Adapting the Lean Enterprise Self-Assessment Tool for the Software Development Domain

Teemu Karvonen[1], Pilar Rodríguez[1], Pasi Kuvaja[1], Kirsi Mikkonen[2], Markku Oivo[1]
[1]Department of Information Processing Science, University of Oulu, Finland
[2] Ericsson R&D Center Finland, Jorvas, Finland
e-mail: teka@iki.fi, pilar.rodriquez@oulu.fi, pasi.kuvaja@oulu.fi, kirsi.mikkonen@ericsson.com, markku.oivo@oulu.fi

Abstract **Lean principles have attracted the attention of software development companies due to their potential to improve competitiveness. However, the application of such principles in the software domain is still in its infancy. This paper presents a proposal for adapting the Lean Enterprise Self-Assessment Tool (LESAT) to guide the transformation of software development companies toward Lean. LESAT, developed by the Lean Advancement Initiative (LAI) at the Massachusetts Institute of Technology (MIT), has been widely used in other domains. In this study, concepts and expressions of LESAT were analyzed and mapped to software development following the ISO/IEC 12207 standard. Seven assessment items concerning life-cycle processes were modified from the original LESAT. The modified LESAT for software was compared with a lean assessment approach called "*Lean amplifier,*" which has been developed and successfully used in practice by Ericsson R&D in Finland. The results indicated that LESAT may complement lean assessment in the software domain at enterprise level, involving the entire value stream. Moreover, they clearly emphasized the role of leadership in the transformation.**

Keywords: **lean; lean software development; lean transformation; assessment; LESAT; enterprise; software engineering**

I. INTRODUCTION

Ever since lean principles were first used in elucidating the characteristics of the Toyota Production System (TPS) [1], [2], [3], they have fascinated enterprises and researchers around the world. The concept of "lean" has evolved and expanded since the groundbreaking results of the International Motor Vehicle Program (IMVP) [3]. Lean software development (LSD) is the adaptation of lean for software development companies [4], [5]. LSD is often associated with agile methods [6]. However, whereas agile methods mainly focus on team level development activities, lean is often viewed as a platform upon which to scale agile methods because it focuses on the whole *value stream* and on enterprise-level processes and behavior. According to Sutherland "*lean software development views all agile methods as valid, proven applications of Lean thinking to software*" [5, p. xvii]. However, how to proceed when adopting lean in software development companies is still not well understood [7], [8]. Moreover, no structured assessment framework exists that supports the transformation toward LSD, at least in the public domain.

In this paper, we present a study for adapting the Lean Enterprise Self-Assessment Tool (LESAT) [9] for use in software-development organizations. LESAT was developed by the Lean Advancement Initiative (LAI) at the Massachusetts Institute of Technology (MIT) [10]. LESAT aims to guide companies implementing lean by assessing their current status based on lean capabilities, as well as setting targets for future steps. LESAT, originally developed in cooperation with the aerospace defense industry, has been used in other domains such as health care and government organizations. We think that the assumptions underlying LESAT that are based on the MIT's extensive knowledge and experience of lean could also be suitable for gauging software development companies' progress toward lean.

In this study, underlying assumptions from product manufacturing are reviewed, in addition to the language of LESAT, and translated into a domain language and processes more appropriate to software. Seven assessment items in LESAT's life-cycle processes section were modified. The items were mapped to ISO/IEC 12207 standard, and concepts and expressions for lean and agile in relation to software development were constructed. Finally, LESAT for software was evaluated by comparing it with an approach designed by Ericsson to assess its own lean transformation called *Lean Amplifier*. Twelve core principles of lean were applied in the evaluation. The results indicated that LESAT for software could complement LSD organizations' assessments, especially at the enterprise level. They also clearly emphasized the role of leadership as a key enabler of the lean enterprise transformation. However, the life-cycle processes section of LESAT still requires in-depth modifications to adapt it to the characteristics of the software-development domain.

The structure of the paper is as follows: Section II presents related work. Section III introduces the architecture and the main features of LESAT. Section IV presents the adaptation of LESAT for software, including some examples clarifying how the adaptation was undertaken. Section V presents the evaluation of LESAT for software, and section VI discusses the implications of this study. Finally, section VII presents the conclusions, limitations of the study and suggestions for future work.

II. Related Work

A. Lean and lean software development

Toyota is perhaps the most well-known company to apply lean in its TPS [2]. However, even Toyota cannot be considered to have completely mastered lean [1]. Lean principles, as introduced by Womack and Jones [12], include *value*, *value stream*, *flow*, *pull,* and *perfection*. Accordingly, lean seeks the most efficient methods on the basis of maximizing value, minimizing waste, and embracing a continuous improvement mentality. According to lean, value is defined as everything for which a customer is willing to pay, and waste refers to everything that absorbs resources but yields no value. Value stream seeks an optimized end-to-end collection of actions for bringing the product to the customer. Flow refers to the organization of activities as a continuous flow by eliminating discontinuities and unnecessary steps (waste) in the value stream. Finally, pull implies producing only what is really needed by making the customer's needs the primary decision driver.

