# Combining Lean Thinking and Agile Methods for Software Development
# A Case Study of a Finnish Provider of Wireless Embedded Systems

Pilar Rodríguez
University of Oulu
Pilar.Rodriguez@oulu.fi

Jari Partanen
Elektrobit
Jari.Partanen@elektrobit.com

Pasi Kuvaja
University of Oulu
Pasi.Kuvaja@oulu.fi

Markku Oivo
University of Oulu
Markku.Oivo@oulu.fi

## Abstract

*Lean Software Development has attracted a great deal of attention during last years. However, it remains unclear how Lean is implemented in a domain that fundamentally differs from the automotive industry in which it originated. This study provides empirical evidence of how Lean can be combined with Agile methods to enhance software development processes. A case study was conducted at Elektrobit Wireless Segment, which has used Agile from 2007 and began to adopt Lean in 2010. Our findings evidence numerous compatibilities between Lean and Agile. In addition to well-established practices in Agile, Lean thinking has brought new elements to software development such as Kanban and work-in-progress limits, a "pull" and "less waste" oriented culture, and a stronger emphasis on transparency and collaborative development. Scaling flexibility, business management involvement and waste reduction were found as challenges, whilst setting up teams, self-organization and empowerment appeared easier to achieve.*

## 1. Introduction

Agile software development (ASD) has become popular among industrial practitioners because of its promising benefits in terms of decreasing time-to-market and increasing development flexibility and product quality [1, 2]. However, Agile, as traditionally described in methods such as Scrum and eXtreme Programming, has been found to have limitations for scaling to the whole organization [3]. Lean is a new fuzzy term that has emerged in software development, especially in the circles closest to Agile methods, as a means of enhancing software development processes and scaling up Agile [4]. However, it is unclear how to implement Lean thinking—originally developed in the automotive industry—in a domain as software development, in which most items are intangible and where work primarily involves information and knowledgeable workers. Moreover, it is important to understand how Lean thinking can be combined with well-established software engineering practices to achieve the best-of-all approaches.

The literature reveals that the idea of applying Lean to software development appeared already in the 90s [5], well before the Agile Manifesto (http://agilemanifesto.org/, 2001) was formulated. Originally, the focus was on making software development processes more efficient by removing '*waste*'. Nowadays, it is generally believed that as long as Agile and Lean in software development nearly converge, they could complement each other and enhance software development processes [6, 7, 8]. Due to the attention that Lean is causing in the software industry, there is a growing body of literature in the topic [9, 10, 11, 12]. However, as recently noticed by Ebert, Abrahamsson and Oza "*We're still in the early phases of truly understanding how Lean methods impact software development. […]. If everything is called 'Lean', and different methods from Agile to project management are mixed ad hoc, confusion results both in science and practice*" [13].

This paper contributes a better understanding of Lean in the context of ASD by analyzing how Elektrobit (EB), a large Finnish provider of wireless embedded systems, combines Agile methods and Lean thinking in its software development processes. Specifically, EB's Wireless Segment, which has used Agile since 2007 and began to adopt Lean in 2010, is the object of study. During these years, the segment has perceived important enhancements in its software development processes. For example, internal metrics shows that productivity has increased by more than 30% in some areas, customer satisfaction has also improved as a result of a better ability to respond to changes, information and expertise are more properly shared and the work environment better encourages and support the generation of new ideas.

The primary interest of this study is in identifying the essential elements that characterize EB's Wireless Segment Lean and Agile way of working, especially those that the adoption of Lean has brought on top of practices that predate the Lean Software Development movement. In order to have a wide picture of the phenomenon, we are also interested in identifying

978-1-4799-2504-9/14 $31.00 © 2014 IEEE
DOI 10.1109/HICSS.2014.586

challenges that EB wireless segment's personnel has faced when transitioning to Lean and Agile as well as more readily achievable elements. In the context of an assessment exercise, we explored EB way of working by conducting five focus group sessions with EB's software processes experts and a personnel survey filled by 49 employees of the segment. The paper makes two primary contributions. First, it offers deeper insight into Lean Software Development, and the essential elements that define its combination with ASD. Second, although technologies are not universally good but appropriate according to the circumstances, our findings may be useful for other organizations in the process of adopting Lean thinking, insofar as they provide new insights into how Lean Software Development is occurring in practice and challenges potentially faced when adopting it.

The paper is structured as follows. Section 2 reviews related work. Section 3 describes the research setting. Section 4 presents the findings, which are discussed in Section 5. Finally, section 6 presents conclusions, limitations and future research.

## 2. Background and related work

This section introduces Lean thinking as originally developed in manufacturing and reviews prior research in the specific field of Lean Software Development.

### 2.1. Lean thinking and its five core principles

Lean thinking emerged in the Japanese automotive industry in the 1940s. Shortly, it focuses on maximizing value and minimizing waste in production processes. The difficulties of analyzing it arise because there is no common definition of Lean neither in manufacturing nor software development [10]. Thus, Lean has been differently described by various authors [14, 15, 16] and adapted to different domains such as aeronautics [17], health care [18] and retail clothing [19]. In this study, Lean Software Development is explored through the lens of the five principles of Lean thinking as originally identified by researchers from the *International Motor Vehicle Program* (MIT-IMVP) [15]. Since there is not a standardized definition of Lean thinking, we believe these principles can help us to simplify the analysis to the roots of Lean, avoiding secondary interpretations bias. These principles are:

- *Value*, understood from a customer's perspective, is the core focus of Lean. Everything done in the organization should produce value to the customer. Thus, if something absorbs resources but produces no value, it is considered *waste* and has to be removed.
- *Value stream* is the end-to-end collection of actions required to bring a product from customer order to customer care, ensuring that each activity provides customer value.
- *Flow* means that the value stream does not have discontinuities so that activities are organized as a continuous 'flow' enabling smooth deliveries.
- *Pull* implies producing products (or part of products) only when they are really needed (just-in-time), on demand on the customer.
- *Perfection* centers in the concept of continuous improvement to achieve zero defects.

