Building Lean Thinking in a Telecom Software Development Organization: Strengths and Challenges

Pilar Rodríguez Kirsi Mikkonen Pasi Kuvaja Markku Oivo Juan Garbajosa

ABSTRACT

The potential shown by Lean in different domains has aroused interest in the software industry. However, it remains unclear how Lean can be effectively applied in a domain such as software development that is fundamentally different from manufacturing. This paper explores how Lean principles are implemented in software development companies and the challenges that arise when applying Lean Software Development. For that, a case study was conducted at Ericsson R&D Finland, which successfully adopted Scrum in 2009 and subsequently started a comprehensible transition to Lean in 2010. Focus groups were conducted with company representatives to help devise a questionnaire supporting the creation of a Lean mindset in the company (*Team Amplifier*). Afterwards, the questionnaire was used in 16 teams based in Finland, Hungary and China to evaluate the status of their transformation. By using Lean thinking, Ericsson R&D Finland has made important improvements to the quality of its products, customer satisfaction and transparency within the organization. The study makes two main contributions to research. First, the main factors that have enabled Ericsson R&D's achievements are analysed. Elements such as 'network of product owners', 'continuous integration', 'work in progress limits' and 'communities of practice' have been identified as being of fundamental importance. Second, three categories of challenges in using Lean Software Development were identified: 'achieving flow', 'transparency' and 'creating a learning culture'.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management – *life cycle, productivity, programming teams, software process models.*

D.2.10 [**Software Engineering**]: Design – *Methodologies.*

General Terms

Management, Design, Economics, Human Factors.

Keywords

Agile software development, lean software development, le-agile, method adoption, process improvement, process introduction.

1. INTRODUCTION

The interest of the software intensive industry in applying Lean thinking has grown significantly in recent years [25]. The potential that Lean has exhibited in terms of profitability, time-to-market and product quality [3] has generated a demand for greater knowledge in *Lean thinking* in industries other than manufacturing. For example, in the domain of clothing industry, Zara has reported reduced lead time via a business model based on collecting input from customers daily and using Lean inventories. Shorter lead time has enabled Zara to deliver new items to stores twice a week (as much as 12 times faster than its competitors) and to bring in almost 30000 designs each year, as opposed to the 2000–4000 new items introduced by its competitors [30]. In healthcare, the application of Lean has reported significant improvements in reducing patient waiting lists, floor space utilization and lead-time in laboratorial tests [2]. However, the intangible nature of software development, its dynamism and its dependency of knowledgeable workers, whose work primarily involves the use of information, challenges the applicability of Lean ideas [10]. In consequence, Lean is open to interpretation in a domain such as software development that differs fundamentally from manufacturing.

Lean Software Development, initially regarded as one of the Agile methods [8], is now acquiring an identity of its own. Conferences and special issues in well-known journals include topics on Lean Software Development (e.g. [16], [5], [9] and [7]). Accordingly, there is a growing body of literature not only documenting case studies [29, 18, 19] but also investigating specific elements of Lean, like flow [17, 22]. However, as recently noted by Ebert et al. [9], '*We're still in the early phases of truly understanding how "lean methods" impact software development*', and it remains unclear how software companies that are moving towards Lean interpret and implement Lean Software Development in practice.

This paper attempts to contribute to better understanding of Lean in the context of software development by analysing the transformation to Lean of Ericsson R&D Finland. Ericsson is one company in a consortium of companies currently implementing Lean as part of a large research programme in Finland[1]. Ericsson R&D has exhibited commitment and achieved progress in its transformation towards Lean. Primary Ericsson R&D Finland launched a transformation to implement Scrum in 2009 and started its conversion to Lean in 2010. During the last three years, Ericsson R&D transformation leaders, team members and managers have been investigating ways to extract maximum benefit from Lean ideas in the context of software development. Lean principles have been widely discussed and experimented with inside the company. Thus, Ericsson R&D Finland, through a process of learning by experimenting, has come up with its own interpretation of Lean Software Development. Although transition to Lean requires time and resources, the effort has been worthwhile. Using a process that combines features of both Lean and Agile, improvements in the quality of the products, customer satisfaction and transparency within the organization have been achieved. Moreover, build times have been reduced over ten times and the number of commits per day has increased roughly five times. This paper analyses how Lean and its principles have been implemented in Ericsson R&D in Finland and identifies challenges, as well as means of making the transformation to Lean in software development more achievable.

The rest of the paper is structured as follows. Section 2 provides an overview of Lean concepts and presents related work. Section 3 describes the research setting, including objectives of the study, a description of Ericsson R&D Finland and steps carried out in the research. Results of the study are presented in Section 4. Section 5 concludes the paper and presents limitations and future research.

2. LEAN AND SOFTWARE DEVELOPMENT

Although Lean has been widely discussed and used for more than two decades [21, 32, 28], there is no common definition of Lean [28]. This deficit in specification is even greater in Lean Software Development because of the freshness of the topic [6]. This section provides a brief overview of Lean thinking as originally described by MIT researchers, as well as related work in Lean Software Development.

2.1 Lean Thinking

Lean is the product of an incremental development of techniques that have gradually evolved since the late 1940s from the Toyota Production System (TPS). Although difficult to specify, five core principles, first introduced by Womack and Jones in 1996 [33], are considered to govern Lean thinking:

1. *Value* is defined as everything for which a customer is willing to pay. Value, understood from a customer viewpoint, is the core concept of any Lean company. Its counterpart, *waste*, is everything that consumes resources but produces no value to the customer. Lean seeks continuous identification and removal of waste.
2. *Value stream* is the optimized, end-to-end collection of actions required to bring a product from customer order to customer care, ensuring that each activity provides value.
3. *Flow* means that activities are organized as a continuous 'flow', eliminating discontinuities in the value stream and enabling smooth deliveries.
4. *Pull* implies that everything is built only when it is needed, making customer order/market the main decision driver. Thus, unnecessary intermediate and unfinished product inventories must be eliminated.
5. *Perfection,* or *Kaizen* in Japanese, pursues continuous enterprise-level improvement based on the concept that there is no end to striving for perfection.

Different practices and tools such as Kanban, Poke-Yoke or 5 Whys have been designed for implementing these principles, and have, to a greater or lesser extent, crossed the boundaries of a manufacturing context.

