Continuous and Collaborative Technology Transfer: Software Engineering Research with Real-time Industry Impact

Tommi Mikkonen

University of Helsinki, Helsinki, Finland

Casper Lassenius

Aalto University, Espoo, Finland

Tomi Männistö

University of Helsinki, Helsinki, Finland

Markku Oivo

University of Oulu, Oulu, Finland

Janne Järvinen

F-Secure, Helsinki, Finland

Abstract

Context: Traditional technology transfer models rely on the assumption that innovations are created in academia, after which they are transferred to industry using a sequential flow of activities. This model is outdated in contemporary software engineering research that is done in close collaboration between academia and industry and in large consortia rather than on a one-on-one basis. In the new setup, research can be viewed as continuous co-experimentation, where industry and academia closely collaborate and iteratively and jointly discover problems and develop, test, and improve solutions.

Objective: The objective of the paper is to answer the following research questions: How can high-quality, ambitious software engineering research in a collaborative setup be conducted quickly and on a large scale? How can real-time business feedback to continuously improve candidate solutions be gained?

Method: The proposed model has been created, refined, and evaluated in two large, national Finnish software research programs. For this paper, we conducted thematic interviews with representatives of four companies who participated in these programs.

Results: The fundamental change is in the mindset of the participants from technology push by academia to technology pull by companies, resulting in co-creation. Furthermore, continuous cooperation between participants enables solutions to evolve in rapid cycles and forms a scalable model of interaction between research institutes and companies.

Conclusions: The multifaceted nature of software engineering research calls for numerous approaches. In particular, when working with human-related topics such as company culture, development methods or tool infrastructure, many discoveries result from seamless collaboration between companies and research institutes.

Keywords: Technology transfer, collaborative research, software engineering research, public–private partnership

1. Introduction

Public-private partnership (PPP) research programs [1] require considerable up-front planning and promise for research to be done. For instance, EU programs such as the Framework Program 7[1] and its sequel, Horizon 2020[2], call for research proposals that can be hundreds of pages long, with the goal of describing the results of long-term collaborative research between academia and industry in detail. The time window of these projects spans a number of years, with tens to hundreds of millions of euros invested in each program.

While such preplanned and predefined projects can be a good match for certain kinds of research, software engineering research that aims at new abstractions, concepts, methods, and approaches that can be rapidly put into practice require a different approach. In such research, a close collaborative relationship between academia and industry is required. In many ways,

Email addresses: tommi.mikkonen@helsinki.fi (Tommi Mikkonen), casper.lassenius@aalto.fi (Casper Lassenius), tomi.mannisto@helsinki.fi (Tomi Männistö), markku.oivo@oulu.fi (Markku Oivo), janne.jarvinen@f-secure.com (Janne Järvinen)

[1] https://ec.europa.eu/research/fp7

[2] https://ec.europa.eu/programmes/horizon2020/

the context resembles the domain of software intensive products where agile [2] and lean [3] development are the norm and responding to change over following the plan is embraced [4]. Still, research project proposals often require a Gantt chart to outline the complete project schedule — something that is often a complete mismatch with reality when executing the project.

The same is true of the traditional models of technology transfer [5]. Many such models assume a somewhat mechanistic relationship between industry and research institutes, where tasks are mostly carried out in isolation rather than in collaboration [6]. Many problems require more frequent interaction, however, even in one-to-one relationships, whereas multi-party projects situations are more complex [7]. Furthermore, chances are that when the research results are out, even if in the exact form that was prescribed, the needs and expectations have changed so that the outcome might essentially be useless for industry. Setting aim at a moving target is obviously risky, and therefore it is advisable to introduce both a risk management plan and frequent check points to ensure that the results are truly in line with the expectations.

In a field such as software engineering, where change is constant, a different, more rapid and collaborative approach for research and technology transfer is needed [8]. In this setup, models must be able to incorporate the close collaboration of numerous actors from both industry and academia. The situation is further complicated by the fact that in reality not only one but several initiatives are run in parallel at varying stages of maturity and business readiness. In fact, while some of the activities can be run internally by a single company with no support from research institutes or other companies, a consortium can aim at more extensive results on a larger scale. Shneiderman has pointed out in his recent keynote [9] that achieving the best results in such setups requires both basic and applied research. Furthermore, multi-disciplinary research combining technical and human aspects is often needed to solve problems that emerge during the work [10].

In this paper, we propose a technology transfer model for software research that aims to produce a continuous stream of results that have a direct, preferably measurable business impact in companies that participate in the activities. Unlike the technology transfer models of for example Pfleeger [11] and Gorschek et al. [12], which only separate two parties (industry and academia), the proposed model is geared towards large consortia, consisting of numerous companies and research institutes that share an interest in a common research topic. The model has been developed in two consecutive research programs in Finland, Cloud Software (CS, 2010–2013) and Need for Speed (N4S, 2014–2017). Numerous research institutes and companies participated in both programs, with the total volume of over 800 person years and over 100 million euros (MEUR) in budget. Both programs had ambitious goals geared towards leveraging technology innovations to create new business and opportunities for companies as well as top-of-the-line research.

The rest of this paper is structured as follows. In Section 2, we discuss background and related work. In Section 3, we present two case projects in which a new form of collaboration was tried and improved. In addition, we present results from company interviews regarding their observed results of using the new way of collaborating. In Section 4, we propose a new model for collaboration, extracted from the case projects and interviews. In Section 5, we present an extended discussion regarding the results and observations and address the validity of the research. In Section 6 we draw some final conclusions.

2. Background and Related Work

Technology transfer, or the process in which scientific findings, discoveries, and results are transferred from a research institute to a company where they are adapted to business needs, requires numerous activities. Even if various other sources have probably had an effect on the results [13], the linear technology transfer model originally proposed by Bush [14] takes place in four steps — basic research, applied research, development, production, and operation.

Obviously very generic in nature, the model is applicable to almost any field of research, particularly technology. As the model only looks at the maturity of technology, it is agnostic to the number of actors; however there is no elaboration regarding how collaboration over the different phases should happen. The same is true for the so-called reversed linear model [15], where the same steps as in the linear model are used, but the motivation is based on industry pull and needs — in this case those of Bell Labs and ATT [15]. Similar to the linear model, the model of collaboration in the reverse linear model is largely undefined.

When considering these coarse-grained steps, technology must often be experimented with in various ways during the process, for instance, first in cooperation with academia and then internally by the company in pilot projects. Eventually, the technology becomes business as usual, although more direct adoption is also possible. In addition to things that are directly associated with the technology in question, there are various other steps that do not focus on the technology itself, such as protecting technologies via patents and copyrights and establishing development and commercialization strategies such as marketing and licensing to existing private sector companies or creating new startup companies based on the technology. In general, Mansfield's [16, 17] research papers on academic research and industrial innovation are early contributions to the large body of literature on the economic benefits of university research. A landmark of collaboration and technology transfer from academia to industry is the Bayh–Dole Act of 1980, which permits a university, small business, or non-profit institution to elect to pursue ownership of an invention in preference to the government [18]. The Bayh–Dole Act is generic and applies to almost any field of research [19].

When placing the focus on different domains of research, special considerations have been raised. In the field of software technologies in general, to the best of our knowledge the earliest proposed model is that of Redwine and Riddle [20], which describes how software technologies generally mature. The work is based on case studies in the 196s and 1970s, and therefore the baseline resembles earlier work, proposing six

