feature

technology transfer

A Model for Technology Transfer in Practice

Tony Gorschek and Claes Wohlin, *Blekinge Institute of Technology*

Per Garre, *Danaher Motion Särö AB*

Stig Larsson, *ABB Corporate Research*

Successful technology transfer requires close cooperation and collaboration between researchers and practitioners. A seven-step transfer model embodying this philosophy emerged from two academic-industry partnerships.

Technology transfer, and thus industry-relevant research, involves more than merely producing research results and delivering them in publications and technical reports. It demands close cooperation and collaboration between industry and academia throughout the entire research process. During research conducted in a partnership between Blekinge Institute of Technology and two companies, Danaher Motion Särö AB (DHR) and ABB (see the "Industry Partners" sidebar), we devised a technology transfer model that embodies this philosophy. We initiated this partnership to conduct industry-relevant research in requirements engineering and product management. Technology transfer in this context is a prerequisite: it validates academic research results in a real setting, and it provides a way to improve industry development and business processes.

Our model involves seven steps, each building on the ones before it. Although previous transfer models, such as the one by Shari Lawrence Pfleeger,[1] inspired our model, its complete development evolved over time. These steps emerged during a long-term joint commitment, and we added new steps, as needed, along the way. This evolution also dictated the activities we performed in each step. For example, the best way to validate new technology depends on the company's needs and which validation processes the company trusts, as well as the researchers' need to validate new technology for academic purposes. Considering industry preferences when performing validation is important for success.[2]

In this article, we present our technology transfer model, and we report our experiences and lessons learned for each of the seven steps involved. Figure 1 outlines this model, showing how all seven steps are relevant and interdependent for overall transfer success.

Step 1: Identify potential improvement areas based on industry needs

We began by assessing current practices, observing domain and business settings, and identifying the demands imposed on industry.[3] Observation of the real world before formulating research questions is critical.[4] Research must connect to the needs that on-site practitioners perceive, or their commitment could be difficult to obtain.

During this stage, we identified several potential areas for improvement in product management and requirements engineering. We subsequently analyzed these areas and prioritized them according to perceived importance and dependency.[5] Here, *dependency* means transferring future improvements in a certain order—avoiding "hitching the cart in front of the horse," you might say.

Lessons learned

- Researchers' on-site presence helps base the research agenda on real industry-relevant issues. It also helps build the technical, organizational, social, and cultural understanding necessary to do a good job.
- Having a friendly, easily recognizable presence gives researchers extensive access to all practitioner groups.
- Information gathering should be balanced to avoid giving more attention to the most vocal practitioner groups at the expense of the quieter ones.
- It's essential to involve all affected parts of the organization, including upper management, middle management, engineers, support, marketing, and sales.

Step 2: Formulate a research agenda

According to the prioritized needs identified in the previous step, we formulated a research agenda, in close cooperation with our industry contact persons, who were middle-management practitioners. At this time, these contact persons increased their commitment and moved from participants to active *champions*. They began contributing substantially with input and facts, and they made it possible for us to quickly and easily access the company organizations. The researchers also had a regular on-site presence—at first learning and observing, and then gradually more actively exchanging ideas and concepts with multiple practitioners, not only the champions. Doing your homework and learning the domain establishes a common understanding and vocabulary. Close proximity between researchers and practitioners is a critical factor for success.[4]

In our case, the main needs identified as high priority revolved around moving from bespoke customer-centered development to market-driven, product-centered development. One of the main challenges in a market-driven environment is handling large volumes of requirements from multiple sources. These include both internal sources (developers, marketing, sales, support personnel, and bug reports) and external sources (users, customers, and competitors—whose information is often gathered through surveys, interviews, focus groups, and competitor analysis).[6] This large volume of requirements makes initial screening important to decrease the risk of overloading in the evaluation and realization process.[7] In addition, the requirements themselves come in all different shapes and sizes, and at different abstraction levels. Some are very general and sound more like goals than requirements; others are detailed and technical. It's important to be able to handle all of them, checking them against product strategies and roadmaps and prioritizing them. In our case, both companies had similar needs and thus a vested interest in addressing many of the same issues.

