Process Improvement Archaeology

What Led Us Here, and What's Next?

Michael Unterkalmsteiner and Tony Gorschek, Blekinge Institute of Technology

// The results of past improvement initiatives can shed light on an organization's evolution and can represent a learning opportunity for future process improvements. Researchers tested this in an applied research collaboration with the Swedish Transport Administration. //

WHILE IN EVERY organization corporate culture and history change over time, intentional efforts to identify performance problems are of particular interest when trying to understand the current state of an organization. The results of past improvement initiatives can shed light on the evolution of an organization and represent, with the advantage of perfect hindsight, a learning opportunity for future process improvements. We encountered the opportunity to test this premise in an applied research collaboration with the Swedish Transport Administration (STA), the government agency responsible for the planning, implementation, and maintenance of long-term rail, road, shipping, and aviation infrastructure in Sweden.

The agency was formed in 2010 in order to render more efficient the (until then separate) road, railway, and maritime agencies. To achieve the government's goal of increasing productivity and innovation in the construction market, the STA targets that by 2018, 50% of the total project volume will be realized in the form of design-build (D-B) contracts, allowing the agency to focus its efforts on core competencies and outsource other activities to suppliers.[1] Also in the software industry, outsourcing of IT services is a common and growing business practice.[2] D-B contracts are particularly attractive for the STA as solution providers can innovate, develop, and use components that work as an integrated whole, while also taking over responsibilities and risks for activities that were previously carried out by the client[3]—i.e., the STA.

However, this form of project delivery also requires increased competence in requirements engineering, communication, and management since the design work is outsourced to a solution provider. The STA (and its precursor organizations) recognized this need in the early 2000s and invested resources to improve in the area of requirements engineering. Today, the STA is very conscious about the importance of requirements engineering and management as part of its overall development processes. Therefore, we decided to perform *Process Improvement Archaeology* (PIA) to better understand the current processes and devise new directions for further improvements addressing the currently encountered challenges. Before we look at the PIA steps and its results in more detail, we provide the context and motivation for focusing the investigation on requirements engineering in particular.

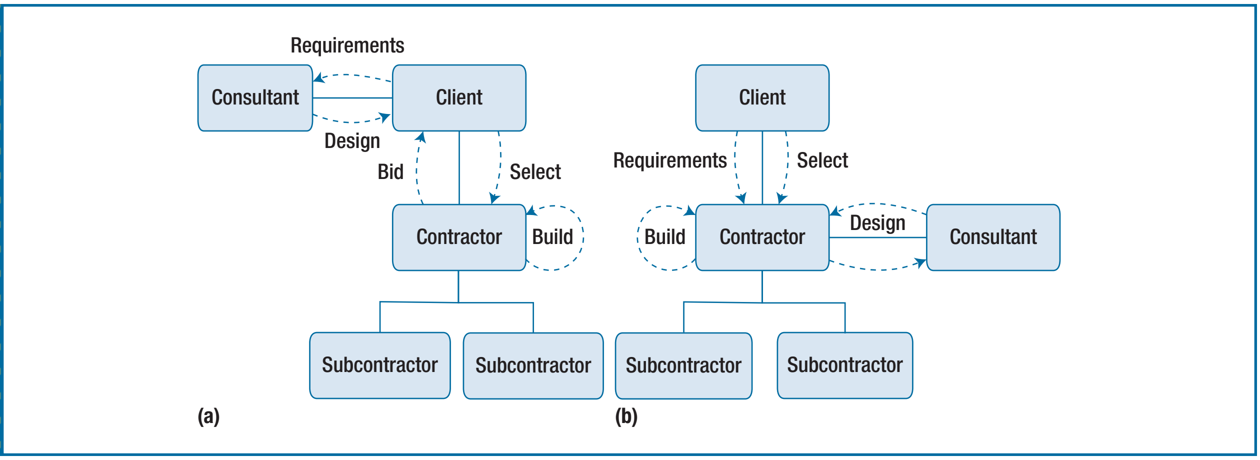

FIGURE 1. The (a) design-bid-build (D-B-B) and (b) design-build (D-B) project delivery approaches. Eight studies investigated the impact of the project delivery approach on the cost and schedule of project completion.[10] Six studies determined advantages in both cost and schedule control for the D-B approach, while only two studies observed advantages in cost and one study observed advantages in schedule control for the D-B-B approach.

