

Hitting the Target: Practices for Moving Toward Innovation Experiment Systems

Teemu Karvonen¹✉, Lucy Ellen Lwakatare¹, Tanja Sauvola¹, Jan Bosch²,
Helena Holmström Olsson³, Pasi Kuvaja¹, and Markku Oivo¹

¹ University of Oulu, Oulu, Finland

{Teemu.3.Karvonen, Lucy.Lwakatare, Tanja.Sauvola, Pasi.Kuvaja,
Markku.Oivo}@oulu.fi

² Chalmers University of Technology, Göteborg, Sweden

Jan.Bosch@chalmers.se

³ Malmö University, Malmö, Sweden

Helena.Holmstrom.Olsson@mah.se

Abstract. The benefits and barriers that software development companies face when moving beyond agile development practices are identified in a multiple-case study in five Finnish companies. The practices that companies need to adopt when moving towards innovation experiment systems are recognised. The background of the study is the Stairway to Heaven (StH) model that describes the path that many software development companies take when advancing their development practices. The development practices in each case are investigated and analysed in relation to the StH model. At first the results of the analysis strengthened the validity of the StH model as a path taken by software development companies to advance their development practices. Based on the findings, the StH model was extended with a set of additional practices and their adoption levels for each step of the model. The extended model was validated in five case companies.

Keywords: Software development · Agile development · Feedback loops · Innovation experiment systems · Continuous deployment

1 Introduction

Traditionally, R&D's assumptions about desired product functionality are based on a list of requirements or product backlog items that are gathered by product management. However, customer needs might change very rapidly, and they are often difficult to identify. This may lead to a situation in which R&D spends time and effort on developing product functionality that doesn't add value for customers. To solve this problem, agile methods [1] offer a set of practices that allow for shorter development cycles and more frequent interaction with customers. In conjunction with agile methodologies, approaches, such as innovation experiment systems (IES) [2] and continuous experimentation [3], emphasise data collection practices and continuous validation with customers in order to improve R&D accuracy and

customer responsiveness. However, while these approaches are attractive to companies in the software industry, they require an evolution of the company's current ways of working. Typically, and as recognised by Olsson et al. [4] in the Stairway to Heaven (StH) model, software development companies most often evolve from traditional development to agile R&D, from agile R&D to continuous integration (CI), from CI to continuous deployment (CD) and from CD to R&D as IES. While the first step in the StH model is characterised by long feedback loops and slow cycles, the later steps enforce fast feedback, rapid cycles and data-driven development practices in which feature value is continuously validated with customers. As recognised in this research [4], a number of opportunities and challenges are associated with the evolution from one step to the next.

In this study, and based on multiple-case study research, we investigate how five Finnish software development companies evolve their software development practices according to the steps in the StH model. In particular, we identify the benefits and the barriers they experience when moving beyond agile practices and towards IES. Therefore, our research questions are:

RQ1 What are the benefits and barriers that software companies experience when moving towards IES?

RQ2 What are the key practices that software companies need to adopt in order to evolve their software development practices according to the StH model?

The contribution of this paper is threefold. First, we strengthen the validity of the StH model, as introduced by Olsson et al. [4], in terms of the typical evolutionary path that software development companies take when advancing their development practices. Second, we extend the model with a set of practices required for climbing the steps in the StH model. We also identify the four levels at which these practices can be adopted. Third, we validate the extended model in five Finnish software development companies using a multiple-case study approach involving qualitative interviews at each company.

The rest of the paper is structured as follows. Section 2 introduces related work that is relevant to this study and, most importantly, we present the StH model. In Section 3, we extend the StH model and introduce the practices that companies apply when evolving towards IES. Section 4 presents our case study design and research method. In Sections 5 and 6, we validate the extended model and discuss the case study findings. Section 7 concludes this paper and suggests topics for future research.