2.2 Lean in Software Development

Although universal applicability of Lean principles is still the subject of debate, especially in knowledge work such as software development, most studies suggest that Lean principles could be applied to virtually any system [29, 33]. Recently, Poppendieck and Cusumano claimed that, '*if Lean is thought of as a set of principles rather than practices, then applying lean concepts to product development and software engineering makes more sense and can lead to process and quality improvements*' [23]. In software development, authors have made different interpretations of Lean principles, although commonly they have interpreted Lean in the context of Agile Software Development (ASD). Table 1 lists well-known interpretations. Although customized, they are not fundamentally different and all exhibit conformity with the core five principles of Lean.

Table 1. Interpretations of relevant Lean principles in software development.

Lean Software Development Principles (Poppendieck and Poppendieck 2003 [24])	**Lean Software Strategies** (Middleton 2005 [19])	**The Lean Thinking House** (Larman and Vodde 2009 [15])	**The Kanban Principles** (Anderson, 2010 [1])
- Eliminate waste - Build quality in - Create knowledge - Defer commitment - Deliver fast - Respect people - Optimize the whole	Interpretation based on Womack and Jones's five Lean principles: value, value stream, flow, pull and perfection	Two pillars: respect for people and continuous improvement. 14 principles: long-term, flow, pull, less variability & overburden, stop & fix, master norms, simple visual mgmt., good tech, leader-teachers from within, develop exceptional people, help partners be lean, Go-See, consensus, reflection & kaizen.	-Visualise the workflow -Limit WIP -Manage flow -Make process policies explicit -Improve collaborativel y (using models and the scientific method

In recent years, progress toward Lean Software Development has been mainly driven by industry pioneers familiar to some extent with ASD. Consequently, Lean Software Development itself has not been extensively researched and there is a lack of a developed understanding on which of its elements are positively applied in practice, as well as ways to combine principles of Lean with ASD [7]. Some of the Agile Manifesto principles resemble Lean thinking. For example, '*simplicity – the art of maximizing the amount of work that need not be done – is essential*'; '*at regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly*'; further, '*our highest priority is to satisfy the customer through early and continuous delivery of valuable software*'. However, '*welcoming changing requirements, even late in development'* seems to be antagonistic to Lean thinking. Some early publications analyse Lean and Agile experiences. For example, Mehta et al. [18] present a case study in which relevant Lean principles are applied in an IT department. The results show that Lean principles are aligned with many best practices of software engineering. Wang et al., analysing the purposes of applying Lean in ASD, identified six different strategies for application and introduced the term *Le-agility* in software development [31].

3. CASE STUDY DESIGN

In this study, a case study method was used because of its ability to provide a deeper understanding of phenomena that are little known, such as Lean Software Development [26].

3.1 Objective and Research Questions

The objective of the study is to understand *how Lean thinking is implemented in software development companies,* which differ substantially from those in which Lean was originally developed. Two research questions drive the study:

RQ 1 How are Lean principles interpreted and implemented in a software development domain?

RQ 2 a) What elements of Lean thinking are challenging the implementation of Lean in software development? b) What are the elements of Lean thinking that are more easily achievable in a software development context?

3.2 The Case Company and Its Context

The study was conducted mainly in Ericsson R&D Finland, which is a pioneer within Ericsson in the adoption of Agile and Lean on a large scale. Besides cooperating sites in Hungary and China, which are also transforming towards the same Lean way of working, participated in the second phase of the study. Ericsson is a world-leading provider of telecommunications equipment and related services to mobile and fixed network operators. Forty percent of the world's mobile traffic is handled by Ericsson systems. The sites studied are focused on developing a complex mobile network product. The product is mature (ten years old) and based on a platform that is utilized also by other products. Therefore, careful control of the interface with the platform is needed. As programming languages, RoseRT/RSA-RTE, C++ and Java are used. Engineers are highly experienced with profound knowledge of the product. The development is carried out in a large distributed multisite context.

From 2010, Ericsson R&D has been transforming its processes from following basic Agile principles to complementing them with Lean principles. The transformation in Finland involves around 400 people. As Figure 1 illustrates, Lean thinking underpins ASD. Moreover, Lean is a means of achieving a learning organization and ensuring that improvement is a continuous and never-ending process.

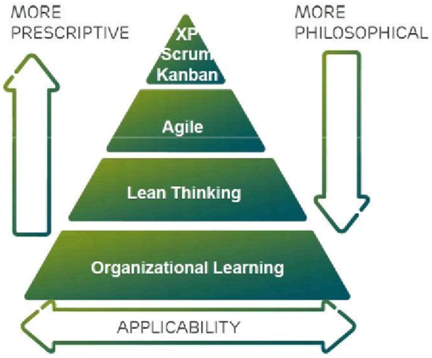

Figure 1. Lean and Agile in context at Ericsson R&D.

Unlike other organizations transforming towards Lean [33], Ericsson R&D was not facing a crisis that triggered its adoption. Rather, the transformation was driven by the company's desire to remain a strong player in the highly competitive telecom industry. It was recognised that people were too satisfied with the status quo and no one challenged current practices. Thus, although Ericsson R&D was doing well, the company realized that it could do better. With the goal of reaching a customer-focused, cross-functional and value-thinking organization, four specific motivators drove the transformation towards Lean: to create most value, to improve responsiveness, to build in quality and to empower people [11].

During the transformation, changes have impacted the whole development chain, from the earliest product release phases to maintenance [20]. Moreover, Agile and Lean have inculcated a profound change of culture and thinking into the company that goes way beyond just processes and tools [11]. Big projects have given room to flexible releases. Agile feature-oriented development, with an end-to-end user story focus, is used to provide flexibility in managing the product and a better ability to serve multiple stakeholders. Development is structured so that it supports continuous flow. Continuous Integration is a key element in achieving continuous deployment and smaller end-to-end deliveries. Testing is done in parallel with development. As result, the status of the software quality at all levels is visible and feedback times between traditional test phases have been reduced from months to weeks.

Cross-functional teams have replaced the traditional functional silo-based organization (system/development/test). Product owners help to ensure alignment with the larger common goals. End-to-end product responsibility is stressed at all levels. Although deep expertise is still highly needed, broader competence is valued more than specific narrow competencies. Thus, the emphasis is on creating a collective knowledge culture. Training sessions and retrospectives in both *hard* and *soft* aspects[2] of software development are frequently arranged to support organizational learning. Coaches are available on all levels. These initiatives help to provide first-hand evidence and experience that enable rapid learning by means of fast feedback cycles. All teams are approximately equal in competence, permitting any team to pull the top priority item from the backlog.

