Research Prototypes versus Products: Lessons Learned from Software Development Processes in Research Projects

Dietmar Winkler, Richard Mordinyi, and Stefan Biffl

Institute of Software Technology and Interactive Systems, CDL-Flex,
Vienna University of Technology
Favoritenstrasse 9/188, 1040 Vienna, Austria
`{dietmar.winkler,richard.mordinyi,stefan.biffl}@tuwien.ac.at`

Abstract. Software and systems development in industry typically focus on constructing high-quality products by using traditional or agile software processes and applying established tools and methods. Most projects have to handle more or less stable requirements but usually build on a proven architecture. On the other hand, research projects typically aim at investigating new ideas, facing promising research directions, showing feasibility of novel approaches or building prototypes for demonstration purposes. Obviously there seems to be a big gap between industrial projects and research projects. Anyway – after a period of research – there is the need to enable the transition from prototype to real products, comparable to industrial developed software products. The main challenge is bridging the gap between research prototypes and industry products, typically out of scope of a research project. As we have to handle these challenges in a long-running research project, this paper aims at identifying risks, challenges and candidate solutions to identify how to bridge the gap from research to industry. Main result of this paper is an adapted software engineering process that has been initially evaluated in context of our research project.

Keywords: Software Development Processes, Research Projects, Industry Product, Prototyping.

1 Introduction

Typical industry projects follow defined software engineering processes [16], e.g., traditional and/or agile development approaches, using well-defined methods and tools [1][6] covering important steps of the software development project, e.g., requirements elicitation, architecture definition, code construction, testing, and delivery/deployment of new and updated software products. Based on standardized software processes, process tailoring approaches support the application and adaption of software processes (out of the box) to a specific organizational or project context [5]. Based on adapted processes, methods, and tools companies typically implement and use well-established approaches within an organization, following a common goal: i.e., delivering high-quality (software) products to customers at an optimum cost/benefit ratio for all involved stakeholders.

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 48–59, 2013.

On the other hand, research projects typically focus on different topics and different goals, e.g., investigating new ideas or facing promising research directions. Based on several stakeholders, e.g., funding organizations, principal and application industry partners, researchers, and developers, various and partly conflicting interests and goals are observable. Unclear, unstable, and frequent changing requirements are additional challenges to be addressed within a research project: new ideas come up frequent and can result in fundamental changes of the solution concept and – as a consequence – of the prototype solutions. However, typical outcomes of research projects are concepts, feasibility studies of novel approaches, and prototype applications for demonstration purposes. An important issue focuses on the empirical evaluation of concepts and research prototypes [3], i.e. demonstrating that the solution work like expected. In contrast to industry projects and products, research projects and prototypes typically have strong limitations regarding application capabilities in industry, and, thus, require appropriate processes, methods, and tools to transfer prototype solutions to industrial products. Based on these basic differences, we derived an important research questions:

How can we bridge the gap between (a) research projects and industry projects and (b) research project prototypes and industry products?

In this paper we report on challenges and candidate solutions based on experiences from our research project, i.e., CDL-Flex[1], a seven-year research project, started in 2010. The main objective of the project is to support engineers in large-scale engineering projects to (a) better collaborate and exchange data between different disciplines and (b) to improve the engineering process [4]. The project focuses on the automation systems domain (e.g., hydro power plants and steel mills) where engineers coming from various disciplines, e.g., the mechanical, electrical, and software domain, have to collaborate and exchange data efficiently. Efficient data exchange is a pre-condition for change management [18], even if different disciplines are involved. Please see Section 3 for a more detailed description of the research project. After three years of research, our industry partners claim to apply selected use cases in (his) industry environment. Nevertheless, the outcome of the research project is still classified as a (research) prototype. Thus, we need some mechanisms to transfer the prototype use cases to an industry product. In this paper we address these challenges (over time) and present a candidate solution for a software development process to support the transfer from research results to industry solutions, and report on findings and lessons learned after a three year period of our research project.

The remainder of this paper is structured as follows. Section 2 presents some related work in context of software development processes. Section 3 introduces to the CDL-Flex research project in more detail. We highlight research challenges in section 4 and present our candidate solution and first results in section 5. Finally, section 6 concludes and identifies future work.