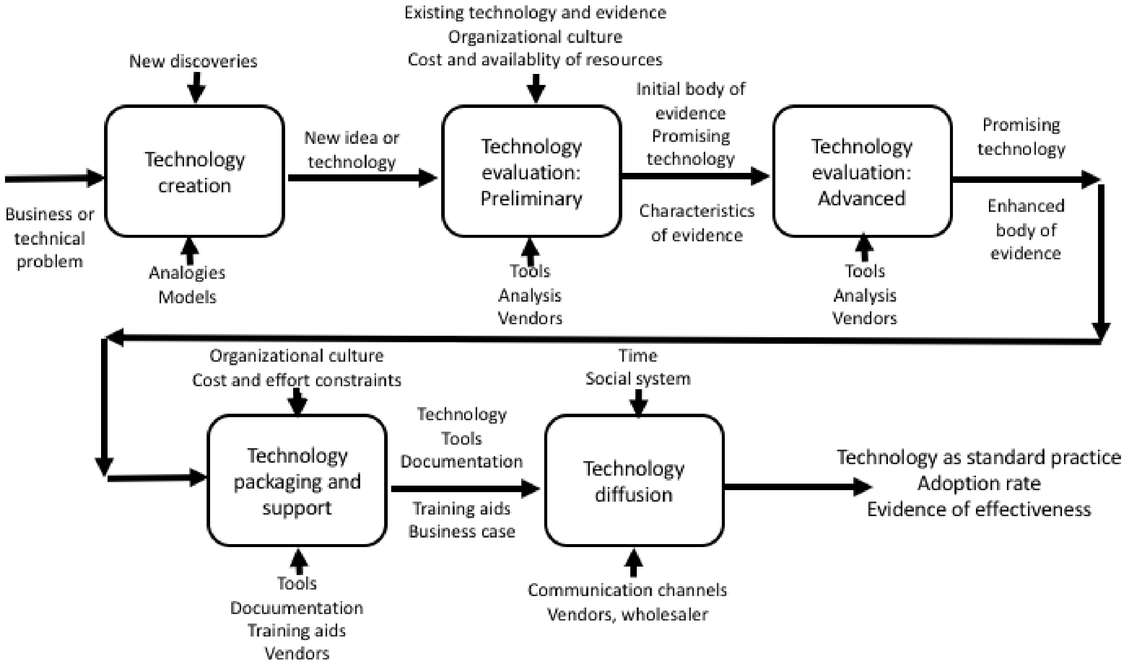

Figure 1: Technology transfer model proposed by Pfleeger [11]

steps that take technology from research towards popularization and large-scale use. A similar, although slightly simpler transition process is proposed by Zelkowitz [21] in the context of NASA software engineering research, and work on process improvement in particular can be regarded as an early form of technology transfer for software engineering practices [22, 23]. Still, despite the models, many of the key software engineering research results, such as work of Parnas et al. [24, 25] on modularity, are more directly associated with industry needs and practices rather than academic results stemming from basic research.

A prominent technology transfer model for software engineering research in particular is that proposed by Pfleeger [11], although it is also applicable in the field of software technologies as well. The model proposes a five-step process (Figure 1). In the process, each step is associated with activities and well-defined input and output. This makes the process easy to understand and follow, given that all the information is available. The steps are somewhat similar to those in earlier models, where maturity increases on the way from basic research to commercial, everyday use. Furthermore, like the earlier model, the motivation for the transfer seems to be a technology push from research, which companies adopt as a part of the process.

Another software engineering-specific approach to technology transfer has been proposed by Gorschek et al. [12]. Consisting of seven steps (Figure 2), this model is more collaborative than Pfleeger's, as the roles company and academic research play are identified, calling for explicit cooperation and interaction over different phases. In comparison to other models, the fundamental difference of Gorschek et al.'s model is that it acknowledges industry pull as an important characteristic of a technology transfer model for software. In fact, the model includes a spirit of technology diffusion, where there is room for different roles for companies and for academic research, thus including elements of both technology push and industry pull.

In general, the above models are somewhat mechanistic in the sense that they assume static needs and require following a rather sequential approach with only limited iteration where the transfer is between two parties that are intimately involved in the process. Examples include subcontracted research and cases where the goal is to create completely new technologies. Furthermore, based on a 2002 paper by Kaindl et al. [26], it seems that in general both academia and industry could improve their interaction to promote and put new research results in the field of requirements engineering into practice. In fact, as pointed out by Basili et al. [27], successful technology transfer in the field of software engineering calls for champions, or humans that are committed to embrace and preach the new technology, although a number of success factors can be identified [28]. Still, as proposed by Beecham et al. [29, 30], the value and relevance of such collaboration should only be judged by the results and outcomes.

In contrast to models that assume a somewhat static environment, introducing new technologies, methods, and tools has become a daily business in software intensive businesses. Things happen seemingly without any scientific evidence, and in a way that completely overlooks the proposed technology transfer models. For instance, in the field of web programming, the fashionable JavaScript libraries change rapidly [31], and following the latest trends in leveraging cloud computing opportunities [32] calls for constant screening of available options. The same goes for engineering practices that are under constant improvement, following Agile [2], Lean [33], and, more recently, Continuous * [34] and DevOps [35] ideals. Further-

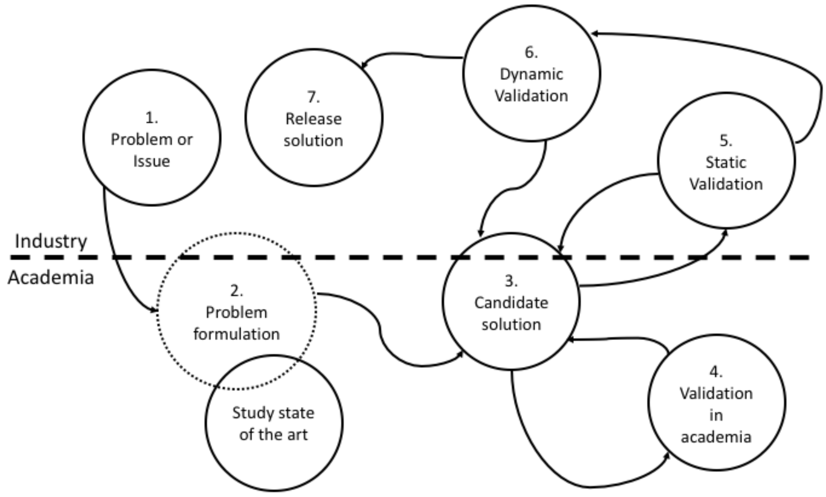

Figure 2: Technology transfer model of proposed by Gorschek et al. [12]

more, instead of two stakeholders, many publicly funded research projects include several companies and research institutions, which all have their individual interests in participating in research.

Such projects act as catalysts for an on-going change where transactions are increasingly being replaced by partnerships in the field of research. Supported by industry-university-government relationships [36], or the so-called Triple Helix as discussed by Etzkowitz and Leydesdorff [37] and Gustafsson and Jarvenpaa [38], this form of networked research is creating entirely new opportunities for collaboration compared to traditional two-party models [39].

In a setup where a large consortium consisting of research institutes and companies is working for the same goals, numerous technology transfer initiatives are continuously and simultaneously conducted in parallel. Because the above models are either single or two-party activities, managing many such activities in parallel is difficult for both industry and the academic side, especially when the number of participants per activity increases. Therefore, setups in which academia and industry constantly interact in partnerships calls for a different approach for technology transfer than those proposed for more transactional, collaborative cases. Because this type of an approach has several characteristics of action research [40] (research is initiated to solve an immediate problem or to improve the way issues are addressed and problems solved) it is only natural that it is well-suited for producing data for empirical software engineering research [41]. While one can argue that this is a reflection of the difference between applied and basic research — the so-called Pasteur's quadrant where, in contrast to pure applied or basic research, the work is called use-inspired [6] — it has been pointed out that such distinction is highly artificial and arbitrary [42]. Rather than going for such distinction, one can consider factors such as technology transfer readiness levels [43] to determine if and when to adopt new technology.

3. Technology Transfer 2.0: Continuous Software Engineering Research

In this section, we introduce an approach for industry-academia research collaboration that produces results in a continuous, agile fashion, previously briefly reported in [44].

3.1. Motivation

The motivation for the new approach stemmed from the observation that many software engineering problems are *wicked*[45, 46], where the problem and the context in which the problem arises change as the problem is being partially solved. Therefore, it is not possible to define and solve problems in isolation, as the company context is intimately connected to the problem. In fact, the problem may not even surface outside the particular company context. This makes progress similar to process improvement; changes can be made only iteratively, as their adaptation is recreating the context in which new improvements must fit.

The situation is further complicated by the fact that software engineering problems emerge in contexts that are already populated with tasks awaiting attention. Consequently, there is very little room for extra activities, such as new ideas or improvements in the company pipeline. From the company perspective, many research activities qualify as such, making it easier not to adopt new ideas even if research suggests that doing so would be helpful in the long run [26].

To introduce surviving solutions and improvements in the company context, there must be industry pull — in other words, the company must be ready and willing to experiment with and adopt new ideas and techniques.

3.2. Goals for Project Setup

Agile/continuous empirical research with a long-term perspective. As proposed by Shneiderman [9], several approaches to research are needed in the field of software engineering.

Adopting the lean and agile mindset, the key goal for our research setup was to ensure a constant source for industry problems available for academic research and a constant stream of potential solutions available for companies. The motivation was to enable rapid validation and instant experimentation for both parties, with minimal overhead in terms of cooperation.