As a large R&D centre developing a complex product, Ericsson R&D Finland felt the need to have a common but still flexible development approach for all teams. The approach had to support a great deal of freedom of action level as well as clear alignment of intent, direction and vision. Instead of following a defined and detailed process, a flexible Scrum framework was selected. According to Ericsson R&D's experience, people are the core of the transformation. Consequently, personal initiative and self-organization are encouraged, as opposed to the earlier top-down control. Thus, guided by the Scrum framework, team members actively participate in the transformation, assisting in setting up teams and deciding the practices that work best for them. Besides Scrum, Kanban is used as a method to implement the Lean mindset, particularly in product maintenance. '*Pull mindset, team working, team empowerment, and continuous improvement have become part of the everyday activities. Best practices from Scrum have been selected to complement the Kanban implementation*' [27]. To conclude, seating and facilities have been also comprehensively modified to support the Lean way of working [12]. Individual offices have gradually given way to team spaces, raising the concept of the *R&D team area*, as shown in Figure 2. Team members work at a common table, sharing almost

[2] *Hard aspects* of software development involve languages, tools, techniques, architectures, modelling, processes, etc. *Soft aspects* are aspects such as interaction between people, engagement, team dynamics, creativity and self-organization.

everything. Teamwork, communication and collaboration are supported also by the daily use of whiteboards, flip charts, information radiators, and video connections, chat rooms and wikis for distributed teams. Despite the substantial resistance encountered when the work space change was instituted – understandable in view of the long history of individual offices – it has showed benefits on promoting teamwork.

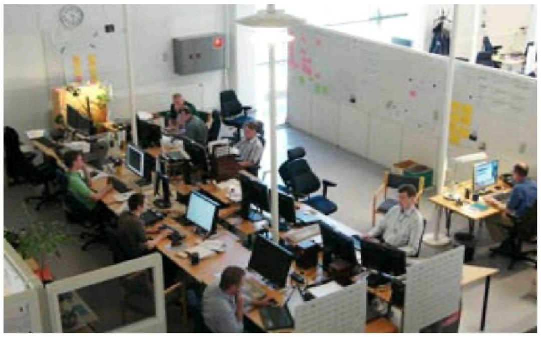

Figure 2. View of a team area, illustrating the R&D team area concept for Scrum teams (from [12]).

3.3 Research Method Steps

The research consisted of two phases, in which researchers acted as investigators rather than participants.

3.3.1 Phase 1: Focus Group Sessions

The first phase of the study used 17 focus group sessions [14] to explore Ericsson R&D's interpretation of Lean Software Development. The goal of the sessions was to define a set of statements that represent the essential aspects of Ericsson R&D's Lean way of working. The resulting statements composed a tool called *Team Amplifier,* which was afterwards used for evaluating the transformation inside the company and supporting the creation of a Lean mindset. Two main reasons motivated us to use focus groups. First, focus group is a flexible exploratory research method that enables discussions between experts who query and explain each other. Outcomes of those interactions offer valuable insights that are difficult to get through other methods such as individual interviews [14]. Moreover, focus groups allow obtaining rich and in-depth information covering a relatively wide range of participants in a short period of time [14]. Second, group discussions are commonly used at Ericsson R&D. Participants were used to conduct focused discussions where open and safely state their opinions and actively listen each other. Thus, potential weakness of focus groups such as unfocused discussion or dominant personalities limiting discussion were not noteworthy.

The research started with a half day *material walkthrough workshop* organized to introduce the researchers to the specific Lean/Agile processes at Ericsson R&D Finland, as well as specific practices and terminology. Existing process documents, usually in the form of PowerPoint presentations, were provided and used as an input in the study. Thereafter, focus group sessions were conducted with experts leading the transformation, as well as practitioners directly impacted by Lean Software Development in different roles, such as developers, testers, Agile coaches, Scrum masters and product owners. This research design enabled diversity in participant perspective at all hierarchical levels and facilitated triangulation. Discussions aimed to answer the following question: *Which are the essential aspects defining our Lean way of working that should be included in the Team Amplifier?* Each group session lasted between 2-3 hours and had a number of participants between 5 and 15. A total of 21 company representatives and three researchers participated in the sessions (see Table 2). After plenty of discussion, topics emerged spontaneously, which were recorded in the form of statements, using Excel spreadsheets. Examples of statements are: '*There are no "functional silos" in our company, for example between Marketing and Engineering functions*', '*Lessons learned and practices from inside and outside Ericsson are actively shared, evaluated and further improved within the organization*' and '*Managers and leaders know the Value Stream Map for the product they are working with*'. Usually new topics related to the statement under discussion were surfacing. Moreover, statements created but considered later to be of little interest were accordingly dropped. The Excel spreadsheet was shared across all focus group participants so that individuals had the chance to reflect on the statements before the next session. Researchers participated in a small number of the sessions in order to decrease their possible impact on the dynamics of the group, while still enabling them to follow the process. Thus, the Excel spreadsheet was also shared with the researchers after every focus group session. Additionally, the MIT solution for assessing the leanness of an enterprise and its capacity to change according to Lean principles, LESAT, was reviewed in a couple of sessions [13].

Table 2. Focus group design.

Focus Group Variation[3]	# Sessions	# Participants	Estimated effort[4]
Traditional face-to-face	11 (~22 hours)	*Company internal* Between 6 and 15 participants from 5 different sites including different profiles, from transformation leaders to team members	165
	2 (~6 hours)	*Including researchers* 7 company representatives 3 researchers	60
Synchronous on-line	4 (~8 hours)	*Including researchers* 2 process managers 3 researchers	40

As result of this process, a set of statements (114) was collected. The statements were afterwards analysed from the perspective of the five core principles of Lean. The 114 statements were coding under the codes "*value and waste*", "*value stream*", "*flow*", "*pull*" and "*perfection*". In cases where a statement fit under multiple codes, we selected the first relevant one and then mentioned other principles that also apply. For example, we coded the statement "*Everyone is encouraged to identify waste and challenge the current practice*" under the codes "*value/waste*" and "*perfection*". During the coding process the code "*people*" emerged since many statements referenced to aspects related to people mind-set, empowerment and organizational culture. Also sub-codes under main codes emerged as important concepts for implementing Lean principles were identified (e.g. "*value/defer commitment*", "*people/teamwork*"). Company power point presentations as well as notes took by the researchers during the sessions, were used for interpreting the findings. Finally, a company representative (the

[3] Traditional face-to-face focus groups occur when all participants are in the same place at the same time. Synchronous on-line focus groups may be in different places but at the same time and require computer mediation to establish the session [14].