---

[1] CDL-Flex: Christian Doppler Laboratory „Software Engineering Integration for Flexible Automation Systems", `http://cdl.ifs.tuwien.ac.at`

2 Related Work

This section summarizes related work on engineering processes in Section 2.1, methods and tools in Section 2.2, and product/process maturity levels in Section 2.3.

2.1 Software Engineering Processes

Research projects typically focus on an experimental development of new process, methods, and tools to gain knowledge in defined areas by using scientific methods [2]. Nevertheless, it has to be shown that concepts, derived from research activities work in a defined context. Thus, candidate outcomes of research in software engineering result in creating software solutions and prototypes focusing on defined purposes. Based on the nature of research projects, i.e., handling (a) highly instable requirements and (b) instable architecture approaches, the flexibility of engineering processes is a key requirement for research projects. Thus, an important question is which software processes are flexible enough to enable the construction of prototypes and products in context of research projects and – after successfully evaluating these prototypes – which processes aims at bridging the gap between research visions, prototype products, and industry products. In industry, several software process approaches, either traditional or more flexible (agile) process approaches are available for constructing industry products [16]. Among others, the Rational Unified Process [8] or V-Model XT [14] are candidates for application in research projects. Nevertheless, the structure of these processes and the pre-condition of more or less stable requirements might hinder successful application in research projects, where concepts, architecture, and implementation may change frequently. Agile processes, e.g., Scrum [15] or eXtreme programming, seem to fit well to research projects as they focus on user interaction and flexibility and support fast feedback-loops of individual stakeholders. Nevertheless, a stable baseline of tools, methods, and development environments is a pre-condition for software development. Unfortunately (early) research projects do not provide this kind of stable baseline. More current approaches, e.g., Lean Development or Kanban [9] can also provide an organizational framework for software construction – nevertheless, similar critics apply for research project application – there must be something stable to build on.

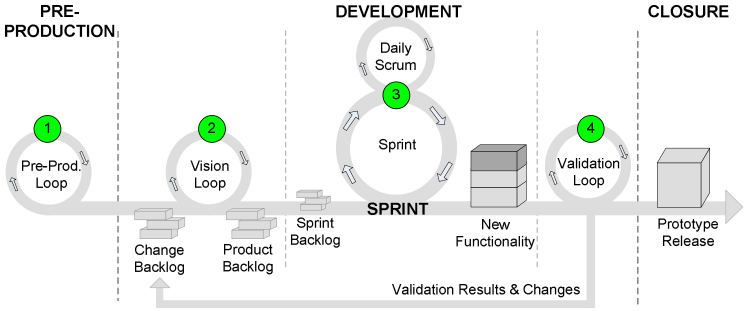

Fig. 1. Extended Scrum Process Model according to [10]

From the authors' perspective, research projects also include creative work to find new and promising solutions – similar to game development domain [10]. Thus, processes derived from game development are promising candidates for application in research projects. In Musil *et al.* we introduced a modified agile process approach based on Scrum including (1) pre-production, (2) vision, (3) product development, and (4) validation loops [10]. Figure 1 presents an overview of this adapted software engineering process:

(1) *Pre-Production Loop*. Goal is (a) identifying candidate use cases (e.g., during workshops with industry partners and researchers) based on visions and current needs and (b) cost-value considerations [3] for selecting most valuable use cases.
(2) The *Vision Loop* focuses on the product backlog maintaining product vision and changes from industry partner/researcher feedback.
(3) The *Sprint* enables developers in constructing and evolving the use case according to product and sprint backlogs.
(4) Finally, the main goal of *Validation Loops* include (a) use case verification and validation, (b) in-depth industry partner/researcher feedback, and (c) stimulation of new ideas and visions as a baseline for updating backlog for next iterations.

Based on our previous work [10] the extended Scrum process approach seems to be the most promising approach for handling research projects. Nevertheless, this process does not specify the exit/transition point from prototypes to industry products.