### 2.2. Previous work on Agile and Lean software development

The potential exhibited by Lean in terms of productivity, time-to-market, product quality and customer satisfaction [20] have aroused the interest of the software industry. Although, the universal application of Lean principles to knowledge work like software development is under debate [10], more authors agree that whilst specific practices and tools need to be adapted, Lean principles could be virtually applied to any domain [10, 21]. Lean thinking in software development started as early as the 1990s with concepts such as Lean software production [5]. However, Lean Software Development is known today through its promotion by the Agile community [1], where it has progressively acquired an identity of its own [7]. As philosophies Agile and Lean have some differences [22]. However, the particularities of software products, such as its value proposition and malleability, open new opportunities for combining Agile and Lean in a software domain [8]. Thus, Poppendieck considered Lean thinking a "*platform upon which to build agile software development practices*." [23]. Similarly, Edogmus said that "*if we avoid nit-picking based on semantics, it's easy to come up with a many-to-many mapping* [between the two approaches]" [6], and Coplien and Bjornwig argue that Agile and Lean complement each other by addressing different components of systems development [24].

Recent studies reviewing the body of knowledge of Lean in software development reflect the freshness of the topic. Based on 30 experience reports, Wang et al. [7] examined the purposes of applying Lean in ASD identifying six strategies: non-purposeful combination, using Lean to interact with other business areas while keeping Agile in software development, directly using Lean in software development processes to facilitate the adoption of Agile, using Lean in software development to improve Agile processes, transforming from Agile to Lean, and synchronizing Agile and Lean.

Although Wang et al.'s study offers significant insights into why Lean is applied with ASD, it does not deeply explore how the combination is actually implemented. More recently, Jonsson's systematic literature review [25] reveals that although Lean Software Development is a promising approach, the low amount of available studies and the dominance of some authors make difficult to draw reliable conclusions.

Most published knowledge in Lean Software Development are in form of books, which provide a diversity of interpretations of what Lean Software Development is and how it should be used (e.g. [4, 24, 26, 27]). Perhaps, the most widely acknowledged generalization of Lean Software Development principles is the seven principles compiled by Poppendieck and Poppendieck: eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build in integrity, and see the whole [4]. Other authors have stressed on Lean architectures [24] or using Kanban to bring Lean thinking [26]. However, there is also a growing body of scientific studies documenting case studies [9, 10, 11, 12] and analyzing specific elements of Lean [28]. Table 1 summarizes previous work, which as our study address the "*how*" of Lean Software Development.

**Table 1. Studies on Lean Software Development**

| Study | Scope | Main elements |
|---|---|---|
| Middleton 2001[29] | Two software development teams | Reducing WIP<br>Stop-the-line<br>Key factors: Organizational alignment and people aspects |
| Middleton Faxel and Cookson 2005 [30] | Case study Timberline Inc. | WIP limits and small batches dev.<br>Cross-functional teams<br>Standardized procedures<br>Transparency, data driven decisions |
| Mehta Anderson and Raffo 2008 [9] | IT department, web-based sales system | Frequent builds using cadence<br>Substantial time on upfront tasks<br>Integrated product teams<br>Transparent development process |
| Staats, Brunner and Upton 2011 [10] | Wipro Tech. (From waterfall to Lean) | Iterative development<br>Streamlined communication<br>Visual control boards<br>More standardized tasks |
| Middleton and Joyce 2012 [11] | Dev. team at BBC London using Kanban for one year | Transparency, Kanban boards and information radiators.<br>Daily standup meetings help to smooth workflow and to identify bottlenecks |
| Trimble and Webster 2013 [31] | Dev. team at NASA moving to Lean and Agile | Scrum, short delivery cycles<br>User centered development<br>Automatized unit testing<br>Continuous integration, "hackathon"<br>Tools support (JIRA) |
| Rodríguez et. al 2013 [12] | Ericsson R&D Finland | Scrum, Kanban, continuous integra.<br>Transparency using R&D team areas<br>Avoiding extra standardization<br>Empowerment and learning |

Although few, yet they reveal similitudes on how Lean is implemented in software development as well as conflictive aspects. Recurrent elements are Kanban [26], sometimes combined with Scrum, in which is called Scrumban, work in progress (WIP) limits, frequent builds through continuous integration, social and 'people' aspects such as team work and self-organization, development transparency (i.e., where everyone can see the entire development chain) and continuous improvement. Discrepancies are also observable in aspects such as work standardization and effort allocated to upfront analysis and architectural work. Our previous work is also aligned with these findings. In a case study with Ericsson we found that its combination of Agile and Lean is based on "*well known agile elements such as product owners and continuous integration* [and] *newer elements like emphasized transparency, R&D team areas, value stream mapping and WIP limits*" [12].

## 3. Case study design

As the study is exploratory, we used a case study strategy that facilitates deep understanding of the phenomenon in its natural context [32].

