

Modeling the Software Prototyping Process in a Research Context

Mika SAARI, Jari SOINI, Jere GRÖNMAN, Petri RANTANEN, Timo MÄKINEN
and Pekka SILLBERG

*Tampere University, Faculty of Information Technology and Communication Sciences,
Pori, Finland*

Abstract. The paper examines the Third Mission of universities from the point of view of company collaboration in the prototype development process. The paper presents an implementation of university-enterprise collaboration in prototype development described by means of process modeling notation. In this article, the focus is on modeling the software prototyping process in a research context. This research paper introduces prototype development in a university environment. The prototypes are made in collaboration with companies, which offered real-world use cases. The prototype development process is introduced by a modeling procedure with four example prototype cases. The research method used is an eight-step process modeling approach. The goal was to find instances of activity, artifact, resource, and role. The results of modeling are presented using textual and graphical notation. This paper describes the data elicitation, where the process knowledge is collected using stickers-on-the-wall technique, and the creation of the model is described. Finally, the shortcomings found in our existing practices and possibilities for improving our prototype development processes and practices are discussed.

Keywords. Software process, process modeling, knowledge management, modeling methods, prototyping, modeling

1. Introduction

It is a common conception that the modern university serves three main purposes: teaching, research, and a broader social function. The latter of these functions, commonly dubbed “The Third Mission” [1-4], is considered to include measures contributing to social influencing and interaction. Nevertheless, multiple views in terms of defining the Third Mission exist, and Henry Mugabi [5] for example, compiled a selection of the varying definitions present in recent research literature in his dissertation. Moreover, the concept has been increasingly integrated into university strategies as well as operations pertaining to regional development [6]. Universities serve to produce and share knowledge and technological expertise, contributing in their part to the modernization and success of enterprises, and the “Third Mission” is often associated with tech-driven collaboration between the university and enterprises in addition to its social function.

The social significance of the Third Mission is widely acknowledged, and an international evaluation indicator conducted by UNESCO [7] places Finland among the top countries regarding collaboration between university and enterprises. In Finland, the most common manifestations of this kind of collaboration are various research and development projects that are often funded externally, examples of which include

contributions by the Academy of Finland, Business Finland, various foundations, the European Union, and other international sources. For small and medium-sized enterprises (SMEs) [8] in particular, collaborative projects with a university offer significantly better opportunities to participate in R&D activities for instance, as opposed to relying on internal funding and expertise only.

This paper introduces the model of prototype development practices that have been applied in research projects between university and enterprises (mostly SMEs) in Finland. The practices have been applied for many years in several collaborative projects between Tampere University, Pori unit, and regional enterprises. In this paper, the model is presented with the help of a few cases serving as concrete examples. Promoting and contributing to the Third Mission of the university from the perspective of regional development, the format introduced in the present paper is but one example of collaboration between the university and enterprises.

The paper is structured as follows: In Section 1, background information along with work related to the subject are presented. The research approach in terms of the applied model is presented in Section 2. Section 3 introduces the process knowledge with the help of examples, after which Section 4 describes the creation of the Prototype Development Process (PDP) model in selected cases. Section 5 discusses the observations and challenges confronted during the use of the model. Finally, Section 6 summarizes the paper.

2. Related studies on the subject

This section describes some points of view, factors, and experiences, which relate to the collaboration context between universities and enterprises. Earlier research results in this field are examined below.

In their research, Basili et al. propose guidelines on how collaborative research could succeed between public sector research and industry. In [9], a couple of arguments are present, which are worthy of examination. First, they say that there is not enough research considering real development contexts. Second, they suggest that software engineering research needs to foster context-driven research if it seeks to evolve towards a brighter future. The main contextual factors are human, organizational, and domain-related. In [9], they believe that practical software engineering, the big picture, and suitable solutions are mostly born from bottom-up research and a succession of case studies rather than from top-down research.