2.2 Methods and Tools

In industry projects methods and tools represent the foundation for product development. Basically, constructive methods and tools support engineers in building software documents (e.g., model-based or test-driven) while analytical approaches support defect detection, verification, and validation (e.g., reviews, inspection, and testing). Typically standardized methods and tools are available organization-wide in repositories for selecting and reusing them within the organization. In research projects, where several stakeholders collaborate (maybe in different research organizations), every researcher is using his/her own toolbox, which fits best to the individual requirements or individual preferences. Thus, there is a large base of different tools in a heterogeneous research landscape. While this approach might be suitable for research prototypes, where a small subset features are in the scope of research, this heterogeneity hinders efficient product development in industry. As a consequence an agreement on the most relevant methods and tools has to be established to support the transition from research prototypes to industry products.

2.3 Prototype and Product Maturity

A main issue of prototypes and products focus on the maturity of deliverables. Maturity levels are well-known in context of processes based on CMMI [7] or ISO 15504 (SPICE) [17] to estimate the capability levels of individual processes and the maturity of products and/or organization. Nevertheless in context of prototypes and products maturity levels are typically based on verification and validation results (e.g., results from test runs and acceptance tests). However, maturity levels based on the quality of prototypes and products seem to be a reasonable approach to assess individual work products, i.e., prototypes or products.

To support the transition from research prototypes to industry projects we see the need to introduce (a) defined software engineering process approaches, enabling flexible handling of requirements and stakeholder needs, (b) defined sets of methods and tools for application in research projects as well as industry projects, and (c) assessment approaches of prototype/product maturity levels with respect to apply the solution in an industry context.

3 The CDL-Flex Research Project

This section introduces to the CDL-Flex research project including the basic goals of the project, addressed research areas, and involved stakeholders.

3.1 Project Goals

Engineering projects in the automation systems domain, e.g., hydro power plants and manufacturing systems, depend on the knowledge of experts from a wide range of different disciplines and domains, e.g., mechanical, electrical and software engineering [12]. Individual knowledge is embodied in a heterogeneous set of domain-specific tools and data models. Weakly integrated tools and data models hinder efficient collaboration and data exchange between disciplines [4]. Main goal of the project is to support engineers and managers in large-scale engineering projects to overcome this technical gap between individual tools and the semantic gap of individual data models to better collaborate and exchange data in heterogeneous engineering environments.

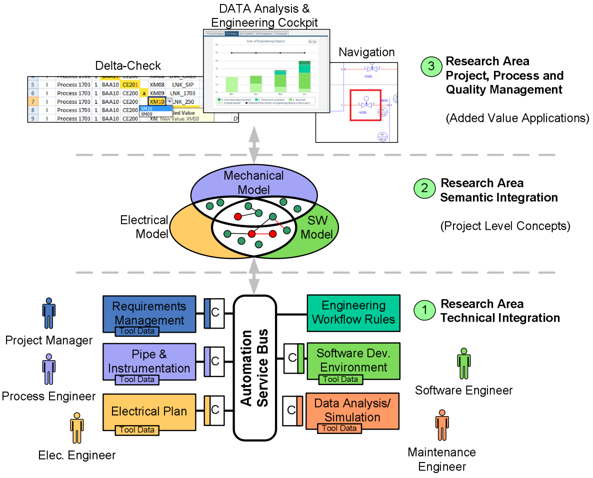

Fig. 2. Application Level-Concept of the Automation Service Bus

Based on the technical and semantic foundation, added-value components support end users and project managers in applying specific use cases in context of engineering projects in a heterogeneous environment. Figure 2 presents the application level-concept of the project with focus on three different research areas:

(1) *Technical Integration of Tools* represents the technical foundation to enable individual tools interacting with each other. For instance, a change request by the electrical engineer will be propagated across a middleware platform (i.e., the *Automation Service Bus, ASB*) to other affected engineers. See Winkler *et al.* for a more detailed description of this basic change management process [18].
(2) *Semantic Integration of Data Models.* Semantic heterogeneity of data models, caused by individual tools, hinders efficient data exchange. Note that the circles in Figure 2 represent discipline-specific data models. Common concepts, i.e., overlapping areas, are the foundation for mapping data models to each other. See Moser *et al.* for a more detailed description the common concept approach [13].
(3) *Added Value Application.* The technical and semantic integration of tools and data models enables added-value components, e.g., change management across domain and tool borders [18], project observation and control as well as comprehensive data analysis with the Engineering Cockpit [12], or efficient navigation between different tools, which are typically not connected to each other [11].

Note that these applications have been built as research prototypes (proof-of-concept) to show the feasibility of the underlying architecture (technical and semantic integration) and to enable added-value components for research and industry partners.