### 3.1. Objective and research questions

The goal of the study is to explore how software intensive companies combine Lean thinking and ASD in practice. Three complementary research questions drove the study as follows:

- *RQ1. What elements characterize the combination of Agile methods and Lean thinking in software development?*
- *RQ2. What challenges are potentially faced when combining ASD and Lean thinking?*
- *RQ3. What elements are more easily achievable when combining ASD and Lean thinking?*

### 3.2. The case company and its context

The study was conducted at the software development units of EB Wireless Segment. EB is a provider of embedded systems for the wireless and automotive industry. The company has more than 1600 employees in seven countries. Its wireless segment employs approximately 600 people distributed mainly in Finland and the United States. The segment offers wireless solutions for customer-specific devices in a not mass-manufacturing scale, to customers such as telecommunication manufacturers and public defense authorities. The wireless segment is organized around projects with a wide scope, ranging from prototypes to

complete validated solutions. A software development project involves from 3 or 5 up to 20 teams working simultaneously, with 2 to 15 teams directly dealing with software development. Systems are developed for operating systems such as Linux, Android and Windows using languages such as C++.

The initial implementation of Agile in EB Wireless Segment started in 2007 through pilots with selected teams and projects. In 2008, Agile, specifically Scrum, was taken into a wider context and the segment was organized around Scrum teams. However, as EB was maturing through its use of Agile, it was evident that scaling Agile extended well beyond just forming Agile teams. Thus, since 2010 Agile has been complemented with Lean principles, which are based on more holistic enterprise thinking. EB Wireless Segment could be included in the category E of Wang, Convoy and Cawley classification [7], '*purposeful application of lean approaches in ASD - Comprehensible application of Lean approaches to transform Agile processes*'. Currently, all software development teams use Scrum or Kanban as primary method. Projects are organized around product owners, Scrum or Kanban masters and teams. Approximately 100 Scrum masters have been trained, most of whom have been certified. As EB is also developing hardware other methods in addition to Scrum and Kanban are used. EB Wireless Segment has also progressively changed to a leaner structure, reducing the number of organizational levels to 4 or 5.

### 3.3. The research process

The research was driven by the company's desire to assess its transition to Lean and Agile. It was composed of two phases. First, EB Wireless Segment's way of working was explored using focus groups. Next, the segment was evaluated using a survey tool resulting from previous phase. Thus, the first phase focused on RQ1 and the second on RQ2 and RQ3.

**Phase 1: Focus group sessions.** Focus groups were conducted with experts guiding the transformation and employees applying Lean and Agile in different roles. For the design of the focus groups, we followed the guidelines of Kontio et al. [33]. As depicted in Table 2, five sessions were conducted, involving seven company representatives and three researchers. Company participants included two process managers, responsible for coaching Lean and Agile and ensuring processes quality; two program managers, responsible for programs consisting of multiple projects; one project manager, responsible for concrete projects; and two team members, one developer and one tester. The process managers participated in all sessions, whilst the rest took part in the last two sessions.

**Table 2. Focus groups design**

| Variation | Sessions | Participants | | Effort |
|---|---|---|---|---|
| Traditional face-to-face | 5 (~10 hours) | Attending all sessions | 2 process managers<br>3 researchers | 70 (p*h) |
| | | Attending last two sessions | 2 program managers<br>1 project manager<br>2 team members | |

The question "*What are the essential elements that define our Lean and Agile way of working, which should be included in a survey to assess our transformation?*" directed the discussions. As the number of questions that could be included in the survey was limited, only those elements considered as most important were included. As advised by Kontio et al. [33], a guide was used to focus the discussions. Based on company's internal material and literature on the topic, the researchers prepared statements in advance that could be part of the assessment survey tool. For example, as the concept of waste is relevant in Lean thinking, the statement '*We reduce wasteful activities frequently*' was included in the guide. The guide had 57 statements. During the focus group, the statements in the guide were displayed on a big screen and discussed one by one. It was then decided whether the statement under discussion was relevant to EB Wireless Segment or not. New topics related to the statement under discussion usually surfaced and, similarly, elements considered in the guide but found to be of little interest were dropped. Sessions were conducted until there was an agreement on the survey, which was finally composed by 97 statements.

Afterward, the survey statements were coded following the guidelines by Miles and Huberman [34, pp.54], to identify the most important elements of the Lean and Agile combination at EB Wireless Segment. During the coding process, descriptive codes emerged as important concepts were identified (e.g. "customer value", "prioritization", "tool", "communication", "teamwork"). Descriptive codes were subsequently clustered into themes such as "transparency" and "flexibility" and according to the five principles of Lean when possible, under the codes "value/waste", "value stream", "flow", "pull" and "perfection" using interpretative codes. The coded survey was analyzed to answer RQ1 and a company representative (the second author) reviewed the primary findings for validation. Due to confidentiality, the survey cannot be made publicly available. However, Appendix A shows a fragment of it, including the coding analysis.

**Phase 2: Lean and Agile assessment survey.** In the second phase, the survey was used to assess EB Wireless Segment processes, identifying challenges and more easily achievable aspects. In addition to the

statements designed during the focus groups, questions about demographics and respondents' profile were included. Statements were measured by asking the respondents to rate the extent to which he/she agrees with each of the 97 statements using a five-point Likert scale, from *strongly agree* (5), *agree (4)*, *neutral (3)*, *disagree (2)* to *strongly disagree* (1), including also an option "*I cannot answer/not applicable*" (0). Statements were designed in a positive way to define elements that should be in place. Four representative units of the segment were selected for the assessment. After piloting the questionnaire to check its legibility, an e-mail request was sent to 226 of the segment's staff. 49 responses were collected. Table 3 shows information about respondent's profile and experience in software development and ASD.

