Abstract

Software producing organizations face the challenges of changing demands, rapidly evolving technology, and a dynamic ecosystem in which their products and services need to operate. While the challenges remain at the intersection between technology and business, research focuses on three streams within the literature. Because of the pervasiveness of software, the challenges are observed everywhere in the economy, whether it is logistics, online marketing, or e-health. Furthermore, they are applicable to any software producing organization, whether it is a software startup creating apps, a software producer in its growth period, or a larger producer that manages its platform and ecosystem. Therefore, over the course of three days, we aim to bring together scientists and practitioners from the software production domain. The meeting of different sub-communities leads to the creation of a joint research agenda for the further development of the field.

1. Description of the Seminar: Topics and Goals

Software producing organizations (SPO) face challenges every day. Whether they are open source consortia or commercial software product companies, they all face the challenges of changing demands, rapidly evolving technology, and a dynamic ecosystem in which their products and services need to operate. The challenge to make these organizations successful is multi-disciplinary (see Table 1). First, there exist technology challenges, such as eliciting and prioritizing requirements, dealing with platforms and technology standards, and operating in complex technology landscapes that constrain and enable their technology. Secondly, there exist adoption challenges: organizations need to find ways to convince their target users to adopt their technologies and to coordinate evolving technologies to provide the most valuable end-user experience. Thirdly, there exist business model challenges, where these organizations must find ways to maximize profit from their innovations and technologies. Because of the pervasiveness of software, the challenges are observed everywhere in the economy, whether it is logistics, online marketing, or e-health. Furthermore, they are applicable to organizations in every stage of development, whether it is a software startup or a software giant that has influenced the market consistently for decades.

Software producing organizations are successful in advancing society. However, there exist some particular problems that still require solving.

- Location of Startups - First, the European market faces a very different ecosystem from companies in silicon valley: a fragmented market, a fertile but less ambitious startup
ecosystem, and competition from many sides. If the European software industry is to succeed, a fertile ecosystem needs to be created that can bring forth the next Google, Amazon, or Apple in Europe.

- **Innovation for growth** - There is presently a trifecta of different IT innovations: Internet of Things [22], Big Data, and Augmented Reality [23]. These three innovations will continue to challenge the status quo of IT in society and continue to challenge software companies to innovate, whether they are building traditional business applications or innovative open source solutions.

- **Ecosystems everywhere** - Software producing organizations have managed to innovate, like no others, in chains. It is hard to find any software technology that does not stand on the shoulders of a multitude of layers of platforms. Software producing organizations need to use these ecosystems to innovate and provide new solutions. Innovations in this domain, such as app stores, can change complete industries [8].

*Table 1: Overview of Software Business Challenges and its multi-disciplinary research field.*

<table>
<thead>
<tr>
<th>Dimensions of software business challenges</th>
<th>Technology challenges</th>
<th>Adoption challenges</th>
<th>Business model challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>eliciting and prioritizing requirements</td>
<td>provide the most valuable end-user experience</td>
<td>maximize profit from their innovations</td>
</tr>
<tr>
<td>Research Domain</td>
<td>Engineering &amp; technology research</td>
<td>Technology management research</td>
<td>Business research</td>
</tr>
<tr>
<td>Dominant Research Stream</td>
<td>Platform Orchestrators</td>
<td>Growing Software Producers</td>
<td>Startups</td>
</tr>
</tbody>
</table>

In Section 6.1, we provide an overview of the relevant literature along three dominant research streams and the selected research methods within the literature. Thereafter, Section 6.2 suggests five central research challenges and presents the seminar’s goals.

1.1. **Lifecycle Challenges of Software Producing Organizations**

SPOs go through several stages of maturity during their lifecycle. At each stage, the SPO meets several challenges, many of which presenting academic challenges as well. The stages that we explicate here are the startup phase, the growth phase, and the platform orchestration phase.

1.1.1. **Startups**

Starting organizations have several relevant challenges to tackle [9, 84]. Startups are operating in a dynamic environment and have the opportunity to shake up existing technology value chains, change
society for the better, and bring technological innovation to market rapidly. Simultaneously, they run the risk of failing for one of the many reasons that startups fail [12, 48, 68]. According to a recently published research agenda for software startups [101], the following three challenges for startups should be addressed.