In context-driven research it is necessary to have intense collaboration between industry and academia. There need to be cycles of model building, experimentation, and learning in software engineering research. Usually researchers face some problems in identifying the challenges of collaborating organizations. After overcoming these challenges, the gap between the state of the art and the requirements for a solution must be assessed. In developing a solution, it is important to clearly define working assumptions in order to achieve applicability and scalability in context [9].

Another major part of this collaboration relates to funding. There are a couple of ways to start funding a research project between industry and academia. Companies can give grants to students/researchers for a specific project, build collaborative projects with academia in short- or long-term existing relationships, and there is an opportunity to build up a wider network of partners between industry and academia [10]. Also, governments are encouraging this kind of collaboration because it improves innovation

efficiency and thereby enhances wealth creation [11]. As a result, several countries have already put innovative programs in place [9].

Industry-academia collaboration benefits those organizations that do not have their own R&D facilities. Companies can utilize the knowledge of academic resources to understand their modern-day software engineering problems. Industry has noted that it can support innovation and development processes when collaborating with researchers [9].

Companies are increasingly investing in software development, although their core areas of expertise are defined around business areas and systems rather than software. However, most companies do not have the necessary resources and know-how to develop effective solutions to software engineering problems. This makes collaboration between industry and academia very useful [9].

The aim of a study by Wohlin [12] was to gather experiences and lessons learned from successful collaboration between industry and academia in two different environments. First, it was performed in Sweden and included a six-year collaborative project. The collaboration partners were five different-sized companies from various sectors and the Blekinge Institute of Technology. Later the study was replicated in Australia. Industry roles included product managers, project managers, developers, and testers. Academic roles included professors, researchers, and students at different levels.

The results from Wohlin's study [12] were that support from company management is crucial. There must be a champion at the company who argues for the cause, and not only a person assigned the responsibility for the project. There are different levels of understanding between different categories of people (for example, people in industry, senior researchers, and students). Social skills are particularly important in long-term collaboration.

In the study by Carver [10], there is an example of industry-academia collaboration. The background of the study was that the challenges faced by the companies were too labor intensive, lacked context-specific validation, or were not embedded into existing tools or design processes. Collaboration between industry and academia produces successful results when there is a good connection between academic and industry partners, there are the right collaborators on both sides, the timing of the interaction fits the requirements on both sides, and it is understood that the process from research prototypes and an academic publication to a deployed solution takes time.

In [13], the author interviewed researchers to understand their needs and problems in research-based projects. University research was more concerned with community issues, and companies had clients that were constrained by formal agreements. The industry groups had defined roles and responsibilities; the research teams were more dynamic. Industrial companies used formal development methods, but most university projects did not. The milestones for industrial projects were set by the customer, while the milestones for university projects were usually set by the funding partners. After the interviews, the author decided, for a number of reasons, including the uncertainty about the research objectives, that university researchers were unable to use a well-defined software development method.

The above-described examples provide the research background for the context that is discussed in this paper. The following section concentrates on explaining the basis of the process modeling, which is relevant background information for the description later, in Section 4, of the implementation of the university-enterprise collaboration with the help of the process modeling notation.

3. Research approach – An eight-step model

This section deals with software process modeling in a research context. Modeling is an approach for analyzing and understanding a complex phenomenon resulting in a model, which is a simple and familiar structure that can be used to interpret some part of reality [14]. When the phenomenon to be analyzed is a software process, information is captured and classified into a model with the help of a process-modeling schema [15], i.e., a meta-model specifying the concepts, relationships, and rules [16] used when modeling processes. The basic concepts related to the software process include activity, artifact, resource, and role [17].

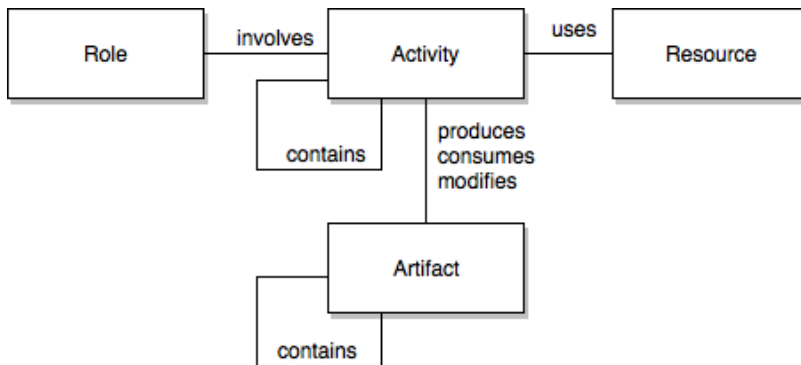


Figure 1. The basic concepts related to software processes.