**Table 3. Survey design**

| Sample | Responses | Respondents profile [a] | Experience |
|---|---|---|---|
| 4 units (226 staff) | 49 (Response rate 22%) | Developers (53%)<br>Testers (12%)<br>Scrum masters (10%)<br>Architects (10%)<br>Project managers (8%)<br>Support functions (8%)<br>Build&integration (4%)<br>Business manager (2%)<br>Product owner (2%)<br>Release manager (2%)<br>Quality manager (2%)<br>Other (6%) | *SW development*<br>> 10 years: 59%<br>2-10 years: 17%<br>< 2 years: 23%<br><br>*Agile methods*<br>> 2 years: 36%<br>6-24 months: 30%<br>< 6 months: 32% |

[a.] Respondents could select more than one role.

The results were analyzed by calculating the average of each statement. "*I cannot answer/not applicable*" responses were omitted from the analysis. Statements were shortened by lower averages and those with an average lower than 3 (most of the respondents disagreed or strongly disagreed) were considered as challenges in applying Lean and Agile. In the case that more than one statement belonged to a specific topic, the average of the group was calculated. Statements with an average higher than 3.5 (most of the respondents agreed or strongly agreed), were similarly analyzed to identify more easily achievable elements.

# 4. Findings

## 4.1. RQ1: Agile and Lean combination

Discussions during the focus groups mainly concentrated on elements already known in ASD. However, a Lean flavor extending Agile concepts was clearly perceived, and some distinctive elements of Lean were also pointed out. For example, although the idea of product owners analyzing the "voice of the customer" is not new and has been practiced for more than a decade as part of Agile adoption, the emphasis on doing only things that add customer value extends the responsibility of caring about value not just to product owners but also to everyone else inside the organization. Figure 1 summarizes the findings of this phase. Findings are grouped by elements already considered in the literature of ASD, and elements brought by Lean thinking. Overall, we did not find a radical change from ASD to Lean Software Development but rather an incremental improvement in which Agile is not abandoned when Lean is adopted. Two aspects appeared supporting transversally EB way of working, transparency and people-oriented culture.

**Value and Value Stream**. 32% of the statements of the survey were related to customer value aspects. A network of business/product owners was established to manage customer value. Business owners work at higher level, whilst product owners work directly with development teams. Business owners define the release/feature content and priority the product backlog. Product owners collaborate closer with teams to prioritize and define user stories. Although features and user stories are defined from a customer's point of view (in problem domain), architects and system designers also participate in release and sprint planning meetings to support business/product owners to bring to bear an architectural and technical perspective when depicting the product. Moreover, in the case of large projects, one team is allocated to do upfront planning, analysis and architectural work in a Kanban mode, for those features that present a complicated Definition of Done. Release plannings are conducted every four weeks for both Scrum and Kanban teams. Scrum cycles last one or two weeks depending on the team. A unique product backlog is kept for each product using JIRA as the product management tool. At the end of each release/sprint, business/product owners validate and accept features and user stories respectively from a customer perspective. Although all these elements are part of Scrum, it was found that focusing on customer value is emphasized more by Lean thinking. Thus, discussions stressed that everything done in the organization should provide customer value, extending the responsibility of caring about value from product owners to everyone in the organization.

Speed and flexibility, key in ASD, were emphasized during the discussions. Discussions focused on the importance of having short lead-time between business opportunities and deliveries, and having room for changes in releases to include latest customer/business requests. In the software intensive

