

Citation details: Lane, M, Agerfalk, P and Fitzgerald, B (2012) Identifying lean software development values, *Proceedings of ECIS Conference*, Barcelona, June 2012.

IDENTIFYING LEAN SOFTWARE DEVELOPMENT VALUES

Lane, Michael T., LERO¹ - The Irish Centre for Software Engineering, University of Limerick, Limerick, Ireland, michael.lane@ul.ie

Fitzgerald, Brian, LERO¹ - The Irish Centre for Software Engineering, University of Limerick, Limerick, Ireland, brian.fitzgerald@ul.ie

Ågerfalk, Pär J., Department of Information Science, Uppsala University, Uppsala, Sweden, par.agerfalk@im.uu.se

Abstract

Agile software development (ASD) has emerged as a practice-led initiative which offers great promise in improving software productivity. However some confusion exists as to its relationship with Lean Software Development (LSD). Some treat LSD as more or less synonymous with ASD whereas others view LSD as a different concept. The definition and positioning of LSD relative to ASD is important as it gets to the heart of software development as craft versus science debate. The purpose of this paper is to identify core LSD values that 'define' LSD much as the agile manifesto values unified and defined so-called "lightweight methods" for ASD. We posit that LSD is more management philosophy than method and illuminate this through a genealogical analysis of the origins of LSD. We identify principles from various sources on the application of lean thinking in different domains. Synthesizing these principles we derive a candidate set of lean values that characterise LSD. Although immediately valuable to practitioners seeking to apply lean values in agile projects, future research will use this value-set to assess elements of "lean thinking" in the practices and principles of various agile methods so as to facilitate optimal applicability of these methods in a lean context.

Keywords: Lean software development, Agile software development, Lean thinking, Lean values.

¹ This work was supported, in part, by Science Foundation Ireland grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Centre (www.lero.ie)

1 Introduction

Agile software development is an umbrella term used to identify "lightweight" methods. Many such methods emerged to counteract the perceived weaknesses of approaches that were based on the traditional systems development lifecycle or "Waterfall" model. These "lightweight" methods sought to avoid wasteful activities in software development. For example, waterfall phases decry backtracking thus enabling the growth of small issues into major rework items that could be resolved by iterative end-customer reviews of individually completed features(Highsmith, 2002). Research is ongoing to determine the effectiveness of these methods(Abrahamsson et al., 2009).

One particular "lightweight" method was named "Lean development" (LD) (Cohen et al., 2004, Charette, 2003, Highsmith, 2002). This is considered a top-down strategic approach to software development in contrast to other agile methods(Highsmith, 2002). The application of lean thinking to software development approaches was also reported under the labels: "lean hardware/software development" (Hou, 1995) and "lean software development"(Middleton, 1995, Middleton, 2001, Middleton et al., 2005, Middleton and Joyce, 2011, Raman, 1998, Poppendieck and Poppendieck, 2003). This paper uses the term lean software development (LSD) as an umbrella term to refer to methods that claim to apply lean thinking. Analysis of agile software development research, indicates that very few research efforts have been made into LSD (Dingsoyr et al., 2008). Confusion exists about whether LSD should be considered an agile method or a different form of development that has a close relationship to the concept of agile development. Various agile methods recommend practices and techniques that support lean thinking (Hibbs et al., 2009, Poppendieck and Poppendieck, 2003). Windholtz in (Middleton and Sutton, 2005) presents empirical research to support his view that one of the most widely adopted agile methods, extreme programming (XP), embraces lean thinking. However, the aforementioned confusion is evident in a separate contribution from Middleton and Sutton (2005) which proposes that XP emulates craft work rather than lean production. However they note that XP has got potential to become more effective, especially if it incorporates strategic "top-down" thinking and domain orientation. Fowler (2008) maintains that development teams should not consider whether to perform Agile software development or Lean software development. Instead they should aim to apply lean thinking in their implementation of an agile method - "developers should not do agile *or* lean - they should do agile *and* lean". The view is taken here that LSD is not one of the agile methods - it is a broader concept that considers software development from an overall business perspective and as such, may inform the construction, adaptation and application of software methods.

The purpose of this paper is to induce a set of higher-order values that underpin lean principles and practices. An exploration of the origins of lean concepts and their application in different domains identifies a series of principles reported in the literature on the lean paradigm. An hermeneutic interpretive process is followed to synthesize a compilation of these principles into a set of lean values that can be leveraged to explore the relationship between LSD and ASD and contribute to the resolution of the aforementioned confusion around these two approaches. Such an effort is of imminent importance to both practice and academic research. An increase in the conceptual understanding of LSD is clearly required to facilitate rigorous research in the area. Understanding the relationship between LSD and ASD is also required in order to successfully employ "agile *and* lean" as suggested by Fowler (2008). Finally, opportunities for further research that may leverage the lean value set are described and potential applications of the set in industry settings are proposed.