The results of modeling are presented using a textual or graphical notation. There are several approaches for eliciting information for process models [15] such as interviews and artifact analyses. Process modeling can be prescriptive or descriptive. While a Prescriptive Process Model (PPM) describes how a process should be performed, a Descriptive Process Model (DPM) describes how it actually is performed [18].

In this study our aim is to model software development practices performed in an academic context. We follow the DPM approach proposed by Becker & al. [19]. The approach consists of eight steps grouped into two phases:

Set-up phase

1. Objectives and Scope
2. Define Schema
3. Select Language
4. Select and Tailor Tools

Execution phase

5. Elicitation
6. Create Model
7. Check Model
8. Check Process

We will apply the DPM approach in the following way: The data for the models is collected through interviews with the developers involved in the processes. The schema shown in Figure 1 guides the data collection, the results of which are shown on stickers on the wall during the work (Figure 2 in Section 4). In the resulting models, the activities are represented as rectangles with rounded corners. Stick figures represent roles and different icons represent resources. Artifacts appear as parallelograms, cylinders, and document symbols. The associations between activities and artifacts are represented by continuous arrows and the links between activities and roles and resources are dashed. Gray symbols and dashed rectangles represent aggregations (Figures 3-6 in Section 4). Graphical representations of the models are produced by a free online diagram software, draw.io. (DPM Steps 2-4).

The objectives and scope of the modeling are presented in the introduction of this paper (Step 1). The following section provides an example of data elicitation (Step 5) including the modeling results (Step 6). After that, the possibilities for improving the modeled processes are discussed (steps 7-8).

4. Process elicitation and resulting models

This section describes the steps 5 and 6 of the DPM approach. Process knowledge is highlighted in this section, which introduces four different prototype development processes (PDP 1-4).

Information for the process models is collected from four cases:

- PDP 1 - Verification of customer complaints related to bus routes [20].
- PDP 2 - Verification of customer complaints related to garbage collection. [20].
- PDP 3 - Data collection in a public indoor swimming pool [21, 22].
- PDP 4 - Passenger counting in a free-to-ride bus [23].

Common to all of these example cases is the software development resulting in a working prototype. The development process starts from the idea of collecting certain data with certain equipment. Then the idea is validated – can it be viably implemented? If the answer is yes, the implementation phase produces the first working prototype. Usually prototype implementation includes software coding and the implementation of hardware from off-the-shelf devices. The working prototype is tested in a laboratory and if the device displays sufficient reliability, the device is moved to real-world testing. The development processes and testing phases usually produce data. The overall outcome from these prototype development processes has been academic output such as a research paper. Knowledge of the prototyping process is visualized as illustrated in Figure 2.

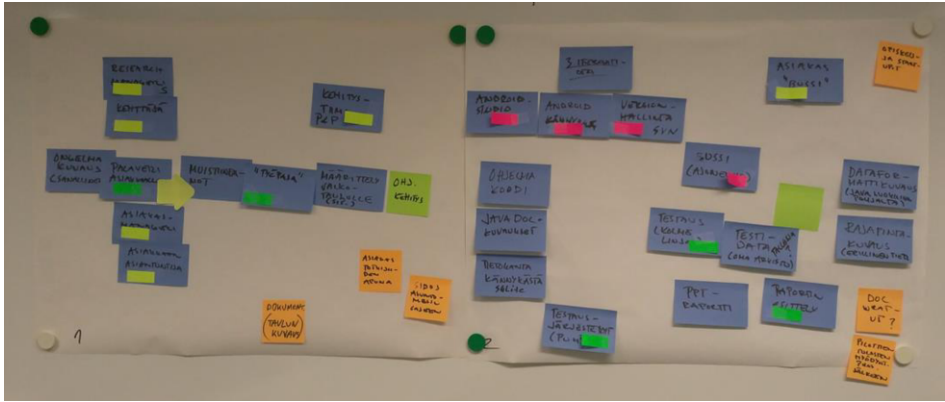


Figure 2. Whiteboard and notes.