3.2 Project Stakeholders

Research projects typically involve a set of different stakeholders who have to collaborate to achieve defined (but different) goals, for instance:

- *Project Sponsors, e.g., public agencies,* typically focus on basic research, drive research goals forward, and enable good publications in the field(s) of research.
- *Principal Industry Partners* contribute with resources, e.g., additional funding, knowledge in the application domain, and customer contacts. Added values are the permission to use research results in their own business area as a product or as input for consulting activities after project completion.
- *Additional industry partners* who support researchers with real-world use cases, business domain knowledge, and test data. Main interest is getting challenges solved for application in their own business domain.
- *Researchers* in addressed areas focus on research challenges and publication in related communities.
- *Open Source Community (OSS).* Parts of the middleware platform are available as open source contribution[2] with the goal to make research findings public and usable within the OSS community. Note that there are ongoing plans for publication under Apache license.

---

[2] Open Engineering Service Bus (OpenEngSB): `http://www.openengsb.org/`

- *Developers*. Based on the project setting, we introduced two main development groups, i.e., an open source development team and an industry development team responsible for (principal) industry partner related use cases.
- *Power-Users* are application domain experts, who provide (a) domain knowledge and (b) feedback to early prototypes, and (c) support researchers in finding most valuable use cases based on industry needs for future research (research vision).
- *End-Users* should work with the application, i.e., the product, in daily business. Note that end-users typically require stable and working software products.

These basic groups of stakeholders have been introduced quite early in the project to enable effective and efficient prototype development, as required by research contracts. After three years of research and prototype building new roles, e.g., product management and quality assurance teams (QA team) have been established.

4 Research Challenges and Questions

Based on the related work and research/industry best practices we identified two main research challenges in context of the research project:

RQ1. How can we bridge the gap between research projects and industry projects? While systematic and established processes, methods, and tools are available in industry context, research projects have to be more flexible (e.g., changing architecture and requirements). Thus, a main challenge is to find a well-defined process to handle research and industry projects to support prototypes and products development.

RQ2. How can we transfer research project prototypes to industry projects? Research prototypes typically include strong limitations for industry application regarding stability, performance, and user acceptance. Thus, this question focuses on a classification and/or assessment for prototypes/products evaluation.

5 Solution Approach, Results, and Lessons Learned

This section summarizes the CDL-Flex solution approach, initial results, and presents lessons learned after three year of research and prototype construction.

5.1 Prototype and Product Maturity

A first step towards a successful transfer from research prototypes to industry projects is identifying maturity steps (levels) in prototypes and products. Nevertheless, the contribution and the “quality” of basic research (i.e., prototype development) and industry product development vary over time. Figure 3 presents the five basic maturity levels (or steps of development), implemented in the CDL-Flex research project:

- *Level 1 – Research Vision.* This development step includes creative processes, brainstorming, and workshops with industry partners to get ideas, visions and current needs of industry partners to be addressed in the research project.
- *Level 2 – Research Concept.* Based on initial ideas and visions basic concepts (concrete use cases including test data and test cases) are developed – mainly by researchers. Feedback cycles on the concepts enable early validation of the ideas

and visions and ensure that concepts meet individual requirements of industry partners. These prototypes are mocked (i.e., demos without real functional behavior) to simulate the expected behavior based on the initial concepts. Main goal is to receive feedback, e.g., on the user interface and the planned behavior.