Collaboration at scale. To enable a constant flow of problems and candidate solutions between companies and research institutes, two types of scaling must be achieved. First, one company must be able to obtain results from various research parties. Second, one research institute must be able to provide the same results to several companies. In both cases, it is desirable that only the bandwidth invested by the party would be the limiting factor in collaboration and not the process that would define alternating roles along the way. Furthermore, the mindset would be such that academics are a joint research team instead of being competitors. Allowing academics to cooperate on joint problems can also be a fruitful source of cross-pollination of intermediate results.

Strong business and academic impact. The third essential goal of the programs was to really go after things that would have a strong business impact but without risking long-term academic research. A part of this impact would of course be defined by the long-term perspective that was already mentioned. But to also attract SMEs and other small organizations in the program, it is essential to provide immediate impact – during the program that is – whenever possible. In addition, strong academic results were deemed a necessity to maintain researchers interest in participating in the programs. At the same time, all the research efforts were to have a direct link to company problems, as exposed by their representatives. Furthermore, priorities in terms of the different needs were set by companies.

Vision driven long-term goals. Any visionary endeavor requires a strong vision, and research programs are no exception. A joint vision that all the parties commit to is therefore a prerequisite for initiating any ground-breaking research initiative, and all parties must have something to gain. Furthermore, the vision must justify a long enough program to build trust over long-term collaboration, as otherwise in-depth research and development simply cannot be realized.

Business case driven short-term goals. To serve the short-term interests of companies, the implementation of the vision is decomposed into smaller units called business cases. Business cases – the name reflects the fact that originally it was assumed that all business cases would be measurable and quantifiable in terms of investment and output, which turned out to be impossible – are the daily manifestation of research work, and executing them both introduces new results and incorporates them in the companies businesses. The size and the complexity of a business case can vary, but it is essential that all business cases have more than one organization involved.

Self-organizing research activities. It was almost immediately recognized that the content of business cases could vary considerably. To enable versatile activities in business cases, which can introduce unique needs, we decided that the size and the length of a business case should be defined by the participants of the case in question. The same goes for the number of partners involved in the activity. This setup created flexible modes of participation by the companies; sometimes a company would act as a driver and at other times some of the research institutes could be driving the business case. The only restrictions for a business case were that there should always be more than one participant and that no business case is populated by only research institutions.

Time-boxed, enforced interaction. As acknowledged by agile development approaches, interaction is one of the strongest means to communicate results. Consequently, to keep a large consortium in sync, time-boxed checkpoints are needed. These checkpoints require the participation of all parties so that many different ways of interaction can be enabled.

3.3. Case Projects and Their Organization

The goals outlined above have been experimented with, refined, and evaluated in two national Finnish software research programs. In the following, we first introduce the case projects and then focus on their organization.

3.3.1. Case Projects

We experimented with these ideas in two large-scale Finnish research initiatives: Cloud Software [47] (CS, 2010-2013) and Need for Speed [48] (N4S, 2014-2017). The initiatives were jointly funded by the Finnish Funding Agency for Technology and Innovation (Tekes) and the participating organizations, including both companies and research organizations. Therefore, the projects relied on PPPs, following the Triple Helix approach [38], where the industry, research institutes, and funding agencies act in a collaborative fashion. Numerous research institutes and companies, including both large enterprises and SMEs, have participated in both programs, with a total volume of over 800 person years in effort and over 100 MEUR in budget. While there are several different ways to form such alliances, flexible funding models that support collaborative research are an important enabler for this type of approach. In the end, they enable the collaborative setup, optimally drive the formation of critical mass, and help create established baselines for aspects such as intellectual property rights, consortium agreements, and extending research to commercial collaboration.

Both of the above programs had ambitious goals, geared towards leveraging technology innovations to create new business and business opportunities for companies. The programs and their goals are briefly introduced in the following.

Cloud Software. The goal of the Cloud Software program was to adopt cloud computing [49, 32] and associated technologies [50] as a paradigm in Finnish software-intensive companies. The program approach was multifaceted including cloud business, cloud technologies and lean and agile software enterprise as major building blocks for the cloud transformation, supported by integrated security and superior user experience.

Need for Speed. The N4S program was built around three main themes for the software intensive industry [51]: 1) a paradigm change from product business to delivering value in real-time; 2) deep customer insight to improve the hit-rate of businesses; and 3) mercury-like rapid business experiments

[51] that are aimed to help organizations discover new business opportunities. As concrete technologies, these call for a sophisticated delivery pipeline for deploying applications [52], whose introduction to numerous companies was one of the core technical aims of the project.

3.3.2. Program Organization

Both programs started by defining the research goals in a Strategic Research and Innovation Agenda (SRIA), a vision document that serves as a roadmap for the research work carried out within the scope of the programs (as an example, the reader is referred to [53]). For practical purposes, the SRIA defines joint targets for all the participating organizations. Therefore, based on the SRIA, the programs introduced a continuous planning model that was based on business cases, research sprints, quarterly reviews (Q-reviews), and yearly planning cycles. These are addressed in more detail in the following.

Business cases. Day-to-day research is organized in terms of business cases or strategic research items that fall under the umbrella of the SRIA. Business cases, or strategically important research topics for one or more companies, were defined by industrial partners and were supposed to introduce business impact that is critical for future success, with elements of both research and business experimentation included in each business case. Business cases are all shared, with a company leading the effort. To foster cooperation, business cases were decomposed into research themes to foster cooperation so that one theme would serve several business cases. Themes had their own pace, with no requirement for synchronization with other business cases or themes. However, for each theme, there was a shared checkpoint every three months called a quarterly review. Correspondingly, each period of three months was referred to as a research sprint.

Research sprints. At the beginning of the research sprint, the research team, with members from each participating organization of a business case, defines a set of realistic tasks and outcomes over the next three months. For each on-going business case, it is also possible to fine-tune the focus based on industry priorities, resource constraints, and research capacity. This self-organized way of working rather than project steering based on plans ensures that each party can focus on activities that are most essential for them. The outcomes of research sprints vary depending on the business cases they address and the phase of the research. The results may include, for example, design plans, method descriptions, case studies, research articles, technical demonstrations, or business experiments. These artifacts are stored on the programs intranet, the main storage as well as access and sharing point for the programs research assets. These assets are classified into software offerings (e.g., IPR, patents, and copyrights), other intellectual property (e.g., product concepts, business models), internal/operational (e.g., new tools, processes, and techniques), and external/relationship (e.g., business relationships, networks, and ecosystems developed). At the end of each sprint, teams present the highlights of their work in a quarterly review meeting.

Quarterly reviews. The enforced, time-boxed synchronization points of the programs, called Q-reviews, played a major role in fostering cooperation, planning the research sprints, and sharing the results. A key motivation behind the Q-reviews was to ensure transparency to attain the best possible impact. Each Q-review consisted of themed workshops, training events, presentations, strategy discussions, and planning sessions. The value of the continuous planning process was extremely visible in the programs Q-review meetings when key results are presented and demonstrated at the so-called Demo Bazaar where the newest results are demonstrated to the whole program. Furthermore, transparency introduced by Q-reviews enabled sharing information and results and facilitated new innovation, sometimes even beyond existing plans. Each Q-review concluded with a retrospective session to collectively reflect and learn what went well and what could be improved in the next Q-review meeting

Yearly planning. The PPP model that was assumed in the programs received its funding decisions on a yearly basis. This meant that for every year, new (or revised) plans were made, building on business cases. In addition, results were reported to the funding agency to demonstrate the advancements in the project. Therefore, in analogy to software development, sprints can be regarded as the mechanism for keeping the actual work ongoing, and yearly reporting and planning corresponds to software releases.

Program management. The programs employed the concept of a leadership troika to manage the big undertakings effectively. The Focus Area Director (FAD) is responsible for the program execution, the Academic Coordinator (ACO) commissions research activities, and the Program Coordinator (PCO) coordinates the programs operational work. In addition, the program has a steering group consisting of major industrial partners and an extended steering group that includes academic partners. In addition to the leadership troika, communities of practice [54] (CoP), with a nominated leader/coordinator pair from industry and academia, were formed around the topics that generated the most interest. This way, research results and ongoing activities were easier to follow in a more focused fashion. In addition, CoP leaders had some reporting duties to the leadership team.