First, further research is necessary into business model design [13, 14, 25, 27, 34] of software startups. Current research is geared towards supporting startups in pivoting [15, 24], i.e., changing the business model or technology completely based on market feedback. Furthermore, study is being done on what qualities a minimum viable product should have [16]. Startup research also needs to look at legal challenges [20]. More research is needed on how startups can facilitate change without going under first. There are examples of startups that have dubious reputations (Uber), simply because society is not ready for the innovations that these startups bring. Further analysis is needed on which of these startups provide a positive effect to society. We need to find out whether models for assessment of organizational responsibility are applicable to startups as well [19].

Secondly, while research has established that the environment of software startups is extremely dynamic, unpredictable and even chaotic, research has also shown that the engineering practices are mostly ad-hoc leading to poor quality realization of concepts [102, 103]. The research should improve our fundamental understanding of software based startups in the form of novel startup evolution and pattern models, high-quality empirical software engineering research on new methods, frameworks, processes, techniques and engineering practices. Research should also develop tools to support startup processes as well as to improve the software engineering practices and offer new ways to improve the quality and effectiveness of the concepts on which the tomorrow's startups are built.

Thirdly, when looking at the startup ecosystem, there are many opportunities for research into factors that enable startups, such as economic factors and enablers [21, 104], startup education [11], the role of incubators, and policies that can encourage or discourage innovation through startups.

1.1.2. **Growing Software Producers**

The software industry is typically populated by small to medium enterprises, with several somewhat larger keystone firms [31]. These firms are struggling with challenges [78] of maturing technology (i.e., legacy systems, technical architectures, professionalization of requirements engineering and product management functions), business development, hybrid business models and licensing [42], and dealing with their position in software ecosystems.

With regard to maturing technology, SMEs have had to deal with different technological transitions. Together with little guidance offered by existing development methods, SPOs have little guidance during the early activities within the development process [95, 96]. An area that has received abundant attention is the transition to Software as a Service [51, 52, 53, 59, 81, 82] and cloud software [54]. Also, there is increased research into the app economy [56, 39, 49] and the role it plays in the success of software production businesses. Another challenge is product development: the how
and what of software production processes. One of the areas that has received large amounts of
attention is agile product development [16, 70] and also continuous software engineering [64].

Business development can be interpreted as strategic development of a software production
business. These include larger decisions, like going international, forging strategic partnerships,
outsourcing parts of the software production processes, etc. The field of internationalization and
software export has received large amounts of attention but remains challenging, no matter theory available [28, 79, 83]. Furthermore, forging strategic partnerships currently is seen as an art [80], but not as a well founded practice yet [57] and more research is needed in this area. Also, outsourcing the most important process of software production, i.e., product development [10, 26], has been studied, yet organizations still experience many challenges when doing so.

Business model design is a topic that has gained abundant attention in different fields, and the
software business domain is no different [61, 63, 65, 66, 76]. It remains interesting to study how businesses can be run in a sustainable manner with propensity for growth. A relevant sub-area of this domain is software pricing, which still has not received sufficient attention [56, 94].

Finally, an issue that is becoming more prevalent in every industry is gathering product performance
data in the field to enable continuous development and improvement of products and customer
experience [46]. The collection of software operation knowledge and enactment on it is a continuing theme that deserves more attention in future research [90].

1.1.3. **Platform Orchestrators**

Larger firms in the software industry have challenges in the domains of ecosystem orchestration,
runtime partnerships, dealing with openness, managing an organization’s portfolio of products,
technology management, and mergers and acquisitions.

Larger firms have the challenge to mobilize small players around them to collaboratively serve a
market [8], forming software ecosystems. One of the main challenges for these organizations is how
to organize the ecosystem and govern these [73, 69]. Relevant questions center around ecosystem
health [23, 32], ecosystem analysis [74], and open source [92]. Organizations also need to be aware
of the influence of open application markets and app stores [39, 49].