2 Research Method

In his work on learning organizations, Senge (1990) described practices as the activities that practitioners perform whereas principles are the "guiding ideas and insights" that inform them on the rationale behind

the activities that they select and implement. This paper proposes to extend this two-tier perspective to embrace the concept of "values" – the latter being more broad abstract beliefs which govern principles (Figure 1). The three-tier values-principles-practices model is evident in the Agile Manifesto(Fowler and Highsmith, 2001). It outlines a set of values to represent the broad abstract beliefs of the signatories. These values are then supported by a set of principles that offer guidelines to assist in the concrete application of the values. These principles are then applied by practices of actual agile methods.

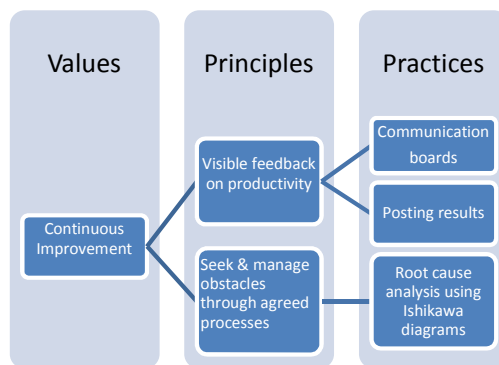


Figure 1. Three-tier representation (see Table 2 for the complete set of values)

Ambler (2002) describes a method-specific use of this model. He frames his description of the Agile Modelling (AM) method from a top-down perspective by presenting a set of high-level values that represent the fundamental concepts that govern the method. These values are supported by a set of principles that relate the values to their more concrete application in software development. Finally, he proposes a set of practices that may be performed to follow the direction of the principles. In keeping with Karlsson & Ågerfalk (2009) this three-tier model is mapped onto the method rationale framework of Ågerfalk & Fitzgerald (2006), which is then used as a tool to understand the conceptual linkages between values, goals (principles) and method fragments (practices).

As stated above, a result of this paper is a set of higher-order values that underpin lean principles and practices. A bottom-up approach is taken to identify a candidate set of values. Experiences and opinions of various authors are analysed in order to clarify and understand the different sets of lean principles that these authors recommend as necessary to guide lean thinking in different situations. These lean principles are synthesized into a set of values intended to abstract the beliefs and intentions of the principles.

The identification of a representative set of commentators on lean thinking was approached from a genealogical perspective. The main sources of information related to the earliest application of lean thinking (operations management), were lean training courses and Ohno(1988). The next phase was to generalize lean operations. Womack and Jones (2003) served as a launching pad to conduct citation searches for relevant papers, websites, books and conference proceedings. These searches were influenced by interest in the application of lean thinking to software development. Inconsistencies in reporting Charette's lean development method highlighted confusion about the relationship between lean and agile in software development and motivated a review of his work in project management and lean thinking. The proposal that Scrum and Lean were two separate approaches to address complex adaptive systems(Sutherland, 2008) prompted investigation into product development. Finally, the software focus directed the search to review sources related to the term "lean software development".

Analysis of the above sources revealed several sets of lean principles. These principles were analysed in an iterative fashion in order to induce a set of values. As expected, overlaps were encountered as aspects of different values are addressed in many of the principles. It was decided to consider what value was

most prevalent in each principle and to assign the principle to that value. This is considered reasonable as the motive was not to build an exact mapping of principles to values, but rather to use the principles as a basis to help uncover a representative set of lean values. The analysis reviewed the sources of each principle to fully understand the meaning of the principle in the context of the paper in which it was presented. A description of each principle was created and this enabled an initial definition of each value being addressed. The iterative synthesis of the values involved both convergence from narrow to broad (as similar values were grouped to a higher-order value) and in some cases divergence from high-level to specific. Further details on this process are presented in section 4.

Clearly there are limitations with this approach. The initial literature review is not easily generalized or replicable. The identification of values was subject to the bias of the authors' beliefs and experiences. However, it is hoped that these beliefs and experiences are grounded in an appreciation of software engineering research and application and that the proposed value set resonates with members of these communities. This research is not intended to present the definitive set of lean software development values. It is intended to present a candidate set that may be used to facilitate further research into the relationship between agile software development and lean software development.

3 Evolution of Lean Software Development

Methodology (or the study of methods) has proposed that software methods can be viewed as formalized approaches created in clinical environments and in many cases, subsequently tailored to become both situational methods to suit certain environments and methods-in-action adapted as a result of the influence of various forces in a particular development situation. Many methods have been based upon a higher-order construct of problem-solving known as the systems development lifecycle or colloquially "the waterfall model". (Fitzgerald et al., 2002).

As stated earlier, this paper proposes that LSD should also be viewed as a broad concept that informs the construction, adaptation and application of many methods: a management philosophy that has evolved from different domains including operations management, product development, project management and software development. This section reviews research into the application of lean thinking in these domains. In the interest of brevity, it is not feasible to describe every principle identified. However, to support descriptions of the reported opinions of selected commentators from each domain, a selection of principles are described in the context that they have been proposed by the original source. To supplement this narrative and support further inquiry, each source is mapped to a codified list of its associated principles (Table 1). Each principle referred to in the following sections is supported by its associated table 1 code e.g. *On* refers to principle *n* of Ohno (1988); *SB_n* refers to Spear and Bowen (1999).