3.4. Industry Experiences

To study how the companies viewed the research programs and how they felt technology transfer was organized, we conducted a semi-structured interview at four participating companies. The focus of the interview was on the execution in one business case only, putting the emphasis on interactions and cooperation with research institutes. The interview questions are shown in Appendix A. The interviewees were selected based upon having played a key role in the respective company during the execution of the above programs. The model proposed by Gorschek et al. [12] was used as the baseline model in the interviews, but it was shown to the interviewees only towards the end of the interview to reflect their findings vis-á-vis the model. All the interviews lasted 60–90 minutes and were recorded for analysis. For the purposes of this paper, we present the results in the form of narratives, each of which has been checked by the corresponding interviewee.

3.4.1. Company A: "Towards Continuous Experimentation"

Company context. Company A is a large multinational business-to-business organization that provides embedded systems to corporate customers. The company has over 100,000 employees globally and has adopted agile and lean software development in several of its development units in different parts of the world. The case described below was done in one development organization within the company with about 200 employees.

The company had a new service-based product intended for emerging markets. The product was recently acquired from another company, and there was a need to quickly elicit and test customer needs and grow both the organization and the features aggressively to pursue a hot, growing market.

Research needs and execution. The organization had to scale up both the development organization and find an efficient way of quickly understanding customer needs and wants in an emerging market for which the elicitation of requirements was very difficult. Therefore, a trial-and-error mentality had to be adopted.

The company aimed at introducing a culture of direct customer contacts and continuous experimentation to get the development organization close to the customers and to rapidly implement and test new features to support sales and to iteratively understand the real market needs. The steps that were taken are 1) workshops and discussions with researchers on continuous experimentation; 2) pilot experiments with customers; and eventually 3) a continuous experimentation culture implemented for several teams in the organization.

Technology transfer reflections. The company considered the work with continuous experimentation an important success. The experience both helped the organization break its old ways with a big distance between development and customers and provided direct business benefits through the application of continuous experimentation that helped drive the product in the right direction. The company saw clear benefits in the program structure from the point of view of collaboration with both academic partners and other companies. With respect to other companies, the program provided a natural place to benchmark and share experiences in trying out new methodologies. The academic partners brought an outsiders viewpoint and experiences from several companies, providing value both as sparring partners and as sources of knowledge. Collaboration with academics provided the additional benefit that, due to a well-working non-disclosure agreement (NDA) structure in the program, they could also be at least partly involved in business-critical activities that for natural reasons could not be shared with other companies.

According to the company, the traditional way of technology transfer would not have been an option here, as the ways of working had to be defined and improved on the fly, that is, during implementation. Key to the success was the close collaboration with the academic partner and the rapid improvement loops within the company. Therefore, a sequential model would neither support the industry need for rapid development of the new ways of working based on immediate feedback nor help the researchers gain a deep understanding of the solution in context.

Summary and concluding remarks. The company informant commented that the way of working in Cloud Software and the N4S program was a huge improvement from traditional, long research projects in which results are not seen until late in the project. However, the programs would have benefited from even more tight CoP around very specific problems.

3.4.2. Company B: "Introducing DevOps"

Company context. Company B was a Finnish cyber-security company that operates globally. It has over 1000 employees. The company provides software-based security solutions to its customers. The company has a long history of using agile and lean software development methods. It has implemented company-wide transformation to using agile and lean practices.

The business case for Company B was related to introducing DevOps as a company-wide program. The idea of DevOps emerged from the agile culture and test automation. There was interest in increasing the automation of deployment and saving effort for real customer oriented work.

Research needs and execution. Despite impressive results in agile and lean transformations, there was the problem of not being able to devote enough time to creating customer value and solving customer problems. The cycle times were much faster than with traditional methods but still not fast enough for the customers and the dynamic market.

The company aimed at a considerable reduction in deployment time. Time and effort from development to operation was planned to be reduced. The plan was to deploy DevOps widely in the company.

The research was started with an investigation of the DevOps phenomenon by an academic partner. This was done in collaboration with the company and other partners in the N4S consortium. Candidate solutions were directly put into use in the real industrial environment. Technological solutions were experimented with in several teams without static or dynamic validation. New solutions were continuously put into use. Each team had the authority to implement DevOps in their own way.

The case was not a traditional technology development project. The aims were similar to those of traditional software process improvement projects. The whole cycle of introducing new practices was run in rapid continuous cycles. Every three months, the company participated in the N4S Q-reviews. The meetings provided an opportunity to interact with researchers and other N4S companies that were also introducing DevOps.

Technology transfer reflections. DevOps involves a thorough cultural change. Development and operations are traditionally different organizations and are associated with different organizational cultures. A DevOps transformation includes many technical aspects, but cultural issues – especially across two different organizations – are a major challenge. The nature of the transformation would be a challenge for the traditional technology transfer model because piloting cultural changes in academia or small pilots would be close to impossible. Rapid cycles and feedback from real industrial environments is crucially important.

The results of introducing DevOps were excellent. If a deployment time was previously months, it now became weeks. If deployment time was weeks, it now became days or even hours. This means not only savings but also the ability to put more effort into creating customer value and being able to quickly take advantage of market opportunities. Business decisions are no longer dependent on development and deployment delays. Decisions can be made much faster.

When asked about the traditional technology transfer model, the interviewee asked a counter question: What if the time interval between step 1 and step 7 (of the Gorschek model) would be a maximum two weeks? It was clear that a sequential flow in the technology transfer creates challenges for time to market unless driven by weak signals, thus calling for predictive, uncertain scenarios in the early phases of research. Ideally, they would like research to provide candidate solutions and they would simply pick the best solutions and deploy them directly in industry environments with very fast cycles.

Summary and concluding remarks. Overall, the company was very satisfied with the way of working in N4S, and its personnel participated very actively in the N4S quarterly meetings. The investment of two days for a consortium meeting every three months was seen as profitable. The benefits of collaborating and interacting with researchers and other companies were seen as very advantageous not only during the project but potentially reaching far into the future.

3.4.3. Company C: "Introducing Evidence-Based Release Planning"

Company context. Company C is a Finnish SME (around 200 employees) that operates globally. The main product is a web development framework that is available under an open source license. The main sources of revenue include direct customer projects built on top of the framework and providing support to the most important customers.

The company has access to a vast amount of data. Data sources include traditional market research, developer and customer feedback, open source community activity, and more in-depth technical data such as framework API usage data from existing installations. However, this vast amount of data meant that it was overkill to start planning a new release relying purely on facts, and therefore decisions were to a large extent educated guesses of senior management

Research needs and execution. Fundamentally, the company was suffering from two problems: 1) decision making did not really consider the data, making the role of marketing and sales challenging as the company was not really reacting to market needs; and 2) decisions were too coarse and far-reaching, making releases large, complex and slow to implement.

The companys goal was to create and improve ways of utilizing data coming from various sources in company level decision making and to leverage that to create smaller and faster releases. These releases would then be targeted more clearly to certain use cases or customer demand. Furthermore, feedback would be received more quickly, thus validating the needs of the customers. The following steps were taken to this end:

1. Identify data sources (including also feasibility studies and experiments done in cooperation with academic partners)
2. Design and implementation of new data collection systems (including in particular a mechanism to collect data associated with API calls, done in cooperation with academic partners); all data now goes to the same data base.
3. Setting up analysis processes, and defining analysis methods (partially in cooperation with academic partners)
4. Analysis of the results
5. Creating backlog items to steer product development, based on the outcome of the analysis.

These were run several times during the program, with each step introducing new data, new experiments considering some special needs such as UX or security, and adding new analysis methods. In particular, the goal was to make the process more systematic by adding automated steps, which has fostered new research cooperation with research institutes.

Technology transfer reflections. A key element in the program was the strong cooperation between the company and the research institutes. It was mentioned that the ability of research institutes to envision and propose the benefits of the results was vastly appreciated by the company when it was presenting its business case and related problems. Moreover, the company acknowledged that this requires that the researchers are deeply involved in everyday issues in software development and not only act as specialists in a certain field.

Another important element was the time frame. It was explicitly pointed out that it would have been impossible to commit to a research project that would deliver results only after a year, for instance. Instead, results – which need to address the most important everyday needs – were expected every quarter, as scheduled by the master frame of the program.