[4] Overall estimation of the effort (Persons*Hours).

second author) reviewed primary findings for validation. Results are presented in Section 4.1.

3.3.2 Phase 2: Team Sessions

In the second phase of the study, the *Team Amplifier* was used to evaluate 16 teams based in Finland, Hungary and China in order to identify strengths and challenges in their respective Lean transformations, as well as to coach Lean concepts further. Although all teams were following the overall transformation as described in Section 3.2, they were at different levels of Lean adoption maturity. Details of the teams are included in Table 3. Overall, teams in Finland and Hungary were more experienced in the use of Lean methods, having used Lean for almost three years, whereas in China teams were just starting up.

Table 3. Team Amplifier sessions.

Finland and Hungary 8 sessions involving around 42 people. 2 hours/session	
Development Team 1,2	2 hours
Fault Handling Team 1	2 hours
Tools Team	2 hours
Management Team	2+2 hours
Agile Coaches Team	2+2 hours
Product Owner Team	2 hours
China 14 sessions involving around 56 people. 3-4hours/session	
Development Teams 1,2,3	Distributed design team, amplifier session after 2-4 sprints, follow-up amplifier session after 6 months.
Product owner Team	Amplifier session in the beginning and a follow-up session after 6 months
Fault Handling Teams 1,2	Local team using kanban. Amplifier session before the first sprint, follow-up amplifier session after 6 months.
Development Team 4,5,6	Distributed design team, amplifier sessions after 2-3 sprints

Team sessions were conducted according to the following dynamics. The *Amplifier* questions were discussed one by one (applicable ones depending on the team type: product owners, developers, coaches, etc.). The question under discussion (in the form of a statement) was displayed on a big screen and team members individually gave their personal opinions on the current status of the team with regard to that statement by voting it. The ratings used were: 0= *Not applicable/Do not understand/Do not see*; 1= *Not yet demonstrated*; 2= *Basic knowledge and skills*; 3= *Good knowledge and skills* and 4= *Highly developed knowledge and skills*. Subsequently, an open discussion took place in which everyone explained the reasons for his/her personal rating. Finally, team members agreed on areas for improvement. Examples of representative *Team Amplifier* statements discussed during the sessions are shown in Table 4. A company representative (the second author) facilitated the sessions. Afterwards the results were analysed by identifying the top three challenges and achievements for each team according to the voting results, and analyzing patterns between teams.

4. RESULTS

This section presents the results of the study. The main elements of Ericsson R&D Finland's Lean Software Development are analysed in Section 4.1. They are structured according to the core five principles of Lean [33] described in Section 2.1. Section 4.2 presents achievements and challenges including sites in Finland, Hungary and China.

4.1 Lean Thinking at Ericsson R&D

Our main research question is RQ1: *How are Lean principles interpreted and implemented in a domain such as software development that differs substantially from that in which Lean was originally developed?*

Ericsson R&D Finland implements Lean principles in the context of ASD. Thus, many of the fundamental elements of its process come from Agile methods such as Scrum. This is quite common approach in software development, as identified by previous studies [25, 31]. In the specific case of Ericsson R&D Finland, the use of ASD, at a more prescriptive level, is nowadays guided by Lean principles. Therefore, Agile is not abandoned when Lean is adopted but rather is incorporated into a process that combines the elements from both approaches that work better for Ericsson R&D. In the classification established by Wang et al.[31] of Lean application in ASD, we could consider Lean Software Development at Ericsson R&D as a case of category E: '*From agile to lean: comprehensive application of Lean approaches to transform Agile processes*'. Next we describe the main elements that emerged as important in Ericsson R&D Finland processes during the focus group sessions. They are also summarized in Table 4, including some examples of statements recorded in the sessions that evidence the findings.

4.1.1 Value and Value Stream

The principle '*Everything that we do in the organization is adding value'* guides each activity in the Lean development chain. Several strategies were identified for implementing this principle in practice. First, a *network of product owners* has been established to make the voice of the customer actively heard throughout the whole product lifecycle. Product owners guide the development and ensure that customer value is transparent for everyone. Product owners are value-driven and focus on the highest priority goal at any one time. Managers and leaders are responsible for setting the vision and the organizational goals. However, release and portfolio management must consistently involve and update product owners and other stakeholders throughout the entire programme. Thus, information on market trends and business situation is disseminated to everyone inside the company. On the other hand, product owners continuously challenge stakeholders'/customers' needs to find the best possible solution (what is internally called *bells & whistles versus bare bones solutions*). Although each product has a unique product owner, several *proxy product owners* support in attending development teams, creating *product owner teams*. Product owners and proxy product owners act as a unique interface with the organization for any work request towards the development team in order to keep them focused on what really produces customer value. Scrum masters or team coaches also shield the development team and coach staff in the surrounding organization to support them. Product owners are responsible for creating a clear and inspiring product/feature vision together with the team. At a more practical level, techniques such as INVEST (Independent, Negotiable, Valuable, Estimable, Small, Testable) user stories, providing clear acceptance criteria, manageable size (more granular at the top and more general towards the bottom) and MMFs (minimum marketable features) are applied to create the product backlog content. Development team members constantly collaborate with product owners (e.g. grooming the backlog, asking for early feedback, releasing 'Definition of Done' including non-functional requirements, etc.). We might say that the role of the product owner is quite close to the role of the *Chief Engineer* in Lean manufacturing (called *shusa* in the TPS) [33].

Product owners are expected to have full awareness of feature dependencies, budget, resources and schedule all the time.

Another idea from Lean thinking that Ericsson R&D Finland applies is *to manage products as a whole*. Release and portfolio management focuses on the big picture, collaborates with other areas (like hardware and platform units) and takes responsibility for the customer deliverable as a whole. Development teams are similarly coached for the same ends. For example, product owners/proxy product owners paint the big picture of the product for the development teams. Managers use techniques from Lean such as Value Stream Mapping for the products they are working with, looking for bottlenecks and acting on the highest priority one. Fostered by a culture of living with uncertainty, the norm '*making decisions at the last responsible moment*' is applied to irreversible decisions, i.e. options are kept open until the last responsible minute, when maximum information is available.