Lessons learned

- Doing your homework means learning company and domain-specific vocabulary and understanding the practitioners' situation. This is necessary to build trust.
- Problem formulation is the basis for the research agenda—that is, what is to be done and improved. As practitioners participate in this process, they formalize their roles as industry champions for the research and make long-term commitments to it.

Industry Partners

Danaher Motion Särö AB develops and sells software and hardware equipment for navigation, control, fleet management, and service of automated-guided-vehicle systems. More than 50 AGV system suppliers worldwide use DHR technologies and expertise in their own products. The headquarters and R&D center are located in Särö, Sweden, and there are 85 employees there.

ABB is a leader in power and automation technologies that help utility and industry customers improve performance while lowering environmental impact. The ABB group of companies operates in about 100 countries and employs approximately 102,000 people. The transfer of new methods for requirements engineering involved one of the ABB development centers in Sweden. The product development part of this organization has 200 employees.

Both companies are participating in a joint six-year research project with the Blekinge Institute of Technology in the area of process improvement and requirements engineering. The collaboration in requirements engineering began in late 2002 with DHR; ABB joined in late 2003.

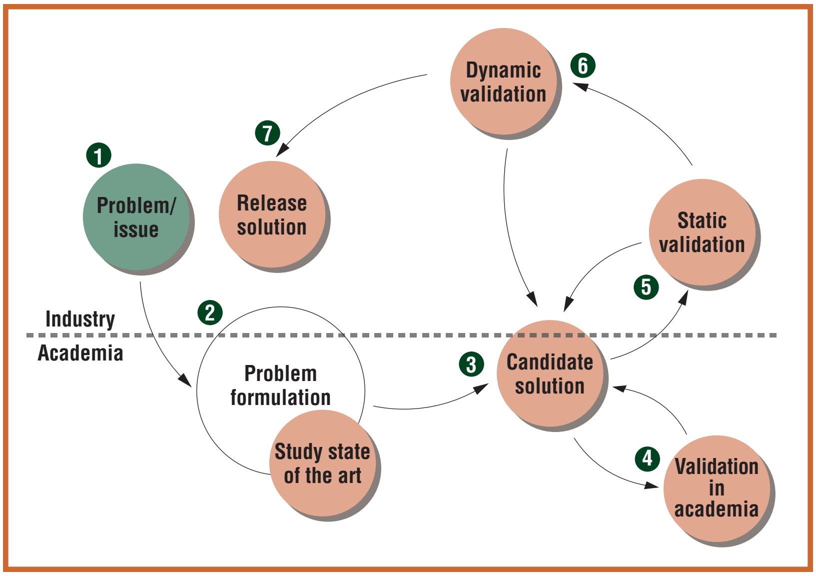

Figure 1. Overview of research approach and technology transfer model. 1. Identify potential improvement areas based on industry needs, through process assessment and observation activities. 2. Formulate a research agenda using several assessments to find research topics, and formulate problem statements while studying the field and the domain. 3. Formulate a candidate solution in cooperation with industry. 4. Conduct lab validation (for example, through lab experiments). 5. Perform static validation (for example, interviews and seminars). 6. Perform dynamic validation (for example, pilot projects and controlled small tests). 7. Release the solution step by step, while remaining open to smaller changes and additions.