2 Related Work

Today, most companies have adopted agile methods, and different flavours of the methods have become the *de facto* way of working in the software industry [5]. In allowing for more flexible ways of working with an emphasis on customer collaboration and speed of development, agile methods help companies address many of the problems associated with traditional development [6]. As a way to further advance agile development practices, companies are moving from release cycles of 6 to 12 months to more frequent software releases [7, 8, 9, 10]. To achieve this, companies increasingly

adopt practices such as CI [11], continuous delivery [12] and CD [13]. In empirical studies, Claps et.al [13] and Leppänen et al. [14] recently identified multiple benefits and challenges associated with the adoption of CD. They noted that some of the benefits include faster feedback, more frequent releases, reduced risk for each release and improved productivity and quality; some of the challenges include customer preferences to non-frequent release, domain constraints and manual testing. The adoption of these practices reflects an evolution in which companies move beyond agile practices towards R&D practices characterised by short release cycles, frequent customer validation and fully automated testing and deployment practices. Although the same agile R&D principles apply, moving beyond agile practices means: a) integrating business strategy planning, operations and other corporate functions into shorter development and release cycles [4], [15]; b) utilising automated testing practices that allow for frequent builds [12] and c) implementing continuous experimentation and innovation with customers [2, 3, 4] to better understand real customer needs. The specific aspects involved in going beyond agile as well as more holistic views of agility have been discussed in recent SE studies [15, 16] and especially in the context of lean software development [17]. As recognised in these studies, the main motivation for companies moving beyond agile is that, even though agile practices can improve R&D efficiency and product quality, they are insufficient for achieving benefits in a business ecosystem [18] and at the enterprise level [16]. To realise benefits at these levels, companies need to scale the benefits they experience at a team level, that is shorter development and feedback cycles, to include product management and customers. In order to better understand this evolutionary path, we outline the StH model below. The model describes the steps that companies may take when moving towards IES and it works as the basis for our discussion on how to improve company competitiveness and customer responsiveness.

Based on significant empirical experience as well as numerous studies that have described the transition from traditional development to more agile ways of working and beyond, the StH model [4] describes the typical evolution path for software development companies that are evolving their ways of working. In capturing this transition in five steps, the model reflects much of the prominent research in the field, and it helps understand the way in which most companies advance their software development practices. Based on empirical research as well as the authors' previous experiences of working with software development companies, the model also outlines the actions that companies need to take when climbing the different steps and advancing their ways of working. In previous research [4], the model has been used as a tool to identify where the company is in its evolutionary path and what actions it needs to take to advance. It has also been useful for describing the fundamental change that software development goes through when a company attains the final step on the stairway and when R&D is viewed as an experiment system in which customers are involved in continuous, real-time validation of software functionality [19].

The StH model views evolution from the point of view of four stakeholders: 1) the R&D organisation, 2) the validation and verification organisation, 3) the customers and 4) the product management organisation. In the StH model, the 'traditional development', step A, is characterised by long development cycles. Development processes are sequential and teams are typically large and separated into disciplines

[20]. In step B, the R&D organisation starts adopting agile development practices, typically by introducing smaller cross-functional development teams that work in shorter cycles [1]. However, at this step, product management and system verification still work according to the traditional development approach. In step C, practices for CI are adopted, including automated builds and automated testing [11]. In this step, both R&D and system verification work in short cycles and there is always a shippable product. In step D, CD is adopted and the customers are involved in short cycles with frequent software releases [13]. Code changes are pushed to the customer allowing instant feedback on new functionality. In step E, companies adopt data collection mechanisms to continuously learn about customer behaviour and product use. Feature experiments are run on a continuous basis and the collected data steer the R&D organisation [2, 3]. Rather than being specified by the product management in the early phase of development, requirements evolve based on data collected from real-time customer use. In this step, the entire organisation, including product management, is involved in short feedback cycles. In each of the steps, the level of integration of and interaction between company functions increases. Thus, steps D and E cannot be achieved without R&D, product management and customers that work in short development cycles. One implicit premise of the StH model is that evolution starts from traditional development. While this is typically the case for large-scale software development companies, it might not be the situation for smaller companies and new software start-up companies. As described by Ries [21], those types of companies are typically created much closer to the last steps in the model, i.e. the CD and the IES steps. However, as a model that pictures the general evolution path, StH depicts the different steps that are relevant for most companies and the evolution that most often occurs between those steps.