After three years of implementing Lean, Ericsson R&D has understood that *transparency* is the essential factor that makes this approach work synergistically. Difficulties in visualizing value streams and work items in software development, as well as software development volatility, demand continuous and fluent collaboration and communication between managers/leaders, product owners/proxy product owners and development teams. Accordingly, seating and facilities have been completely reformed in order to promote transparency in the development chain. Managers and leaders stay close to teams. Managers are seated in the R&D team areas, like any development team member. Also, they apply the Lean principle *Go-and-see* to make themselves aware of tools, technologies and features the teams are working with. They also participate in team ceremonies where they actively listen and ask questions. To capture relevant requirements, product owners are also in continuous close collaboration with relevant stakeholders/customers. Similarly, they have their own space in R&D team areas so as to have a closer relationship with teams. Some of the product owners are also seated with their respective team. The concept of the R&D team area in Ericsson R&D Finland is reminiscent of the *cellular manufacturing* and *work-cells* in Lean manufacturing. The purpose of both is the same, to increase transparency, facilitate flow and promote learning. Release plans are up to date, reflecting the current feature status. The goal is that they are visible, shared (e.g. via a release burn-up or burn-down chart) and understood by all team members and all stakeholders.

Contrary to the commonly held (mis)understanding that the focus of Lean is essentially the elimination of waste or *muda*[5], our empirical evidence suggests that the concept of value and build quality is much more strongly emphasized. The concept of waste did come up in some discussions but always oriented towards a wide perspective of continuous improvement, rather than specifically reducing costs. For example, the statement '*Managers and leaders encourage identifying waste and challenging the current practice*' was recorded, attaching elimination of waste to the concept of continuous improvement.

4.1.2 Flow and Pull

Three elements were found essential for achieving flow. First, during these years of Lean transformation, the organizational structure has been flattened to remove 'functional silos', for example between marketing and engineering functions. At present, a maximum of three layers can be distinguished in Ericsson R&D Finland and even as few as two in some areas. As a result, information feedback loops between different parts of the organization are currently shorter and unnecessary handovers performed between teams and at organizational levels have decreased. Second, considerable effort has been devoted to creating a successful Continuous Integration (CI) environment. Developers can now integrate code on a second-by-second basis (unit and component tests) and teams on a minute-by-minute basis (team builds, static analysers and simulated smoke test). Moreover, system builds are run daily (system level testing). The feedback loop between feature/system level testing and coding has significantly improved. Build times have been reduced over ten times and the number of commits per day increased roughly five times up to 2011. CI allows checking at any time whether the product meets what the customer really values. Third, limiting work in progress (WIP) is a Lean practice that has been applied from portfolio management to team level to facilitate flow. Unfinished work is kept to a minimum and portfolio management already applies limits in the development WIP. Development teams have their WIP limits as well. Limits have had to be adjusted multiple times until a suitable limit was found for each team. Although competence between teams is approximately equal, teamwork and capacity are not exactly the same and each team needed to experiment to find what limits work best in its specific context. At a personal level, people are also coached to help them limit their personal WIP (e.g. do not take on too many tasks at the same time), stay focused (e.g. start and finish meetings on time, stick to the goal) and keep a sustainable pace.

According to Lean thinking, *standardization* is a key concept for achieving flow. In a software development environment, Staats et al. [29] found that standardized approaches led to fewer defects, less rework and improved productivity in Lean Software Development. However, they also recognized this task as a challenge because '*the lack of task repetition within software services obscures the degree to which tasks can be specified and then standardized*'. In our study, we found that Ericsson R&D Finland is indeed trying to avoid extra standardization when implementing Lean. Of course, standards continue to be used in some areas such as coding standards, partially standardized Definition of Done and semi-standardized processes following a flexible Scrum/Kanban framework. However, teams have the flexibility to select the practices that work better in their specific context, within the Agile, Scrum-oriented framework. In Ericsson R&D Finland's experience, over-standardizing processes and extensive mechanisms to ensure predictability and control prevent flexibility and closeness to the customer, resulting in organizational silos with multiple handover-related challenges. Instead of aiming for standardization, Ericsson R&D Finland is, by deploying Lean, learning how to stop worrying and live with uncertainty by creating a continuous improvement and learning organizational culture.

4.1.3 Perfection

Challenging the status quo, in aspects such as organizational needs, assumptions about tools and practices and personal behaviour, is one of the foundations of the company. In fact, as previously explained, it was the main driver triggering the transformation at Ericsson R&D Finland. Based on Lean principles, many techniques have been introduced for continuously improving both *hard* and *soft* aspects of software development. The purpose is that conflicts are not hidden and team is able to handle them. *Team retrospectives, communities of practice gatherings* and *open spaces* are frequently organized in

[5] For example, the title of Womack and Jones's 1996 book is *Lean Thinking. Banish* ***Waste*** *and Create Wealth in Your Organization*.

order to challenge current practice. Managers also promote and support the Lean principle of *stop-the-line*, a system under which everything is stopped when a problem is found in order to identify the root cause and work towards preventing it from happening again in the future. A *network of Scrum masters and coaches* has been created to help teams see their conflicts and decide what to do about them. Scrum masters, coaches and managers help remove impediments that development teams are not able to resolve by themselves and promote the finding of solutions for impediments that go beyond organizational borders. Moreover, people are coached to challenge themselves and improve their own individual performance. Ericsson R&D Finland's experience shows that solving impediments with proper scope and at the right time brings benefits in terms of avoiding making unnecessary major improvements upfront.

Promoting a *learning organization* is another pillar of the company. One of the most important means of encouraging a continuous learning culture is the concept of *Communities of Practice* (CoPs). CoPs are used, together with open spaces, for discussing problems and solutions, sharing good practices and seeking new ideas regarding a specific role/practice/topic. The purpose is to incorporate as many opinions as possible into discussions. Moreover, CoPs help to promote a culture of transparency in the organization. Team *experiments*, '*learning through fast feedback loops and safe/fast failure*', are also praised. Managers and leaders demonstrate openness to feedback and proposals. Technical aspects of continuous learning and excellence are also supported by appropriate training programmes. Software *craftsmanship skills* are encouraged in techniques such as coding standards, refactoring, unit testing, test-driven development, behavior-driven development, continuous integration, frequent reviews, collective ownership, simple design and coding/testing dojo.