Step 3: Formulate a candidate solution

After establishing a research agenda, the collaboration with industry continued with the design of a candidate solution. We designed a requirements engineering model called the Requirements Abstraction Model (RAM).[8] The purpose of this model is to incorporate possible solutions for many of the needs identified during the assessments at DHR and ABB; it primarily offers product planning and product management support. RAM is a multilevel requirements abstraction model, with a supporting process that aids practitioners in handling requirements in a structured and repeatable way during requirements elicitation, analysis, refinement, and management. The nature of the model is to use the fact that requirements come at different abstraction levels instead of trying to flatten all or mix different types in a document. Using RAM makes requirements abstraction (checking them against strategies) and breakdown (refinement to a testable format) part of the analysis and refinement work. Hence, it's possible to compare and prioritize requirements, because they are homogenous at each abstraction level

We created this candidate solution (RAM) in collaboration with practitioners. The researchers' main responsibility was to monitor the state of the art in research and combine this knowledge with new ideas and angles. Another reason to collaborate with practitioners is to keep research focused on real industry needs. A common problem is that research solutions don't fit with present business and development methods,[9,10] thus increasing cost and raising the bar for technology transfer.

Lessons learned

- Besides being a valuable resource, practitioners can provide a reality check, making sure a candidate solution is realistic and fits current practices and the company's situation.
- In formulating a candidate solution in collaboration with practitioners, commitment and trust are key. Moreover, the champions need to communicate and share ideas and information with colleagues, preparing for a change in the mind-set throughout the organization.
- Creating new solutions to identified issues is tempting. It's important that the researchers act as the link to the state of the art in research, ensuring that techniques, processes, and tools already developed and validated aren't ignored. In our case, this meant building on and refining some research results obtained by others, and adding new technology as necessary.

Evolution and transfer preparation through validation

As we formulated the candidate solution, we recognized a need for evaluation. So, we introduced several validation steps to accomplish this goal. The idea is to refine the candidate solution, test it for usability and scalability, and determine whether it addresses the needs satisfactorily. In addition, the validation steps gradually prepare for technology transfer. In this case, the solution itself must evolve on the basis of feedback from validation, but the validation steps can also prepare the company for change. Preparation means showing the people in the organization that using the new solution is more advantageous than doing business as usual. This is critical for getting commitment to the technology transfer—something researchers often miss.[9]

Step 4: Conduct lab validation

We evaluated RAM in a university lab environment both before and during static validation in industry. We conducted the evaluation in an experimental setting using software engineering graduate students as subjects. These students used the model and performed requirements engineering activities as we envisioned industry practitioners performing them. The results made it possible to catch some issues without using industry resources. In addition, we received early input on the model's usability and scalability. Early validation in a lab environment helps ensure that pressure to transfer all research results indiscriminately doesn't gain the upper hand.[11]

Lessons learned

- An initial practical test of the candidate solution in a lab environment can provide fast, valuable feedback, identifying obvious flaws so that you can fix them before industry piloting.
- The results are useful for presenting the candidate solution to practitioners and management and convincing them of both manageable risks and potential benefits.
- Lab validation isn't the same as live industry use. Realizing the limitations adds to the validity of the lab evaluation when presenting it to industry.

Step 5: Perform static validation

Static validation involved widespread presentation of the candidate solution in industry and collecting feedback from practitioners. The first phase of static validation presented the model (RAM) to all industry personnel involved in the initial process assessment in step 1. Practitioners representing developers, designers, project managers, product managers, and marketing and sales viewed the candidate solution through several seminars. This allowed the practitioners who would use the model to voice their opinions early. In addition to receiving feedback from practitioners, which further improved the model, the seminars also let us give practitioners feedback. Involvement and two-way communication between researchers and practitioners not only can improve results; it can also lay the foundation for meeting critical technology transfer goals such as shared commitment between researchers and practitioners—that is, getting support for process change at all levels of the organization.[4] In addition to holding seminars, we conducted several follow-up interviews.