The master frame was considered essential for other reasons, particularly as it kept fostering possibilities for further cooperation. For the company in question, there were even more opportunities to follow and grasp than the company was able to absorb. For instance, topics related to continuous delivery were considered, but in the end there was not enough bandwidth to follow that track as well. The same goes for tracks that promise results but fail to be urgent in the companys present situation and scope. However, the company proposed that a magazine summarizing the most important research opportunities and the preliminary results on a layman level be produced for the consortia participating in the program. The company pointed out that in their view, they were only executing steps 1, 6, and 7 of the Gorschek model. In other words, all the validation took place in production. If something was not suitable, it was eliminated from production when this was observed, and, in contrast, if something was deemed useful, it was left integrated in the product process. Clearly, academic partners and research institutes were executing tasks such as candidate technology creation, but in accordance with the company experiences, this had been done in the background so that candidate solutions were readily proposed by researchers when needed.

In general, the company claimed that this approach – going directly to production – is actually necessary to maintain company interest and momentum. Something that would be only framed for research needs was not seen as practical, as this would not have been given enough priority by the company.

Summary and concluding remarks. In general, the company found the model of collaboration well-suited to their needs, and instantly stated that they would be ready to collaborate with other companies and research institutes in a similar setup. In addition, the employees of the company were inspired to participate in the academic line of research. To this end, the company was involved in numerous publications during the program, and some of the publications include company personnel as authors. In addition, at least one doctoral defense is to be finalized by the company personnel.

3.4.4. Company D: "Towards Agile Data Warehousing"

Company context. Company D was a mid-sized Finnish software company with some 600 employees at the time of the interview. The company's core competence is delivering projects as a service, where each project has unique characteristics. At present, there are close to 200 active projects, most of which are tailor made, relating to custom software systems. Approximately half the company's revenue comes from the public sector, which has very specific needs regarding laws and structures unique to Finland. Such projects have only one customer organization, while they may and in many cases do have a large number of end users. The size of the projects varies remarkably, ranging from two to hundreds of man months per project.

One particular kind of service the company offers its customers is solutions to help utilize data in their business decisions. Such data warehouse projects are very labor-intensive, and there was a need to improve the methods, tools, and processes, for example, by automating tasks

Research needs and execution. The data warehouse solution projects previously required several manual tasks with different tools to be carried out, for example, to create a metamodel for the data warehouse. It was decided to seek a way to increase the degree of automation in these projects. In addition, good practices and tools to manage project resources, such as version control tools, needed to be put into more active and systematic use. The goal was to increase the productivity of the data warehouse and analytics projects with the aim of gaining at least a six-fold increase in the speed of development.

The research problem was framed as a concrete business case and a decision to develop a robust and well-thought out solution. At this stage, no targeted survey on academic literature was conducted, although the systematic literature surveys on related topics conducted in the N4S program provided a good understanding. One of the main benefits of these surveys was understanding that no real solutions actually existed and what was being envisioned would actually be state-of-the-practice. A number of candidate solutions were created iteratively in multiple real client projects. The academic partners took part in the real projects with a scientific viewpoint. The solutions were put directly into use by the clients. The clients saw the value in contributing to the research by offering researchers access to their case and data, even if it was not directly beneficial to the project itself.

Technology transfer reflections. The initial problem originated from industry. The creation of solution candidates was mainly done in industry, partly based on academic literature, and their validation was conducted in industry in collaboration with the academic partners. The clear difference was that the solutions did not originate from academia and nor was the validation primarily conducted there. Instead, the validation of the feasibility of the solutions came directly from applying the solutions in real client projects. Furthermore, the work was much more iterative around phases 1, 3, 6, and 7, with the distinction that phase 3 was more clearly on the industrial side that what is depicted in the figure.

Some challenges experienced related to some initiatives from the academic partners, such as requests for data collection, that were acted upon in the case but were never actually used for research. In the beginning of the program, the research collaboration was considered by some merely as a cost, and it was difficult to fit any activities into the already full calendars of the company employees. However, as the work progressed, these concerns were mitigated. Actually, the major lesson from the N4S program was to include stricter rules for the companies to participate in the quarterly meetings with a minimum number of participants, such as five. This would have helped to take even better advantage of the collaboration, as in hindsight it seemed some opportunities were missed. There could have been more internal marketing of the collaboration opportunities, for example, selling the internal ideas in the meetings to the academic partners and communicating the forms of academic collaboration to the companies.

The business results from the program were considered extremely valuable. The case company would have started part of the development regardless of the program, but the research collaboration contributed positively to the way the activities were viewed within the company and gave an ideology that enhanced the motivation, as the research attitude appealed to certain individuals, thus motivating them to work on the topic and even to join the company.

The company was able to contribute to more than 10 scientific publications, and one doctoral thesis was completed by an employee of the company during the program. Further, at least two more employees have started or are considering a doctoral thesis.

Summary and concluding remarks. In all, the model for the N4S program was considered very good, and the program was large enough, having the needed critical mass. As a suggestion for improvement, a more direct benchmarking with the competitors taking part in the program would have been beneficial. Academic partners could act as anonymizers in such a comparison. In addition, there could even be a friendly competition for the companies in terms of the academic results, such as publications, that would stem from their case.

4. Proposed Model

Based on the company interviews, it seems that many of the elements of the technology transfer models are present in the research approach used in the case projects, but they are executed in a parallel and collaborative manner (see Table 2 for direct comparison to the closest models of technology transfer). The consensus was that the companies were only executing steps 1, 6 and 7 of the Gorschek model (Figure 2). However, when there are numerous actors involved, it is possible that some phases are not paid attention to in some of the organizations. For instance, it is only natural that a company might completely overlook technology review done for another company but is ready and willing to monetize the results. In addition, some of the issues in the industry may emerge when academic research obtains new results. Finally, even if mainstream operations are not involved in technology transfer as such, the programs have served as important door-openers for academics to peek into the true internal side of a company, enabling the study of mainstream operations and possibly the identification of other issues that in turn might motivate further collaboration.

Based on experiences while working on the project and on the interviews discussed above, we have incorporated the views into the technology transfer model presented in Figure 3, with the company cases discussed above presented in the light of the model in Table 1. At the core of the model are research themes, which were obtained by refining business cases to an actionable level of detail. Research activities were then formed around the different teams, enabling each party to collaborate on the research themes with numerous other organizations. While at times the roles of individual contributors were mixed – it was not uncommon that a researcher participated in company operations directly or that a company engineer acted as the lead author of a research article – at the principal level the organizations had different roles in the process. These are addressed in the following.

Companies. Upon expressing interest in business cases, companies are open to obtaining candidate solutions from research institutes and from other companies working on the same research theme. They then test and experiment with new tools, methods, and techniques in their own context, thus producing data and experiences that are representative of their particular context. This output is made available to other companies and research institutes, enabling peer-level collaboration between companies.

Research institutes. The role of research institutes is to feed companies with new solutions and ideas, some of which originate from the academic literature, experimental research, and collaboration with other companies. This input is then turned into experiences and feedback by companies, who evaluate it in different ways. Research institutes may also participate in hands-on industry-specific operations, if that helps in the evaluation. For instance, new kinds of tools might require expert assistance to evaluate or be available only as pre-versions that cannot be directly embedded in company operations. The feedback and experiences from various companies enables research institutes to summarize and generalize the results.

Funding, facilitation, and steering. While companies and research institutes collaborate on a daily basis, certain actions are needed to coordinate the work. In our case, this coordination included satisfying the requirements of the funders, enforcing quarterly physical interactions, prioritizing the different business cases over time, and changing the direction of the work if deemed necessary. However, these activities should not overlook peer-level issues, and therefore the decisions should not be dictated but made in consensus among the program participants.

5. Discussion

As for practical use, the model of collaborative research proposed in this paper has so far resulted in breakthroughs in a number of participating companies. The companies themselves have reported 2–5 times more frequent releases and considerable savings. Unfortunately, due to the restructuring of the Finnish ICT innovation system that took place in the summer of 2015, some of the enablers of this model were disabled, particularly in terms of funding, which was in part cut due to the economic situation and partly because funds were directed differently to support particular startup companies rather than established enterprises. Consequently, the direction that collaborative research has been heading towards is smaller consortia that still work in accordance to the baseline set in joint projects described in this paper. However, larger consortia that could take the model to the EU scale have not been easy to form, as in general such projects require considerable up-front planning rather than self-organization and continuous deployment of results in practice.