Furthermore, organizations need to be aware of the power distribution in the ecosystem and the
ways in which power can be used to maximize revenues in a software market [91]. Furthermore,
larger firms need to deal with the intricacies of ‘coopetition’, where organizations may be competing
in one domain while having to collaborate in another [8, 62, 67].

Openness is a challenge organizations need to deal with as well. The influence of open source, open
platforms, and open development processes on the software industry is significant [40, 67, 71, 77,
85]. Increasingly, organizations are partly or even completely open, leading to relevant hybrid
business models. Another domain related to this is the function of standards for becoming successful
in software business [88].
As organizations grow larger, their products become legacy. Typically, an organization may be working on several versions of one product at the same time, and even complete product portfolios [75]. With continuous technology improvement, organizations also need to make sure technological innovations are implemented accordingly [41]. Innovations in domains such as Internet of Things, data ecosystems [72], and augmented reality have a continuous influence on the success of current software products.

Finally, as organizations grow they realize that their growth strategies can be broader than just autonomous growth. One particular growth strategy that is popular in software business currently, is the practice of mergers and acquisitions [29, 33, 36, 44]. Further analysis is needed on why mergers and acquisitions are generally more successful in the software industry.

1.1.4. Research Methods in the Software Production Domain
Typical research methods in the domain are surveys, case studies [17], in-the-field experiments [97], structure literature reviews [98], and design research [93]. Overall, the research community follows a data-driven approach, as suggested by the research paradigm of evidence-driven software engineering [99]. Depending on the research challenge, the community benefits from both, qualitative and quantitative research approaches.

Survey data is used in the domain consist of large surveys where organizations are periodically asked to provide financial and process data about their business on the one hand. On the other hand, surveys are in-depth comparative case studies, that are used to understand differences across demographics [18, 35, 38].

Case studies in the software business domain are typically short-term research projects that observe one or several companies closely, doing interviews, participant observations, software study, and artefact study. Often, these studies help to establish lessons learned and develop new frameworks [100].

Structured literature reviews help the community to aggregate and to synthesize prior evidences and research results. Therefore, prior research streams can be investigated and commonalities as well as differences can be identified. Building on this aggregated view, further synthesize leads to the development of new artifacts, such as framework or methods [96].

More recently, design science research has gained more momentum. It helps the discipline to systematically extract requirements, derive clear design principles and suggest a corresponding instantiation of such artifact [96].

The research in software business would benefit from a more structured and foundational methodological body of knowledge, which is one of the topics that will be handled in the Seminar.
The time to bring together experts in software business research is now. Consolidating their current research, output in the form of research proposals, and establishing a research agenda for the future is of vital essence for the software industry, and the European industry in particular. The Dagstuhl seminar will provide a foundational event and milestone for the software business community.

1.2. **Goals of the Seminar**

We find five goals for the Dagstuhl seminar, that we wish to achieve:

- Establish a joint research agenda in order to strengthen cross-community research efforts
- Examine expertise reports for practitioners to increase accessibility of research data and results
- Define target outlets to find and communicate a common denominator
- Initiate collaborative research projects to improve research relevance and rigor
- Investigate instrument needs to foster discussion with industry.

**Research Challenges**

As a result, five Research Challenges have been identified that will be tackled during the Dagstuhl Seminar:

- **RC1. Prioritization and Alignment of Research Agendas** - There are many researchers active in the domain, many of them working on related topics. In the seminar, we aim to determine a heartbeat for the community, and align and prioritize research topics.

- **RC2. Lack of Research Data** - There exists a lack of easily accessible research data for the software business community. There are few case studies, international surveys, etc. that can support researchers in building up a foundation of theory under software production processes.

- **RC3. Acknowledgement of the Research Community** - The research community is highly active, but also highly dispersed across different domains. Publications end up in business journals (Journal of Management Education, MISQ), technical journals (Empirical Software Engineering, Information and Software Technology, Journal on Software and Systems), and several books have been published. We need joint publication outlets and events to collaborate at and spearpoint our research efforts [86].