3.1 Influence of operations management on LSD

The Toyota Production System (TPS) is often cited as one of the key catalysts to the worldwide interest in lean approaches to different disciplines. Many TPS concepts have been practiced by manufacturers worldwide (Spear and Bowen, 1999). This system evolved within the manufacturing environment of Toyota in response to the challenges facing the company to be cost-effective and adaptive in order to initiate and sustain a competitive advantage in the automotive sector (Hibbs et al., 2009, Coplien and Bjornvig, 2010, Ohno, 1988, Holweg, 2007).

Many early developments such as parts interchangeability, work standardization and division of labour led to the industrial revolution in which craft workers were replaced by mass production. The US automotive

<i>Operations Management</i>	
Ohno(1988)	O1: Continual emphasis on waste reduction; O2: Flow; O3: Built-in quality; O4: Respect for people; O5: Caution in slow-growth environment; O6: Production levelling; O7: Autonomation; O8: Respond to needs; O9: Cost reduction
Spear and Bowen (1999)	SB1: Detailed specification of every task; SB2: Unambiguous communication procedures between employees; SB3: Single pathway for every product; SB4: Strict adherence to scientific method for improvements
Womack & Jones (2003)	W1: Value; W2: Value Stream; W3: Flow; W4: Pull; W5: Perfection
<i>Product Development</i>	
Reinertsen and Shaeffer (2005)	R1: Batch size reduction; R2: Tolerate necessary variability; R3: Maintain flow; R4: Pull; R5: Fast powerful feedback loops; R6: Avoid over-reliance on requirements; R7: Invest in flexibility; R8: Achieve adequate failure rates; R9: Understand economics of waste; R10: Avoid sub-optimization
Liker and Morgan (2006)	L1: Establish customer-defined value to separate value added from waste; L2: Maximize design space(explore alternatives); L3: Levelled process flow; L4: Rigorous standardization; L5: Chief Engineer system; L6: Balance functional and cross-functional expertise; L7: Develop engineering competence; L8: Integrate suppliers; L9: Continuous improvement; L10: Build culture of relentless improvement; L11: Adapt technology to fit people and process; L12: Simple visual communication to align organization; L13: Support standardization and learning with powerful tools.
<i>Project Management</i>	
Charette (2003)	C1: Customer satisfaction; Best value for money; C2: Active customer participation; C3: Team effort; C4: Everything can change; C5: Domain solutions; C6: Avoid duplication; C7: 80% solution now preferable; C8: Minimalism; C9: Needs determine technology; C10: Measure product growth in features; C11: Use lean appropriately; C12: Worker development
<i>Lean Software Development</i>	
Hou(1995)	H1: Defined; H2: Configuration Management, H3: User Involvement, H4: Transparent, H5: Tailorable, H6: Rapid Development, H7: Scalable, H8: Defect-free, H9: Continuous improvement, H10: Manufacturability, H11: Supportability, H12: Scalable architecture, H13: Hardware/software codesign
Middleton(2001)	M1: Continual improvement; M2: Empowered workers; M3: Defect prevention; M4: Visual quality measures; M5: Automatic quality measurement devices(often self-developed)
Poppindieck & Poppindieck (2003)	P1: waste elimination; P2: learning amplification; P3: deferment of decisions; P4: optimal speed of delivery; P5: team empowerment; P6: integrity construction; P7: systems thinking
Middleton, Flaxel & Cookson (2005)	MF1: Small iterations; MF2: Requirements elicitation; MF3: Requirements chunking to support resource allocation; MF4: Units of work to enable Takt time establishment; MF5: Low distance between collaborators; MF6: Consistent roles, work practices; MF7: Defect prevention; MF8: Feedback on productivity & errors; MF9: Impartial data collection; MF10: Multi-disciplinary teams;
Middleton & Joyce (2011)	MJ1: Levels of WIP; MJ2: Pull work only when needed; MJ3: Level out workload; MJ4: Stop to fix problems; MJ5: Continuous improvement; MJ6: Make process visible; MJ7: Ensure technology serves people & process;