5.1. Lessons Learned and Detailed Observations

Software research with continuous, direct business impact is reachable. Despite initial claims against the model, such as we cannot conduct systematic, reliable research within three months, the research sprint approach has facilitated the division of the work effort and has resulted in smaller, more manageable portions without compromising systematic and validated research methods. Furthermore, based on the authors subjective view, the rate of adoption of results in industry has in our experience been much more rapid than in the previous models of collaboration between academia and industry, although this is difficult to measure precisely. The same phenomenon is also visible in the results of the industry interviews reported in this paper.

Scaling is natural and built-in. In our setup, numerous research institutes worked with one company to satisfy their different business cases, and a single research institute can work with numerous companies if their business cases are similar. This amplifies learning among companies, as results from different setups are made available. Furthermore, research institutes can experiment with their technology and obtain empirical results from several companies. This also supports the relevance of the research.

Fast pace of interaction is preferred. A trend in the interviews was that at the beginning of the program, the Q-reviews

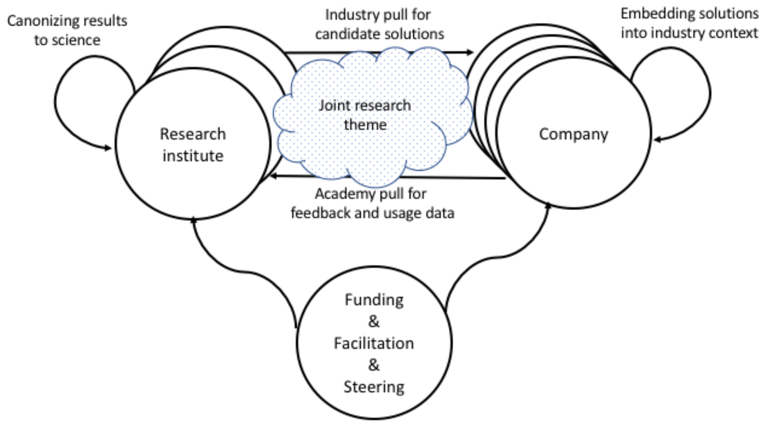

Figure 3: Technology transfer model for multi-party research.

were somewhat awkward and confusing, but after participating in them once or twice, the participants all realized that this is the event where all the action is. Consequently, to increase bandwidth and to participate in as much research as possible, some companies started to bring in as many people as possible so that no important development would go unnoticed. In addition, a common hope was that reviews would not take place only once in a quarter but more like sprints in, for example, Scrum, that is, once a month or every two weeks, as this would help integrate the results into company operations. A counter-argument for shorter program-specific research sprints is that the same (or similar) activities could also be executed at the level of CoP or research teams that work on the same topic. We also need to remember that Cloud Software and N4S were very large consortia where a quarterly Q-review rhythm was considered optimal. Smaller programs may find other rhythms more suitable.

Software research is not only technical but also about culture change. Simply transferring technical artifacts from a research institute to a company to monetize them is not a good fit to many software engineering research problems. While algorithms, tools, and technologies match well with such an approach, software engineering is also about people, process, and methods, which cannot be studied in isolation in research institutes. Instead, the industry context is an essential ingredient in such work to create meaningful research.

Attitude change towards co-creation. Probably the biggest defining factor that was mentioned in the interviews was the attitude change towards co-creation, involving both personnel working in companies as well as academics. The former truly appreciated the academic catering of potential results to utilize, to the extent of feeling like a kid in a candy-store at quarterly reviews when results were presented. As for academics, being able to present meaningful, applicable results to problems at hand in the industry requires keeping a close eye on industry needs and creating preliminary results in anticipation of what could lead to a potential breakthrough with a company. Furthermore, companies have had to reconsider their whole technology transfer chain, and it was pointed out that those companies that introduce business involvement in research in the earliest phase are the ones to benefit the most. Business cases and the fact that also companies get funded are also an important mechanism for creating commitment to achieve in the attitude change.

Company cooperation and benchmarking via academic partners. A number of companies — particularly SMEs that had a limited bandwidth to invest in Q-reviews — raised interest in following what other companies are working on via academic partners. That way, they could still follow what results others were producing and potentially later on catch up by working either directly with them or with academics who were involved. In addition, larger companies, many of which were competitors of each other and sometimes reluctant to work in direct cooperation, would initiate benchmarking over various topics, done by academics, to learn how they were positioned in comparison to each other (e.g., [56]). That said, not all inter-company collaboration need be such. Instead, the programs have produced results and activities that have been executed in collaboration by authors from companies that are competitors (e.g., [57]). At the same time, data collected from companies has resulted in numerous high-end academic forums upon further processing in academia (e.g., [58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71]).

5.2. Threats to Validity

A threat in this study is the bias in answering the interview questions, especially as each interviewee has played a key role in the respective company during the execution of the research. Still, we believe that the answers are as unbiased and honest as the interviewees were able to provide. In addition, there is also the issue of potential misunderstanding between the interviewees and the researchers. We alleviated this problem partly by member checking [72], that is, by having the interviewees review the results included in the paper. Another threat originates from directly using the data from the interviews as such without detailed triangulation.

Table 1: Case companies' relation to the proposed technology transfer model. Note that only one business case per company is addressed in the table, whereas in practice, several cases were run in parallel. For further detail, the reader is referred to Section 3.4. above.

	Canonizing results to science	Industry pull for candidate solutions	Academy pull for feedback and usage data	Embedding solutions into industry context
Company A: "Towards Continuous Experimentation"	Publications including company personnel; documented best practices based on results from a number of companies working on the same business case.	Scale up the development organization; find an efficient way to quickly grasp new market needs.	Key to the success was the close collaboration with the academic partner, and the rapid improvement loops within the company.	Continuous experimentation culture implemented for several teams in the organization.
Company B: "Introducing DevOps"	Number of academic publications that generalize the results of deploying DevOps.	Initial investigation of the DevOps phenomenon; jointly defined candidate solutions designed to meet company needs.	Real-life feedback from deployments; first-hand opportunity to witness the increasing pace of development cycles.	Candidate solutions were directly taken into use in the real industrial environment.
Company C: "Introducing Evidence-Based release planning"	Number of research articles with company authors; one company specific doctoral dissertation.	Create and improve ways of utilizing data coming from various sources in company level decision making, and leveraging that to create smaller and faster releases.	Stepwise introduction of new data collection facilities, each step introducing new data, new experiments, and new analysis methods.	Systematic, automated data collection regarding the use of new and existing features in place.
Company D: "Towards Agile Data Warehousing"	Number of academic publications where company authors are involved and that describe the generalized solution; one doctoral thesis from the company completes, two inspired to start.	A number of candidate solutions were created iteratively in multiple real client projects. The academic partners took part in the real projects with a scientific viewpoint. The solutions were taken directly into use by the clients.	The clients saw the value in contributing to the research by offering the access to their case and data for the researchers, even if not directly beneficial to the project itself.	Agile data warehouse projects were enabled, improving methods, tools, and processes for such projects using automation.

In addition, the researchers who conducted the interviews were participants of the program as well, representing the research institutes. This causes potential bias among the researchers. Again, this threat was mitigated by allowing the interviewees to review the findings from their case.

The main threats to external validity are the limited number of interviewees and the possibility that the studied organizations are not representative in the first place. Therefore, it may be that not all companies share the experiences as such. To mitigate this problem, the companies were selected so that they operate in different markets and so that their business cases cover the entire spectrum of the programs. The results may not be directly applicable in other companies or research programs as such. However, given the close relationship between the proposed technology transfer model and the fundamental aspects of how companies develop their products based on research results, it is very much possible that a similar setup could be successfully used in other research programs.

6. Conclusions

The role of technology transfer in the field of software engineering is clearly in a state of transition. In contrast to some other research fields, a lot of research is done inside companies to foster new business opportunities. At the same time, PPP plays an important role in collaborative research.

Based on the results of the case studies reported in this paper, the collaboration between industry and academia requires new models for technology transfer that emphasize the role of co-creation and co-learning instead of following a sequential, well-established path from academic research to products that a company takes to the market. In fact, to a large extent, learning is mutual, bi-directional, and very productive; the most recent estimate that includes data from five enterprises that participated in the N4S program is that the results have enabled yearly savings of 40 MEUR in these organizations only. We take this as hard evidence of the benefits of the proposed model. At the same time, collaboration provides tremendous research opportunities in the field of empirical software engineering, as companies appear to willingly provide data to researchers if they see that the research serves their interests, too.