A Look over the Tea Cup's Rim

Requirements engineering is a central part of the development of software-intensive products, governing planning, effective implementation, and product quality.[4] This is emphasized even more in projects where the design and implementation are outsourced to a software supplier.[5] Research and practice in this area have therefore sought to improve and validate requirements-engineering concepts in various domains, adapting them to the particular context. A witness to the growing knowledge base on requirements engineering is the 53 systematic literature reviews that were performed between 2006 and 2014, covering nearly 8,000 primary studies.[6] Since requirements engineering can be seen as a branch of systems engineering,[7] our conjecture is that there is a large potential for transferring and applying this knowledge to the construction industry, given the growing need for requirements-engineering competence, as illustrated next.

The predominant paradigm for the development of large infrastructure projects in the 20th century has been the design-bid-build (D-B-B) project delivery system.[8] In this paradigm (see Figure 1a), the project is separated into a design phase and a construction phase.[9] The client commissions a consultant to produce bid documents and technical specifications that meet the client's requirements. The bid documents are used to elicit and then select an offer from competing contractors. The winning contractor is commissioned to implement the project according to the specifications.

The D-B paradigm (see Figure 1b) was the predominant form in preindustrial times and is now witnessing a renaissance in the wake of downsized in-house project management capabilities and costly disputes between design and construction parties.[8] In this project delivery system, the client deals with a single contractor, responsible for both design and construction services.[8] In contrast to D-B-B, design and construction run in parallel, leading to shorter project delivery times and lower total costs (see Figure 1).

The reason why we are interested in studying construction projects is the central role requirements play in the D-B paradigm.[9] In the D-B-B paradigm, requirements can be refined during the design phase of the project, where the client and consultant define needs and solutions collaboratively.[11] However, the D-B paradigm requires that the client's needs are precisely described such that they can be universally understood and interpreted[11] by all involved stakeholders (client, contractor, and consultant). A well-defined and commonly understood scope has been identified as the most important D-B project characteristic.[12]

If these observations sound familiar to the software engineering ear, it is because high-quality requirements are equally important for companies who outsource design, implementation, or testing of software.[13] Given these parallels in