- **RC4. Focus on Methods** - The community is typically doing research in the empirical domain, whether it is the execution of case studies, surveys, or even experiments in an app store. There lies an opportunity to structurally improve methodical education in the domain of software production.

- **RC5. Address the Gap Between Industry and Science** - Many of the topics researched in the software business community appear never to make it to the tables of CXOs, i.e., the entrepreneurs that lead software producing organizations. We need new instruments to actively approach the industry and provide them with knowledge and data [87].
Synergies and Research Questions

The scientific seminar is a combination of expert lectures in the morning and working sessions in the afternoons. All [three] research streams mentioned above are assumed to stay relevant for the next 5-10 years within the software business research community. While we also expect the community of software business to continue its growth, we need to avoid duplication and benefit from synergies across sub-communities. Therefore, we derive the following five research questions:

- What are the synergies between the sub-communities within software business research?
- How can we increase the accessibility to research data?
- What are commonly accepted and desirable research outlets of the community?
- How can we advance the methodological rigor of our research domain?
- What instruments are needed to foster the discussion with practitioners?

The suggested Dagstuhl seminar will bring together:

- Researchers working on tools and methods for software businesses
- Experts in developing and managing software ecosystems
- Specialists in mergers and acquisition of software businesses

Impact on the Research Community

The organization of the Dagstuhl seminar helps to bring the community on software business together. The software business community currently advances the field through different conferences and workshops. Examples are the International Conference on Software Business (ICSOB), International Workshop on Software Ecosystems (IWSECO), International Workshop on Software Product Management (IWSPM), International Workshop on Software Startups (IWSS) and others. Therefore, the seminar can help the development and organization of joint conferences in the future.

We also seek to bring the community and its sub-communities closer together. There are different workshops and conference. Participants of one event may not be aware of the results from another. Hence, this Dagstuhl seminar helps to raise awareness of this community and its different facettes. The Dagstuhl seminar provides a great vehicle to bring the community together, while also developing a cornerstone for future reference.

Besides facilitating inter-group knowledge exchange related to research context, we also see this as an opportunity to share learnings related to research rigor and education. A research exchange related to current research activities also leads to the exchange of prior experiences. Therefore, we can learn jointly as a community from our mistakes and further increase the rigor of this research community. The results of one sub-community may trigger further research in another. Fundamental research related to software spark new application-driven research and vice-versa.

The applying researchers are well-known in the area of software business research. They have chaired many international conferences and advance the research community otherwise. Examples
conferences are the International Conference on Advanced Information Systems Engineering CAiSE 2012 (General Chair - Brinkkemper), European Conference on Information Systems ECIS 2013 (General Chair - Brinkkemper), International Conference on Product Focused Software Process Improvement PROFES 2015 (General Chair - Abrahamsson), PROFES 2016 (Program Chair - Abrahamsson), IEEE International Technology Management Conference ICE 2016 (General Chair - Abrahamsson), International Conference of Software Architecture ICSE 2018 (Program Chair - SEIP track Bosch), International Conference on Software Engineering ICSE 2018 (Program Chair SEIP track Bosch), Design Science Research in Information Systems and Technology DESRIST 2017 (Program Chair - Maedche). Also, the applying research scientists are part of the Editorial Board in notable Journals, such as Journal of Information and Software Technology (Brinkkemper), Journal of Database Management (Brinkkemper), Journal of Systems and Software (Bosch), Science of Computer Programming (Bosch), Journal of Software: Evolution and Process (Abrahamsson), e-Informatica Software Engineering Journal (Abrahamsson) [...]. Other notable engagements are the co-founding of Advisory Board SoftwareVOC\(^1\) (Brinkkemper), co-founding the International Software Product Management Association ISPMA\(^2\) (Brinkkemper), founding of the international Software Startup Research Network SSRN\(^3\) (Abrahamsson)

---

\(^1\)SoftwareVOC - a national platform for the Netherlands Software Industry with about 250 member companies

\(^2\)ISPMA - an international not-for-profit association for the education and certification of product managers in the software industry world-wide

\(^3\)SSRN - an international not-for-profit network for researchers focusing on engineering aspects of software startups. Web-site: www.softwarestartups.org
2. Relation to Previous Dagstuhl Seminars

Within the last four years, we find ten seminars that intend to improve software in one way or another. We suggest three classes of seminars. First, we find seminars that are related to the software system itself. Seminars focused on the structure and functionality of a software. Second, we find seminars that suggest improvements related to the development itself. Often, they seek to improve task accomplishment of the software developers themselves. Third, we find seminars that are related to specific elements of software as a business, its management and improvement. Following, we present each class and their seminars in further depth.