In this study, we use the StH model as the theoretical basis from which to explore the benefits and barriers experienced by five Finnish companies as they climb the steps described in the model. The StH model is outlined in Fig. 1. Although the StH model has been widely used in many software development companies, and it has been referred as ‘the typical evolution path’, so far the validation of the model has been limited to use of the multiple-case study method. However, in the absence of research that can validate the typical evolution steps towards IES, the assumptions that the StH model makes about companies that take those steps seem to be well aligned with recent SE literature and practitioner reports describing the companies’ strategic goals and experiences in adopting agile and CI practices. According to Claps et al. [13], so far only a few companies have succeeded in deploying software continuously to their customers. At Gap Inc. [7], the transformation to agile was started by first selecting a pilot project that made a big investment in the company’s CI system. At Conject AG [8], the transition from the traditional six-month release cycle to the continuous flow of small releases was enabled by aligning coding and testing activities to the same short cycles, by test automation and by implementing the CI system. At Rally Software [9], the transition from an eight-week release cadence to continuous delivery of software was enabled by abandoning time-boxed Scrum sprints and by adopting lean practices, such as Kanban and Kaizen. By first developing a better understanding of the entire process, the company was able to

make many changes to the development process. In addition, the company's Sales, Marketing, Support, Technical Account Managers and User Learning teams were affected. NASA Ames Research Center [10] was able to move from a six-month delivery cycle to a three-week cycle within two years due to the evolution of its development practices. This two-year evolution process was described as a journey from traditional to lean and then to agile.

3 Extending the Stairway to Heaven Model

In this section, we extend the StH model by integrating it with practices that are important for companies that are evolving towards IES (Fig. 1). Those practices are suggested based on empirical research as well as the authors' previous experiences of working with software development companies. To categorise the practices, we use the Business, Architecture, Process, Organisation (BAPO) approach, i.e. interdependent software development concerns as outlined by Linden et al. [22]:

- **Business:** How to make a profit from your products;
- **Architecture:** The technical means to build the software;
- **Process:** The roles, responsibilities and relationships within software development;
- **Organisation:** The actual mapping of roles and responsibilities to organisational structures.

The purpose of this extension is to allow for a more precise analysis of both the company's current ways of working and the practices they may need to adopt to further evolve. There are four levels at which the adoption of these practices can take place.

- **Not adopted:** The practice is not adopted or it is abandoned.
- **Team:** The practice is adopted in some teams. Some teams inside the organisation can be ahead of the rest of the organisation.
- **Product:** The practice is adopted at the product organisation/program level. Some product organisations can be ahead of the rest of the organisation.
- **Institutionalised:** Practice is fully adopted; it is the standard way of working throughout the entire organisation.

The application of the extended StH model in five case companies is demonstrated in Section 5. It should be noted that there is no "one and right" way for companies to evolve towards IES. Different software engineering processes have to be tailored to fit the particular business goals of the organisation, the specific context of the organisational culture, etc. Thus, we don't consider the practices we present below as prescriptive in that they have to be deployed in a certain way in a company. Rather, they are descriptive, and they suggest actions that are needed when advancing between the different steps in the model. However, when implemented in a company context they require careful adjustment to fit the particular company context.