4.1.4 People

Respect people is a central aspect of the Lean company culture. There is trust that all teams and individuals are doing their best and have the conditions to do their best. A balance between personal goals and team and organizational goals is pursued. A clear example of how this culture has been embraced is the extreme care with which measurements are selected. Measures are made of only what adds value and 'one level up' (e.g. team instead of individual). Teamwork and self-organization are also widely promoted.

Table 4. Main characteristics of Lean Software Development at Ericsson R&D Finland.

Value and Value Stream	
Elements for implementing Value and Value Stream	- Management involves and updates product owners and other stakeholders throughout the entire programme regarding organizational vision and goals. - A network of 'product owners' and 'proxy product owners' makes customer value transparent for everyone. - *Bells & whistles* versus *bare bones* solutions. - Product is managed as a whole through constant collaboration with other areas such as hardware and platform. - Value Stream Maps are used to produce a big picture for R&D teams. - Transparency, supported by R&D team area concept, is a cornerstone.
Team Amplifier statements	- 'Backlog is constantly updated and prioritized according to latest information and driven by value.' - 'Release plan is up-to-date reflecting the current feature status and is visible, shared (e.g. via release burn-up or burn-down chart) and understood by all team members and all stakeholders.' - 'Teams see the big picture, collaborate with other teams and take responsibility for the product as a whole.'
Flow and Pull	
Elements for Implementing Flow and Pull	- No functional silos inside the organization. - Short Continuous Integration feedback cycles with a satisfying test coverage. - Limiting work in progress from management to development to maintain a sustainable pace. - Full awareness of feature dependencies.
Team Amplifier statements	- 'No unnecessary handovers performed within the team and among teams.' - 'Scrum Masters encourage the team to have a short CI feedback cycle with satisfying test coverage.' - 'We keep the unfinished work to a minimum.'
Perfection	
Elements for Implementing Perfection	- Continuously challenging the status quo. Continuous improvement and continuous learning organization. - Communities of practice and open spaces. - Experiments, fast feedback loops and safe/fast failure. - Root cause analysis of problems and stop-the-line. - State of the art tool environment and technical excellence. - Design and provide Agile/Lean training based on needs.
Team Amplifier statements	- 'Learning and practices from inside and outside Ericsson are actively shared, evaluated and further improved within the organization.' - 'Care about excellence in whatever you do.' - 'Product value stream shows our bottlenecks and we act on the one with the highest priority.'
People	
Other key elements	- Respect people. - Self-organization. - Team work.
Team Amplifier statements	- 'We trust that every team and individual is doing their best and they have the conditions to do their best.' - 'Team members actively make use of each other's competencies and help each other grow.' - 'Teams have clear goals and all team members are committed to them.'

Managers and product owners do not push people in any direction but empower the team and encourage team organization and collaboration. Ericsson R&D Finland recognizes that success depends on teams taking more initiative and responsibility. Indeed, its experience is that, the more freedom is given to the teams, the more responsibility they are prepared to take. When team members have responsibility, they can make decisions faster, speeding up the development process. Coaches support and teach teams on how to make decisions effectively and to follow them up. Building teams is easy but to work as a genuine team and not as individuals requires a mindset change that is not as easy to attain. For this reason, team members are actively involved in setting up the teams. Some tips related to teamwork were intensely discussed during focus group sessions (see Team Amplifier statements in Table 4).

4.2 Achievements and Challenges

The *Team Amplifier* was used to identify achievements and challenges when implementing Lean Software Development in Ericsson R&D Finland and in Ericsson R&D cooperating sites in China and Hungary. Although each team had its own situation, some patterns were identified among teams that are to some extent generalizable. Our second research question can be divided in two sub-questions:

RQ2.a) What elements of Lean thinking are challenging the implementation of Lean in Software Development?

Several areas of improvement, or 'battles', were found during team sessions. Three categories of challenges emerged most frequently among teams:

1. Achieving flow: Flow has been easier achieved in product maintenance, where the nature of the work consists of a constant flow of customer service requests. The use of Kanban, together with a flatter organizational structure, has decreased the number of unnecessary handovers. So, Fault handling teams have significantly speeded up their responses to faults and service requests from customers. However, achieving end-to-end flow in product development has been found more challenging, mainly owing to the number of decisions involved in the process that can create handovers. It was found that flattening the organizational structure has considerably decreased the number of unnecessary handovers performed within the team and between teams. However, it has been also realized that existing decision points need to be further developed to support readiness and enable flow. Moreover, in order to achieve real flow, the whole organization must work in the same Lean mode. For example, managing the product as a whole is difficult if the culture of neighbouring areas such as Human Resources or Marketing is not aligned with Lean thinking. Transformation is expanding and the whole organization is moving in the same direction. A seamless organization is needed to obtain end-to-end benefits. Additionally, decreasing work in progress as well as reducing large work items, from customer deliveries to team tasks, has been found critical for achieving flow but also challenging. Large work items produce problems in planning sessions when translated into creating small tasks at team level. Ultimately, a company culture of flow must be created. It was found that 'push culture' remains, mainly in teams starting the transformation, interrupting them from their sprint goal and preventing flow.

2. Transparency: Although transparency in the development chain has increased since the introduction of Lean Software Development, there is still a tendency in development teams to limit collaboration outside the team to Agile needs only. For example, collaboration for improvements outside teams was found to be rare. This represents a risk, since it can lead to merely local optimizations at the expense of the overall system. Keeping systems thinking in mind when teams concentrate on one sprint at a time, is tough, especially at the beginning of the transformation; this was the case in teams in China.

3. Create a culture of continuous learning: At the beginning of the transformation, wider competences were promoted. Some people misunderstood this to mean that everyone should become a generalist. However, the focus was to encourage progressive learning through fast feedback loops, simple and safe experiments and the use of one another's competences in order to help each other grow. Communities of practice have been found very effective in this sense, mainly in technical and engineering practices. However, it was also found that high pressure to deliver to the customer limits time available for simultaneous learning. Some teams had difficulty finding time to conduct experiments. Learning requires time and commitment and without any of those elements it is not possible to create a culture of continuous learning.

RQ2.b) What are the elements of Lean thinking that are more easily achievable in a software development context?

Three main achievements were found during the team sessions.