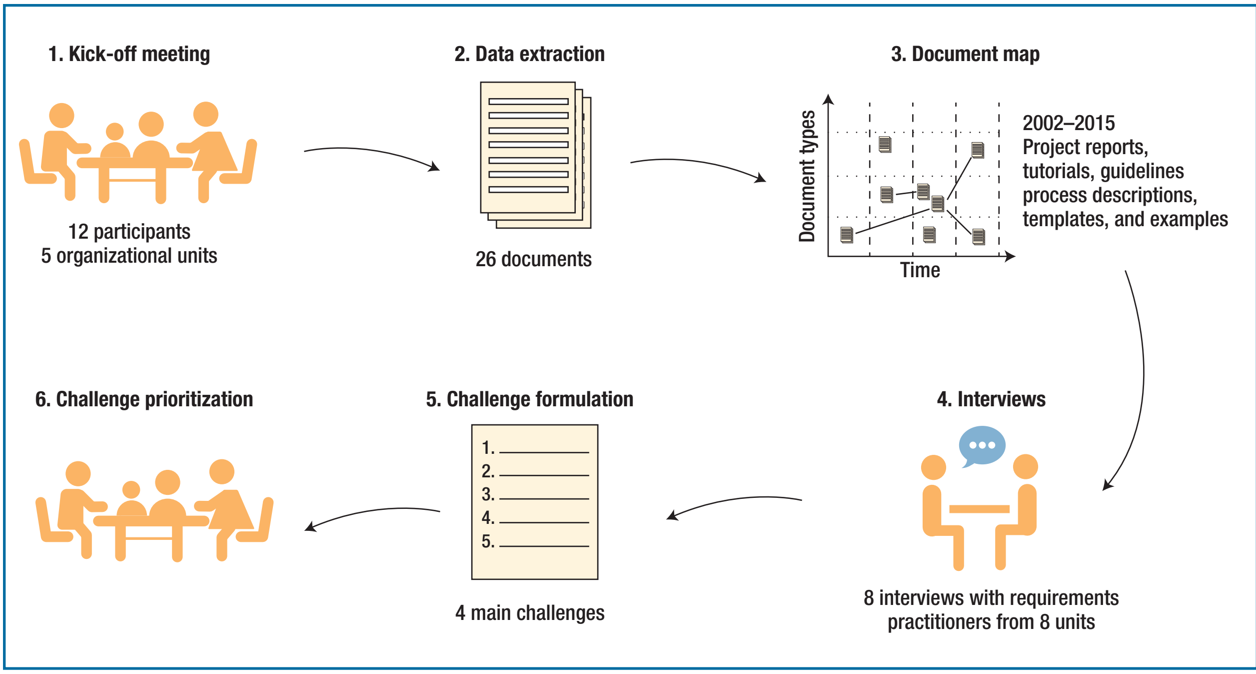

FIGURE 2. The six steps of Process Improvement Archaeology (PIA).

requirements-engineering management between construction and software engineering projects, we sought to understand the STA's current approach to requirements engineering, and the evolutionary steps in their improvement efforts as they provide justification for the current state and indicate directions for future improvements.

Process Improvement Archaeology

The goal of PIA is to learn from past improvement initiatives and their outcomes and identify, based on current data, challenges and improvement opportunities. We developed this method, which is inspired by our previous work on postmortem analysis,[14] to cater to the STA's decentralized improvement strategy, where independent initiatives work toward a common goal. This lack of central organization allows for maximum flexibility in the organizations' units to plan improvements but makes it more difficult to coordinate and benefit from synergetic effects.

Figure 2 illustrates the six PIA steps. We select the participants of the kick-off meeting in step 1 based on their current involvement in improvement initiatives related to requirements engineering and management. It is important here to identify a diverse set of participants from different organizational units. We elicit from the participants a starting set of documentation produced in past investigations. Similar to literature reviews with snowball sampling,[15] the goal is to collect a diverse set of documentation, from as many different sources and authors as possible.

In step 2, we analyze the documentation, extracting the type of the artifact, creation date, authors, purpose and outcome of the investigation, and references to other investigations not included in the original starting set, which we in turn look up and analyze too.

Based on the extracted information, we create in step 3 a document map that helps us understand relationships, or the lack thereof, between investigations. The structured representation of the investigations lets one identify those efforts that match the overall goal of the PIA. For example, if the goal is to better understand why improvement efforts seem to not have any discernable impact, one would select those past initiatives that are not referred to by more recent ones. Another strategy could be to group initiatives according to their purpose or outcome and check whether they refer to each other or not; this allows one to identify initiatives that would benefit from a closer collaboration. Finally,