Table 1: Lean principles from various sectors/authors

industry was revolutionized by these concepts leading to its dominance of the sector world-wide. By 1945, the Toyota Motor company launched a programme to "catch up with America". Severe constraints including a small home market, limited land availability, skills shortages and limited funds for capital investment motivated Toyota to seek more innovative production mechanisms to mass production approaches in order to pursue their goal. (Ohno, 1988). The TPS was built upon proven operations management techniques, quality concepts, analyses of innovations adopted by other industry sectors and continuous learning and refinement as a result of the introduction of various initiatives intended to reduce waste and increase value to the customer. Ohno (1988) challenges the use of large batch production in the pursuit of economies of scale, stating that large batches both increase waste (such as defects) and inhibit product diversity (Holweg, 2007). Waste reduction (O1)² is a key principle of the TPS. Seven major sources of waste are identified: transportation, overproduction, over processing, defects, inventory, motion and waiting. Levelling customer demand (O6) is key to enabling the transformational flow of parts through the supply chain. The Just-in-time (JIT) method is used to enable flow (O2). This demands that a part is produced at the time it is needed. It requires upstream activities that have rapid set-up systems in order to respond to demand and downstream activities to have levelled their schedules to enable predictable flow. Ohno (1988) notes that the arrival of just enough inventory just in time enables waste reduction, promotes consistency and reduces the chances of employees being overburdened by workloads. For these reasons, the principle of flow is emphasised by many authors. Built-in-quality (O3) refers to the concepts of mistake-proofing in order to prevent the occurrence of defects. A related principle specifically called out from the works of Ohno (1988) is "autonomation" (O7) which refers to the "intelligent machine" (suspends work and alerts operators upon detection of an adverse quality event) to promote quality. An example of the application of this concept in software development would be a build verification test that reacts to any test failures by automatically stopping the build and waiting for developer intervention. Cost reduction (O9) demands that selling price is dictated by the market - not on a cost plus basis (Selling price = profit - cost). Other principles highlighted by Ohno include the empowerment of workers (O4), awareness that high-growth success criteria (such as mass production) do not apply in slow-growth economies (O5) and awareness that all improvement initiatives should be a response to a clear opportunity or need (O8). Spear and Bowen (1999) investigates the difficulties of TPS replication encountered by organizations worldwide, despite Toyota's willingness to offer open access to their manufacturing systems. They postulate that this difficulty emanates from imitators placing an excessive focus on the practises of the TPS and not enough emphasis on the overall implementation of the system. They propose that the implementation of TPS can be defined as the compliance to four rules: detailed specification of every task (SB1); unambiguous communication procedures between employees (SB2); a single pathway for every product or service (SB3); strict adherence to the use of the scientific method in pursuing improvements (SB4). This leads to an organization equipped to pursue continuous learning. The TPS promotes the encapsulation of work so that modifications to an area do not lead to unintended adverse consequences in other parts of the system. Liker and Morgan (2006) note that this continual pursuit of innovation at the "gemba" or place of action had a consequence that the overall system was not formally documented for many years.

Womack and Jones (2003) synopsise their interpretation of the TPS and additional lean research into five principles: value, value stream, flow, pull and perfection. Value (W1) could be considered customer-defined satisfaction with an information system or product features. Value stream (W2) refers to the mapping of every step from initial transformation of raw materials to delivery of final product in order to identify and tackle value and waste. Steps in a software method could be analysed to identify value added, non value-added and necessary non-value added activities being performed in a software project. Flow (W3) refers to the establishment of a process of value-adding activities resulting in the delivery of a

Comment [ML1]: Must remove from submission - there is no table 1

² Space constraints preclude a full explanation of each principle. Principles are keyed to the referenced work as follows: *On* refers to Ohno (1988); *SBn* refers to Spear and Bowen (1999). See Table 1 for complete set of principles.

product or value. Flow in operations management promotes the reduction of batch sizes in order to have a product flow through production out to a customer. The analogy for software is a feature. Middleton (2005, 2011) acknowledges the difficulty of establishing a "standard feature" and proposes the use of feature sets or "minimum marketable features" to deliver value. Pull (W4) refers to the establishment of production to react to customer demand. The process begins with the final customer being synchronized with product completion. No upstream work is delivered until the immediate succeeding downstream operation signals a need (via Kanban). (Ohno, 1988). Hibbs et al. (2009, p.113) presents the application of pull in Scrum. Perfection (W5) relates to the continual improvement of standard work.

Comment [ML2]: That should be 2009 - change in submission

3.2 Influence of Product development on LSD

Liker and Morgan (2006) explain that lean principles extend beyond the manufacturing production department. Many industries have adopted different techniques from the TPS practices. They propose that lean adoption in a non-repetitive environment such as services would be better served by using the experiences of the Toyota Product Development system (TPPD). Studies of this system are synthesized into a generic lean product development framework of management principles that address three areas: process, people and tools/technology. Process thinking promotes a complete understanding of what value is from the customer's perspective (L1), strong up-front design to ensure the correct plan is followed (L2), effective levelling of planned activities (L3) and establishment and adherence to standard processes (L4). People management fosters a culture of empowerment and technical and domain expertise (L6, L7) that enables workers (including suppliers (L8) to continually improve both the product and the process. Tools/technology management insists that tools must be tailored (L11) to address the problem. Tools must be used to support communications at all levels and must be leveraged to support standardization and continual learning within the organization (L12, L13). An overarching characteristic of the TPPD is the promotion of continual learning by the workforce (L9). The application of certain lean techniques in order to remove some waste from a particular task is secondary to an organization gaining momentum in the empowerment of its workforce to constantly pursue continual improvement (L10) and evolve new standard activities that form the basis for future waste reduction initiatives. Liker's interpretation of TPPD is consistent with recommendations on the application of total quality management to software development: innovative approaches to customer care, continual process improvement, judicious use of tools and technologies and strong awareness of human factors (Kan et al., 1994).