1. Creating a culture of continuous improvement: Generally, creating a culture of continuous improvement was found to be easier than creating a culture of continuous learning, since learning is a step further that requests more maturity. It is true that such practices as retrospectives have been used for quite a long time in Ericsson R&D Finland and Hungary (from its Agile conversion), with the result that teams have already interiorized their dynamics quite well. However, less mature teams in using Agile and Lean, such as teams in China, have also quickly embraced a culture of continues improvement. Interestingly, not only retrospective routines but also improvement actions were well established as part of their daily routine. Some teams also indicated that constant and free discussion on improvements is more useful than the use of retrospectives. The main challenge emerging here was how to focus improvement actions. Many opportunities for improvement may be discovered but careful focus is needed to select the improvements that have highest priority. A tendency was found to focus improvement actions on 'daily issues'. One reason could be that it is easier to take actions for improvements in a team scope than to remove impediments that go beyond team and even organizational borders. Evidently, more organizational transparency, which was found as a challenge, is needed in the last case.

2. Involving people in the transformation: Teams were found to be highly motivated to work towards Lean and Agile, which was considered an encouraging and positive insight. People have more fun at work and are motivated by the benefits achieved in Ericsson R&D Finland (site using Lean longer). To make visible benefits and success stories is a mean for spreading the transformation. As a result, the whole organization is moving in the same direction.

3. Creating a team culture: Although there is still much individual work in some teams and resistance to pair work was also found, coaching techniques were identified as providing very good results in creating teams that genuinely work as such, rather than as individuals. Feedback during the teams' sessions was very positive, reflected in the good results of statements such as '*Team members give and openly receive both positive and constructive feedback to each other*', '*Team members actively make use of each other's competencies and help each other grow*' and '*Team members have fun together*'.

During its journey, Ericsson R&D Finland (promoter of the transformation) has learned some valuable lessons. First, Kanban and Scrum are good methods if supported by the right thinking for guiding Lean transformation. Second, experience is the real source of learning. First-hand experience from teams relating to how they cooperate and how Lean impacts on them, is the real source of learning. Only through extensive experience and deep analysis of the specific context is it possible to understand how well the new way of working is tuned into the situation. Third, visibility and transparency are essential for achieving Lean principles in a domain such as software development, where work items are less perceptible than in manufacturing. Finally, good leadership and coaching skills beyond simply selecting methods and tools and trying to practice them are needed. In line with the findings of Ebert et al. [10], the experience of Ericsson R&D Finland is that superficial inclusion of Lean, based simply on operational tools, yields frustration when people realize that the benefits are not attained.

5. CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

Contrary to the general belief that a crisis has to precede the adoption of Lean, in Ericsson R&D Finland the transformation has happened smoothly, without the spur of a previous disaster. Before the application of Lean thinking, the company was using Agile methods under the umbrella of Scrum. Lean thinking was posteriorly adopted to complement Agile. Specifically ASD usage, at a more prescriptive level, is guided by Lean principles. The study evidences numerous compatibilities between Lean and Agile. Thus, the way of working at Ericsson R&D Finland today is a combination of elements from both approaches that works best for them. An incremental process improvement is clearly recognized where well known agile elements such as product owners and continuous integration are combined with newer elements like emphasized transparency, R&D team areas, value stream mapping and work in progress limits.

In order to ensure that each activity in the development chain provides customer value, a network of product owners, to manage products as a whole, considering the end-to-end development chain, and transparency, have been identified as being of fundamental importance. Having a seamless organization, Continuous Integration and limiting work in progress have emerged as important elements for achieving flow. A culture of continuous learning and improvement is promoted through initiatives such as creating communities of practice inside the organization, conducting frequent retrospectives and open spaces for challenging the status quo and applying Lean techniques such as stop-the-line. Finally, respect for people, empowerment and self-organization as well as teamwork were found to be core elements of the company Lean implementation. Evaluation of 16 teams identified positive experiences as well as challenges when applying Lean Software Development. Attaining flow, transparency and creating a learning culture were found to present more difficulties. On the other hand, creating a culture of continuous improvement, involving teams in the transformation and creating a team culture were seen as readily achievable if appropriate coaching techniques are applied to support the change.

Related to threats to validity, like any case study, our study involves a set of threats that need to be considered when interpreting the results. In particular, our study explores Lean Software Development in the context of one company. Thus, it is possible that our observations cannot be fully generalized to other settings (external validity). However, since the phenomenon of Lean Software Development is relatively unexplored and Ericsson R&D Finland can be considered as an early adopter that has achieved a successful transformation (some benefits have already been perceived), we believe that the results of the study can bring new insights that may help guide organizations pursuing a similar endeavour. On the other hand, the question that may arise is whether Ericsson R&D Finland's Lean implementation is actually fully conformant with Lean Thinking. Our study does not focus on epistemological concerns, mainly because, as in other domains in which Lean has been applied, there is no universally accepted definition of Lean in software development. However, Ericsson is part of a large initiative investigating ways of applying Lean thinking in software development [4] and we are convinced that Ericsson R&D Finland is consciously trying to transform itself to Lean Software Development. In an attempt to minimize threats to construct validity (to what extent operational measures represent the concepts being studied), a material walkthrough workshop was organized at the beginning of the research, which helped researchers and participants *to speak the same language*. Moreover, multiple data sources (process documentation and focus groups) were used during the study and participants' triangulation lent the data greater accuracy and validity. Finally, a typical danger in case study research is the impact of preconceptions of the researchers in data collection and analysis. In order to minimize this threat, the researchers took part in just a few of the focus groups, and investigator triangulation was used in both phases. The fact that the Lean initiative was occurring contemporaneously with the research study helped also to avoid retrospection bias.

The work here suggests several directions for future investigation. Our exploratory study provides an overall picture of how Lean is implemented by a software-intensive organization. However, elements of this phenomenon that have been identified as essential, as well as strengths and challenges when moving towards Lean Software Development, are topics of interest that should be studied in greater detail. For example, how to apply the principle of pull was briefly discussed during focus groups sessions and therefore, moderately investigated during team evaluations. Another topic that merits further study is sources of waste in software development. Moreover, the present study concentrates on the experience of just one company, although staff with clearly different profiles took part. To conduct similar analysis of other software intensive companies' Lean Software Development implementation would enable *analytical generalization* by extending results to cases that have common characteristics.

6. ACKNOWLEDGMENTS

This article is based on the work carried out in the ICT SHOK Cloud Software program financed by the Finnish Funding Agency for Technology and Innovation (Tekes) and Tivit OY. The graduate school on Software Systems and Engineering (SoSE), funded by the Ministry of Education in Finland and by the Academy of Finland, has also partially supported the work.

