Coburg, Germany.

This article is a revised version of a paper presented at the EDUCON2017 conference held in Athens, Greece, 25-28 April 2017. Article submitted 14 July 2017. Published as resubmitted by the authors 28 August 2017.