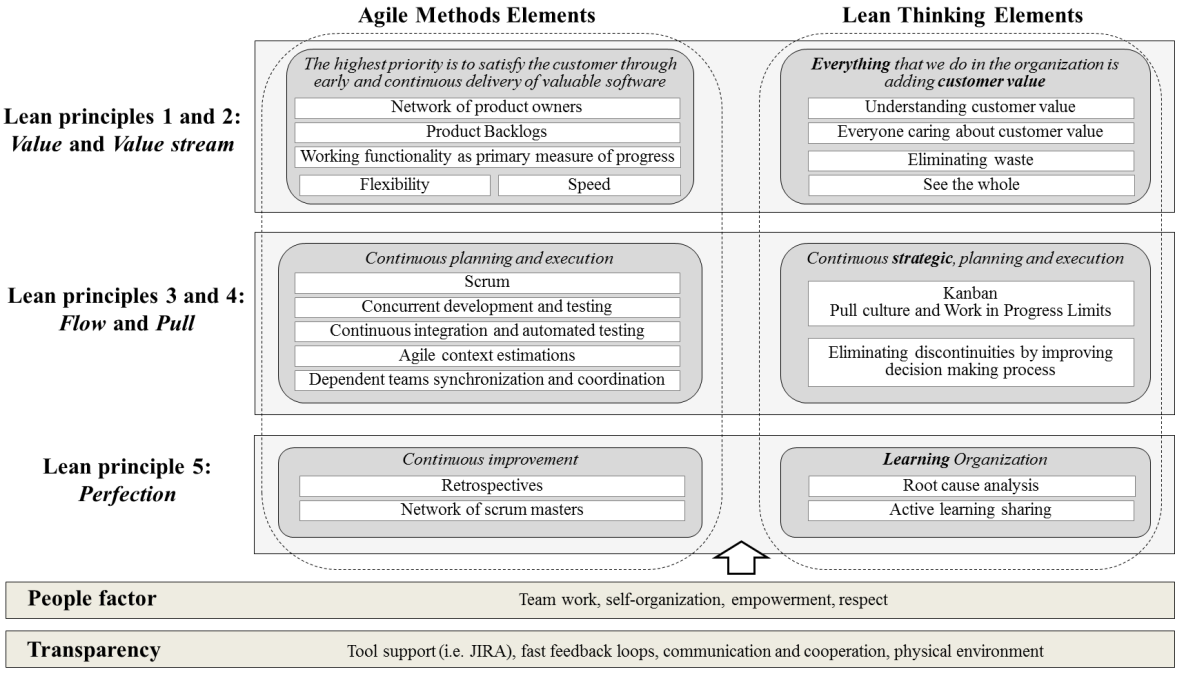


**Figure 1. Main elements characterizing EB's Wireless Segment Agile and Lean way of working**

industry flexibility is perceived as a must to meet customer needs. In a manufacturing context, flexibility has been often considered a source of waste that may make Lean and Agile not fully compatible [22]. Flexibility in software development does not derive from development volume, but from changeable customer needs. The fact that software is easily malleable and value is not limited to a single time-bound effort [21] enables new ways of combining flexibility and Lean thinking in a software context [8].

Finally, eliminating waste and seeing the whole were identified as newer elements brought by Lean thinking. Regarding waste, employees at EB wireless segment are trained to identify and reduce wasteful activities in their work. Delayed decision making, which usually causes unnecessary handover during the development process, excess WIP in the form of backlog work waiting to be done and unused features were discussed as typical sources of waste in software. Excess WIP was found to be especially relevant because it easily becomes obsolete being a potential source of bugs. On the other hand, seeing the whole was found important to recognize that systems are not simply the sum of their parts. It was discussed that organizational entities cannot work independently because the outcome is related to their cooperative capabilities to create value. Interacting components are important for reaching smooth value streams and avoiding local optimizations. Previous studies have indicated that "*the implementation of [product owner roles] has frequently led to violate the principle of optimizing the whole*" [21]. In manufacturing, Lean does not consider an intermediate role for defining and prioritizing work for separate teams as the role of Product Owner, but a Chief engineer takes the responsibility of the product as a whole. EB Wireless Segment uses a network of product owners but promotes to take responsibility for the product as a whole already from implementation teams. The statement "*Implementation teams see the big picture, collaborate with other teams and take responsibility for the product as a whole*" reflected this aspect.

**Flow and Pull**. Continuous planning and execution, which is extended from development teams to organizational strategy (internally called '*strategic agility*'), is a key element of the EB Wireless Segment way of working. At implementation level, two weeks Scrum sprints, continuous integrations and test automation support continuous planning, cadence and smooth deliveries. At business level the strategy of the segment is reviewed partially monthly and wholly quarterly to review the progress of the plans, results and key performance indicators. The purpose is to transform from annual clock orientation toward continual reconsideration of long-term targets, based on implementation feedback. Long term targets are divided into short term plans and executed in a Scrum mode. This way of working enables EB Wireless Segment to reduce the elapsed time between making a decision and seeing the consequences (shorter feedback cycles), endorsing continuous learning and improving the ability to adapt to customer changes.

Continuous planning and execution requires a high level of synchronization among business managers, product owners and project teams. Techniques such as Scrum of Scrum meetings, common iteration schedules for Scrum teams and fluent communication and cooperation inside and outside teams support synchronization and coordination. Estimations also play an important role. The emphasis is not on ensuring predictability and control. Contrarily, uncertainty is managed by continuous learning and adaptation through smaller and accurate estimations.

The principle of pull has been emphasized with the adoption of Kanban. Kanban is increasingly used for visualizing the workflow, measuring lead-time and explicitly limiting the WIP. Results of the assessment survey indicated that approximately 40% of EB Wireless Segment is using Kanban as the development method. For preventing discontinuities, Scrum or Kanban masters and Product Owners act as firewalls to protect teams from external interruptions so that teams can focus on the highest priority tasks. Especial focus is also put on the decision making process. Delays in decision making were considered a source of waste preventing flow. Thus, some of the focus group discussions stressed that decision making should not cause delays in the definition of releases and in the work of development teams.

**Perfection**. Agile continuous improvement has been extended to continuous learning with the adoption of Lean. Besides retrospectives and Scrum masters facilitating the work of the teams and eliminating impediments (already considered in Agile methods), Kanban and its ability to visualize queues, techniques to find the root cause of the problems as well as promoting sharing learning inside the organization were discussed during the sessions. One problem attributed to retrospectives is that if opportunities for improvement are not seriously considered, improvement activities begin to accumulate without positive results. Thus, EB Wireless Segment wanted not just to assess whether retrospectives were in place but also whether teams were really changing their way of working based on the results of retrospectives and finding the root causes of problems to avoid repeating the same issues from retrospective to retrospective.

**Transparency**. Transparency was the most stressed element at EB Wireless Segment in the discussions. In EB Wireless Segment's view, transparency is the key enabler to identify opportunities for improvement and to achieve aligned goals. So, almost all statements in the survey made in one way or another reference to transparency. Visibility to customer value at all levels and in all directions (from management to everyone and from everyone to management) was stressed. Statements such as "*Customer feedback is received and provided continuously during the whole development lifecycle*", "*My team has visibility of the plans and roadmaps that they are contributing to*" and "*Stakeholders attend and are interested in sprint reviews and demos*" resulted from the focus groups. Other statements evidencing the importance of transparency were "*All team members participate in Stand-up meetings*" and "*The members of my team daily make available the status of our project into JIRA, wiki or other similar tool*". Tools such as JIRA and wikis were found important to support transparency, communication, cooperation and short feedback loops. Effort has been invested in creating an environment that can be easily understood by all team members, independently of the member profile. Tools help share knowledge and spread management's top-down vision and strategy directions, which is discussed and improved as a consequence of people empowerment. Continuous integration, code review and code analysis tools help also teams to rapidly receive feedback in development issues such as bugs.