The second phase of static validation presented the research results and model exclusively to upper management in a series of interactive seminars. Upper management controlled the resources needed for taking the next steps; if they couldn't see the benefit of the solution, the quality of the research would be irrelevant in terms of transferring it to practice. High-quality research is not sufficient by itself. Management support is crucial,[4,10] and this fact cuts both ways. On the one hand, selling an idea (in this case, the model) to management often refines arguments and can be a good control point in terms of committing resources for technology transfer. On the other hand, at this early stage the research is unproven in practice, so it's difficult to show objective facts such as metrics to motivate the transfer. These seminars also showed that the model had the support of operative personnel and middle management in the organization.

The bottom line for static validation wasn't merely to sell the model to industry representatives. The seminars and interviews gave invaluable feedback regarding the model's usability and scalability. For example, we stripped away about 15 percent of the model's contents (requirement attributes, validation steps, and so on) during this stage. What had worked in theory (and gotten past lab evaluations in step 4) needed to be slimmed down for implementation in practice. Very early during static validation, we realized that a smaller model that produced good-enough requirements and decision-support materials was preferable to a larger model that risked not being used at all.

Lessons learned

- Widespread presentation of candidate solutions in the organization has several purposes: getting feedback and ideas for improvements, validating understanding and coverage, and giving feedback to the practitioners involved in the assessment phase in step 1.
- Anchoring future change at all organizational levels is crucial.
- Upper-management seminars and interactive sessions are critical for assessing the risks and relative benefit of the changes proposed in transferring the candidate

If upper management couldn't see the benefit, the quality of the research would be irrelevant.

The relative success of dynamic validation increased the model's credibility.

solution. It's important to show that researchers developed the candidate solution in cooperation with practitioners in the organization, and that it has their support.

- Researchers must not be afraid of tailoring, or even downscaling, the candidate solution at this stage. Change is part of validation and refinement, and it's natural as the ideas presented mature and grow over time.

Step 6: Perform dynamic validation (piloting)

We performed dynamic validation through two pilot projects, both performed at DHR. The first was limited in scope: using the model to elicit, analyze, refine, and manage about 75 requirements. Researchers and an industry project manager (champion) conducted this work. The product specified using the model was an in-house product configuration tool, and the customers were also situated in-house.

The success of this first pilot motivated DHR to continue the dynamic validation with a larger pilot. The second pilot involved using the model in a real development project, without the direct involvement of the researchers or industry champions, although the model used in the first pilot was reused to some extent. This project had 18 developers and ran for about four months (3,500 person-hours). The project manager was the primary requirements engineer; the developers used the requirements produced with the model in their work.

Both pilots gave valuable feedback, although not many new critical problems or potential difficulties were reported. The primary feedback was that tool support, as well as tool adaptation, was important for model scalability, and that it's important to incorporate training efforts in future model implementations. The main reasons there were no surprises during piloting included the static validation and the continual collaboration between researchers and industry champions. The relative success of dynamic validation increased the model's credibility and built trust for us working with the model in the eyes of practitioners and management.

Lessons learned

- Piloting in industry makes it possible to realistically evaluate a candidate solution without giving up control. Piloting minimizes risk, because the pilot is a limited test.
- Pilots can provide input for further improvements and indicate what is needed during the full-scale transfer.
- Because pilots are limited in scope, they might not catch all potential problems, such as scalability issues.
- To some extent, pilots precede proper transfer (for example, training isn't formalized, and the solution itself isn't released). Therefore, the company might use the candidate solution differently from what researchers initially intended.

Step 7: Release the solution

After gauging the results from the static and dynamic validations, the team of researchers and industry partner representatives decided to go forward with the actual implementation of the model: the official release. DHR would fully implement the model and incorporate it in the official development and management process. ABB implemented a trial release of the model, initially limiting it to one large release project; full implementation is pending results from this trial release. Despite this difference, the technology transfer process and the techniques used to accomplish it were largely the same for both companies. We had refined RAM through the validations and pilots, but it was rather general in nature, a least common denominator suiting both companies. A central part of the rationale behind the model was that one size does not fit all—an important research lesson from working with two companies. For this reason, we designed RAM so that it can fit different organizations, depending on that organization's needs—taking into account, for example, differences in types of requirements, specific domains, and so on.