The application of lean has spread and today is widely applied in different domains such as the retail clothing industry [13], health care [14], and the aerospace industry [15]. Zara, the Spanish fashion company, has considerably reduced lead time via a business model based on collecting and sharing input from customers daily and using lean inventories [13]. In the domain of heath care, Young and McClean have reported improvements when applying lean, "*especially in the areas of safety, delay and cost-effective delivery of care*" [14, p. 383]. As early as the 1990s [16], researchers began exploring the potential of lean to make software-development processes innovative, responsive to business changes, and economically efficient. LSD has began to attract attention again more recently [5], [17], [18], primarily due to the expanding industry need for scaling agile software development.

LSD has been mainly promoted by the agile community, being initially considered just another agile method. However, today LSD is acquiring its own identity [7]. Poppendieck considered lean as a "*platform upon which to build agile software development practices*" [19, p. 1]. Coplien and Bjornwig argued that although agile and lean have fundamental differences, they also complement each other [20]. Although lean, as contemporarily understood, is a new topic in software development, there is already evidence of the industry's interest in adopting this approach [21], [7].

While lean principles [12] are under debate in a knowledge activity as software development [22], the seven principles compiled by Poppendieck and Poppendieck [5] are widely acknowledged. These principles are 1) *eliminate waste*, doing only what adds customer value without delays; 2) *amplify learning* using frequent feedback loops; 3) *delay commitment*, deciding as late as possible; 4) *deliver as fast as possible*, responding to real customers' needs; 5) *empower the team*, providing an expert workforce and delegating responsibility to workers; 6) *build integrity in*, focusing on quality from the start and avoiding having to correct defects by testing the product at the end of the development process, and 7) *see the whole*, focusing on achieving an overall goal.

This work utilizes the definitions and views of Womack and Jones [12] on lean, and Poppendieck and Poppendieck's perspectives of LSD [5] for adapting and evaluating LESAT for software.

B. Lean transformation and assessment

Organizational change refers to "*the process by which organizations move from their present state to some desired future state to increase their effectiveness*" [23, p. 88]. Changes in organizations can happen at different levels, from automating manual tasks to rethinking the nature of the organization and defining new business models. Evidently, risks associated with organizational change increase as the scope of the transformation broadens. Lean highlights the importance of considering the whole enterprise in the transformation. Thus, a lean enterprise uses lean not just in some areas but in everything it does [3]. Consequently, adopting lean is often part or the organization's strategy [24], leading to transformation of the enterprise and impacting disciplines such as sales, human resources, etc.

Assessments have been identified as essential when transforming toward lean [10]. Evaluating the enterprise's performance or its operational efficiency is not enough for guiding the lean transformation. Assessment approaches that reflect internal organizational capabilities and provide practical information for improving the organization's internal processes are also considered necessary. Doolen and Hacker [24] previously reviewed the existing assessment methods for guiding lean transformation. However, identified assessment methods focus on specific practices in a manufacturing environment and cannot be directly applicable to software development organizations.

In the software engineering domain, CMMI-DEV [25] and ISO/IEC 15504 [26] have become a standard way to assess an organization and its capabilities. CMMI-DEV V1.3 also includes support for agile methods. However, they have not been designed for conducting lean transformations and are hardly applicable. Assessment methods closer to LSD have emerged during recent years for evaluating the agile transformation of software-development organizations [27], [28], [29], and [30]. However, again, these methods did not explicitly consider lean aspects.

This study aims to help fill the knowledge gap on approaches for assessing the transformation toward lean in software-development organizations. Considering the research efforts of the LAI (http://lean.mit.edu/) of the MIT, we propose the adaptation of one of its models, LESAT, for software development.

The LAI at the MIT proposed a holistic approach composed of a Lean Enterprise Model (LEM) [31], the enterprise Transition-To-Lean (TTL) roadmap [32], and the LESAT [9]. LEM is a general framework that provides an overall vision of lean. The TTL roadmap provides a practical framework for assisting organizations in their effort to transform into lean. In the domain of LSD, LAI-MIT's TTL roadmap has inspired the recent work of Kuusela and Koivuluoma [8], who presented a lean transformation framework for software-intensive companies. Kuusela and Koivuluoma stressed that "*At intervals, lean assessments are useful to get a picture how lean transformation is progressing. From time to time it is vital to check if corrective actions are needed, major or minor ones*." [8, p. 4]. LESAT is the LAI-MIT's solution for assessing the leanness of an enterprise and its capacity to change according to lean principles. We believe that LESAT, originally designed in the domain of the aerospace industry and adapted to other domains such as government organizations [33] and health care [34], can also provide the foundations for assessing lean transformations of software development companies.