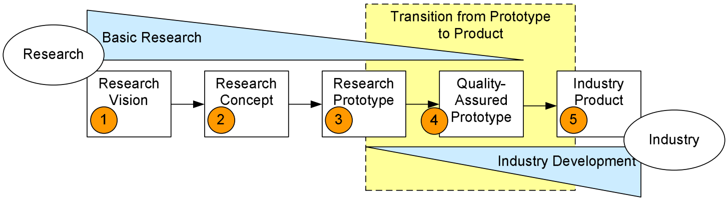

Fig. 3. Maturity Levels in Research/Industry Projects

- *Level 3 – Research Prototypes* include real functional behavior based on the research concept and industry partner feedback. An initial research prototype is based on agreed use cases, test cases, and test data to show the feasibility of the solution. Although basic quality assurance activities have been applied, these prototypes focus on providing the basic functionality with strong limitations to stability, robustness, usability, and fault tolerance.
- *Level 4 – Quality-Assured Prototype.* To enable more stable, robust and fault tolerant systems, additional implementation effort and extended quality assurance approaches, e.g., integration and acceptance testing, are required. Typically, these tasks are out of scope of researchers (who want to show the feasibility of the concept) and have to be executed by other stakeholders, i.e., industry development and quality assurance teams. Note that both teams usually have to be paid.
- *Level 5 – Industry Product.* The final maturity step focuses on real industry products where industry-related methods for development and quality assurance apply. In our project industry products are typically developed by our principal industry partner, supported by the industry development team at the CDL-Flex.

Note, that the impact of basic research contributions decreases over time and impact of industry contributions increases (see Figure 3). The most interesting part is the transition phase, involving (early) research prototypes (level 3, quality-assured prototypes (level 4), and – finally – industry products (level 5).

5.2 Software Development Process

To support individual requirements, derived from individual prototype/maturity levels, we applied the extended Scrum process model (see Figure 1) to the individual development steps (maturity levels) and highlight main contributions of involved stakeholders. Figure 4 presents the proposed extended process approach to enable prototype and product development and the transition between prototypes and products. Basically, researchers, industry partners, and power users derive a set of research vision use cases during workshops and discussions (1; “Pre-Production Loop”). Based on selected use cases researcher develop concepts and implement initial (mockup) prototypes for discussion and feedback (2; “Vision loop”). Note that these initial research prototypes are typically developed by students during their university work (e.g., diploma thesis or internships). Thus, the quality of the prototypes varies and the prototypes are not usable in industry context. Nevertheless, main results of this step are (a) prototypes for feasibility studies and (b) sets of more concrete requirements and features for research prototype implementation.

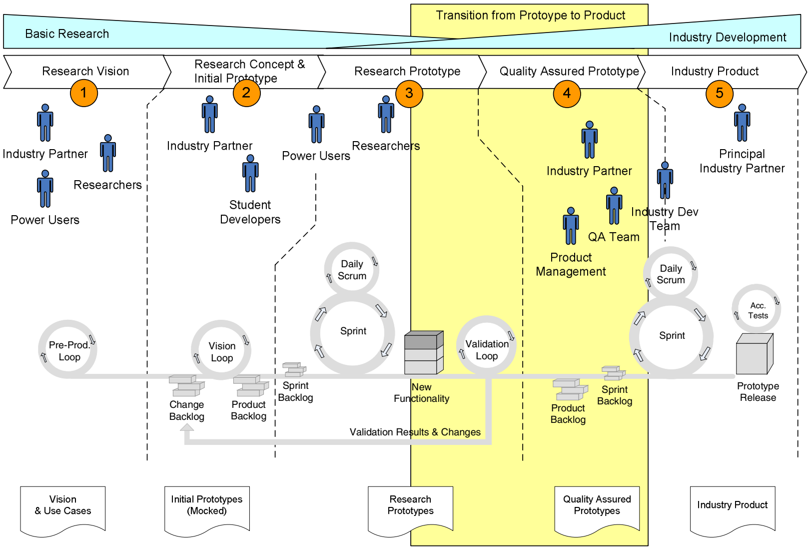

Fig. 4. Proposed Development Process

Based on these more or less stable requirements (derived from the vision loop) research prototypes (3) are developed by student developers. Additional feedback cycles (supported by industry partners, power users, and researchers) enable the improvement of research prototypes ("Validation Loop"). Nevertheless, these type of prototypes lack in robustness, stability, and fault tolerance because the goal is to show the feasibility of the prototype on functional level. To enable application in industry context, more stable and robust prototypes are necessary (4; "quality assured prototypes"). Thus, we nominate product owners, industry development and quality assurance teams for individual use cases and/or industry partners. Note that team members are responsible for the quality-assured prototype and are recruited as professionals. After a successful pilot application the quality assured prototype will be transferred to our principal industry partner, who is responsible for the product and have to integrate the solution in his product portfolio.