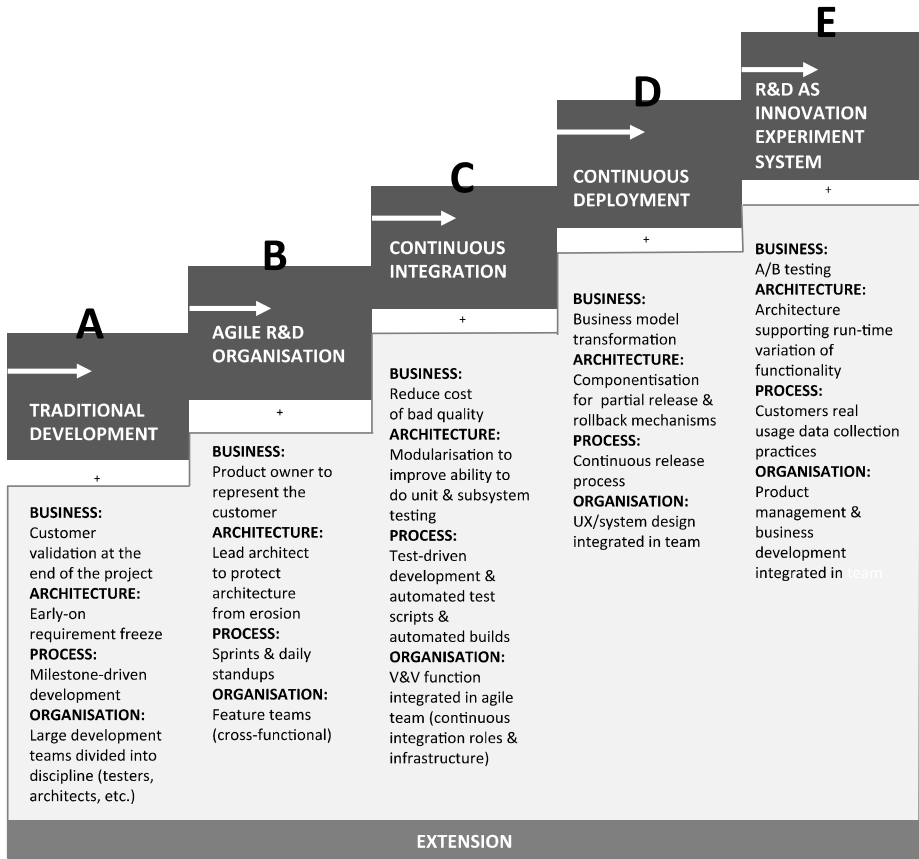


Fig. 1. The Stairway to Heaven model [4] and (+) the extension

4 Research Design

4.1 Research Method

This study is an in-depth multiple-case study that adopts an interpretive approach [23]. It includes empirical data from five case companies in the ICT sector in Finland, allowing for a cross-case analysis of the data. The case study method is a suitable research approach for an overall study in which researchers act as investigators and control over the context is not possible [24]. The case study approach is also beneficial for creating a rich understanding of people’s experiences.

4.2 Data Collection and Analysis

This paper reports on a three-month (November 2014–January 2015) multiple-case study involving five Finnish software development companies that are moving

towards IES. The main data collection method used was semi-structured individual interviews with open-ended questions [23]. Altogether, the study included 24 interviews. In all of the interviews, we reused the original StH study [4] set of interview questions. The interviews had four main themes: (1) organisation and current way of working, (2) customer interaction mechanisms/models, (3) strengths and weaknesses in ways of working and (4) benefits and barriers as experienced when moving towards IES. In companies A, B, C and D, we conducted five interviews in each company and, in company E, we conducted four interviews. In companies A, B, C and D, the interviews were conducted face-to-face and in company E the interviews were conducted via videoconference. The data collection involved case company stakeholders from Product Management, R&D, Validation & Verification and Sales & Marketing (Table 1). All of the interviews were held in English. The duration of the interviews ranged from 90 to 120 minutes; the interviews were recorded and transcribed. During the interview, the three researchers shared the responsibilities; one researcher mainly asking the questions and two researchers took notes. Section 4 describes the case companies that participated in the study in more detail as well as their position in relation to the StH model.

Table 1. Case companies and interviewees' roles

Case	Industry	Interviewees' roles
A	Embedded systems and R&D services	1) Special device senior manager, 2) Special device product owner, 3) Sales and account manager, 4) Senior specialist in software, 5) Quality manager in wireless segment
B	Telecommunications	1) Test automation manager, 2) Senior developer, 3) Program manager, 4) Operations manager of the local site, 5) Technical coordinator
C	Telecommunications	1) System verification engineer, 2) Program manager, 3) Software architect, 4) Product line manager, 5) Software engineer
D	Factory automation	1) Project manager, 2) Program manager, 3) User experience (UX) designer, 4) Product manager, 5) Developer
E	IT services	1) Product owner, 2) Project manager, 3) Technical service owner, 4) Technical lead

The data analysis was performed by three researchers in continuous collaboration, following the general techniques for case study analysis suggested by Runeson et al. [23] using the QSR NVivo tool.¹ During the analysis, all of the materials, including transcripts, field notes, audio files and other related material, were stored in NVivo. All of the transcribed interviews were carefully read and coded by themes. For this study, three main levels of codes were applied for each of the 24 interviews: 1) 'barriers' (barriers that prevent companies from moving towards IES), 2) 'benefits' (anticipated or identified benefits of moving towards IES) and 3) 'practices' (practices

¹ Qualitative data analysing software (<http://www.qsrinternational.com/>).

for moving towards IES). The results were synthesized in two phases by adopting within-case analysis and cross-case analysis, as described by Yin [24].