III. LESAT IN A NUTSHELL

The purpose of LESAT is to assess an enterprise's current status in terms of lean (capability), as well as to set targets for a desired future state. LESAT focuses not only on lean characteristics, but also on lean behavior and, especially, leadership for initiating and fostering the journey of transforming into a lean enterprise. LESAT is composed of assessment matrices that contain those aspects that are important in lean transformation. The capability maturity model (CMM) [25], which was developed by the Software Engineering Institute, was considered in the design of LESAT [10]. However, "w*hereas the CMM/CMMI models focus on systems and software development, LESAT considers all processes of an enterprise, encompassing all enterprise practices with an emphasis on leadership and integrative management processes*" [10, p. 19]. LESAT is organized into three assessment sections as follows:

1) A *lean transformation/leadership section*, which contains processes and leadership attributes nurturing the transformation toward lean principles and practices.

2) A *life-cycle processes section*, which focuses on processes responsible for product creation from conception to post-delivery support.

3) An e*nabling infrastructure section*, which assesses those processes that provide and manage the resources for enabling enterprise operations.

Fig. 1 illustrates the LAI's view on the lean enterprise, as well as the relationship between the processes. The enterprise leadership processes provide the basis on which the enabling infrastructure processes and finally the life-cycle processes are built on.

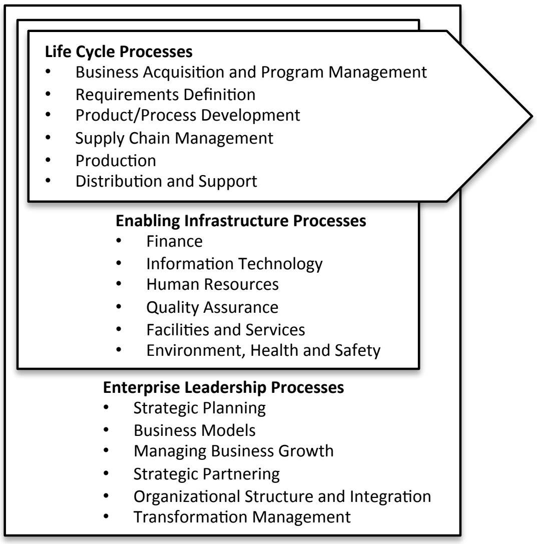

Figure 1. Process architecture view of lean enterprise [10]

Fig. 2 shows LESAT's maturity matrices. LESAT assesses 54 lean practices (items). Each item contains a description of the lean practice, in addition to lean indicators and descriptions of five capability levels for guiding the assessment. The capability levels range from least capable (Level 1) to world class (Level 5).

Organization of LESAT Maturity Matrices

Section I – Lean transformation/leadership
I.A Enterprise's strategic planning (three lean practices)
I.B Adopt a lean paradigm (four lean practices)
I.C Focus on the value stream (four lean practices)
I.D Develop a lean structure and behavior (seven lean practices)
I.E Create and refine a transformation plan (three lean practices)
I.F Implement lean initiatives (two lean practices)
I.G Focus on continuous improvement (five lean practices)
Section II – Life-cycle processes
II.A Business acquisition and program management (four lean practices)
II.B Requirements definition (two lean practices)
II.C Develop product and process (three lean practices)
II.D Manage the supply chain (three lean practices)
II.E Produce the product (two lean practices)
II.F Distribute and service the product (four lean practices)
Section III – Enabling infrastructure processes
III.A Lean organizational enablers (five lean practices)
III.B Lean process enablers (three lean practices)

Figure 2. Organization of LESAT's maturity matrices [10]

An example of LESAT's lean practice is presented in Fig. 4. This example assesses the enterprise's lean capability to provide customer support to become an integrated part of the customer's business network (in accordance with the principle of considering the whole value stream). The most appropriate capability level is decided upon by using lean indicators and capability-level descriptions for the specific practice. The C checkbox is marked with the current state of enterprise capability, and the D checkbox is marked with the desired future state. The gap between the current and the desired states can be calculated and presented in a numerical/graphical form and exploited for planning and implementing process-improvement initiatives.

IV. Adaptation of LESAT for Software

This study is based on LESAT version 1.0, which was publicly available at the time of initiating this study [35][1]. First, the concepts underlying LESAT and lean principles were analyzed. Then, LESAT was adapted to the software development domain, including the addition of comments and mapping for ISO/IEC 12207 processes [11] and lean and agile software-development practices and principles. Finally, LESAT for software was evaluated as described in section V.