This whiteboard and the Post-It notes are one way to highlight development process practices [24]. The notes are color-coded: Yellow markers are roles, green denotes activity, red is for resources, and the blue notes are artifacts. The orange notes describe issues and improvement ideas that came up during the data elicitation process. Knowledge of the development process is collected by means of this whiteboard and note notation (stickers-on-the-wall technique).

Further, this section presents the creation of a model for each of the PDPs. First we introduce the developed model for PDP 1 and 2. Figure 3 presents the university – company interaction. The model includes six steps which have been identified from the development process. The steps start from requirements definition and end with the publication of results. The gray boxes – the software development step and the development artifacts – are discussed more in the subsections on the bus case and the garbage collection case.

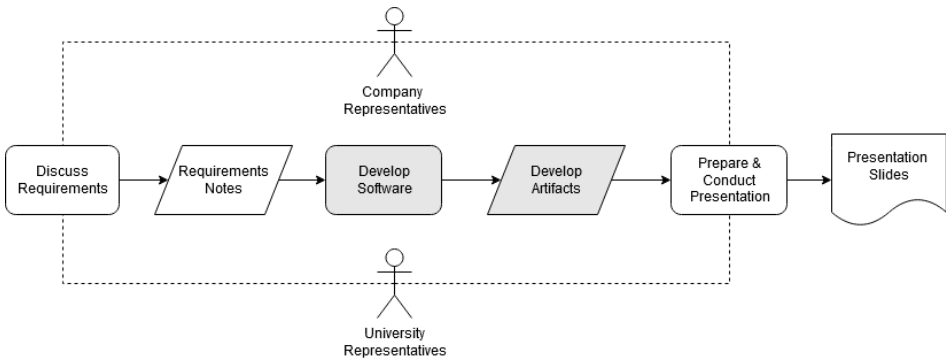


Figure 3. The illustration of university – company interaction during development (common to all PDPs).

The model includes all the main factors. The university representatives are the research group including the project manager and software/hardware developers. This group has the main responsibility for the prototype development. The company representatives are involved in the development process in the role of instructor. In the

presented PDPs 1-4, the company representatives are active at the starting and ending points: at the start with the definition of requirements and at the end where the results are presented to them. They also provide the testing environment if the testing is done at their company. However, they are not involved in the development process itself.

4.1. Verification of customer complaints related to bus routes (PDP 1)

The bus case (PDP 1) was established for handling customer complaints. The bus company had received complaints from customers that the buses did not stop to pick up customers or did not come at all. The prototype was developed to collect photos with time stamps at defined bus stops. This prototype was implemented in mobile phones and the main idea of prototype was to collect photos of the bus stops as the bus approached. The project group at the university developed a working prototype, which was then tested in the buses [20].

The development process is illustrated in Figure 4, expanding the previously presented steps “Develop Software” and “Develop artifacts”. The development group in this process consists of only university personnel. The development process starts with a design discussion – the first activity, which produces the first artifact: the whiteboard sketch. The results are then used in the software construction phase. This activity produces the second artifact: software code with documentation. After that the test activity starts, which produces the collected data artifact. The coding and testing activities could be iterated several times. Figure 4 also includes the resources used during the development process: programming language, test device with GPS, camera and network, and the testing environment – the bus itself.