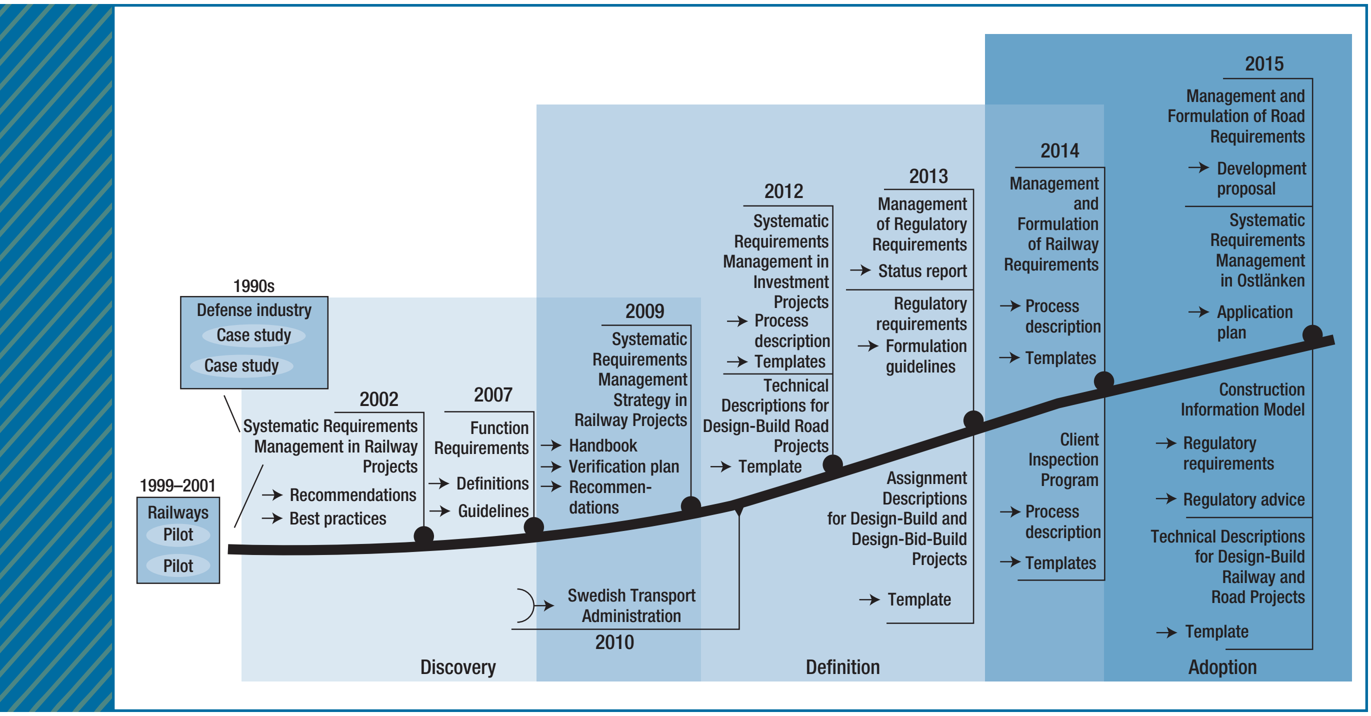

FIGURE 3. The development and adoption of requirements engineering at the Swedish Transport Administration (STA), illustrated by the various documents produced in internal studies and improvement projects. We analyzed 26 documents (arrows indicate the types) dated between 2002 and 2015. After the consolidation of several government transportation agencies into the STA in 2010 and the strategic shift to D-B project delivery, an increase occurred in improvement projects and the associated documentation, targeting requirements-engineering practice.

"fertile" investigations can be identified by looking at often-referred-to documents. This allows one to identify those initiatives that seem to influence the overall improvement direction in an organization.

This analysis allows one to identify central improvement initiatives, their purpose, and the involved stakeholders, which are then interviewed in step 4. The concrete interview questions depend on the goal of the PIA. A typical goal would be to understand the challenges of implementing recommendations that were produced in an improvement investigation. Another goal could be to understand why initiatives with similar purposes are not using each other's results and findings.

In step 5, we compile a list of challenges, based on the results from the conducted interviews, and form an interest group. Together with this group, we prioritize and develop plans to address the identified challenges in step 6.

Learning Requirements Engineering

We now look at the evolution of requirements engineering at the STA, the output of step 3 of the PIA described earlier. Figure 3 illustrates the progression and key documents that can be associated with three overlapping phases in which the STA discovered, defined, and started adopting systematic requirements management.

Discovery Phase

Investigations in the early 2000s drew on experiences from the defense industry on how to perform systematic requirements management, where central concepts such as requirements types and levels, traceability, and verification and validation were analyzed and considered for adoption in railway projects. In pilot studies, customer requirements were broken down into system and functional requirements in order to define alternative designs with different capacities and cost levels.

While the pilot studies concluded that systematic identification and analysis of requirements were useful, it took several years until these principles were evaluated on a broader basis. An investigation in 2009 reported on the outcome of different strategies to introduce systematic requirements management (SRM) within the investment division of the railway agency. SRM is a process with the following six requirements management concepts: identification, formulation, systematization, acceptance, verification, and validation (a seventh concept, change management, was added to the process in 2012). A successful strategy was the introduction of a spreadsheet-based requirements-monitoring plan that allowed the follow-up of the implementation of requirements. Also, an effort was made to adapt existing procedures and templates to raise the quality of the specified requirements.

However, few of the changes were eventually adopted. In addition, it turned out to be difficult to find a project where the process could be piloted, mostly due to not being able to identify a project in the starting phase, but also because it was difficult to change the culture and habits in long-running projects.