A. Conceptual analysis of LESAT

When analyzing the original version of LESAT, the following four key concepts were identified as the most frequently cited terms: a) extended enterprise, b) value stream, c) stakeholder, and d) suppliers (supplier chain, supplier network, and key suppliers). Therefore, we investigated whether the same concepts apply in the software domain.

Leveraging extended enterprise emphasizes the capability to extend lean to cover the entire value stream. LESAT's glossary [9] defines an extended enterprise as "*all businesses along the value stream that contribute to providing value to a customer*," and value stream as "*the specific activities required to design, order, and provide a specific product, from concept to launch, order to delivery, and raw materials into the hands of the customer*". Closely related to the extended enterprise, the concept of an *ecosystem* seems to be quite popular in contemporary software business language [37, 38]. An extended enterprise can be also defined as consisting of a regular business-to-business chain of suppliers and customers.

Value stream has a very specific meaning in the lean literature (e.g., Womack & Jones's lean principles [12]). However, the implementation of the value stream is always context specific. In the software domain, Messerschmitt and Szyperski [38] defined the software value chain as consisting of the following types of organizations: application software provider, application service provider, infrastructure service provider, system integrator, business consultant, industry consultant, infrastructure software supplier, end-user organization, and information content supplier. All of the aforementioned types of organizations may be seen as potential organizations in the extended enterprise in the software industry.

LESAT's glossary defines stakeholders as "*all those who have an interest in an organization, its activities, and its achievements. These may include customers, partners, employees, shareholders, owners, government, and regulators.*" The definition of stakeholders can again vary depending on the context. Rajala, Rossi, and Tuunainen [39] listed typical stakeholders in the software business as development partners, distributors, employees, financiers, management, shareholders, subcontractors and suppliers.

Overall, these key concepts are also commonly applied in the software domain literature. However, some content and terminology needed to be revised to adapt LESAT from the aerospace to the software-industry context as described in the next section.

B. LESAT's adaptation

Based on the original LESAT, the review of the literature, and the ISO/IEC 12207 standard, the adaption of LESAT to the software domain required the following actions:

1) Seven of the 54 assessment items in the original LESAT were modified to conform to terminology and processes in the software domain (i.e., 87% of assessment items remained the same as the original LESAT since they were considered suitable for software development). The changes mainly affected the life-cycle processes section of LESAT due to the major differences (production and manufacturing processes) between the aerospace industry and the product-development characteristics of the software industry. The sections lean transformation/leadership and enabling infrastructure were almost directly applicable to the domain of software development.

2) The life-cycle processes section was mapped to ISO/IEC 12207 (software life-cycle processes) [11]. The ISO/IEC 12207 standard was applied because it is a widely used standard in the software-engineering domain. Each assessment item in LESAT was revised using the ISO/IEC 12207 standard definitions and mapped to the appropriate software process terminology. Thus, production and manufacturing processes were translated to software implementation processes as described in clauses 7.1.1 (Software Implementation Processes) to 7.1.7 (Software Qualification Testing Process).

3) Several comments for the software industry and for lean and agile software development methods were added to clarify the assessment of lean practices, indicators, and capability levels from a software development point of view. For example, Scrum was presented in the LESAT section I.D as an optional way to develop a lean structure and behavior in a software organization.

Fig. 3 presents an example of the modification of an assessment item and its mapping to ISO/IEC 12207. The purpose of this assessment item was to evaluate the enterprise's capability to utilize the lean principle of continuous learning and perfection in production and manufacturing processes. For its adaptation to the software-development domain, software-implementation knowledge and processes were introduced to demonstrate corresponding lean behavior.

[1] Recently, the LAI published LESAT 2.0, which is only available to the LAI's community members [36].

Process	**Enterprise's practice**	**Enterprise's characteristics**	
II.E **Produce product**	II.E.1 Utilize **production knowledge** and capabilities for competitive advantage	Strategic leveraging of **manufacturing** capability	
	II.E.2 Establish and maintain a lean **production system**	Defect-free **production** pulled by the customer	

ADAPTATION

↓

II.E **Implement software**	II.E.1 Utilize **software-implementation knowledge** and capabilities for competitive advantage	Strategic leveraging of **software-implementation** capability	ISO/IEC12207: Clause 7.1.1 - 7.1.7
	II.E.2 Establish and maintain lean **software-implementation processes**	Defect-free **development** pulled by the customer	ISO/IEC12207: Clause 7.1.1 - 7.1.7

Figure 3. Example of LESAT modification and mapping

An example of the modifications made in the assessment of the enterprise's capability levels is presented in Fig. 4 (original LESAT) and Fig. 5 (LESAT for software). This example shows how post-delivery services must be considered when adapting LESAT to the software domain.