**People oriented development.** The people factor has been highly emphasized in ASD [35]. Respect people is also a pillar of Lean thinking [16], which ranges from more philosophical to more practical aspects such as humanity, respecting culture and customs, enhancing quality of life and enhancing individual creativity and teamwork. In the specific case of EB Wireless Segment, team work is strongly promoted. In fact the whole segment is organized around Scrum/Kanban teams with a common team goal. Because software development relies on knowledgeable and creative people, teams' self-organization and empowerment is also strongly promoted. When team members have responsibility, they can make decisions faster and speed up the development process. The people factor was represented in the survey through statements such as "*All my team members commit to sprint goals*", "*Differences between individuals are respected in my team*" and "*Implementation teams enjoy responsibility, trust and respect and not scapegoating*."

## 4.2. RQ2 and RQ3: Assessment results

In the second phase of the study, the Lean and Agile assessment survey was conducted to assess EB Wireless Segment way of working. Table 4 shows the results of this phase. Overall, strengths refer mainly to elements already considered in ASD, where EB has longer experience. However, key elements brought by Lean such as transparency and focus on customer value scored also highly in the survey.

**Table 4. Top five strengths and challenges**

| Strengths | Avg. | Challenges | Avg. |
|---|---|---|---|
| 1.Implementation set-up | 4.16 | 1. Flexibility | 2.69 |
| 2. Respect people | 3.90 | 2. Business mgmt. tasks | 2.72 |
| 3. Self-organization | 3.71 | 3. Waste reduction | 2.82 |
| 4.Focus on customer value | 3.68 | 4.Syncrhonization and coordination | 2.86 |
| 5.Transparency | 3.63 | 5. Short feedbak loops | 2.87 |

**Challenges.** Flexibility, in the form of capacity to include changes in products during the development, scored the lowest in the survey (2.69). Agile provides thoughts for increasing velocity and flexibility. However, to realize in a practical manner, flexibility needs to be through in the whole value stream, which according to the results of the survey remains as a challenge. On the other hand, short feedback loops from teams up to highest business levels are important in Agile and Lean. However, the results of the assessment showed that in practical set-ups it is not easy to involve business management to prioritize the backlog, defining the feature content and accepting the developed features as done. Statements regarding to these tasks scored 2.93, 2.47 and 2.78 respectively. Eliminating waste is other key principle in the Lean thinking. Assessed teams perceived that they were not able to remove waste so much (2.82), even they could identify it (3.6), because of their complex project set-up. Finding resources and time for carrying out activities to eliminate waste appeared as a challenge. Regarding synchronization and coordination, one of the bottlenecks was identified in the collaboration and synchronization of the hardware and software teams. Deliverables may vary from minutes or hours in software up to weeks in hardware development. This topic resulted in the survey in a quite low score (2.86). Finally, teams did not see the feedback loop as short as expected (2.87). Long feedback loops were found to be cause by challenges to involving business management with development and to integrating suppliers with development tasks.

**More ready achievable elements.** Set-up at implementation level, referring to elements of Agile methods such as roles, Scrum ceremonies and practices such as continuous improvement and automated testing, appeared to be in good situation. Thus, statements such as "*The role of the Product Owner is clear to me"*, "*Architects and System Designers are available for defining Feature/Release content*", "*The Scrum Master is available for my team when needed*" and "*Continuous Integration environment is available*" scored 4.79, 4.3, 4.03, and 3.82 respectively. Similarly, statements designed to assess people related aspects such as "*Differences between individuals are respected in my team*", "*I am empowered to make decisions by myself*", and "*Implementation teams are self-organized*" scored quite high with 4.0, 3.83, and 3.78 respectively. Regarding to customer value and transparency, some statements evidenced the good status of these elements. "*All features delivered are relevant to our customers*" and "*Projects complete with a satisfied customer*", scored 3.67 and 3.71 respectively. "*All team members participate in Stand-up meetings*" (4.14), "*It is clear to me in which Release feature delivery belongs*" (3.63), "*My team has visibility of the plans and roadmaps that they are contributing to*" (3.37) got also good scores.

## 5. Discussion

Although Lean Software Development was initially mainstreamed by Poppendieck's work [4], recently much more diversity has been introduced [13]. The combination of Agile and Lean has been questioned in manufacturing [22]. As most of previous work, our study confirms that Lean and Agile thinking are not that dissimilar in a software domain. Thus, software intensive companies select those elements from Agile and Lean that suit well in a software development context, creating their own interpretation of Lean Software Development. What EB Wireless Segment considers works better for them includes a combination of elements from ASD, to achieve flexibility, Lean thinking, to scale Agile and make software development processes more efficient, and good practices of software engineering. The "*softer*" side of software development has been highlighted during last years [35]. So, it may be a misconception that Agile and Lean Software Development serve as easy excuses for irresponsibility with no regard to the engineering side of the software discipline. However, it is interesting to observe that technical and human aspects were balanced in the survey. Thus, EB Wireless Segment recognizes the importance of the social aspect of software development, but without forgetting engineering practices, similarly important. For example, from a technical perspective practices such as continuous integration, concurrent development and testing, test automation and static code analysis were considered as essential during the focus groups.