To work smoothly in practice, the model needs to allow one research institute to serve several companies, and vice versa. The enablers of such an approach include investing sufficient time to cooperate and a set of research themes that are of interest to several research institutes and companies. Finally, the mindset of research institutes needs to change, as it is too late to start to consider candidate solutions only when a potential need arises for it in a company. Instead, there must be a general understanding of company needs, including long-term and short-term views, and new trends, solutions, and results that emerge in the research field.

Table 2: Comparison of Technology Transfer Models.

	Generic models [14, 15, 55]	**Redwine and Riddle** [20]	**NASA SEL** [22, 21, 23]	**Plfeeger** [11]	**Gorshek et al.** [12]	**Proposed model**
Domain and focus	Generic.	Technology.	Software engineering.	Software engineering.	Software engineering.	Software engineering, software technology.
Organization	Undefined.	Undefined.	Single party.	Two parties, academy and industry, distinct roles.	Academy and industry collaboration, distinct roles.	Scalable multi-party, academy and industry, multi-party teams.
Mode of operation	Technology push, industry pull, collaboration.	Technology push.	Technology push.	Technology push, industry pull.	Technology push, Industry pull.	Technology push, industry pull, collaboration.
Life cycle	4 phases: basic research, applied research, development, production & operation.	6 step technology maturation.	Various models[3], typically including training and experimentation, development, pilots, gradual use.	Sequential, 5 key activities.	Sequential, 7 key activities.	Iterative with rapid cycles.
Problem formulation and validation	Basic research, industry needs.	Basic research.	Basic research, industry needs.	Business need; Technology creation with incremental evaluation and packaging.	Identification of an industry problem; Candidate solutions built in academia, gradually increasing industry involvement.	Industry-academia consortium in collaboration; Continuous feedback through frequent releases even to end-customers.

References

[1] V. Kouwenhoven, The rise of the public private partnership: a model for the management of public-private cooperation, Modern governance: New government-society interactions (1993) 119–130.

[2] J. Highsmith, A. Cockburn, Agile software development: The business of innovation, Computer 34 (9) (2001) 120–127.

[3] M. Poppendieck, T. Poppendieck, Lean software development: an agile toolkit, Addison-Wesley Professional, 2003.

[4] K. Beck, Embracing change with extreme programming, Computer 32 (10) (1999) 70–77.

[5] T. J. Allen, Managing the flow of technology: technology transfer and the dissemination of technological information within the R and D organization, Massachusetts Institute of Technology, Cambridge, MA, 1977.

[6] D. E. Stokes, Pasteur's quadrant: Basic science and technological innovation, Brookings Institution Press, 2011.

[7] B. Shneiderman, J. M. Carroll, Ecological studies of professional programmers, Communications of the ACM 31 (11) (1988) 1256–1258.

[8] J. Buxton, R. Malcolm, Software technology transfer, Software Engineering Journal 6 (1) (1991) 17–23.

[9] B. Shneiderman, High impact research: Blending basic and applied methods, in: 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, IEEE, 2013, pp. 3–3.

[10] J. Preece, H. D. Rombach, A taxonomy for combining software engineering and human-computer interaction measurement approaches: towards a common framework, International journal of human-computer studies 41 (4) (1994) 553–583.

[11] S. L. Pfleeger, Understanding and improving technology transfer in software engineering, Journal of Systems and Software 47 (2) (1999) 111–124.

[12] T. Gorschek, P. Garre, S. Larsson, C. Wohlin, A model for technology transfer in practice, IEEE Software 23 (6) (2006) 88–95.

[13] B. Godin, The linear model of innovation: The historical construction of an analytical framework, Science, Technology, & Human Values 31 (6) (2006) 639–667.

[14] V. Bush, Science: The Endless Frontier, 1945. Reprint, North Stratford, NH: Ayer Co., 1995.

[15] A. Ione, The Idea Factory: Bell Labs and the Great Age of American Innovation by Jon Gertner. Penguin Press, New York, NY, USA 2012. 432 pp. Trade. ISBN-13: 978-1-5942-0328-2 (2013).

[16] E. Mansfield, Academic research and industrial innovation, Research policy 20 (1) (1991) 1–12.

[17] E. Mansfield, Academic research underlying industrial innovations: sources, characteristics, and financing, The review of Economics and Statistics (1995) 55–65.

[18] H. Blackwell, Emerging energy and intellectual property – the often unappreciated risks and hurdles of government regulations and standard setting organizations, The National Law Review (2012) 5–22.

[19] D. C. Mowery, R. R. Nelson, B. N. Sampat, A. A. Ziedonis, Ivory tower and industrial innovation: University-industry technology transfer before and after the Bayh-Dole Act, Stanford University Press, 2015.

[20] S. T. Redwine, W. E. Riddle, Software technology maturation, in: Proceedings of the 8th international conference on Software engineering, London, England, 1985.

[21] M. V. Zelkowitz, Software engineering technology infusion within NASA, IEEE Transactions on Engineering Management 43 (3) (1996) 250–261.

[22] F. McGarry, R. Pajerski, G. Page, S. Waligora, V. Basili, Software Process Improvement in the NASA Software Engineering Laboratory, Tech. rep., Software Engineering Institute, Carnegie-Mellon University, Pittsburgh, PA, United States (1994).

[23] V. R. Basili, F. E. McGarry, R. Pajerski, M. V. Zelkowitz, Lessons learned from 25 years of process improvement: the rise and fall of the NASA software engineering laboratory, in: Proceedings of the 24th International Conference on Software Engineering, ACM, 2002, pp. 69–79.

[24] D. L. Parnas, On the criteria to be used in decomposing systems into modules, Communications of the ACM 15 (12) (1972) 1053–1058.

[25] D. L. Parnas, P. C. Clements, D. M. Weiss, The modular structure of complex systems, IEEE Transactions on Software Engineering (3) (1985)

259–266.

[26] H. Kaindl, S. Brinkkemper, J. A. Bubenko Jr, B. Farbey, S. J. Greenspan, C. L. Heitmeyer, J. C. S. do Prado Leite, N. R. Mead, J. Mylopoulos, J. Siddiqi, Requirements engineering and technology transfer: obstacles, incentives and improvement agenda, Requirements Engineering 7 (3) (2002) 113–123.

[27] V. R. Basili, M. K. Daskalantonakis, R. H. Yacobellis, Technology transfer at motorola, IEEE Software 11 (2) (1994) 70–76.

[28] C. Wohlin, A. Aurum, L. Angelis, L. Phillips, Y. Dittrich, T. Gorschek, H. Grahn, K. Henningsson, S. Kagstrom, G. Low, P. Rovegard, P. Tomaszewski, C. van Toorn, J. Winter, The success factors powering industry-academia collaboration, IEEE Software 29 (2) (2012) 67–73.

[29] S. Beecham, P. OLeary, I. Richardson, S. Baker, J. Noll, Who are we doing global software engineering research for?, in: 2013 IEEE 8th International Conference on Global Software Engineering (ICGSE), IEEE, 2013, pp. 41–50.

[30] S. Beecham, P. O'Leary, S. Baker, I. Richardson, J. Noll, Making software engineering research relevant, Computer 47 (4) (2014) 80–83.

[31] A. Taivalsaari, T. Mikkonen, The web as a software platform: Ten years later, in: Proceedings of WEBIST'17, INSTICC, 2017.

[32] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, A view of cloud computing, Communications of the ACM 53 (4) (2010) 50–58.

[33] M. Poppendieck, Lean software development, in: Companion to the proceedings of the 29th International Conference on Software Engineering, IEEE Computer Society, 2007, pp. 165–166.

[34] B. Fitzgerald, K.-J. Stol, Continuous software engineering: A roadmap and agenda, Journal of Systems and Software 123 (2017) 176–189.

[35] P. Debois, DevOps: A software revolution in the making, Journal of Information Technology Management 24 (8) (2011) 3–39.

[36] A. N. Link, G. Tassey, Cooperative research and development: the industry, university, government relationship, Springer, 1989.

[37] H. Etzkowitz, L. Leydesdorff, The triple helix–university-industry-government relations: A laboratory for knowledge based economic development, EASST Review, Vol. 14, No. 1, pp. 14-19, 1995.

[38] R. Gustafsson, S. Jarvenpaa, Extending community management to industry-university-government organizations, R&D Management, Feb. 2017.

[39] H. Etzkowitz, Enterprises from science: The origins of science-based regional economic development, Minerva 31 (3) (1993) 326–360.

[40] D. E. Avison, F. Lau, M. D. Myers, P. A. Nielsen, Action research, Communications of the ACM 42 (1) (1999) 94–97.