The term "*spares*" did not fit in the terminology of immaterial software products. It was replaced with the term "proactive maintenance approach," i.e., improved released product quality and reduction of dedicated resources for maintenance work, as a corresponding indicator for the lean behavior of an enterprise.

Lean practice	II.F.4 Provide Post Delivery Service, Support, and Sustainability. *Providing customer solutions*
Lean Indicators (Examples)	• Customer feedback is proactively maintained and used to predict any emerging service issues and enhance future designs. • Spares levels are reduced in line with short predicable lead times for replacement spares.
Level 1 ***□C □D***	High level of spares necessary because of unknown failure rates and long lead times for spare replenishment.
Level 2 ***□C □D***	Collection of data on failure trends permits both determination of service interval points for preventative maintenance and a reduction of spare part levels.
Level 3 ***□C □D***	The enterprise is increasingly involved in addressing customer maintenance solutions. Spare levels are reduced through common platforms; root cause analyses, are fed back into product design.
Level 4 ***□C □D***	The enterprise is part of the customer's maintenance solution by ensuring availability through replacement of critical components before failure.
Level 5 ***□C □D***	The enterprise has become part of customer's business solution via warranting of product performance.

Figure 4. Example of original LESAT version 1.0

Lean practice	II.F.4 Provide Post Delivery Service, Support and Sustainability. Providing customer solutions.
Lean Indicators (Examples)	• Customer feedback is proactively maintained and used to predict any emerging service issues and enhance future designs. • Dedicated resources (reserve buffer) for postdelivery support are reduced in line with short predicable lead times for responding customer requests.
Level 1 ***□C □D***	Resources for maintenance work are kept high or new product development is constantly interrupted because of unexpected errors in released product.
Level 2 ***□C □D***	Error history data is collected and used for more accurately estimating need for dedicated maintenance and error correction resources.
Level 3 ***□C □D***	The enterprise is increasingly involved in addressing customer maintenance solutions. Resources dedicated for maintenance can be reduced by root cause analyses, adaptive design, and common platform solutions.
Level 4 ***□C □D***	The enterprise is part of the customer maintenance solution by ensuring availability through updating critical components before failure.
Level 5 ***□C □D***	The enterprise has become part of customer's business solution via warranting of product performance.

Figure 5. Example of modified LESAT for software version

The complete version of LESAT for software, including all changes made in the adaptation, can be downloaded via the following link: http://goo.gl/9Odqv. Furthermore, the summary sheets to facilitate the assessment and record the current and the desired capability level for each lean practice contained in the LESAT matrices are available for download via http://goo.gl/Iis5k.

V. Evaluation of LESAT for Software

To evaluate its suitability, LESAT for software was compared with the approach used by Ericsson for guiding its lean transformation. Ericsson is a multinational telecommunications company in the middle of transforming its processes according to lean principles. Therefore, Ericsson can be considered as an early adopter of LSD. *Ericsson's lean amplifier* was internally designed for gaining insights into how teams are evolving in Ericsson's agile and lean ways of working. Although Ericsson's lean amplifier has been specifically designed to meet Ericsson's needs, it was selected as a practical benchmark for the software domain due to its strong background in applying lean principles in a real software-development context.

A. Comparison

The objective of the evaluation was twofold: a) to evaluate how LESAT can support the lean transformation of software-development companies and b) to gather insights for further development of LESAT for software. Table 1 summarizes the main characteristics of both approaches.

Table 1. Comparison of LESAT for software and Ericsson's lean amplifier

	LESAT for software	Ericsson's lean amplifier
Assessment method	Capability maturity model Self-assessment by the leadership of an enterprise	Assessment based on a set of statements discussed in focus groups Individual or in-group completion of questionnaires (statements)
Design objective	Assess current capability and set target for desired state. Continual improvement in the capability of the enterprise to meet its lean objectives	Identify strengths and weaknesses to agree on improvement areas and to help build a growth plan for individual roles, teams, and the organization and build a continuous improvement mindset at all levels and foster a learning organization
Evaluation scale	Tailored maturity definitions (five levels from one to five)	Fixed evaluation definitions (five levels, from zero to four)
Structure	Divided in three sections based on processes	Divided into eight amplifiers based on roles and units

To further compare LESAT for software and Ericsson's lean amplifier, the number of their respective assessment items by main sections was analyzed. 28 from 54 assessment items in LESAT assess lean transformation and leadership processes (see Fig. 6). This represents a 52% of the assessment items of the whole assessment tool.