Definition Phase

This phase was fueled by the 2010 merger of the different transportation agencies into the STA and the declared goal to increase the number of D-B projects. Since then, processes have been defined and templates have been produced to support this goal (see Figure 3). In order to promote the new processes and templates, already in late 2010 a competence network was founded; as of 2016, it includes more than 130 members from the planning, investment, and maintenance divisions of the STA. The SRM process for investment projects was officially approved in 2012. The stated purpose is to

- ensure that facility requirements are known to all stakeholders when decisions are made about those requirements,
- ensure that no requirements are overlooked during the project, and
- ensure that the facilities' compliance with the requirements can be evaluated.

While the process description is prescriptive in what activities should be performed and when, it does not define how the work is to be conducted. The role of the requirements expert is to coordinate the process implementation and adapt it to the projects' needs.

Besides developing process descriptions, considerable effort was made to collect and synthesize documentation from existing D-B projects into templates that capture reusable domain knowledge. For example, the template from 2012, "Technical Descriptions for Design-Build Road Projects" (extended in 2015 to also include railway projects), exhaustively lists and describes all relevant components of a road project in a hierarchical structure. Each component contains generic blueprints for defining the scope, functions, requirements for materials, and means to verify the implementation that need to be specified for the particular project.

Adoption Phase

In this latest and current phase, we can observe how requirements engineering and management is becoming a central activity at the STA (see Figure 3). For example, the clients' inspection program covers how and when to verify deliveries against requirements related to a set of product attributes (e.g., compliance with safety and security regulations and standards). This program would not be effective without the SRM process. Other indications for the adoption of the SRM process are the extension of the Management and Formulation of Railway Requirements initiative to road projects and the use of SRM in several large infrastructure projects, one of which is Ostlänken, budgeted for US$7 billion with planned completion in 2028.

Requirements Engineering—the Common Ground between Engineering Disciplines

The document analysis illustrates how the STA's decision to increase the number of D-B projects also resulted in increased interest in fostering requirements-engineering competence and expertise. The SRM process, illustrated in Figure 4a, is thereby a common theme that evolved over more than a decade. We decided therefore to focus the PIA interview (see step 4 in Figure 2) on this process and its actual application. We conducted three interviews with experts communicating the process, who directed us then to five more interviewees with expertise on working with the process.

The main goal of SRM is to increase the likelihood of building the correct product from the beginning, reducing cost overruns due to rework and loss of information, and increasing planning and development efficiency. In the interviews, we focused on identifying the encountered challenges in implementing SRM, as discussed next, and identifying software engineering solutions to these challenges and reviews of the solutions (see Figure 4b).