Comparing our findings with previous studies several similarities can be found. The network of product owners for managing customer value is also discussed in [12]. Similarly to Middleton's work [29, 30, 11], we found WIP limits as an important element for achieving flow in a software development context. Actually, Kanban is more and more used and helps to visualize and manage WIP, and create a pull (versus a push) culture. Also, as mentioned in previous studies [12], continuous integration and test automation appeared as keys to support flow by frequent and smaller builds. Also, the concepts of transparency and

streamlined communication, emphasized in previous studies [30, 9, 10, 11, 12], were found essential. What is different in the case of EB Wireless Segment is its strong reliance on JIRA for supporting transparency. EB uses JIRA not just for tracking the status of the product (features and user stories) but also for making visible everything in its product development cycle without restrictions in sharing information, from development tasks to business goals and strategy.

Based on the results of the assessment, some lesson can be learnt. On the one hand, we learnt that setting up Agile and Kanban teams appears as a relatively simple task. Aspects such as self-organization and focusing on customer value got also good scores in the assessment. However, involving business management and achieving company level agility remains still as challenges, which challenges also to achieve short feedback loops and prevents flow. Thus, although Lean thinking supports in scaling Agile from development teams to software development units, embracing also other parts of the organization such as hardware was found still as a challenge.

## 6. Conclusions, limitations and future work

This study examines how Lean and Agile are applied in software intensive organizations by analyzing the case of EB Wireless Segment. The results show a balance between social and technical aspects, where Lean thinking not just guides the way of scaling Agile but also adds new elements to software development processes. In addition to practices well established in Agile over the past few years, Lean thinking brings elements such as WIP limits and the pull concept, mainly by the usage of Kanban, the concept of removing waste, the principle of see-the-whole and a greater emphasis on transparency and collaborative development. On the other hand, whilst setting up teams and establishing self-organization were relatively easy to achieve, this study found it was much more challenging to scale flexibility, reduce waste and involve business management in the Agile and Lean way of working.

Our study suffers the common limitations in case studies [32]. Since the study was conducted in a single company, it is possible that our observations cannot be fully generalized to other settings (external validity). However, Lean Software Development is a relatively unexplored phenomenon. Since EB has showed some success in its transformation, our results can bring new insights and guide organizations pursuing a similar endeavor. It may be also questioned whether the Lean implementation of EB Wireless Segment is actually fully conformant with Lean thinking. Our study does not focus on epistemological concerns. However, we are convinced EB is consciously trying to transform itself to a Lean organization. Regarding data collection and analysis methods, in an attempt to minimize threats to construct validity, a material walkthrough workshop was organized at the beginning of the research, which helped researchers and participants to speak the same language (construct validity). Moreover, multiple data sources (process documentation and focus groups) were used and participants' triangulation lent the data greater accuracy and validity. As future work, elements identified as important, as well as strengths and challenges, are topics of interest to be studied in greater detail. We are conducting similar studies in other companies to enable analytical generalization and extend results to cases with common characteristics.

## 7. References


[1] T. Dybå, and T. Dingsøyr, "Empirical studies of agile software development: a systematic review", Inf. Softw. Technol., vol. 50, pp. 9-10, 2008.

[2] P. Rodríguez, J. Markkula, M. Oivo, K. Turula, "Survey on Agile and Lean Usage in Finnish Software Industry", in ESEM, pp. 139-148, 2012.

[3] K. Vilkki, "When agile is not enough", in 1st Int. Conference on Lean Enterprise Software and Systems, 2010.

[4] M. Poppendieck, and T. Poppendieck, Lean software development: An agile toolkit, Addison-Wesley Professional, 2002.

[5] P. Freeman, Lean concepts in software engineering, IPSS-Europe International Conference on Lean Software Development, Stuttgart, Germany, pp. 1-8, 1992.

[6] K. Vilkki, and H. Erdogmus, "Point/Counterpoint", IEEE Software, vol. 29, no. 5, pp. 60-63, Sept.-Oct. 2012.

[7] X. Wang, K. Convoy, and O. Cawley, "Leagile software development: An experience report analysis of the application of lean approaches in agile software development". Journal of Systems and Software, vol. 85, no. 6, pp. 1287-1299, 2012.

[8] P. Rodríguez, J. Markkula, M. Oivo, and J. Garbajosa, "Analyzing the drivers of the combination of lean and agile in software development companies", 13th Int. Conference on Product-Focused Software Development and Process Improvement, 2012.

[9] M. Mehta, D. Anderson, D. Raffo, "Providing value to customers in software development through lean principles", Software Process Improvement and Practice, vol. 13, no. 1, pp. 101-109, 2008.

[10] B. Staats, D. Brunner, and D. Upton, "Lean principles, learning, and knowledge work: evidence from a software services provider", Journal of Operations Management, vol.29, no. 5, pp. 376-390, 2011.

[11] P. Middleton, D. Joyce, "Lean Software Management: BBC Worldwide Case Study", Engineering Management, IEEE Transactions on , vol. 59, no. 1, pp. 20-32, 2012.

[12] P. Rodríguez, K. Mikkonen, P. Kuvaja, M. Oivo and J. Garbajosa, "Building lean thinking in a telecom software development organization: strengths and challenges", Conference on Software and System Process (ICSSP), 2013.