4.3 Validity and Generalisation of the Results

Generally, in case study research, there are different threats to validity, such as researchers' biases, that can limit the trustworthiness of the results. In our study, we assess three aspects of validity, i.e. construct validity, external validity and reliability, as identified by Runeson et al. [23]. Prior to data collection, the research design that also included the data collection process was carefully considered. The activity involved acquiring the original interview guide, selecting appropriate companies and roles for the interviews and providing all interviewees with introductory materials (e.g. study objectives, the structure of the interview, etc.). This was done to address construct validity, which is concerned with a clear representation of the studied constructs. However, company transition may take even several years. Therefore, interviewees' memories about transition may not be as accurate as they are for more recent events in the company. The companies were selected from a group of leading-edge companies that were participating in a large national research program that aimed to enhance Finnish ICT companies' capability to deliver value in real-time. Convenience sampling was applied. Threats to the reliability of the study findings were mitigated by having at least three researchers involved in all phases of the research, particularly in the data collection and analysis phases. This practice helped reduce the research bias that could arise from having only one researcher participate in data collection and analysis. Additionally, to lower the risk of errors in the interviews, the transcripts that were used for data analysis were sent to the interviewees for review. External validity is mostly concerned with the generalisability of a study's findings. The findings of our study are meant to provide software companies with insights with the intention of helping them move beyond agile practices.

5 Case Study

This section presents the case companies and main findings, individually, for each case company by applying the extended StH model introduced in Section 3. We report benefits (RQ1), barriers (RQ1) and key practices (RQ2) for each company and analyse them in the StH model. Table 2 summarises the findings for each company by applying the StH model and extension. Section 6 presents the cross-case analysis based on the data collected from all five case companies. Three researchers collectively compiled the information presented in Table 2 and analysis was based on the researchers' common interpretation of how practices were adopted in each case company.

Table 2. Adoption of the extended StH model practices. Not adopted (NA), Team (TE), Product (PR), Institutionalised (IN).

		Traditional	Agile	CI	CD	IES
Company A	Business	NA	IN	IN	NA	PR
	Architecture	NA	IN	IN	PR	PR
	Process	NA	IN	IN	TE	NA
	Organisation	NA	IN	IN	PR	NA
Company B	Business	NA	IN	TE	NA	NA
	Architecture	NA	IN	TE	PR	TE
	Process	NA	IN	TE	NA	NA
	Organisation	NA	IN	NA	NA	NA
Company C	Business	NA	IN	PR	NA	NA
	Architecture	NA	IN	IN	PR	PR
	Process	NA	IN	PR	NA	NA
	Organisation	NA	PR	PR	NA	NA
Company D	Business	NA	PR	PR	NA	NA
	Architecture	IN	PR	PR	NA	NA
	Process	IN	PR	PR	NA	NA
	Organisation	NA	PR	NA	NA	NA
Company E	Business	IN	TE	NA	NA	NA
	Architecture	IN	TE	NA	NA	NA
	Process	IN	TE	TE	NA	NA
	Organisation	IN	TE	TE	NA	NA

5.1 Company A

Company A is developing embedded software solutions for specialised markets in the wireless and automotive industry as well as providing R&D services. The focus is on customisable software solutions for the automotive industry and the wireless connectivity of special devices used by specialised market segments such as public safety. As shown in Table 2, Company A is best described as a company with institutionalised practices for CI. Architecture, process and organisation practices were already established in CD in some teams and product programs. We could see that this company as a whole is moving towards CD. Experiences and lessons learned from these practices were proactively used to coach other parts of the organisation in this transition to CD. Interestingly, although this company had not yet institutionalised CD practices, some product programs had already applied IES practices in some selected customer cases. According to the interviewees, the main benefit of moving towards CD is to improve customer feedback cycles and project transparency. The main barriers are considered to be the lack of a suitable business model, test automation and common practices for CD. As identified by the

interviewees, in order for the organisation to move forward, management must invest more in CI build systems and test automation.

5.2 Company B

Company B is a telecommunications equipment manufacturer that also provides services for managing network operations. In this company, we interviewed employees from the R&D organisation who are responsible for developing a compact mobile broadband solution. Company B has institutionalised agile practices, and several teams within the company have already adopted CI practices. According to the interviewees, the main benefit of moving towards CD is that product quality will improve due to frequent and automated test suits. The main barrier is considered to be the issue of how to adjust and align internal and external stakeholders to shorter development cycles. As identified by the interviewees, in order to move forward the level of test automation must be increased, which will require additional resources and investments.