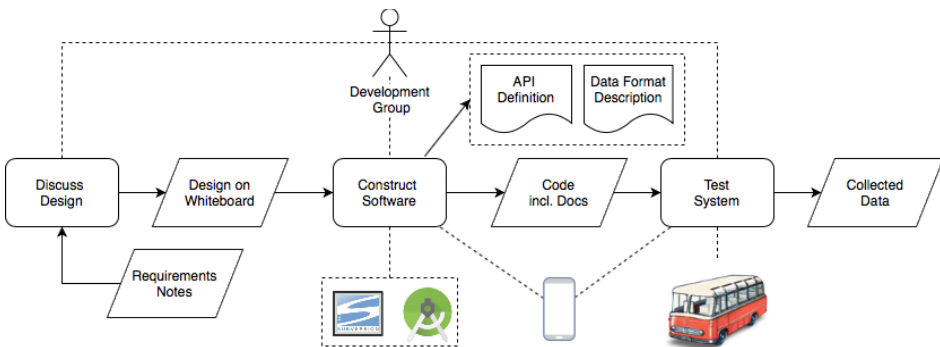


Figure 4. The development process of the bus case.

The presented development process produces a working prototype. In this case, the prototype was introduced to the customer – the bus company.

4.2. Verification of customer complaints related to garbage collection (PDP 2)

The garbage collection case (PDP 2) was similar to PDP 1. The use case was also intended for managing customer complaints. The garbage collection company had

received complaints from their customers that the trashcans had not been emptied. In most cases the reason was a vehicle blocking the garbage collection truck or similarly that the truck was unable to empty the trashcan. The prototype solution in this case used the same idea as in the bus case: namely, put the camera in the truck and take pictures when the GPS registers the right location [20].

Figure 5 shows the activities and artifacts. The biggest difference was in the resources: the cellphone was replaced by a Raspberry Pi with a camera, GPS, and network device. The data collector resource was the MySQL server instead of the phone. In addition, the testing environment was the garbage collection truck itself.

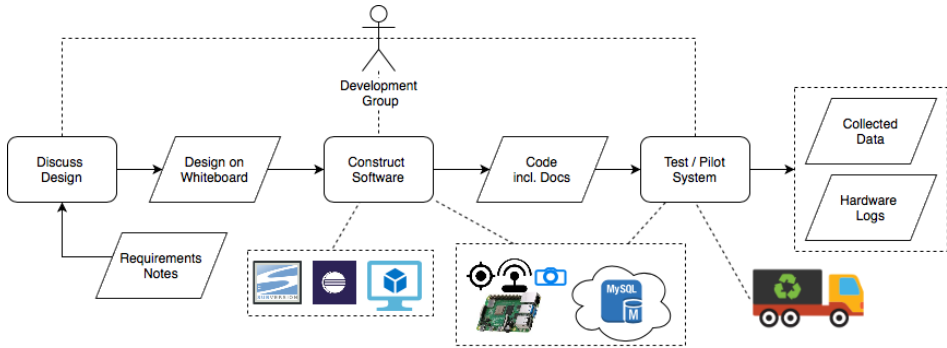


Figure 5. The development process of garbage collection case.

The developed prototype worked and the piloting phase in the garbage truck lasted several weeks. The development process captured data, which in this case were photos. In addition, the device – a Raspberry Pi, generated a test log during the pilot phase.

4.3. Combined model of bus and garbage collection cases

PDP 1 and PDP 2 should be presented together because the second prototype – the garbage truck camera system – utilized the definitions and results of the first prototype. They were also implemented very close to each other in time.

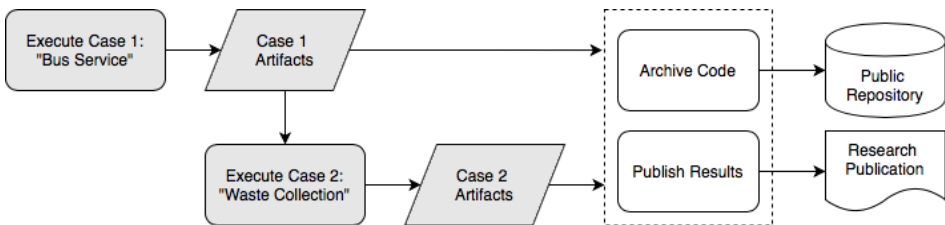


Figure 6. The combined model of the bus and garbage collection cases.