Comment [ML3]: This word is missing from submission

Reinertsen and Shaeffer (2005) also emphasize the difference between product development and manufacturing activities. Process variability, sequence, volatile requirements and attitudes to risk influence the manner in which lean principles are addressed in a research and development environment. Whereas variability in a repetitive environment is to be avoided, this is not the case in a non-repetitive situation. In R&D, variability is desirable in certain circumstances and is to be removed in others (R2). Manufacturing adds value to items in a sequential manner. R&D adds value to an intangible asset: information. As such, this value addition can occur to the same information in different places at the same time, thus necessitating consideration on how the potential parallel addition of value can be exploited. R&D deals with moving targets as market needs and technology capabilities evolve - it must embrace uncertainty (R6) in order to innovate whereas manufacturing aims to enhance productivity by operating under stable constraints. However, despite the aforementioned differences between manufacturing and R&D, they propose that certain lean principles discovered by the manufacturing sector can be applied to product development. Reduction in batch sizes equates to addressing smaller "batches" of information (R1). This can be interpreted as reducing project size. Variability of expected work and task durations are an outcome of risk-taking which is critical to innovation (R8). Enabling workers to be multi-functional (R7) promotes a "pull" strategy to deal with task variability and also encourages the maintenance of flow (R3) rather than a predictive planning approach which can often be foiled by latent requirements.

The main difference between lean product development (LPD) and traditional product development is the focus on customer value provision at each process stage. LPD processes are not well defined in the

literature - it is left to organizations to implement processes to address the LPD philosophies. LPD embraces long-term thinking and encourages sustainable development processes that enable an organization to manage changing customer needs and market conditions. (von Wurtemberg et al., 2011)

3.3 Influence of Project Management on LSD.

Project management consists of a series of overlapping processes or activities that address the needs of nine knowledge areas including "risk management".(PMI, 2008). Koskela and Howell (2002) asserts that such a view of project management is based upon the production theory of transformation and does not address other production theories such as flow and value. Charette (1996) contends that risk management should be considered a central tenet of software project management rather than just one of a number of different areas to be managed. He bases this proposal on the proposition that the complexity of software systems makes them unsuitable for the traditional or scientific problem-solving approach to their resolution. The dearth of reliable information in software projects makes them notoriously difficult to predict and this has led to many notable project overruns. Risk management has been the vehicle used to try and improve software project management control. He promotes the concept of "risk-entrepreneurship", proposing that organizations need to move from a negative defensive view of risk to also embrace positive risks (opportunities). Charette (1996) promotes the "joint approach" that leads to both top-down and bottom-up views of risks in order to incorporate both the detailed technical and external environmental (strategic) aspects of an opportunity or threat. This requires strong leadership and an awareness of the relationship between an information system under construction and the environment in which it is to be used. The Lean Development (LD) agile method supports the application of this approach(Highsmith, 2002, Charette, 2003). It is a top-down strategic approach that proposes a management philosophy rather than practices and techniques. Just as Kichoro Toyoda³ set the "impossible goal" to his workforce to catch up with the US automotive giants in three years, Charette (2003) sets a goal that LD should enable teams to establish and sustain massive productivity gains: 66% less cost, duration and defects than similar projects performed by a CMM level 3 software development team using other approaches. This is consistent with the view that in order to make a successful and sustainable transition to lean, it is important to provoke "kaikaku" or radical improvement by setting extremely challenging goals (Middleton and Sutton, 2005). LD supports the extension of information technology work into the business, thus increasing the ability to provide value to the customer (C1). It is a tactic employed to evolve a "change tolerant" organization (C4) that is capable of predicting the needs of the market. It requires collaboration between customers, marketing, business management, project management and software developers(C2, C3). LD may be summarized as a method that evokes a strategic focus, encompassing the concept of risk entrepreneurship, leveraging the principles and lessons of lean production and employing stretch goals to motivate its adoption(Highsmith, 2002).

3.4 Influence of Software Development on LSD

During the 1990's, various reports were published on different initiatives that applied lean concepts to software development approaches. Hou (1995) evaluated different approaches in order to propose a lean hardware/software method for the development of embedded systems used by the U.S. military. Different criteria were proposed and used to evaluate a series of existing approaches that included spiral processes, incremental approaches and the clean room engineering method. These criteria were generated from process research, lean product development concepts and hardware systems design considerations. Clean room engineering places a strong emphasis on defect prevention(H9), focussing on the root cause of a defect's presence in a system and promoting the correction of the development process so that such a

³ The surname of the owner of Toyota Motor Company is sometimes referred to as "Toyota".

cause is removed into the future. The removal of waste appears to be synonymous with the concept of lean and the presence of defects is one of the more visible elements of waste in a software application.