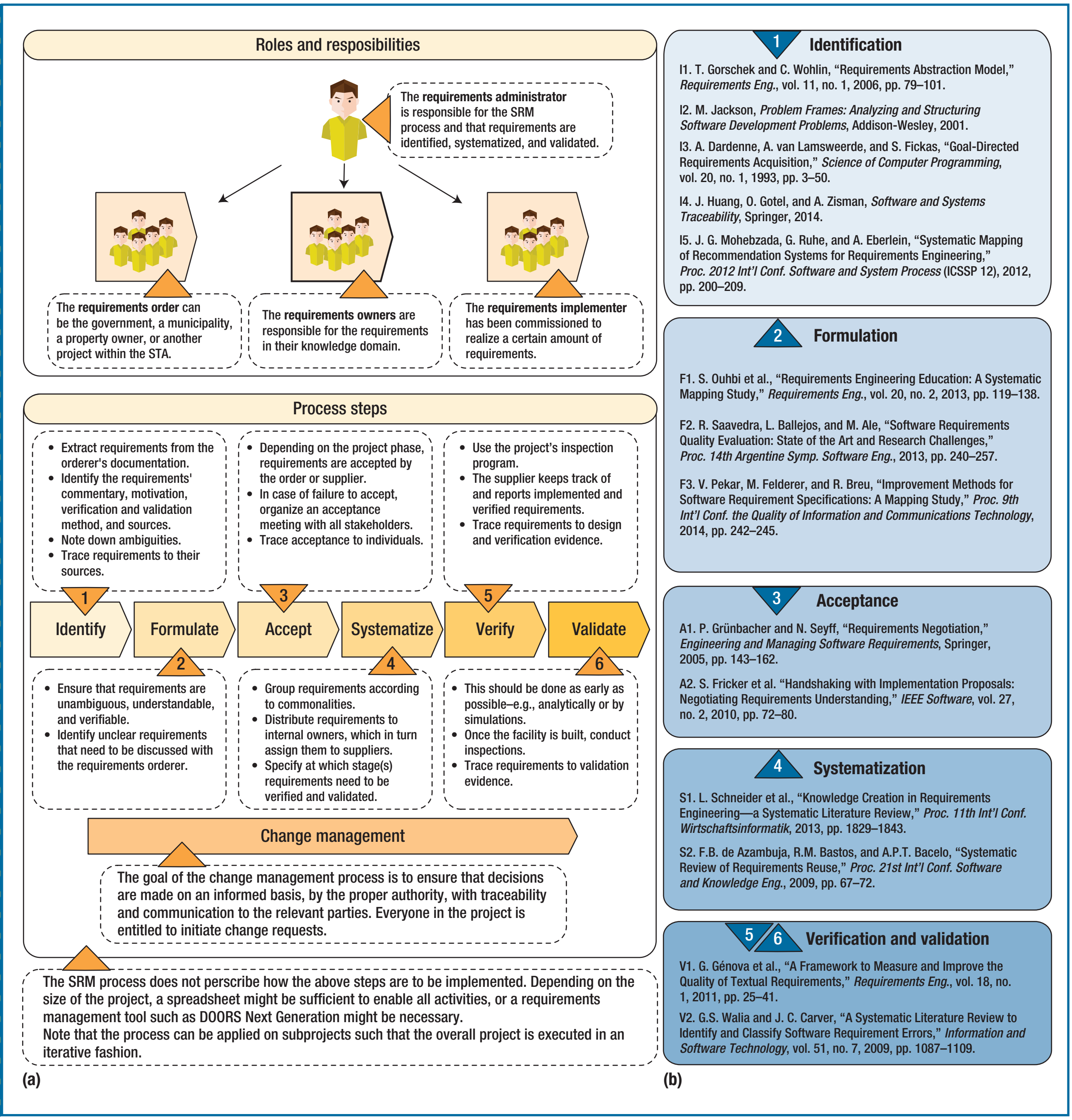

FIGURE 4. Systematic requirements management (SRM) at the STA. (a) Roles and responsibilities and the six process steps. (b) A selection of potential solutions or research reviews from the software engineering literature for each step.

Identification

The STA has a long tradition of designing its facilities together with contractors, following the D-B-B paradigm. The requirements owners (see Figure 4 for a description of the roles) have the technical expertise that leads under the D-B paradigm to requirements that tend to specify solutions rather than define a customer need. One interviewee summarized, "If you choose a solution very early in the process, I don't think you can become a procurement agency. You need to specify your goals, your driving functions, and performance requirements."

One strategy to address this challenge is to place the stated requirements on abstraction levels (see reference I1 in Figure 4) and work up (abstract) those that are on a component level—i.e., close to a solution. This would allow the STA to negotiate product- and function-level requirements with the contractors while still benefiting from in-house technical expertise. In addition, problem framing (reference I2) would be a useful approach to separate requirements from implementations. The combination of and traceability (reference I4) between requirements on different abstraction levels also increases understanding of the requirements, as well as the ability to present only the relevant levels of information to the appropriate stakeholders—in essence, avoiding a flat huge document used by everyone.

Requirements are also elicited from different sources—e.g., the government, property owners, facility users, technical standards, regulations, and laws—that may produce overlapping, conflicting, and not directly comparable needs. These requirements are large in quantity and heterogeneity, making it very costly to analyze them manually. Recommendation systems (reference I5) that support domain analysts to identify conflicting and redundant needs can be useful to improve, in terms of both efficiency and quality, the identification of requirements.