The outcomes of these PDPs were a public repository and research publication [20]. The public repository includes all the developed software code.

4.4. Data collection in a public indoor swimming pool (PDP 3)

PDP 3 handled a prototype system with the purpose of improving the quality of manually collected data. The prototype was a mobile application that the maintenance personnel used to collect and store data from several different meters in a public swimming pool. More information on the prototype can be found in the research articles by Soini [21, 22].

The Prototype Development Process was similar to the one shown in Figure 3. Activities included the discussion of requirements, software coding, and the presentation of the results. In addition, the artifacts were similar: discussion notes, software, and a research publication. The biggest difference to the other cases was that the implemented prototype remained in use after the pilot phase. This “extended piloting” period is handled in the research paper by Soini [22] along with aspects of software evaluation. Also, long-term piloting is examined from the point of view of system developers, administrators (maintenance), and end users [22].

4.5. Passenger counting in a free-to-ride bus (PDP 4)

PDP 4 handled a prototype system with the aim of counting passengers on a free-to-ride shuttle bus. Unlike an ordinary people counter, the customer wanted more information on where and when passengers got on and off [23].

The Prototype Development Process was similar to the one shown in Figure 3. Activities included the discussion of requirements, software coding, and the presentation of the results. Again, the artifacts were similar: discussion notes, software, and a research publication. The difference compared to PDP 1-3 was that this prototype was developed for a real-life use case of collecting statistics about bus passengers on a free-to-ride shuttle bus route at a large public event in the summer of 2018 in Pori, Finland. In this case, the development process ended in the one-month pilot. The outcome of the pilot was the presentation of the pilot results to the customer.

5. Discussion

The process introduced and modeled in this paper has been used to create multiple prototypes and pilot experiments during recent years. Thus, based on the observed results, it can be considered to be successful and fit the needs of our development cases. However, while discussing past projects with the team members several challenges did come up. Furthermore, while collecting data for the model, notes were made of issues that the team members pointed out (the orange notes on the whiteboard in Figure 2).

The first issue was the documentation of intermediate specifications described on whiteboards. The funding or goals of the projects do not especially require extensive intermediate documentation, and in practice, only very rarely has there been the need to study the intermediate specifications created during the process. The low requirements for documentation have probably been one of the main reasons for the bad habits in documentation practices. Generally, the decision of whether to prepare any documentation has been based on the developer teams' "gut feeling" about how complex the specification was. In other words, "proper" documentation has been created for more complex intermediate specifications, but simpler specifications and drawings have not been documented in any way. Nowadays, it is quite simple to use a smartphone to capture

the information on whiteboards, so in the future, it might be better to document everything systematically.

The second issue was the interaction with the customers (e.g., companies) - or the lack of it - during the actual prototype development process. The interaction has often been limited to the use case definition phase, to the organization of practical arrangements of pilots (e.g., agreements on which bus lines could be used for testing prototypes), and to the presentation of the research findings. In the final presentation meetings, the companies have never indicated that they would have liked to be more involved in the process. The feedback from the companies has mainly been related to the research findings, and the developed prototypes, and not to the development process itself. In our case, the companies have often not been software-oriented, which could have had an effect on their interest in the process, and it could also have limited the advantages achievable by involving them further in the process. In addition, the companies did not (directly) invest any resources (money or personnel) in the projects. This might have further reduced their interest in participating more deeply in the development process. Furthermore, as the companies had their actual business to run, there could have been challenges in creating a common schedule for meetings for all the parties involved. Of course, depending on the outcome of the research and pilots the companies can obtain knowledge, business ideas, or even working software to use in their actual business, but during the development these results may be too abstract to evoke deeper interest. In addition, the fact that the issue was not brought up in the meetings does not necessarily mean that there is a lack of interest in deeper involvement from the company side. As the university team did not especially raise the issue, it might be that the companies felt that they simply did not want to interfere in the university practices. Thus, to improve our model, the actual company interest in the prototype development process should be further studied.

