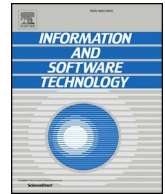




Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Continuous and collaborative technology transfer: Software engineering research with real-time industry impact

Tommi Mikkonen^{*,a}, Casper Lassenius^b, Tomi Männistö^c, Markku Oivo^d, Janne Järvinen^e

^a Department of Computer Science, Gustav Hållströmin katu 2b, Helsinki, Finland

^b Aalto University, Espoo, Finland

^c University of Helsinki, Helsinki, Finland

^d University of Oulu, Oulu, Finland

^e F-Secure, Helsinki, Finland

ARTICLE INFO

Keywords:

Technology transfer
Collaborative research
Software engineering research
Public-private partnership

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 This model is outdated in contemporary software engineering research that is done in close collaboration between academia and industry and in large consortia rather than on a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 and development methods, many discoveries result from seamless collaboration between companies and research institutes.

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 7¹ 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 processes 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, 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].

* Corresponding author.

E-mail addresses: tommi.mikkonen@helsinki.fi (T. Mikkonen), casper.lassenius@aalto.fi (C. Lassenius), tomi.mannisto@helsinki.fi, tommi.mikkonen@tut.fi (T. Männistö), markku.oivo@oulu.fi (M. Oivo), janne.jarvinen@f-secure.com (J. Järvinen).

¹ <https://ec.europa.eu/research/fp7>.

² <https://ec.europa.eu/programmes/horizon2020/>.

<http://dx.doi.org/10.1016/j.infsof.2017.10.013>

Received 12 May 2017; Received in revised form 18 October 2017; Accepted 19 October 2017
0950-5849/ © 2017 Elsevier B.V. All rights reserved.

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 1960s and 1970s, and therefore the baseline resembles earlier work, proposing six steps that take technology from research towards popularization and large-scale use. A similar, although slightly simpler transition process is proposed by Zerkowicz [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 (Fig. 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 (Fig. 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

Download English Version:

<https://daneshyari.com/en/article/6948106>

Download Persian Version:

<https://daneshyari.com/article/6948106>

[Daneshyari.com](https://daneshyari.com)