5.3 Company C

Company C is a manufacturer of data and telecommunication network equipment that also develops a variety of supporting tools for the management of mobile broadband networks. In this company, interviews were conducted with employees involved in the development of a network traffic-monitoring tool. As illustrated in Table 2, Company C can be best described as a company that has well-established agile practices and that has adopted CI and CD practices in parts of the organisation. According to the interviewees, the main benefit of moving towards IES is to improve competitiveness and product quality, as customer feedback would increasingly impact product development. The main barrier for CD is that some company functions still work according to pre-defined milestones, and those functions still support a six-month release cycle. As identified by the interviewees, in order to move forward the current product architecture must be updated from a PC platform to a virtualised cloud computing platform.

5.4 Company D

Company D is developing minerals processing solutions and flow control technology for its customers in the mining, construction, oil and gas industries. In addition, the company develops advanced automation solutions, i.e. distributed control systems for its customers in pulp, paper and power. For the purpose of this study, we interviewed employees involved in the development of a factory automation platform solution. Table 2 illustrates the current situation of how Company D has recently adopted both agile and CI practices in one product program. However, the rest of the company is still primarily using traditional practices. According to the interviewees, it is not possible to move towards CD and IES because their systems are performance- and safety critical. However, while this is the general view, some parts of the systems, e.g.

the user experience (UX) parts, could be improved by applying CD and IES practices. Thus, in order for Company D to move forward, it must identify which modules or parts of the system could be deployed and experimented with in a continuous manner.

5.5 Company E

Company E provides product engineering and IT services to a variety of customers from the telecommunications and consumer electronics industries and the semiconductor industry. In Company E, we interviewed the employees responsible for developing the company's public website. The team is using Scrum as a project management framework with no CI system or automated test cases. Table 2 illustrates how Company E is still mostly applying traditional development practices. However, some teams have adopted agile and CI practices with the intention of having these practices adopted by more teams. According to the interviewees, moving forward would improve product quality and reduce time-to-market. The main barrier for moving forward is the difficulty of aligning the globally distributed development teams. As identified by the interviewees, in order to move forward employees must increase their awareness of and competence in agile software development and the number of cross-functional teams must increase.

6 Discussion

In the previous section, and by using the five steps in the StH model, we identified the current position and practices of each case company. In addition, the extended model was used to identify the practices that companies may apply to advance their practices further. This section summarises and further discusses the benefits, barriers and key practices as identified from the case study interviews and addresses how the interviewees in this study have experienced the evolution. As shown in Table 3, we found multiple similar benefits and barriers that were also identified in earlier studies [4], [13, 14]. Additionally, the table presents the key practices that are needed to move to the next step in the StH model.

In our study, all five companies (A-E) have established agile software development practices within the R&D organisation. Companies A, B and C had already been using agile practices for a couple of years, whereas D and E had only been using them for approximately one year. All of the companies held the opinion that the primary benefit of an agile R&D organisation was that incremental and iterative development allows for more efficient and flexible product development.

Companies A, B, C and D have advanced their software development practices by introducing CI practices, i.e. building and testing software functionality automatically. None of the companies claimed to have achieved fully automated testing. The main benefit of CI is that production quality software is always available internally. The main barrier to moving forward to CI and CD seems to be the high investment and significant effort required to ensure adequate test automation.

Table 3. Summary of benefits, barriers and key practices for moving towards IES

	Traditional -> Agile R&D	Agile R&D -> CI	CI -> CD	CD -> R&D as an IES
Benefits	Short sprints provide the possibility of quickly changing the course of product development.	Provides the ability to build and test products incrementally. Provides high-quality software functionality with production quality.	Customers get fast and incremental delivery of relevant functionality. Customers can perform their own testing and business activities on top of deliveries.	The innovation validation is fast. Immediate feedback is obtained. New business opportunities are identified and development resources are focused
Barriers	It is difficult (complex process) to align different cross-functional teams within the R&D organisations.	There is a lack of team discipline, Test Driven Development (TDD) and module tests for CI test automation.	The shortening of the Validation and Verification (V&V) cycle is complex and expensive. The lack of trust in software quality and missing functions may cause a negative impression.	Customer feedback is integrated into the short development and business planning cycle. It is difficult to conduct experiments in safety-critical systems.
Key practices				
Business	Incorporate product owner to represent customer in development team.	Incorporate supply chain (component and technology suppliers) in the development cycle.	Incorporate lead customers in development. Renew business model, contracts, marketing and sales strategies	Adopt data-driven strategic decision-making model. Implement A/B testing with the customer.
Architecture	Architects monitor and safeguard the integrity of the product architecture in feature-driven development.	Provide modular architecture that can be integrated and tested continuously.	Provide architecture where software functionality can be deployed independently.	Adopt product platform (e.g. virtualisation, cloud technologies) that enables flexible experimentation.
Process	Develop features in sprints, frequent (daily) team meetings.	Adopt test-driven development and daily build practices.	Improve automated system testing and adopt a continuous release process.	Establish a short customer feedback loop and process for data-driven decision making.
Organisation	Adopt and empower cross-functional/feature teams.	Integrate validation and verification (V&V) in cross-functional/feature teams.	Ensure that System/UX design and business development work in short cycles and in alignment with R&D.	Synchronise supplier and customer organisation in short development cycles.