Third, the subject of the usability of the project results came up. In principle, anyone can use the results because the codes and documentation are published as open source, but no studies have been performed on how or if the results are actually used. In general, after the projects (and thus, funding) has ended, the results have been left "as is" without maintenance, bug fixes, or feature improvements. The purpose of the projects was not to create "end products", and usually productization is not one of the project goals, leaving the created software and hardware applications in a state that would require further development into an end-user quality product. Also, it can be slightly challenging to find the material from, for example, the GitHub repository if one does not know exactly what one is looking for. Internally, the published codes and specifications have been reused in future projects when applicable. One potential future direction for research would be a study on how the results of university projects should be published to be most useful for outside parties, and what the crucial elements are that should be published — or are the elements practically the same as in any other prototype development project?

Finally, the participation of university students was discussed. The advantage of involving the students more deeply would be to give the students more meaningful task assignments (for example, for programming or other software engineering courses), but in practice, in the past the participation of students has been rare, and has mainly been limited to PhD students who have been hired by the university or worked at the university on their own funding. In the future, the model introduced in this paper could also be expanded to describe the involvement of students.

6. Summary

The paper examined the Third Mission of universities from the point of view of company collaboration in the prototype development process. The paper presented an implementation of university-enterprise collaboration in prototype development described by means of process modeling notation. The process introduced and modeled in this paper has been used to create multiple prototypes and pilot experiments over recent years. The prototypes were made in collaboration with companies, which offered real-world application problems.

The prototype development process was introduced by a modeling procedure with four example prototype cases. The research method used and presented here was an eight-step process modeling schema. The basic concepts relating to the software process included four factors: activity, artifact, resource, and role. The results of the modeling were presented using textual and graphical notation, and the elicitation of process knowledge the creation of the model were described. Furthermore, certain shortcomings in our existing practices were discovered. Based on the findings the model as well as our prototype development process and practices could be further improved in the future.

In summary, the PDP model provides one concrete and systematic example of how university-enterprise collaboration can be executed in practice. Moreover, the model presented is a real-life indication of how the Third Mission task set for universities can be successfully implemented. Thus, based on the observed results the model can be considered to be successful and fit the needs of our development cases.

References

- [1] F. Schutte, and P. C. van der Sijde, "The University and its region". Examples of regional development from the European Consortium of Innovative Universities. Twente University Press, Enschede, Netherlands. 2000.
- [2] T. Vorley, and J. Nelles, "Building Entrepreneurial Architectures: a conceptual interpretation of the Third Mission", *Policy Futures in Education*, Vol. 7, No. 3, pp. 284-296, 2009.
- [3] I. Niiniluoto, "Yliopistot ja ammattikorkeakoulut yhteiskunnallisina vaikuttajina", In the publication *Vastuullinen ja vaikuttava. Opetus- ja kulttuuriministeriön julkaisuja 2015:3*, pp. 11-30, 2015.
- [4] K. Kankaala, E. Kaukonen, P. Kutinlahti, T. Lemola and M. Nieminen (2004), *Yliopistojen kolmas tehtävä ?*, Edita Publishing Oy. Helsinki, pp.15-42, 2004.
- [5] H. Mugabi, *Institutionalisation of the "Third Mission" of the University. The case of Makerere University*. Academic Dissertation, Tampere University Press, Finland 2014.
- [6] A. Zomer, and P. Benneworth, "The Rise of the University's Third Mission", *Reform of Higher Education in Europe*, (eds.) Enders, J., de Boer H.F. ja Westerheijden, D.F., Sense Publishers. Netherlands, pp. 81-102, 2011.
- [7] UNESCO (2015), UNESCO science report: Towards 2030, p. 74, <https://unesdoc.unesco.org/ark:/48223/pf0000235406/PDF/235406eng.pdf.multi> (Retrieved 4/1/2020 World Wide Web)
- [8] M. Marchesnay, "Fifty years of entrepreneurship and SME: A personal view". *Journal of Small Business and Enterprise Development*, Vol. 18, No. 2, pp. 352-365, 2011.
- [9] V. Basili, L. Briand, D. Bianculli, S. Nejati, F. Pastore and M. Sabetzadeh, "Software Engineering Research and Industry: A Symbiotic Relationship to Foster Impact," in *IEEE Software*, Vol. 35, No. 5, pp. 44-49, 2018.
- [10] J. C. Carver and R. Prikladnicki, "Industry–Academia Collaboration in Software Engineering," in *IEEE Software*, Vol. 35, No. 5, pp. 120-124, 2018.
- [11] T. Barnes, I. Pashby, A. Gibbons, "Effective University – Industry Interaction: A Multi-case Evaluation of Collaborative R&D Projects," in *European Management Journal*, Vol. 20, Issue 3, pp. 272-285, 2002.