[13] C. Ebert, P. Abrahamsson A, and N. Oza, "Lean Software Development", IEEE Software, vol.29, no.5, pp. 22-25, 2012.

[14] T. Ohno, Toyota production system: beyond large-scale production. Portland, Oregon: Productivity Press, 1988.

[15] J.P. Womack, and D.T. Jones, Lean thinking. Simon and Schuster, New York, 1996.

[16] J. K. Liker, The Toyota way: 14 management principles from the world's greatest manufacturer. McGraw-Hill, 2004.

[17] M. Venables, "Boeing: going for lean", Manufacturing Engineer, vol. 84, no. 4, pp. 26-31, 2005.

[18] C.S. Kim, D. A. Spahlinger, J. M., Kin, R. J. Coffey, and J. E. Billi, "Implementation of lean thinking: one health system's journey", Jt Comm J Qual Patient Saf., vol. 35, no. 8, pp. 406-413, Aug. 2009.

[19] N. Tokatli, "Global sourcing: insights from the global clothing industry—the case of Zara, a fast fashion retailer", J. of Economic Geography, vol. 8, no. 1, pp. 21-38, 2007.

[20] B. Bremer, and D. Dawson, "Can anything stop Toyota?", Business Week, vol. 117, Nov. 17, 2003.

[21] M. Poppendieck, M. Cusumano, "Lean Software Development: A Tutorial", IEEE Software, vol.29, no5, pp.26-32, 2012.

[22] J. Ben Naylor, M. Naim, D. Berry, "Leagility: integrating the lean and agile manufacturing paradigms in the total supply chain", Int. Journal of Production Economics, vol.62, no.1, pp.107-118, 1999.

[23] M. Poppendieck, "Principles of Lean Thinking", OOPSLA Onward!, November, 2002.

[24] J. Coplien, and G. Bjornwig, Lean architecture for agile software development. John Wiley & Sons Ltd. 2010.

[25] H. Jonsson, "Lean Software Development: a systematic literature review", IDT Mini-conference on Interesting Results in Computer Science and Engineering, 2012.

[26] D. Anderson, Kanban: Successful Evolutionary Change for Your Technology Business, Blue Hole Press, 2010.

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

[28] K. Petersen, and C. Wohlin, "Measuring the flow in lean software development", Software: Practice and Experience, vol. 41, no. 9, pp. 975-996, 2010.

[29] P. Middleton, "Lean software development: two case studies", Software Quality J., vol.9, no.2, pp.41-252, 2001.

[30] P. Middleton, A. Flaxel, A. Cookson, "Lean software management case study: timberline", Extreme programming and agile processes in software engineering (XP 2005).

[31] J. Trimble, and C. Webster, "From Traditional, to Lean, to Agile Development: Finding the Optimal Software Engineering Cycle". 46th Hawaii International Conference on System Sciences (HICSS), pp. 4826-4833, 2013.

[32] P. Runeson, and M. Höst, "Guidelines for conducting and reporting case study research in software engineering", Emp. Software Eng., vol. 14, no 2, pp. 131-164, 2009.

[33] J. Kontio, J. Bragge, and L. Lehtola, The focus group method as an empirical tool in software engineering. Guide to advanced empirical software engineering. Springer, 2008.

[34] M. B. Miles, and A. M. Huberman, Qualitative data analysis: an expanded sourcebook. Sage Publications, 1994.

[35] A. Cockburn, and J. Highsmith, "Agile software development, the people factor", Computer, vol. 34, no. 11, pp. 131-133, 2001.

## Appendix A. A fragment of the Lean and Agile assessment survey (including coding)

| **ASD and Lean Thinking elements** | | | **Survey statements** (20/97 statements) |
|---|---|---|---|
| **Lean Principles** | | **Themes** | |
| **Elements already in ASD** | *Value and value stream* | Product owners | Business Managers define and prioritize Feature/Releases for Implementation Teams. |
| | | Product backlog | Features are clear and concrete for establishing User Stories without problems. |
| | | Flexibility | There is room to make changes in releases for including latest customer requests. |
| | | Speed | Lead time for business opportunity and delivery is short enough. |
| | *Flow and Pull* | Continuous planning/execution | Plans (such as Release/Sprint plans) are updated at the end of every Sprint.<br>Developers integrate code multiple times per day. |
| | | Synchronization | All Implementation Teams are synchronized in the same iteration schedule. |
| | | Estimations | Implementation Team's velocity is measured and used as input for Release Planning. |
| | *Perfection* | Retrospectives | My team changes the way of working based on the results of retrospectives |
| **Elements brought by Lean Thinking** | *Value and value stream* | Customer value | Everything that we do in the organizations is adding value to our customer. |
| | | Seeing-the-whole | Teams see the big picture and take responsibility of the product as a whole. |
| | | Eliminating waste | No unnecessary handover is performed along the development process. |
| | *Flow and Pull* | Kanban | Kanban runs within a 4 week release cycle |
| | | WIP limits | There is not excess WIP inventory [also a source of waste] |
| | | Discontinuities | Decision making outside my Implementation Team don't causes delays in my work. |
| | *Perfection* | Root cause analysis | The same issues/problems are not repeated from Sprint to Sprint. |
| | | Learning | We actively share learning and best practices in the organization |
| **Other key elements** | *Transparency* | Feedback loops | Feedback loops between implementation teams and business owners are short enough |
| | | Tools support | Information in JIRA /wikis or similar tools can be understood by all team members. |
| | *People* | Respect people | Differences between individuals are respected in my team. |