In our study, only Company A has evolved its software development practices to CD at the product and team levels. Therefore, our findings rely on the interviewees' views of what that transition might require and how it effects the organisation. The main anticipated benefit of CD is that customers receive relevant software functionality faster and incrementally. However, moving to CD seems to require renewing traditional business models as well as identifying customers that are willing to have continuous releases of software functionality. It also seems that the transition from CD to IES might not be feasible for all products and business segments. Interviewees in Company B and Company D consider their current products to be too safety-critical for introducing any experiments with their customers. This finding confirms what earlier research has suggested, i.e. that IES may not be feasible for all products and business segments, and that if pursued, the evolution towards IES requires changes in both the product and business portfolio.

7 Conclusion, Limitations and Future Research

In this paper, we identify the benefits and barriers that software development companies face when moving beyond agile development practices. In particular, and based on a multiple-case study in five Finnish companies, we identify the practices that companies adopt when moving towards IES. Our findings show that all of the interviewed companies have established agile software development practices within their R&D organisations. The primary benefit of doing so is that incremental and iterative development allow for more flexible product development projects in which customer feedback informs the organisation's development efforts and investments. When moving from agile practices to implementing CI at the team level or product level, companies A, B, C and D have improved their R&D capability to continuously integrate and validate software changes at a team and/or component level, and in some cases even at a system level. In this way, the R&D organisation gets faster feedback about functionality and they can avoid the many challenges related to integration of functionality. However, and as recognized by the interviewees, CI is not free. The main barrier is considered to be the high investment and significant effort that are required to ensure adequate test automation and the development of automated test cases. In our study, only Company A has evolved its software development practices to CD, and then only in some of its products and teams. Company A has made initial attempts to transition further to IES by experimenting with customers to validate new software functionality.

The contribution of the paper is threefold. First, we apply the StH model in five Finnish case study companies. This strengthens the validity of the model and describes the evolution path that software development companies may take when advancing their development practices. Second, we extend the model with a set of practices that companies adopt at each step of the StH in order to advance further. In our case studies, we see that companies tend to institutionalise practices in the lower steps of the StH, adopt practices at the product level at the step at which they currently operate, and explore next step practices in individual development teams.

Finally, we validate the extended model and its practices in five Finnish software development companies using a multiple-case study and interpretive approach.

The main limitation of our study is related to the generalisability of the results. Although case study findings may provide important information regarding typical patterns and a set of practices, they cannot be generalised to the entire software development domain.

For future research, we suggest additional case studies that focus on stakeholders, such as customers, suppliers, subcontractors, platform providers, development partners, etc. This might broaden the understanding of the company's evolution path towards IES and expand the StH model view from an internal company perspective to an external ecosystem perspective.

Acknowledgements. This work was supported by TEKES as part of the Need for Speed project (<http://www.n4s.fi/>) of DIGILE (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT and digital business). This work was done in co-operation with the Software Center (www.software-center.se), a Nordic collaboration initiative between eight companies and three universities.