- [12] C. Wohlin, A. Aurum, L. Angelis, L. Phillips, Y. Dittrich, T. Goschek, H. Grahn, K. Henningsson, S. Kagstrom, G. Low, P. Rovegard, P. Tomaszewski, C. van Toorn and J. Winter, "The Success Factors Powering Industry-Academia Collaboration," in *IEEE Software*, Vol. 29, No. 2, pp. 67-73, 2012.
- [13] D. M. P. Dias, "Managing research based software product development in Sri Lankan universities", University of Colombo School of Computing, 2016.
- [14] M. Boman, J. Bubenko, P. Johannesson and B. Wangler, "Conceptual Modeling". Prentice Hall, 1997.
- [15] U. Becker-Kornstaedt, "Prospect: a Method for Systematic Elicitation of Software Processes". PhD dissertation. Technische Universität Kaiserslautern, 2004.
- [16] ISO/IEC 24744:2006. Software Engineering — Metamodel for Development Methodologies. Draft International Standard ISO/IEC FDIS 24744:2006(E), ISO/IEC JTC1/SC7, Montréal, Québec, Canada.
- [17] U. Becker-Kornstaedt and R. Webby, "A Comprehensive Schema Integrating Software Process Modeling and Software Measurement". Research Report 047.99/E, Fraunhofer IESE, Kaiserslautern, Germany, 1999.
- [18] C. Jensen and W. Scacchi, "Discovering, Modeling, and Re-enacting Open Source Software Development Processes: A Case Study". In Silvia T Acuña, S. T. and Sánchez-Segura, M. I. (eds.). *New Trends in Software Process Modeling*. World Scientific Pub. Co., 2006.
- [19] U. Becker, D. Hamann, D and M. Verlage, "Descriptive Modeling of Software Process". Research Report ESE-Report, 045.97/E, Fraunhofer IESE, Kaiserslautern, Germany, 1997.
- [20] J. Grönman, P. Rantanen, M. Saari, P. Sillberg, and H. Jaakkola, "Lessons Learned from Developing Prototypes for Customer Complaint Validation", *Software Quality Analysis, Monitoring, Improvement, and Applications (SQAMIA)*, Novi Sad, Serbia, August 27-30, 2018.
- [21] J. Soini, P. Sillberg, and P. Rantanen, "Prototype System for Improving Manually Collected Data Quality," in *Proceedings of the 3rd Workshop on Software Quality Analysis, Monitoring, Improvement, and Applications, SQAMIA 2014*, Lovran, Croatia, pp. 99–106, 2014.
- [22] J. Soini, M. Kuusisto, P. Rantanen, M. Saari and P. Sillberg, "A Study on an Evolution of a Data Collection System for Knowledge Representation", *EJC 2019: Proceedings of the 29th International Conference on Information Modelling and Knowledge Bases*, 2019.
- [23] J. Grönman, P. Sillberg, P. Rantanen and M. Saari, "People Counting in a Public Event—Use Case: Free-to-Ride Bus", *2019 42th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2019.
- [24] A. Raninen, J. J. Ahonen, H.-M. Sihvonon, P. Savolainen, and S. Beecham, "LAPPI: A light-weight technique to practical process modeling and improvement target identification," *Journal of Software Evolution and Process*, Vol. 25, No. 9, pp. 915-933, Sep. 2013.