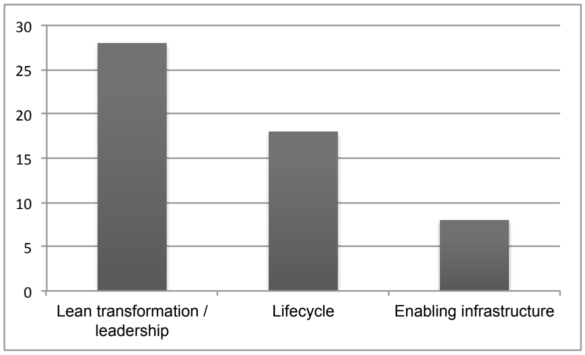

Figure 6. LESAT for software. Number of items in the main sections

As Fig. 7 illustrates, Ericsson's team amplifier, composed by 167 assessment items, takes a more detailed and narrower scope in software-development processes and organizational roles derived from Scrum.

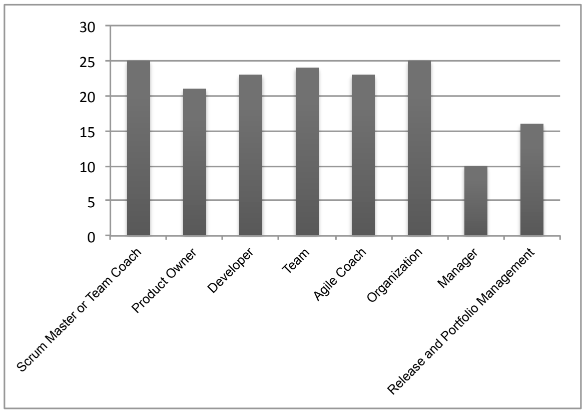

Figure 7. Ericsson's lean amplifier. Number of items in main sections

Although the team and organization amplifiers indicate that the enterprise-level aspect in adopting lean and agile behavior is also considered in the approach of Ericsson, overall the results of the evaluation showed that LESAT for software has an enterprise-wide process scope. Of note, the manager amplifier contains the least number of assessment items in Ericsson's lean amplifier, and LESAT for software has the most assessment items in the leadership/lean transformation section.

B. Distribution of lean principles

LESAT for software and Ericsson's lean amplifier were also compared by mapping each assessment item in both approaches to lean principles as described by Womack and Jones [12] and LSD principles as defined by Poppendieck and Poppendieck [5]. Table 2 shows the degree of coverage of each lean principle in both approaches.

Table 2. Lean principles mapping distribution in LESAT for software and Ericsson's lean amplifier.

		Lean principle	LESAT for software		Ericsson's lean amplifier	
			#[2]	%[3]	#[2]	%[3]
Womack and Jones's principles	1	Value	7	13.0%	11	6.6%
	2	Value stream	9	16.7%	16	9.6%
	3	Flow	9	16.7%	66	39.5%
	4	Pull	3	5.6%	3	1.8%
	5	Perfection	2	3.7%	39	23.4%
Poppendieck and Poppendieck's principles	6	Eliminate waste	1	1.9%	8	4.8%
	7	Build quality in	0	0.0%	26	15.6%
	8	Create knowledge	0	0.0%	31	18.6%
	9	Defer commitment	0	0.0%	5	3.0%
	10	Deliver fast	0	0.0%	17	10.2%
	11	Respect people	1	1.9%	23	13.8%
	12	Optimize the whole	14	25.9%	13	7.8%

The comparison of LESAT for software and Ericsson's lean amplifier with regard to the five lean principles of Womack and Jones showed that they differ in the weighting of the principles "value" and the "value stream." Most of Ericsson's lean amplifier items were identified as either evaluating "flow" or "perfection" principles. These principles can be mapped to either individual or teamwork methods, building a mentality for continuous improvement or challenging the current way of working. Only very few principles were identified for the principle of pull in both approaches.

Considering the seven lean principles according to Poppendieck and Poppendieck [5], it is noteworthy that several principles could not be identified at all in LESAT for software. This is expected since LESAT was not originally designed for software development and for meeting Poppendieck and Poppendieck's views of LSD.

[2] Number of assessment items considering each lean principle. One assessment item can be mapped to one or more lean principles.

[3] Percentage from the total number of assessment items considering each lean principles.

However, this also reveals the clear need of further analysis of LSD principles for improvement of LESAT for software.

Overall, this analysis revealed the weaknesses and strengths of LESAT for software. The weaknesses are clearly in assessing LSD aspects as defined by authors such as Poppendieck and Poppendieck [5], impacting largely on LESAT's life-cycle processes section. The strengths are in a more focused approach for defining value and optimizing the whole enterprise value stream.

VI. Discussion

Although LESAT was designed to meet the needs of the aerospace industry, it has evolved and been successfully used for guiding the lean transformation of companies from different domains [34]. LESAT's design addressed three requirements: "*to assess the degree of leanness of an enterprise and all its core processes,*" "*to provide feedback for improvement and guidance for next steps,*" and "*to be data driven, based on documentable evidence*" [10, p. 5]. It is reasonable to consider that, appropriately adapted, LESAT, which exploits LAI-MIT's extensive knowledge about lean transformation, can also support the lean transformation of software-development companies.