7. REFERENCES

[1] Anderson, D. 2010. Kanban – Successful Evolutionary Change for Your Technology Business. *Blue Hole Press*.

[2] Brandao de Souza, L. 2009. Trends and approaches in lean healthcare. *Leadership in Health Services*. 22, 2, 121-139. DOI=http://dx.doi.org/10.1108/17511870910953788.

[3] Bremer, B. and Dawson, D. 2003. Can anything stop Toyota? *Business Week*, 117 (Nov. 17).

[4] Cloud Software program (2010-2013). Finland. http://www.cloudsoftwareprogram.org/

[5] Conference on Lean Enterprise Software and Systems (last one organized in Tallin, http://less2012.org/).

[6] Conboy, K., 2009. Agility from first principles: reconstructing the concept of agility in information systems development. *Inform. Syst. Res.* 20, 3, 329–354.

[7] Dingsøyr, T., Brede Moe, N., Nerur, S., and Balijepally, V. 2012. Special issue: agile development. *The Journal of Systems and Software*. 85, 6 (June 2012), 1213-1452.

[8] Dybå, T., and Dingsøyr, T. 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50, 9–10 (August 2008), 833-859. ISSN 0950-5849. 10.1016/j.infsof.2008.01.006.

[9] Ebert, C., Abrahamsson, A., and Oza, N. 2012. Special issue in Lean Software Development. *IEEE Magazine*. 29, 5 (Sept.-Oct. 2012). ISSN: 0740-7459.

[10] Ebert, C., Abrahamsson, A., and Oza, N. 2012. Lean Software Development. *IEEE Software*, 29, 5 (Sept.-Oct. 2012), 22-25. DOI= doi:10.1109/MS.2012.116.

[11] Engblom, C. 2011. From Waterfall to Lean and Agile within Ericsson. Keynote at 6th IEEE International Conference on Global Software Engineering, ICGSE 2011 (Helsinki, Finland, August 15-18th, 2011)

[12] Hallikainen, M. 2011. Experiences on Agile Seating, Facilities and Solutions: Multisite Environment. In *Proceedings of the 2011 IEEE Sixth International Conference on Global Software Engineering* (ICGSE '11). IEEE Computer Society, Washington, DC, USA, 119-123. DOI=10.1109/ICGSE.2011.20

[13] Karvonen, T., Rodriguez, P., Kuvaja, P., Mikkonen, K., and Oivo, M. 2012. Adapting the Lean Enterprise Self-Assessment Tool for the Software Development Domain. *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on*. 266-273 5-8 Sept. doi: 10.1109/SEAA.2012.51

[14] Kontio, J., Bragge, J., and Lehtola, L. 2008. *The focus group method as an empirical tool in software engineering*. In: F. Shull et al. (eds.), Guide to advanced empirical software engineering. Springer, 93-116.

[15] Larman, C., and Vodde, B. 2009. Scaling Lean & Agile Development. *Addison-Wesley*.

[16] Lean Software and Systems Conference series http://www.leanssc.org/conferences/ .

[17] Mandic, V., Oivo, M., Rodriguez, P., Kuvaja, P., Kaikkonen H., and Turhan B. 2010. What is flowing in Lean Software Development? *In Proceedings of the 1st International Conference on Lean Enterprise Software and Systems* (LESS),72-84.

[18] Mehta, M., Anderson, D. and Raffo, D. 2008. Providing Value to Customers in Software Development Through Lean Principles. *Software Process Improvement and Practice*, 13, 1 (January 2008), 101-109.

[19] Middleton, P., and Sutton, J. 2005. Lean Software Strategies. *Productivity Press*.

[20] Mikkonen, K., Seikola, M., Jouppila, A. and Engblom, C. 2012. How we learn to stop worrying and live with the uncertainties. Technical report. Available at https://www.cloudsoftwareprogram.org/results/deliverables-and-other-reports/i/27891/1941/ericsson-journey-of-change (date last accessed 11 January, 2013)

[21] Ohno, T. 1988. *Toyota Production System: Beyond Large-Scale Production*. Productivity Press, Cambridge, MA.

[22] Petersen, K., and Wohlin, C. 2011 Measuring the flow in lean software development. *Software: Practice and Experience*, 41, 9 (August 2011), 975-996.

[23] Poppendieck, M., and Cusumano, M.A. 2012. Lean Software Development: A Tutorial. *Software, IEEE*. 29, 5, 26-32, Sept.-Oct. 2012 doi: 10.1109/MS.2012.107

[24] Poppendieck, M., and Poppendieck, T. 2003. Lean Software Development: An Agile Toolkit. *Addison Wesley Professional*.

[25] Rodríguez, P., Markkula, J., Turula, K., and Oivo, M. 2012. Survey on Agile and Lean Usage in Finnish Software Industry. *In Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement (*ESEM '12). ACM, New York, NY, USA, 139-148.

[26] Runeson, P., and Höst, M. 2009. Guidelines for conducting and reporting case study research in software engineering, *Empir Software Eng*, 14,131-164.

[27] Seikola, M., Loisa, H.M., and Jagos, A. 2011. Kanban Implementation in a Telecom Product Maintenance. In *Proceedings of the 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications* (SEAA '11). IEEE Computer Society, Washington, DC, USA, 321-329. DOI=10.1109/SEAA.2011.56

[28] Shah, R., and Ward, P.T., 2007. Defining and developing measures of lean production. *Journal of Operations Management* 25, 4, 785–805.

[29] Staats, B., Brunner, D. and Upton, D. 2011. Lean principles, learning, and knowledge work: Evidence from a software services provider. *Journal of Operations Management*, 29, 5 (July 201), 376-390.

[30] Tokatli, N. 2007. Global sourcing: insights from the global clothing industry—the case of Zara, a fast fashion retailer. *Journal of Economic Geography*, 8, 1, 21-38.

[31] Wang, X., Conboy, K., and Cawley, O. 2012. "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development, *J. Syst. Software*, 85, 6, 1287-1299.

[32] Womack J.P., Jones D.T., and Roos D. (1990) *The Machine that Changed the World: The Story of Lean Production*. New York: HarperPerennial.

[33] Womack, J. P., and Jones, D.T. 2003. *Lean Thinking: Banish waste and create wealth in your corporation*. Revised and updated. Free Press, New York, USA.