**Systems Focus** – Prior seminars achieved great success in order to improve the software systems itself. A first example is the Seminars on “Software and Systems Traceability for Safety-Critical Projects” held in April 2015. Given the long lasting problem of traceability in safety critical software and systems, the seminar started with a new clean slate to identify existing needs and challenges in order to suggest corresponding solutions. A second example is the seminar entitled “Evaluating Software Verification Systems: Benchmarks and Competitions” that was held in April 2014. Generally, the seminar focused on the advancement of comparative empirical evaluation of software verification. More specifically, the seminar discussed and explored different verification tools, bridged the gap between different verification communities, raise mutual awareness and promote collaboration between such communities. A third example is the seminar “Software Engineering for Self-Adaptive Systems: Assurances” held in December 2013 discussed problems of assurances related to self-adaptive systems. Such phenomenon helps practitioners to increase cost-effectiveness of software development while still maintain and/or increasing many functional and quality requirements. Scholars discussed four different perspectives, i.e. criteria for, composition and decomposition of, feedback loops, and perpetual provisioning of assurances.

**Development Focus** – In the past, a selection of seminars had objectives that assist developers in their daily operations to deliver great software. Here, our first example is the seminar entitled “Next Generation Static Software Analysis Tools”. In august 2014, researchers discussed different decision procedures and their integration into the toolbox of developers. The objective was to reduce the widespread of software errors that lead software systems to malfunction, crash or behave incorrect. Second, “Human-Centric Development of Software Tools” is an example of a seminar that focused on assisting software developer in provides tool assistance with enhanced usability. The seminar discussed theories and methods, formulated major research questions and took up the challenge of greater issues related to the community, such as training and reviews methods as well as recruitment. The third seminar example entitled “Software Development Analytics” was held in June 2014. The group of researcher identified the increasing amount of data that is being generated during the software development process as a great means to improve the same. The fourth seminar “Integration of Tools for Rigorous Software Construction and Analysis” from September of 2013 focused on the integration of two approaches in order to enhance the quality of software systems. The approaches state- and machine-based formal methods guide many modelling and analysis techniques used during the construction of software systems.
**Business Focus** – In addition, there have been three seminars that take a rather business oriented view towards software. These seminars are more closely related to the proposed seminar on software business. First, in April of 2014 the seminar “Unifying Product and Software Configuration” sought to combined the configuration expertise from two communities, i.e. the configuration of hardware artifacts and the configuration of software artifacts. Given the trend of combining software, hardware and related services, the group sought to classify existing problems and understand possible modelling approaches. Second, the seminar “Software Certification: Methods and Tools” was held in January of 2013. The researchers investigate the process of software certification in general, and its methods and tools in particular. A certificate has business value to the receiving entity. However, existing approaches are insufficient and more cost-effective approaches are required. Third, in September of 2013 the seminar entitled “Cloud-based Software Crowdsourcing” investigated the novel idea of using crowdsourcing to develop and outsource development tasks during when creating software. The seminar established a theoretical framework, identified a reference architecture and developed a roadmap for software crowdsourcing.

**References**


Organizations are complex systems that need to respond to a variety of changes while operating in a dynamic environment. They involve multiple stakeholders each having a domain-specific perspective that relies on concepts and languages relative to individual...Â We use cookies to make interactions with our website easy and meaningful, to better understand the use of our services, and to tailor advertising. For further information, including about cookie settings, please read our Cookie Policy. By continuing to use this site, you consent to the use of cookies. Got it. We value your privacy.