Formulation

The purpose of written requirements is to document and convey information that is understood and used by people with varying domain knowledge. While the STA holds seminars and education events to support the requirements orderers, the analysts and requirements editors reported in the interviews that they need to spend effort in reviewing and correcting faults that could be detected earlier. A requirements analyst explained that "there is little support for formulating high-quality requirements for those who order them, so we analysts need to improve them."

Formulating high-quality requirements is costly, as many different quality aspects could be considered. It is therefore important to prioritize both quality aspects and critical requirements such that "good-enough" formulation quality can be achieved for the most central ones. Approaches to evaluate and improve requirements quality (see references F2 and F3 in Figure 4) span from computer-based support for correctness, completeness, and consistency checking; term recommendations for glossaries; and ambiguity solving to review processes like perspective-based reading.

Acceptance

The specified requirements are the interface between the orderers (stakeholders articulating a need) and implementers (stakeholders fulfilling that need). The interviewees identified requirements acceptance—i.e., where all stakeholders mutually agree on and commit to the stated requirements—as a critical step in the SRM process where cost overruns due to rework can be prevented early in the project: "We have a lot of people wishing things in the project, and it is hard to know when we actually have accepted a requirement in the project."

Requirements negotiation (see reference A1 in Figure 4) is an important aspect that helps to resolve conflicts among stakeholders, reducing the risk of misunderstandings, making tacit knowledge explicit, and helping to find better solutions. Different conflict resolution strategies can be applied, depending on the goal of the negotiations and the mutual trust of the participating parties. Tools can provide passive support (e.g., by enabling collaboration and communication) or active support (e.g. by facilitating the identification of situations with mutual gain). When negotiating with suppliers, implementation proposals (reference A2), a technique applied in some projects at the STA, can be used to clarify requirements and reduce misunderstandings.

Systematization

The main goal of this step is to ensure that the requirements are managed such that the relevant stakeholders have access to the information they need to fulfil their duties. The challenge here lies in the fact that projects can last decades while contractors change, rendering handover and information transfer crucial in order to prevent knowledge loss. As one interviewee stated, "We have people come and go in the project and lose knowledge."

In global software engineering, strategies to geographically distribute knowledge could be of great value in this context, where project duration and stakeholder turnover are the major barriers against knowledge conservation (see reference S1 in Figure 4). The invested effort to systematically manage requirements over an extended period of time also incites requirements reuse, both within and among projects. To enable requirements reuse (reference S2), approaches to formulate requirements independently from solutions (if it makes economical and technical sense) are needed to not limit innovation and progress.

Verification and Validation

In the SRM process, verification refers to the assessment of whether the design fulfills the specified requirements. (A requirement is something that can be ultimately measured somehow in the final product; i.e., it can be concluded whether the requirement was fulfilled or not.) Requirements verifiability is therefore of the essence since it is the contractor's responsibility to produce evidence of fulfillment. Quality aspects that contribute to the verifiability of requirements are their design independence, traceability, unambiguousness, understandability, completeness, and consistency (see reference V1 in Figure 4). Early focus on improving these quality aspects can reduce the amount of rework, in particular when the product is validated. One possible approach to detect low-quality requirements early is to utilize requirements error and source taxonomies (reference V2) that could help to design focused review processes.

PIA turned out to be a useful and lightweight tool that helped us understand the STA's efforts to establish its requirements-engineering process and its current challenges. Learning requirements engineering at the STA was by no means a straightforward endeavor: an initial discovery phase was followed by a definition phase (fueled by a strategic shift to prioritize the D-B project delivery approach) and the current adoption phase, spanning in total over 15 years.

When studying the STA's efforts, we were surprised by not only the parallels to the software industry but also the long time frames it takes to establish new practices, likely related to the project cycles ranging from 3 to 20 years. In software engineering, product and process ideas can be developed and validated in much quicker cycles, allowing for accelerated organizational learning compared to large infrastructure endeavors. Our goal is therefore to support SRM at the STA with proven techniques from software engineering, applying them in a new context, and advancing both systems and software engineering. The two directions we are focusing on are the identification and formulation of requirements.

First, we want to utilize abstraction to enable both feature-level (what) and component-level (how) requirements to be used in combination. This gives the ability of reuse on higher levels of specification and specifics on lower levels of specification—in essence, decoupling the purpose or needs from the solution, while still maintaining technical control. This has been successfully realized utilizing methods like the requirements abstraction model in industry.