5.3 Software Engineering Environment

In early phases of prototype development, researchers apply methods and tools which seem to fit best to the requirements and individual preferences. Nevertheless, when starting implementing a research prototype a more stable development environment is necessary. Main artifacts and tools in context of our project are:

- *Use Cases*. Use cases represent the most important artifacts in context of the research project, i.e., (a) vision use cases that represent rough ideas and (b) concrete use cases for prototype implementation. Typically use cases are high-level goals (from the perspective of industry partners) which have to be split into several features, represented in backlogs assigned to different maturity levels. Note that a use case includes a brief description, real-world data sets for testing purposes, and success criteria for industry partner acceptance. We use Confluence[3], a collaboration tool, for managing use case and related engineering documents.

- *Features, Issues, Bugs.* We use Jira[4], a project tracking software, to manage individual aspects of the use case, e.g., user stories, issues and bugs. Note that all use case related information are linked from use case descriptions (provided in Confluence) to individual issues (provided in Jira) and individual developers.
- *Jenkins.* To support continuous integration and test processes, we applied Jenkins[5], an open source server to monitor and control the project progress including quality assurance checks with respect to quality-assured prototypes, e.g., by implementing Checkstyle[6] and CodeCover[7].
- *Testing of Prototypes/Products.* Beyond developer testing based on unit tests, we implemented acceptance tests for features and use cases. Acceptance tests, maintained by the QA team, are used to (a) establish an early and common understanding of the use case and (b) to enable automation-supported testing of implemented research prototypes. Based on the maturity of prototypes/products, we applied manual tests during research prototype development and automation supported tests with Selenium[8] at the level 4 ("quality-assured prototypes") latest. This quality assurance strategy enables us in (a) separating individual maturity levels and (b) fast feedback in case of changes.

Table 1. Lessons Learned and Key Findings

	Vision	**Concept**	**Research Prototype**	**Quality Assured Prototype**	**Industry Product**
Outcome	Research Vision	Reserach Concept Mock-Up Prototype Proof of Concept Feasibility Study	Use Case / Features Functional Prototype	Use Case / Features Prototype: robust, stable, and fault tolerant	Use Case / Features Industry product
Maturity Level	n/a	low	low	medium	high
QA approaches	informal feedback	systematic feedback test case definition	test case definition manual tests	automated tests QA metrics	According to engineering process
Users	Researcher	Researcher Developers	Researcher Developers Power Users	Industry Partners Power Users End Users	Industry Partners Power Users End Users
Evaluation	informal discussion	interviews and feedback	basic tests	Automated tests QA metrics Acceptance Tests	Automated tests QA metrics Acceptance Tests
Cost/Value evaluation	Estimation of experts and researchers.	Expected benefits based on state of the practice (Experts)	Basic measurement results from pilot applications,	Comparative evaluations in real world settings (pilot	Comparative evaluations in real world settings (pilot

---

[3] Atlassian Confluence: `http://www.atlassian.com/en/software/confluence`

[4] Atlassian Jira: `http://www.atlassian.com/software/jira/overview`

[5] Jenkins: `www.jenkins-ci.org`

[6] Checkstyle: `www.checkstyle.sourceforge.net`

[7] Codecover: `http://codecover.org/`

[8] Selenium: `http://docs.seleniumhq.org/`

5.4 Lessons Learned

After three years of research and observing/analyzing engineering processes, we identified a set of challenges, risks, and candidate solutions for prototype and product development, which can be addressed by (a) flexible software development processes, based on an extended Scrum process (Section 5.2), (b) a set of tools within the development environment (Section 5.3), and (c) a five-level maturity concept to estimate and asses the maturity of development steps and deliverables. Table 1 presents a brief summary of our key observation and practices applicable in every development step of a single use case. Note that we also include the level of quality assurance, involved stakeholders/users, candidate prototype/product evaluation approaches, and cost/value considerations from industry partner perspective.

6 Conclusion and Future Work

In this paper we reported on our experiences from three years of research work of a seven year research project. The main challenge was to bridge the gap between research prototypes, typically constructed based on visions and research ideas, and industry prototypes, usable in industry context. Based on different requirements and involved stakeholders there is a need for (a) engineering processes that support research projects and industry projects (RI 1) and (b) a maturity concept that enables an efficient classification of deliverables to support a smooth transfer from research prototypes to industry products (RI 2).