When analyzing the three sections of LESAT, it is remarkable that the sections—enabling infrastructure and enterprise leadership—on which the life-cycle processes are built were almost domain independent in the assessment. Consequently, assessment items such as those designed for assessing *strategic planning*, *strategic partnering*, *transformation management*, or *human resources* were retained as they appear in the original LESAT. However, the life-cycle processes section, which includes processes that directly determine the value provided to customers [10], required in-depth modifications. In this work, we adapted LESAT for software according to the ISO/IEC 12207 standard and what is currently known about LSD.

One interesting aspect is the strong focus of LESAT on assessing how value and the value stream are managed. This is important because lean is mainly about how to make organizations deliver as much customer value as possible [40]. Value-Based Software Engineering (VBSE) recognizes the difficulty and challenges on defining value in the software-engineering context [41]. Nowadays, understanding value in software is a much-debated topic with ongoing studies. Devising a definition of value for software development is beyond the scope of this work. Therefore, in LESAT for software, the concept of value conforms to that in LESAT's glossary: "*Value - A product or service's capability provided to a customer at the right time, at an appropriate price, as defined in each case by the customer"* [9, p. 75]. Assessment items related to value and the value stream will continue evolve over time as the definition of value and the concept of lean continue to be expanded upon in the software domain.

It should be stressed that lean assessments should include both tactical and strategic components [24]. LESAT highlights the importance of considering the whole value stream when transforming toward lean and includes assessment items for all business along the value stream. Thus, LESAT for software can guide the transformation of the whole enterprise towards a lean enterprise and not just some areas of the organization. The comparison provided evidence that Ericsson's lean amplifier has been specifically designed for identifying strengths and weakness in LSD as nowadays understood. Thus, Ericsson's approach broadly encompasses all of the principles of LSD as described by Poppendieck and Poppendieck [5] but has a narrower scope, mainly focusing on software-development processes. In the case of LESAT, as it was not designed for the specific context of software, Poppendieck and Poppendieck's principles are hardly enclosed.

Finally, LESAT for software emphasizes the role of leadership. Leadership is essential because it provides the resources needed to eliminate waste and achieve a flowing value stream. Moreover, work on lean transformation has emphasized that the evolution toward a lean enterprise needs both top-down and bottom-up management strategies [8]. The leadership/lean transformation section of LESAT for software can help to initiate and to foster the transformation into a lean software enterprise.

VII. Conclusion, Limitations, and Future Work

This paper presented a preliminary study of the adaptation of LESAT, designed for the purposes of the aerospace industry, to guide the lean transformation of software-development companies. The modifications followed the ISO/IEC 12207 standard. In addition, agile and lean software development principles were incorporated. LESAT for software was compared with the contents of the assessment approach that Ericsson designed to guide its lean transformation. The results revealed differences in the content and the scope of both approaches.

Although LESAT for software required comprehensive modifications, especially in the life-cycle processes section, its focus on leadership and on integrative management processes may enable and promote the lean transformation of software enterprises considering the whole value stream. In particular, its scope at the enterprise level and leadership processes, including strategic planning and infrastructure processes, can potentially complement lean assessments in the software domain.

This paper presents an initial proposal for adapting LESAT for software. However, existing evaluation is still limited and more empirical studies, in which LESAT for software is applied in individual company cases, are needed to validate the tool and make a more comprehensive analysis of its utility.   Moreover, further development of LESAT for software, especially for adapting its life-cycle processes section to LSD, where lean is frequently intertwined with foundations derived from agile methods, is considered necessary. Therefore, it is expected that LESAT for software will continue evolving in both content and process, as a natural result of its continuous usage and a deeper understanding of the phenomenon of LSD and the concept of value in software development.

ACKNOWLEDGEMENT

This article is based on 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 and the Academy of Finland, also supported the work.

REFERENCES

[1] T. Ohno, Toyota production system : Beyond large-scale production. Cambridge (Mass.): Productivity Press, 1988.

[2] J. K. Liker, The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer. McGraw-Hill, New York, 2004.

[3] J. Womack, D. Roos and D. Jones, The machine that changed the world, New York, NY: Rawson and Associates, 1990.

[4] T. Dybå, and T. Dingsøyr, "Empirical studies of agile software development: A systematic review." Information and Software Technology, Jan. 2008, 50(9-10), pp. 833-859.

[5] M. Poppendieck, and T. Poppendieck, Implementing Lean Software Development. Addison-Wesley Professional; 1 edition, 2006.

[6] J. Highsmith, Agile Software Development Ecosystems, Addison-Wesley, Boston, 2002.

[7] X. Wang, K. Conboy and O. Cawley, "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development, J. Syst. Software, 2012.