Middleton (2001) describes five principles of lean manufacturing relevant to software development, emphasizing the importance of defect prevention (M3). He explains that system specifications may be considered work-in-progress and as such constitute a risk of waste. In software development this waste can manifest itself in a defect that could have been quickly resolved during analysis and specification creation but could require much greater effort if uncovered at a later stage in development. He describes reports of defect prevention practices using checklists and root cause analysis. Although the timeline of this experiment was very short, a subsequent two-year investigation of LSD describes how various lean principles were leveraged in order to resolve a systemic problem of ongoing delays caused by defect resolution following code completion (Middleton et al., 2005). Principles applied by the team included: continuous-flow processing (small iterations of fully tested functionality) (MF1); customer-defined value (intense emphasis on requirements elicitation) (MF2); design structure matrix to enable flow (estimation and chunking of requirements using function point analysis to support resource allocation to work) (MF3); Takt time (Grouping of requirements into "units of work" so that a takt time could be calculated and used to assess team productivity rate) (MF4); Linked processes (distance reduction between collaborators and related processes) (MF5); Standardised procedures (consistency of roles, work practices etc. to facilitate resource allocation) (MF6); Eliminate rework (defect prevention using root-cause analysis (MF7); scope management enhanced by increased focus on customer needs and context) (MF7); Posting results (feedback on productivity thus promoting continual learning) (M4, MF8); Data driven decisions (impartial data collection thus avoiding delays due to meetings and disagreements) (MF9); Minimize inventory (decompose product components into stories of 3-5 features of 3-5 units of work to be tackled separately by multi-disciplinary teams (developers/QA/Marketing)) (MF10). After 2 years of applying these principles a review of the organization reported substantial improvements. A more recent 12 month study (Oct 2008-Oct 2009) conducted by (Middleton and Joyce, 2011) proposes that the application of lean principles and practices are context dependent and not subject to a "cookbook" type approach across companies and industries. A notable observation from this study is that the concept of a consistent software development unit of work is very difficult to establish. The concept of takt time used to pace and control the production process was roughly simulated by using minimum market features which resulted in concise cohesive iterations of software development. These iterations, supported by daily meetings, created small feedback loops that provided momentum and control of the process (R5). This interpretation is consistent with the concept of "self-reinforcing virtuous cycles" (Charette, 2003), "punctuated equilibrium" (Sutherland, 2008) and fast powerful feedback loops (Reinertsen, 2005).

Comment [ML4]: WHAT!! Change this to M3.

An influential contribution to the growing body of knowledge on LSD was made by Poppendieck and Poppendieck (2003) in their proposal and analysis of a set of lean principles to support LSD. The emphasis on this work was to provide an application framework of twenty two practices that could enable practitioners to realize lean values. The principles proposed were: waste elimination; learning amplification; deferment of decisions; optimal speed of delivery; team empowerment; integrity construction and systems thinking. (Poppendieck and Poppendieck, 2007) expanded the nature of their earlier tactical software-centric work to incorporate the impact of the broader lifecycle required to incorporate strategic considerations espoused by lean thinking. Table 1 presents a codified list of the lean principles put forward by the various authors reviewed in this section.

4 Findings and Conclusions

This paper reports on a set of LSD values (table 2) induced from principles identified in lean literature across various domains. The principle sets presented in table 1 are used to propose an underlying value being addressed by each principle. Each iteration of the analysis process resulted in related values being

<i>Value</i>	<i>Description</i>	<i>Principles</i>
Continuous Improvement	Visible feedback on productivity. Reviews to propagate learning to wider organizations. Apply rigorous standardization to establish agreed baselines for improvement. Actively seek and manage obstacles through agreed processes, such as root cause analysis. Promote organizational learning and a culture of relentless improvement. Use simple powerful tools.	H9,MF1,M1,MJ1,MJ5,L4,L9,L10,L13,W5,P7,SB4
Business environment awareness	Be aware of why you are doing work. Ensure it adds value to your situation. Future-proof your approach to work against changing market conditions. Consider domain solutions rather than restricting product to one market sector. Don't force lean approach where inappropriate to business needs. Don't optimize locally to the detriment of other aspects of the business.	H5,05,08,C6,C12,R10
Customer value	Ensure customer defines value. Seek out user needs, not just requirements. Early delivery of value. Enable future needs (maintainability). Enable efficient deployment of product features. Provide value for money.	H3,H6,H10,H11,H12,H13,H14,MF1,MF2,MF10,L1,W1,W4,09,C1,C2,C3,C8,P4
Data driven decisions	Impartial data collection to drive decision making and reduce cost of meetings and disagreements. Rigorous scientific approach to continuous improvement.	MF9
Effective collaboration	Reduce distance between collaborators and processes. Simple visual measures of productivity and quality. Balance functional expertise with cross-functional integration. Promote teamwork. Integrate suppliers. Unambiguous communication procedures.	M4,MF5,MJ6,L6,L8,L12,C4,P2,SB2
Effective process	Easily understood method. Enable clear comprehension of method-in-action. Clear mechanisms for resource allocation including consistent roles, names, work practices. Ability to assess team productivity rate. Central expert to integrate overall development ("Chief Engineer"). Scalable approach. Clear product development roadmaps. Fast powerful feedback loops to build momentum.	H1,H4,H7,MF3,MF4,MF6,L5,R5,SB1,SB3
Effective use of technology	Technology must serve a particular need. Configuration management systems and quality measurement devices are examples of key facilities. Pursue automation by using technology to enable workers to effectively adjust system when necessary.	H2,M5,MJ7,L11,07,C10
Embrace change	Defer commitment to scope. Encapsulate features and consider all options carefully. Facilitate emerging requirements.	C5,P3,R6
Flow of value	Stabilize development process to enable levelled flow of value - manage necessary variability. Smooth demand from users. Pull value from users through lifecycle. Apply minimalism (scope items, teams, documents). Allocate resources only when needed.	MJ2,MJ3,L3,W3,O2,O6,C9,R1,R2,R3,R4
Person focus	Empower individuals and teams. Promote continual worker development. Invest in multi-skilling and promote attitude to participate in multiple roles.	M2,O4,C13,P5,R7
Product excellence	Promote strong design culture to avoid premature convergence on incorrect solution. Prototype different options to derive best approach. Deliver functionality in change-tolerant form. Build quality in through practices of jidoka and poka-yoke. Promote deep specialized knowledge of product and process. Promote a culture of excellence.	L2,L7,C11,P6,R8
Remove waste	Promote systematic defect prevention. Eliminate rework through emphasis on customer needs. Holistic focus on quality that motivates the refactoring of legacy code to improve general product. Identify silos of waste present in value stream and improve processes. Promote reusability where appropriate. Take cognisance of the fact that waste can have different forms in different projects/work situations.	H8,M3,MF7,MJ4,W2,01,03,C7,P1,R9