Model tailoring

Tailoring to fit each company was necessary for several reasons. For example, the two companies had different definitions and vocabulary pertaining to products and development, and we had to consider specific practices and processes already in place to align RAM for transfer.

One-day workshop: Tailoring and consensus. The first stage in model tailoring involved a full-day workshop at each company. Participants included managers, developers, and especially product and project managers (the future main users of the model). Process owners were also present. The focus was on adapting the model to suit current practices and, when possible, mapping the roles and responsibilities established by the model to existing ones. We also explored concepts such as "good enough," establishing some examples of requirements through actual specification during the workshop.

We used the model to jointly specify, refine, and analyze requirements, taken from within the company to keep the process as real as possible. As we worked with the example requirements, the researchers and industry champions took turns leading the workshop and documenting the results. This requirements specification segment of the workshop proved to have a grounding effect, keeping discussions from becoming too abstract or theoretical.

Obtaining consensus among all groups represented and ensuring that compromises made didn't impede usability were the main objectives of the workshop—thus, also promoting future commitment.

One-week alignment: Formalization and planning. The information gathered during the workshop was critical for formalizing and planning the model's implementation into the official development and business process. In short, we had to convert workshop results into practice, starting with the development of needed documentation, user guides, example requirements databases (reusing requirements from the workshop and creating additional ones), and tool support. Early in the process of developing and validating the model, we realized that a large, complex, document-heavy model demanding too much from practitioners might have been transferred to the official development and business process, but it would probably have remained on the shelf collecting dust, next to other heavy frameworks and certifications.

The formalization and planning consisted of addressing several items:

Documentation concerned two main items, which we separated to the largest possible extent. Formal documentation addressed all mandatory official process descriptions: role descriptions; process maps; detailed state diagrams of requirements, all mapped to existing processes; glossaries; and so on. These official documents were necessary primarily as reference material, publicly accessible on a need-to-use basis. The user documentation was most important. It consisted of a thin, five-page quick-reference guide.

Training received a high priority, as it does in most process improvement efforts,[9,10] and it benefited from the collaborative research process in general. Practitioners in the two companies had already assimilated many aspects of the model, as a result of their part in creating or validating it. This made training easier, especially in terms of commitment. We used a learning-by-doing strategy for the main parts of the training. The champion led training sessions, and one or two practitioners used the model to engineer new real requirements. We nicknamed this approach *pair requirements engineering* (as in pair programming). This approach had two benefits: First, it provided a good way to learn and gain experience. Second, performing actual work during training sessions decreased overall costs.

Example-driven work practices was one of the main points when RAM was constructed and thus also dominated the training. The company-specific requirements developed during the workshop and the alignment week served as good-practice examples (notice that we don't say "best practice"). The examples themselves constituted an important part of the documentation, in that practitioners could use them during training and later as an information source in their day-to-day work. For this reason, we tried to categorize some of the most common types of requirements and create a database of examples of each category.

Tool support was also important. During the week, we adapted the tool chosen by the respective company. Because the model provides a way to work with requirements, it's largely tool independent. This was evident from the companies' choosing different tools to support the model.

Technical support can greatly influence the success of a technology transfer effort.[11] Practitioners must have continuous access to support from an expert, or at least an expert user. Practitioners who are frustrated or stuck need help fast, or there's a risk that they'll go back to performing requirements engineering as they did previously. Both DHR and ABB have champions who fill the role of expert.

Owner-champion is a role that researchers and practitioner champions to some extent share during research and validation of the solution candidate. However, as the candidate evolves and is released, ownership must transfer to the company and, more specifically, to

A large, complex, document-heavy model would probably have remained on the shelf collecting dust.