[8] R. Kuusela and M. Koivuluoma, "Lean transformation framework for software intensive companies: responding to challenges created by the cloud," Proc. 37th EUROMICRO Conference on Software Engineering and Advanced Applications, pp. 378-382, 2011.

[9] Lean enterprise self-assessment tool – Version 1.0. Cambridge, MA: Lean Aerospace Initiative, MIT, 2001.

[10] D. J. Nightingale and J. H. Mize, "Development of a lean enterprise transformation maturity model", Information Knowledge Systems Management, vol. 3, 2002, no. 1, pp. 15-30.

[11] ISO/IEC. (2008). Standard for systems and software engineering - software life cycle processes. ISO/IEC/IEEE Standard 12207-2008.

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

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

[14] T. P. Young, and S. I. McClean, "A critical look at Lean Thinking in healthcare", Qual Saf Health Care , vol. 17, pp. 382-386, 2008.

[15] T. R. Browning and R. D. Ralph, "Reconceptualizing the effects of lean on production costs with evidence from the F-22 program", Journal of Operations Management, vol. 27, no. 1, pp. 23-44, 2009.

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

[17] P. Middleton and J. Sutton. Lean software strategies: proven techniques for managers and developers. New York, N.Y.: Productivity Press, cop. 2005.

[18] S. W. Ambler. "Scaling agile software development through lean governance", Proc. International Conference on Software Engineering 2009 ICSE Workshop on Software Development Governance, 2009.

[19] M. Poppendieck, "Principles of lean thinking", http://www.leanessays.com/2002/11/principles-of-lean-thinking.html (last accessed June, 28, 2012)

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

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

[22] B. Staats, D. Brunner, and D. Upton, "Lean Principles, Learning, and Knowledge Work: Evi-dence from a Software Services Provider", Journal of Operations Management, vol. 29, no 5, pp.376-390, 2011.

[23] G. R. Jones, Organizational theory, design and change. Pearson Prentice Hall, London, 2004.

[24] T. L. Doolen and M. E. Hacker, "A review of lean assessment in organizations: an exploratory study of lean practices by electronics manufacturers", J. of Manufacturing Systems, vol. 24, no. 1, 2005.

[25] CMMI for Development Version 1.3, Carnegie Mellon, Software Engineering Institute, Pittsburgh, PA.

[26] ISO/IEC, 2003–2006. ISO/IEC 15504 Information Technology – Process Assessment (Parts 1–5).

[27] A. Sidky and J. Arthur, "A disciplined approach to adopting agile practices: The agile adoption framework", Innovations in Systems and Software Engineering, vol. 3, no. 3, pp. 203-216, 2007.

[28] A. Qumer and B. Henderson-Sellers, "A framework to support the evaluation, adoption and improvement of agile methods in practice", Journal of Systems and Software, vol 81, no. 11, pp.1899-1919, 2008.

[29] K. Sureshchandra and J. Shrinivasavadhani, "Adopting agile in distributed development",. Proc. of the 2008 IEEE International Conference on Global Software Engineering, pp. 217-221, 2008.

[30] P. Kettunen, "Systematizing Software Development Agility: Towards an Enterprise Capability Improvement Framework", in press.

[31] Lean enterprise model. Cambridge, MA: Lean Aerospace Initiative, MIT.

[32] Transitioning to a lean enterprise: a guide for leaders – Volume I: Executive overview. Cambridge, MA: Lean Aerospace Initiative, MIT. 2000.

[33] Government lean enterprise self-assessment tool – Version 1.0, Massachusetts Institute of Technology, 2005.

[34] C. L. Hernandez, "Adapting the Lean Enterprise Self Assessment Tool for health care", Thesis (S.M. in Engineering and Management), Massachusetts Institute of Technology, Engineering Systems Division, 2010.

[35] T. Karvonen, "Adaptating the Lean Enterprise Self-Assessment Tool for Software Domain", Thesis (M.Sc in Information Processing Science), University of Oulu, 2011.

[36] LAI releases, LESAT 2.0. Retrieved 4/8, http://lean.mit.edu/news/1096-lai-releases-final-version-of-lesat-20, 2012

[37] J. Bosch, "Software ecosystems taking software development beyond the boundaries of the organization." The Journal of Systems and Software, doi:10.1016/j.jss.2012.03.039, 2010.

[38] D. G., Messerschmitt and C. Szyperski, Software ecosystem : understanding an indispensable technology and industry. Cambridge (Mass.): MIT Press, 2003.

[39] R., Rajala, M., Rossi, and V. K. Tuunainen. "A framework for analyzing software business models." Proceedings of the European Conference on Information Systems 2003

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

[41] S., Biffl, A., Aurum, B., Boehm, H., Erdogmus, P., Grünbacher, Value-Based Software Engineering, Springer, 2006.