Table 2: Synthesized lean values

synthesized into higher-order values. An example of this approach was the emergence and establishment of the value "Effective project management" as underpinning a number of principles. A subsequent iteration resulted in this value being subsumed into a broader value: "Effective process". The synthesis process was not always one of convergence. A final run-through of the values led to the identification of the value "Flow of value". This had been part of the value "Effective process" but it became clear that many researchers had singled out "flow" as a special consideration in lean thinking and therefore was deemed of sufficient relevance to merit consideration as a specific lean value. The proposed set of LSD values is presented in table 2. Linkage between values and their associated principles is shown through the principle codes.

Synthesis of the LSD values reveals an interesting dichotomy of two overarching values that address an external and internal focus respectively: delivery of customer-defined value and reduction of waste. The identification, development and delivery of customer-defined value is a core value that influences engagement with market forces and is supported by a culture of excellence that emphasizes design-oriented processes in order to avoid premature agreement on solutions and promote the development of change-tolerant flexible systems ("product excellence", "embrace change" and "business environment awareness"). The highest-order internally-oriented LSD value ("waste reduction") is supported by the promotion of flow-based demand-driven development that exposes process defects. Focus on people supports effective collaboration and empowerment to continuously improve and further reduce or remove any wasteful activities (such as the production of unused artefacts).

As stated in section 2, limitations of this work include author bias and replication difficulties. However, the proposed LSD value set is not intended to be exhaustive and is open to extension. It is intended to support further research and is an important step in addressing confusion around this software development approach. For example, (Wang, 2011) highlights the need for analysis into the relationship between agile and lean practices. A base set of LSD values could serve to anchor such research efforts. A specific research study using LSD values will be facilitated by leveraging the method rationale analysis framework (Agerfalk and Fitzgerald, 2006) as a lens to relate practices and goals of Scrum to values contained in the LSD value set. The common lineage of Scrum and LSD (Sutherland, 2008) suggests this particular agile method is a good candidate method for initial exploration into the relationship between ASD and LSD. This research approach could also be used to relate other ASD methods to LSD and enhance general understanding of the relationship between these two paradigms. At a more general level, research using the LSD value set could be extended to investigate the relationship between LSD and any software development method. From an industrial perspective, the LSD value set may also be applied to software development projects in order to uncover insights into the "leanness" of a particular approach or "method-in-action". This is particularly important in light of Fowler's (2008) suggestion that developers should focus on lean and agile, not either in isolation.

References

- Abrahamsson, P., Conboy, K. & Wang, X. (2009). 'Lots done, more to do': the current state of agile systems development research. *European Journal of Information Systems*, 18, 281-284.
- Agerfalk, P. J. & Fitzgerald, B. (2006). Exploring the concept of Method Rationale: A Conceptual Tool for Method Tailoring. In: SIAU, K. (ed.) *Advanced Topics in Database Research*. Hershey, PA: Idea Group.
- Ambler, S. (2002). *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*, New York, Wiley.
- Charette, R. N. (1996). The mechanics of managing IT risk (Reprinted from Software Management, December 1993). *Journal of Information Technology*, 11, 373-378.
- Charette, R. N. (2003). Challenging the Fundamental Notions of Software Development. Available: <http://www.itmpi.org/assets/base/images/itmpi/privaterooms/robertcharette/ChallengingtheFundamentalNotions.pdf> [Accessed 03 Oct. 2011].
- Cohen, D., Lindvall, M. & Costa, P. (2004). An introduction to agile methods. *Advances in Computers, Advances in Software Engineering*, 62, 2-67.