[41] S. Easterbrook, J. Singer, M.-A. Storey, D. Damian, Selecting empirical methods for software engineering research, in: Guide to Advanced Empirical Software Engineering, Springer, 2008, pp. 285–311.

[42] N. Rosenberg, Why do firms do basic research (with their own money)?, Research policy 19 (2) (1990) 165–174.

[43] J. C. Mankins, Technology readiness levels – A white paper, Office of Space Access and Technology, NASA (1995).

[44] J. Järvinen, T. Huomo, T. Mikkonen, Running software research programs: an agile approach, in: Proceedings of the 39th International Conference on Software Engineering Companion, poster track, IEEE Press, 2017, pp. 314–316.

[45] H. W. Rittel, M. M. Webber, Dilemmas in a general theory of planning, Policy Sciences (4) (1973) 155–169.

[46] P. Checkland, J. Poulter, T. Williams, D. K. Hitchins, P. Hoverstadt, M. Pidd, D. A. Wren, A. G. Bedeian, K. Brotby, Dialogue mapping: Building shared understanding of wicked problems, Chichester, UK: Wiley Publishing, 2006.

[47] Cloud Software program homepage, in: http://www.cefims.eu/2013/08/cloud-software-programme-tivit/, accessed May 10, 2017.

[48] Need for Speed program homepage, in: http://www.n4s.fi/en/, accessed May 10, 2017.

[49] P. Mell, T. Grance, et al., The NIST definition of cloud computing, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg (2011).

[50] A. Di Costanzo, M. D. De Assuncao, R. Buyya, Harnessing cloud technologies for a virtualized distributed computing infrastructure, IEEE Internet Computing 13(5), 2009.

[51] J. Järvinen, T. Huomo, T. Mikkonen, P. Tyrväinen, From agile software development to mercury business, in: International Conference of Software Business, Springer, 2014, pp. 58–71.

[52] J. Humble, D. Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Pearson Education, 2010.

[53] P. Ahokangas et. al, N4S Strategic Research and Innovation Agenda, 2015, in: http://www.n4s.fi/en/strategic-research-and-innovation-agenda-published/, accessed May 10, 2017.

[54] E. Wenger, Communities of practice: A brief introduction, in: National Science Foundation (US), 2011.

[55] B. Shneiderman, Toward an ecological model of research and development: The choice between basic and applied research is a false one, The Atlantic, Apr. 23, 2013.

[56] M. Ahmad, J. Markkula, M. Oivo, P. Kuvaja, Usage of kanban in software companies: an empirical study on motivation, benefits and challenges, in: 9th International Conference on Software Engineering Advances, 2014.

[57] C. Engblom, M. Setälä, From waterfall to CD in telecom, how can that be done?, Special Session at XP 2015 Leadership and Management Track, May 2015, Helsinki, Finland.

[58] F. Fagerholm, A. S. Guinea, H. Mäenpää, J. Münch, The RIGHT model for continuous experimentation, Journal of Systems and Software 123 (2017) 292–305.

[59] S. G. Yaman, M. Munezero, J. Münch, F. Fagerholm, O. Syd, M. Aaltola, C. Palmu, T. Männistö, Introducing continuous experimentation in large software-intensive product and service organisations, Journal of Systems and Software 133 (2017) 195–211.

[60] V. Myllärniemi, J. Savolainen, M. Raatikainen, T. Männistö, Performance variability in software product lines: proposing theories from a case study, Empirical Software Engineering 21 (4) (2016) 1623–1669.

[61] S. Mäkinen, M. Leppänen, T. Kilamo, A.-L. Mattila, E. Laukkanen, M. Pagels, T. Männistö, Improving the delivery cycle: A multiple-case study of the toolchains in finnish software intensive enterprises, Information and Software Technology 80 (2016) 175–194.

[62] E. Lindgren, J. Münch, Raising the odds of success: the current state of experimentation in product development, Information and Software Technology 77 (2016) 80–91.

[63] F. Fagerholm, M. Ikonen, P. Kettunen, J. Münch, V. Roto, P. Abrahamsson, Performance alignment work: How software developers experience the continuous adaptation of team performance in lean and agile environments, Information and Software Technology 64 (2015) 132–147.

[64] P. Rodríguez, A. Haghighatkhah, L. E. Lwakatare, S. Teppola, T. Suomalainen, J. Eskeli, T. Karvonen, P. Kuvaja, J. M. Verner, M. Oivo, Continuous deployment of software intensive products and services: A systematic mapping study, Journal of Systems and Software 123 (2017) 263–291.

[65] D. Falessi, N. Juristo, C. Wohlin, B. Turhan, J. Münch, A. Jedlitschka, M. Oivo, Empirical software engineering experts on the use of students and professionals in experiments, Empirical Software Engineering (2017) 1–38.

[66] D. Fucci, H. Erdogmus, B. Turhan, M. Oivo, N. Juristo, A dissection of the test-driven development process: Does it really matter to test-first or to test-last?, IEEE Transactions on Software Engineering 43 (7) (2017) 597–614.

[67] N. Taušan, J. Markkula, P. Kuvaja, M. Oivo, Choreography in the embedded systems domain: A systematic literature review, Information and Software Technology 91 (2017) 82–101.

[68] D. M. Fernández, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetrò, T. Conte, M.-T. Christiansson, D. Greer, C. Lassenius, et al., Naming the pain in requirements engineering, Empirical Software Engineering 22 (5) (2017) 2298–2338.

[69] T. Karvonen, W. Behutiye, M. Oivo, P. Kuvaja, Systematic literature review on the impacts of agile release engineering practices, Information and Software Technology, 2017.

[70] E. Laukkanen, J. Itkonen, C. Lassenius, Problems, causes and solutions when adopting continuous deliverya systematic literature review, Information and Software Technology 82 (2017) 55–79.

[71] T. O. Lehtinen, J. Itkonen, C. Lassenius, Recurring opinions or productive improvements – what agile teams actually discuss in retrospectives, Empirical Software Engineering 22 (5) (2017) 2409–2452.

[72] J. W. Creswell, Research Design - Qualitative, Quantitative, and Mixed Methods Approaches, Sage Publications Sage CA: Thousand Oaks, CA, 2014.

Appendix: Interview protocol

In the following, the term *work item* is used to refer to the concrete artifacts produced during the research collaboration. A sequence of such artifacts is to be identified as the primary unit of analysis for a case. The term work item is not meant to be used in the interview as such, but to be replaced by the term(s) the interviewee(s) used.

Task 1: Identification of potential work item sequences.

Task 1.1. Please identify some (important) work items, such as problems or solutions or solution candidates, you / your company worked on in the project.

- For example, posters produced for Q-reviews or Jira items can be used as a basis for refreshing the memory of the interviewee.
- A set of deliverables from the company may have been collected beforehand as a starting point and used here as well.

Task 1.2. Identify each continuation from one work item to another (e.g., a solution was created to a problem, an item was refined, a subproblem was identified and turned to a separate work item, two work items are a part of a larger whole, etc.).

Task 2: Interview. Select one work item sequence and explain its life in the project.

- From where does the initial problem (behind a solution) originate?
- Was the academic literature searched and analyzed in the early stages of the problem formulation?
- Please explain how candidate solutions were searched and created.
- Can you identify different versions or variants of candidate solutions?
- How was the collaboration with the academic partners in early / these stages?
  - What did you do together?
  - What was done at the academic partner(s) in relation to the work item?
- Did you proceed to implement the solution?
  - If yes, explain what you did.
  - If no, why not?
- What did you do to test if the solution is good?
  - What was the role of the academic partners in this, if any?
- What is the status of the solution at the moment?
  - What led to the current situation?
  - Is the solution in real use?
  - How would you characterize the value of the solution?
- What challenges did you encounter during the life (of the work item in the project)?
- How would you explain the main lessons you learned?
- What factors do you think contributed to the success of the item?
- Are you aware of any academic publications or theses (produced within the project) related to the work item?
- Is there anything you would like to add that was not covered in the interview?

Task 3: Discussion about technology transfer. Show the Gorschek et al. model [12] to the interviewee.

- How do you see the model in relation to the life just discussed?
- If you think of your activities in the project more widely, how do they relate to the model?
- How do you see the model as depicting industry-academia collaboration?
- Based on the project, do you have any ideas about how to modify the model to better describe an ideal model for industry-academia collaboration?
- What is the best form for academic collaboration from the perspective of your current position (or the one during the project if more appropriate) in the company?