Lessons learned from previous process improvement initiatives in creative application domains, e.g., game development, can help addressing visions and instable requirements as well as an instable architecture. Figure 4 presented the application of an adapted Scrum process approach [10] in context of research projects. In addition we learned that different method approaches apply in different stages of use case development. Table 1 presented the most important findings derived from project observations. Another important finding was the involvement of different stakeholders, especially in the development process where students can work on individual use case in a defined scope. If research prototypes evolve towards quality-assured research prototypes for pilot applications and industrial products (i.e., an increasing maturity level), professionals are required to enable the construction of high-quality prototypes and/or products. Based on our experience, we believe that the proposed engineering process and the suggested maturity levels can help in better addressing the need of individual expectations coming from research and industry to bridge the gap between research and industry project and research prototypes and industry products.

Future work includes a more detailed evaluation (i.e., a case study) of the purposed process approach to (a) get a more detailed view on the effects of the process and (b) to continue improving the proposed engineering process. In addition we have to evaluate the defined tool-set with respect to applicability in research and industry context in the next phase of our research project.

References

[1] Abran, A., Moore, J.W.: Guide to the Software Engineering Body of Knowledge. IEEE (2004)

[2] Bell, J.: Doing your Research Project: A Guide for first-time Researchers in Education, Health and Social Science, 5th edn. Open University Press (2010)

[3] Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., Grünbacher, P. (eds.): Value-Based Software Engineering. Springer (2005)

[4] Biffl, S., Schatten, A., Zoitl, A.: Integration of heterogeneous engineering environments for the automation systems lifecycle. In: Proc. of 7th Conf. on INDIN, Cardiff, Wales, pp. 576–581 (2009)

[5] Biffl, S., Winkler, D., Höhn, R., Wetzel, H.: Software Process Improvement in Europe: Potential of the new V-Model XT and Research Issues. Journal Software Process: Improvement and Practice 11(3), 229–238 (2006)

[6] Chemuturi, M.K., Cagley, T.M.: Mastering Software Project Management: Best Practices, Tools, and Techniques. J. Ross. Pub. Inc (2010)

[7] Chrissis, M.B., Konrad, M., Shrum, S.: CMMI for Development: Guidelines for Process Integration and Product Improvement. SEI-Series in SE (2011)

[8] Kruchten, P.: The Rational Unified Process: An Introduction. Addison-Wesley Longman (2003)

[9] Ladas, C.: Scrumban - Essays on Kanban Systems for Lean Software Development. Modus Cooperandi Press (2009)

[10] Musil, J., Schweda, A., Winkler, D., Biffl, S.: Improving Video Game Development: Facilitating Heterogeneous Team Collaboration Through Flexible Software Processes. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 83–94. Springer, Heidelberg (2010)

[11] Mordinyi, R., Moser, T., Winkler, D., Biffl, S.: Navigation between Tools in Heterogeneous Automation Systems Engineering Landscapes. In: 38th Annual Conference of the IEEE Industrial Electronics Society, IECON (2012)

[12] Moser, T., Mordinyi, R., Winkler, D., Biffl, S.: Engineering Project Management using the Engineering Cockpit: A collaboration platform for project managers and engineers. In: 9th Int. Conf on Industrial Informatics, INDIN (2011)

[13] Moser, T., Biffl, S., Sunindyo, W.D., Winkler, D.: Integrating Production Automa-tion Expert Knowledge Across Engineering Domains. Int. J. of Distributed Systems and Technologies (IJDST), SI 2(3), 88–103 (2011)

[14] Rausch, A., Bartelt, C., Ternite, T., Kurmann, M.: The V-Modell XT Applied – Model-Driven and Document Centry Development. In: Proc. of the 3rd World Congress for Software Quality, pp. 131–138 (2005)

[15] Schwaber, K.: Agile Project Management with Scrum. Microsoft Press (2004)

[16] Sommerville, I.: Software Engineering, 9th revised edition. Addison-Wesley Longman, International Version (2010)

[17] Van Loon, H.: Process Assessment and ISO/IEC 15504: A Reference Book, 2nd edn. Springer (2007)

[18] Winkler, D., Moser, T., Mordinyi, R., Sunindyo, W.D., Biffl, S.: Engineering Object Change Management Process Observation in Distributed Automation Systems Projects. In: 18th EuroSPI Conference, Roskilde, Denmark (2011)