The researchers' job isn't just research; it's making technology transfer happen.

the relevant on-site champions. This is necessary because researchers can't be present and active in supporting practitioners to the extent necessary. Optimally, with time and use, all the practitioners using the model will collectively own and champion it. This, however, doesn't mean the researchers' presence or participation is at an end. Quite the contrary, researchers must conduct follow-up studies.

Metrics—the collection of data regarding the model's use—are critical for enabling measurement and follow-up studies. During the alignment week, we formulated a plan for which metrics to collect. It was important to enable measurement of the model use. Metrics can indicate benefits or hint at potential problems. In addition to conducting quantitative measurements, we also planned qualitative follow-ups such as interviews and group discussions. This combination of quantitative and qualitative feedback can provide valuable input for future changes and modifications.

Lessons learned

- One size doesn't fit all. A general candidate solution devised through research results won't fit every situation; adaptation and tailoring are often prerequisites to successful transfer.
- Tailoring as a way to get consensus ensures further commitment.
- It's not possible to satisfy every need or solve every problem.
- Supporting a candidate solution involves several tasks: developing lightweight, example-driven, domain-specific documentation and reference guides; incorporating an efficient training strategy, perhaps using pair requirements engineering to streamline costs; choosing, acquiring, and testing tools tailored to specific needs, and offering training on how to use them; and providing technical support to practitioners (champions can fill this role).
- Measurement programs are necessary and need to be planned to gather relevant information regarding the candidate solution's operation.

Looking back, several issues come to mind as critical success factors. Securing commitment, using champions, fostering collective ownership, building trust, and providing the right kind of training are all examples. We will not try to cover all of them here, but rather we mention a few things in addition to the experiences shared thus far—all in perfect hindsight, of course.

First, it's critical to secure *long-term commitment* from both researchers and practitioners. Neither research nor technology transfer is a one-shot deal. Technology transfer happens over time—with small, incremental, and sometimes unplanned improvements to the overall research effort—and is adopted by practitioners continually. This can be frustrating from a research perspective because it's hard to measure the impact of such "improvements." The collaborative research process itself can be seen as implicit technology transfer, under favorable conditions. "Long-term" in this case applies to the lead time required for new technology to be invented and transferred to practice. This is partially because practitioners perform their daily work in addition to working with the researchers. This is not the same thing as a lack of commitment. Rather, practitioners are pressed for time because they operate in a competitive environment.[11] Performing lab validation before industry trials is a good idea because it can help catch issues without consuming resources.

Second, it's important to consider *risk minimization.* The ability to maximize potential benefit and minimize the potential risk is of paramount importance to companies. The validation in iterations improved the model and helped us convince management and practitioners (as well as ourselves) that the risks were acceptable compared to the potential benefit. The researchers' job isn't just research; it's making technology transfer happen. Active participation in some activities that by themselves don't produce research results (papers) can be considered as overhead from a research perspective but as a necessity from an industry perspective. We were fortunate in our endeavors, because both the researchers and practitioners had a pragmatic take on things. Research results were considered important, but from our point of view, the value of these results were directly linked to usability and usefulness in industry.

Finally, in terms of *technology transfer*, we'd like to point out that our technology transfer model was not prepackaged. We added the steps and their contents on demand, as appropriate.

In some cases, this meant additional work—for example, to provide sufficient evidence to management of a particular approach's relative value. Thus, flexibility is an important ingredient of successful technology transfer. Openness to modifying and adding steps and iterations is necessary to satisfy industry demands of risk minimization and relative value evidence as well as researcher needs. Our technology transfer model should be seen as an instantiation of a technology transfer process—more for inspiration than prescription.

Acknowledgments

The collaboration with the industry professionals at ABB and Danaher Motion has been an invaluable learning experience. We thank all involved for their help, patience, and enthusiasm for new ideas, and for seeing possibilities and not obstacles.