Second, it is crucial to formulate requirements such that they are fit for the purpose and support the respective stakeholders in their tasks. Requirements quality needs to be tuned to be good enough while being economically reasonable. Hence, deciding on which quality aspects to improve is equally important as prioritizing requirements that are risky or costly to rework in order to find an acceptable cost–benefit ratio. Such a decision framework would be beneficial for both construction and software engineers.

References

1. T. Ahola et al., "Purchasing Strategies and Value Creation in Industrial Turnkey Projects," *Int'l J. Project Management*, vol. 26, no. 1, 2008, pp. 87–94.
2. B. Meyer, "The Unspoken Revolution in Software Engineering," *Computer*, vol. 39, no. 1, 2006, pp. 124–123.
3. T. Brady, A. Davies, and D.M. Gann, "Creating Value by Delivering Integrated Solutions," *Int'l J. Project Management*, vol. 23, no. 5, 2005, pp. 360–365.
4. D. Damian and J. Chisan, "An Empirical Study of the Complex Relationships between Requirements Engineering Processes and Other Processes That Lead to Payoffs in Productivity, Quality, and Risk Management," *IEEE Trans. Software Eng.*, vol. 32, no. 7, 2006, pp. 433–453.
5. K. Wiegers and J. Beatty, *Software Requirements*, 3rd ed., Microsoft Press, 2013.
6. M. Bano, D. Zowghi, and N. Ikram, "Systematic Reviews in Requirements Engineering: A Tertiary Study," *Proc. 4th Int'l Workshop Empirical Requirements Eng.* (EmpiRE 14), 2014, pp. 9–16.
7. B. Nuseibeh and S. Easterbrook, "Requirements Engineering: A Roadmap," *Proc. Conf. Future of Software Eng.*, 2000, pp. 35–46.
8. M. Konchar and V. Sanvido, "Comparison of U.S. Project Delivery Systems," *J. Construction Eng. and Management*, vol. 124, no. 6, 1998, p. 435.
9. C.W. Ibbs et al., "Project Delivery Systems and Project Change: Quantitative Analysis," *J. Construction Eng. and Management*, vol. 129, no. 4, 2003, pp. 382–387.
10. D.R. Hale et al., "Empirical Comparison of Design/Build and Design/Bid/Build Project Delivery Methods,"

J. Construction Eng. and Management*, vol. 135, no. 7, 2009, pp. 579–587.
11. J.L. Beard, E.C. Wundram, and M.C. Loulakis, *Design-Build: Planning through Development*, 1st ed., McGraw-Hill Education, 2001.
12. A.D. Songer and K.R. Molenaar, “Project Characteristics for Public-Sector Design-Build,” *J. Construction Eng. and Management*, vol. 123, no. 1, 1997, p. 34.
13. T. Gilb, “Requirements for Outsourcing,” *INCOSE Int’l Symp.*, vol. 17, no. 1, 2007, pp. 916–930.
14. M. Unterkalmsteiner et al., “Assessing Requirements Engineering and Software Test Alignment—Five Case Studies,” *J. Systems and Software*, Nov. 2015, pp. 62–77.
15. C. Wohlin, “Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering,” *Proc. 18th Int’l Conf. Evaluation and Assessment in Software Eng.* (EASE 14), 2014, pp. 1–10.

ABOUT THE AUTHORS

MICHAEL UNTERKALMSTEINER is a senior lecturer at the Software Engineering Research Lab at the Blekinge Institute of Technology (BTH). His research interests include software repository mining, software measurement and testing, process improvement, and requirements engineering. Unterkalmsteiner received a PhD in software engineering from BTH. He’s a member of IEEE. Contact him at mun@bth.se; www.lmsteiner.com.

TONY GORSCHEK is a professor of software engineering at the Blekinge Institute of Technology (BTH), where he’s also a research scientist in close collaboration with industrial partners. His research interests include the development of scalable, efficient, and effective solutions for requirements engineering, product management, value-based product development, and agile and lean product development and evolution. Gorschek received a PhD from the BTH Department of Systems and Software Engineering. Contact him at tgo@bth.se; www.gorschek.com.