References

1. Highsmith, J.: *Agile Project Management: Creating Innovative Products*. Addison-Wesley Professional, New York (2009)
2. Bosch, J., Eklund, U.: *Eternal Embedded Software: Towards Innovation Experiment Systems*. In: Margaria, T., Steffen, B. (eds.) *ISoLA 2012, Part I*. LNCS, vol. 7609, pp. 19–31. Springer, Heidelberg (2012)
3. Fagerholm, F., Guinea, A.S., Mäenpää, H., Münch, J.: *Building Blocks for Continuous Experimentation*. In: *1st International Workshop on Rapid Continuous Software Engineering*, pp. 26–35. ACM Press, New York (2014)
4. Olsson, H.H., Bosch, J., Alahyari, H.: *Towards R&D as Innovation Experiment Systems: A Framework for Moving Beyond Agile Software Development*. In: *IASTED Multiconferences - Proceedings of the IASTED International Conference on Software Engineering*, pp. 798–805 (2013)
5. Rodríguez, P., Markkula, J., Oivo, M., Turula, K.: *Survey on Agile and Lean Usage in Finnish Software Industry*. In: *ACM-IEEE International symposium on Empirical Software Engineering and Measurement*, pp. 139–148. ACM Press, New York (2012)
6. Dybå, T., Dingsøy, T.: *Empirical Studies of Agile Software Development: A Systematic Review*. *Information and Software Technology* 50, 833–859 (2008)
7. Goodman, D., Elbaz, M.: “It’s Not the Pants, it’s the People in the Pants” Learnings from the Gap Agile Transformation What Worked, How We Did it, and What Still Puzzles Us. In: *Agile Conference*, pp. 112–115. IEEE Press, New York (2008)
8. Marschall, M.: *Transforming a Six Month Release Cycle to Continuous Flow*. In: *Agile Conference*, pp. 395–400. IEEE Press, New York (2007)
9. Neely, S., Stolt, S.: *Continuous Delivery? Easy! Just Change Everything (Well, Maybe It Is Not That Easy)*. In: *Agile Conference*, pp. 121–128. IEEE Press, New York (2013)

10. Trimble, J., Webster, C.: From Traditional, to Lean, to Agile Development: Finding the Optimal Software Engineering Cycle. In: 46th Hawaii International Conference on System Sciences, pp. 4826–4833. IEEE Press, New York (2013)
11. Ståhl, D., Bosch, J.: Modeling Continuous Integration Practice Differences in Industry Software Development. *Journal of Systems and Software* 87, 48–59 (2014)
12. Humble, J., Farley, D.: *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Professional, Boston (2010)
13. Claps, G.G., Berntsson Svensson, R., Aurum, A.: On the Journey to Continuous Deployment: Technical and Social Challenges Along the Way. *Information and Software Technology* 57, 21–31 (2015)
14. Leppänen, M., Mäkinen, S., Pagels, M., Eloranta, V.-P., Itkönen, J., Mäntylä, M.V., Mannisto, T.: The Highways and Country Roads to Continuous Deployment. *IEEE Software* 32, 64–72 (2015)
15. Fitzgerald, B., Stol, K.-J.: Continuous Software Engineering and Beyond: Trends and Challenges. In: 1st International Workshop on Rapid Continuous Software Engineering, RCoSE 2014, pp. 1–9. ACM Press, New York (2014)
16. Karvonen, T., Rodriguez, P., Kuvaja, P., Mikkonen, K., Oivo, M.: Adapting the Lean Enterprise Self-Assessment Tool for the Software Development Domain. In: 38th Euromicro Conference on Software Engineering and Advanced Applications, pp. 266–273. IEEE Press, New York (2012)
17. Poppendieck, M., Poppendieck, T.: *Implementing Lean Software Development*. Addison-Wesley, New York (2006)
18. Bosch, J., Bosch-Sijtsema, P.: From Integration to Composition: On the Impact of Software Product Lines, Global Development and Ecosystems. *Journal of Systems and Software* 83, 67–76 (2010)
19. Olsson, H.H., Bosch, J.: From Opinions to Data-Driven Software R&D: A Multi-case Study on How to Close the “Open Loop” Problem. In: 40th EUROMICRO Conference on Software Engineering and Advanced Applications, pp. 9–16. IEEE Press, New York (2014)
20. Pressman, R., Maxim, B.: *Software Engineering: A Practitioner’s Approach*. McGraw Hill Education, New York (2014)
21. Ries, E.: *The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Publishing, New York (2011)
22. van der Linden, F.J., Dannenberg, R.B., Kamsties, E., Käsälä, K., Obbink, H.: Software Product Family Evaluation. In: Nord, R.L. (ed.) *SPLC 2004*. LNCS, vol. 3154, pp. 110–129. Springer, Heidelberg (2004)
23. Runeson, P., Höst, M., Rainer, A., Regnell, B.: *Case Study Research in Software Engineering: Guidelines and Examples*. John Wiley & Sons, New Jersey (2012)
24. Yin, R.K.: *Case Study Research: Design and Methods*. SAGE Publications Inc., California (2009)