References

1. S.L. Pfleeger, "Understanding and Improving Technology Transfer in Software Engineering," *J. Systems and Software,* vol. 47, nos. 2–3, 1999, pp. 111–124.
2. S.L. Pfleeger and W. Menezes, "Marketing Technology to Software Practitioners," *IEEE Software,* vol. 17, no. 1, 2000, pp. 27–33.
3. T. Gorschek and C. Wohlin, "Identification of Improvement Issues Using a Lightweight Triangulation Approach," *Proc. European Software Process Improvement Conf.* (EuroSPI 03), Verlag der Technischen Universität, 2003, vol. VI, pp. 1–144.
4. V.R. Basili et al., "Lessons Learned from 25 Years of Process Improvement: The Rise and Fall of the NASA Software Engineering Laboratory," *Proc. 24th Int'l Conf. Software Eng.* (ICSE 02), ACM Press, 2002, pp. 69–79.
5. T. Gorschek and C. Wohlin, "Packaging Software Process Improvement Issues—A Method and a Case Study," *Software: Practice & Experience,* vol. 34, no. 14, 2004, pp. 1311–1344.
6. D.R. Lehmann and R.S. Winer, *Product Management.* McGraw-Hill, 2002.
7. M. Weber and J. Weisbrod, "Requirements Engineering in Automotive Development: Experiences and Challenges," *IEEE Software,* vol. 20, no. 1, 2003, pp. 16–24.
8. T. Gorschek and C. Wohlin, "Requirements Abstraction Model," *Requirements Eng. J.,* vol. 11, no. 1, 2006, pp. 79–101.
9. P. Morris, M. Masera, and M. Wilikens, "Requirements Engineering and Industrial Uptake," *Proc. 3rd Int'l Conf. Requirements Eng.*, IEEE Press, 1998, pp. 130–137.
10. S. Miller, "How Can Requirements Engineering Research Become Requirements Engineering Practice?" *Proc. 3rd IEEE Int'l Symp. Requirements Eng.* (RE 97), IEEE CS Press, 1997, p. 260.
11. H. Kaindl et al., "Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda," *Requirements Eng.,* vol. 7, no. 3, 2002, pp. 113–123.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.

About the Authors

Tony Gorschek is a research associate at the Blekinge Institute of Technology, Sweden. He also works as a consultant and manages his own company. His research interests include requirements engineering, technical product management, process assessment and improvement, and quality assurance. He conducts most of his research in close collaboration with industry. Gorschek received a PhD in software engineering from the Blekinge Institute of Technology. He is a member of the IEEE. Contact him at Blekinge Inst. of Technology, PO Box 520, SE-372 25 Ronneby, Sweden; tony.gorschek@bth.se.

Per Garre is a project requirement coordinator at Sony Ericsson. He was a requirements manager at Danaher Motion Särö AB, Sweden, during his work on this project. His technical interests include project management, requirements management, and aircraft technology. Garre received an engineering degree from Katrienlundsgymnasiet, in Gothenburg, Sweden. Contact him at Sony Ericsson Mobile Communications AB, SE-221 88 Lund, Sweden; per.garre@sonyericsson.com.

Stig Larsson is a senior principal scientist at ABB Corporate Research, Sweden, responsible for the company's product development process improvement initiative. His research interests include product development processes, process improvement, and product integration. Larsson received an MSc in electrical engineering from the Royal Institute of Technology in Stockholm. Contact him at ABB AB, Corporate Research, GFCR/AT, SE-72178 Västerås, Sweden; stig.bm.larsson@se.abb.com.

Claes Wohlin is a professor of software engineering and the provost of the Blekinge Institute of Technology. He is also a visiting professor at Chalmers University of Technology in Göteborg. His research interests include empirical methods in software engineering, software metrics, software quality, and requirements engineering. Wohlin received a PhD in communication systems from Lund University. He is a member of the IEEE and the ACM. Contact him at Blekinge Institute of Technology, PO Box 520, SE-372 25 Ronneby, Sweden; claes.wohlin@bth.se.