- Coplien, J. O. & Bjornvig, G. (2010). *Lean Architecture for Agile Software Development*, UK, Wiley.
- Dingsoyr, T., Dyba, T. & Abrahamsson, P. A Preliminary Roadmap for Empirical Research on Agile Software Development. Agile, (2008). AGILE '08. Conference, 4-8 Aug. 2008 2008. 83-94.
- Fitzgerald, B., Russo, N. L. & Stolterman, E. (2002). *Information Systems Development: Methods in Action*, Berkshire, UK, McGraw-Hill Education Technology.
- Fowler, M. (2008). *AgileVersus Lean* [Online]. <http://martinfowler.com/bliki/AgileVersusLean.html>
- Fowler, M. & Highsmith, J. (2001). The Agile Manifesto. *Software Development*, 9, 28-32.
- Hibbs, C., Jewett, S. & Sullivan, M. (2009). *The art of lean software development: A practical and incremental approach.*, CA, O'Reilly Media.
- Highsmith, J. (2002). *Agile Software Development Ecosystems*, Addison-Wesley.
- Holweg, M. (2007). The genealogy of lean production. *Jnl of Operations Management*, 25, 420-437.
- Hou, A. C. (1995). *Toward Lean Hardware/Software System Development: An Evaluation of Selected Complex Electronic System Development Methodologies*. Cambridge: MIT.
- Kan, S. H., Basili, V. R. & Shapiro, L. N. (1994). Software quality: An overview from the perspective of total quality management. *IBM Systems Journal*, 33, 4-19.
- Karlsson, F. & Agerfalk P. J. (2009). Exploring agile values in method configuration. *European Journal of Information Systems*, 18, 300-316.
- Koskela, L. & Howell, G. The underlying theory of project management is obsolete. PMI Research Conference, June 2002 Seattle, Washington. Project Management Institute, 293-302.
- Lang, M and Fitzgerald, B (2005) Hypermedia systems development practice, *IEEE Software*, 20 (2)
- Liker, J. K. & Morgan, J. M. (2006). The Toyota Way in Services: The Case of Lean Product Development. *Academy of Management Perspectives*, 20, 5-20.
- Middleton, P. (1995). Maintenance management: From product to process. *Journal of Software Maintenance: Research and Practice*, 7, 63-73.
- Middleton, P. (2001). Lean software development. *Software Quality Journal*, 9, 241-252.
- Middleton, P., Flaxel, A. & Cookson, A. 2005. Lean Software Management Case Study: In: BAUMEISTER, H., MARCHESI, M. & HOLCOMBE, M. (eds.). Springer / Heidelberg.
- Middleton, P. & Joyce, D. (2011). Lean Software Management: BBC Worldwide Case Study. *Engineering Management, IEEE Transactions on*, PP, 1-13.
- Middleton, P. & Sutton, J. (2005). *Lean Software Strategies - Proven techniques for managers and developers*, New York, Productivity Press.
- Ohno, T. (1988). *The Toyota Production System: Beyond Large-Scale Production*, Portland Press.
- PMI (2008). *A Guide To The Project Management Body Of Knowledge (PMBOK Guides)*, Project Management Institute.
- Poppendieck, M. & Poppendieck, T. (2003). *Lean Software Development – An Agile Toolkit for Software Development Managers*, Boston Addison-Wesley.
- Poppendieck, M. & Poppendieck, T. (2007). *Implementing Lean Software Development: From Concept to Cash*, Addison-Wesley.
- Raman, S. Lean software development: is it feasible? Digital Avionics Systems Conference, 1998. Proc., 17th DASC. The AIAA/IEEE/SAE, 31 Oct-7 Nov 1998 1998. C13/1-C13/8 vol.1.
- Reinertsen, D. (2005). Journey to lean: Making operational change stick. *Journal of Product Innovation Management*, 22, 539-U2.
- Reinertsen, D. & Shaeffer, L. (2005). Making R&D Lean. *Research Technology Mgt*, 48, 51-7.
- Senge, P. M. (1990). *The Fifth Discipline*, New York, Doubleday.
- Spear, S. & Bowen, H. (1999). Decoding the DNA of the Toyota Production System. *Harvard Business Review*, 96-106.
- Sutherland, J. (2008). *The First Scrum: Was it Scrum or Lean?* [Online]. Available: <http://scrum.jeffsutherland.com/2007/11/is-it-scrum-or-lean.html> [Accessed 21 Sep 2011].
- von Wurtemberg, L. M., Lillieskold, J. & Ericsson, E. Abstract model of LPD: A critical review of the Lean Product Development concept. Technology Management in the Energy Smart World (PICMET), 2011 Proceedings of PICMET '11.: July 31 2011-Aug. 4 2011 2011. 1-7.
- Wang, X. The Combination of Agile and Lean in Software Development: An Experience Report Analysis. AGILE Conference (AGILE), 2011, 7-13 Aug. 2011 2011. 1-9.
- Womack, J. P. & Jones, D. T. (2003). *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*, New York, NY, Free Press.