Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2011 Proceedings

European Conference on Information Systems (ECIS)

Summer 10-6-2011

FROM AGILE TO LEAN: THE PERSPECTIVES OF THE TWO AGILE ONLINE COMMUNITIES OF INTEREST

Lero Xiaofeng Wang

Michael Lane

Kieran Conboy

Follow this and additional works at: http://aisel.aisnet.org/ecis2011

FROM AGILE TO LEAN: THE PERSPECTIVES OF THE TWO AGILE ONLINE COMMUNITIES OF INTEREST

Xiaofeng Wang, Lero - The Irish Software Engineering Research Centre, Limerick, Ireland, xiaofeng.wang@lero.ie

Michael Lane, University of Limerick, Limerick, Ireland, michael.lane@ul.ie

Kieran Conboy, NUI Galway, Galway, Ireland, kieran.conboy@nuigalway.ie

Abstract

In the last several years there has been a noticeable focus shift of agile methods from eXtreme Programming (XP) and Scrum to Lean approaches, which is indicated as "from agile to lean". However our understanding of the driving factors behind this focus shift is very limited. The study proposed in this paper attempts to investigate the shift through a content analysis of two major agile mailing lists. The expected result is a set of driving factors behind the shift from agile to lean, which can contribute to a better understanding of if and why Lean software development is "the next wave of software process".

Keywords: Agile Methods, Lean Thinking, Lean Software Development, Community of Interest, Content Analysis.

1   Introduction

2011 will mark the ten year anniversary of the Agile Manifesto[1]. During these ten years the practice and research of agile methods has become highly prevalent. In recent years some have started to reflect on the agile movement. The term Agile 2.0 was coined to denote some of the evaluations of agile methods (e.g., Miller 2006). A shift of focus of agile methods from eXtreme Programming (XP) and Scrum more recently to Lean software development has been noticed and advocated. Lean software development is claimed to be "the next wave of software process"[2]. Its adoption is believed to be a necessary progression for organisations planning to scale up agile (Anderson 2010). Others claim that Lean software development can resolve issues agile methods failed to address (Birkeland 2010). But our understanding of this focus shift, which is generally termed from agile to lean (Hibbs et al. 2009), if it really exists, is very limited.

The purpose of the proposed study is to uncover the reasons why there is a shift of focus from agile to lean. The study will focus on the views of the agile community in relation to this subject. The research question we ask is: **What are the driving factors behind the shift from agile to lean?** Since the research on the broad topic of Lean software development is considered a nascent area (Dingsøyr et al. 2008), we believe our study is an important addition to this branch of research in general, and on the topic of the shift from agile to lean in specific.

To answer the research question, we will conduct a review of agile and lean software engineering literature, followed by an analysis of the mailing lists of two online agile communities of interest, namely XP and Scrum, as these are two of the most popular agile methods used in the past decade (Dingsøyr et al. 2008). We plan to analyse each mailing list in turn then compare and contrast the findings from the two mailing lists. A common set of themes will be identified which illustrate the driving factors behind the shift from agile to lean.

This research-in-progress paper presents the early stage of our study. Section 2 reviews the background literature and related work. This is followed by a description of the conceptual framework used to frame the subsequent data analysis. The research approach section explains the rationale of analysing the two mailing lists and describes the analysis procedures. The paper ends with expected results and concluding remarks.

2   Agile Methods vs. Lean Software Development

The Agile Alliance and the publication of the Agile Manifesto formally introduced agile methods to the software engineering community in 2001, yet there is no precise, concise and universally accepted definition of *agile method* (Conboy 2006). The literal meaning of *agile method* can be a method that brings agility to software development, or a method that has agility as a quality. In agile literature, however, *agile methods* generally means a family of methods under the umbrella of the Agile Alliance, including: XP (Beck 1999), Scrum (Schwaber and Beedle 2002), Dynamic Systems Development Method (DSDM, http://www.dsdm.org), Crystal Methods (Cockburn 2001), Feature-Driven Development (FDD, Coad and Palmer 2002), Lean Development (Charette 2002) and Adaptive Software Development (ASD, Highsmith 2002). Among these agile methods, XP and Scrum are the two most popular and widely adopted methods (Dingsøyr et al. 2008). XP is originally described by the authors as a lightweight method for small to medium-sized teams developing software in the face of vague or rapidly-changing requirements (Beck 1999). Scrum has been built on the metaphor of a rugby scrum. It sees a software development process as a "chaordic" process similar to the scrum stage, and attempts to control it using a simple project management framework which

---

[1] http://www.agilealliance.org
[2] http://atlanta2010.leanssc.org/

represents an adaptive, quick, self-organising product development process (Schwaber and Beedle 2002, Abrahamsson et al. 2002).

In the last several years, however, lean software development has picked up momentum and become increasingly eminent and popular among agile practitioners and researchers. Lean thinking has its origin in manufacturing and especially the Toyota Production System. There are five main lean concepts: value, value stream, flow, pull, and perfection (Womack et al. 2007). The primary focus is on the identification and elimination of waste from the process. Key lean practices include value stream mapping, kanban, limited WIP (Work-in-Progress), Kaizen (continuous improvement), etc.. Lean software development applies the concepts, principles and practices of lean thinking in software development activities (Poppendieck and Poppendieck 2003, Hibbs et al. 2009). For example, Poppendieck and Poppendieck (2003) specify seven principles of lean software development: eliminate waste, build quality in, create knowledge, defer commitment, deliver fast, respect people, and optimize the whole. An increasing number of software organisations or IT departments have been adopting or are planning to adopt lean thinking and techniques. Many of these groups have already adopted other agile methods, such as XP and Scrum (such as the cases reported in Parnell-Klabo 2006, Smits 2007, Long and Starr 2008, Willeke 2009 and Shinkle 2009). New hybrid practices have emerged, such as Scrumban (Ladas 2009) and the Limited Red Society (Kerievsky 2010).

Viewed originally as just another agile method (Highsmith 2002, Dybå and Dingsøyr 2009), there is an increasing focus on Lean software development and it is viewed as being a method category in itself rather than an instance of agile methods (Hibbs et al. 2009). Poppendieck (2007) claims that Lean provides the theory behind agile practices which is much needed given the fact that the theoretical foundation of agile software development is weak (Abrahamsson et al. 2009). Lean software development is conjectured as the means through which knowledge-based, knowledge-creating companies and organizations can effectively implement agile software development principles and practices (Sutherland and Tabaka 2007). Several studies claim that lean software development has the potential to resolve issues that agile methods have failed to address. Ambler (2009) argues that agile processes based on XP or Scrum can break down when one or more scaling factors are present (such as organizational wide agile adoption and transition or distributed agile development). Instead, a lean governance model based on enablement, collaboration, and motivation can enable agility at scale. Birkeland (2010) presents the experience of a software team shifting from a timebox-based development process (Scrum) to a flow-based process (pull-based, kanban), to show that flow-based process can better manage certain types of software development projects and activities, e.g. maintenance.

Therefore, the focus shift from agile to lean, as demonstrated by these studies, is not simply a shift of methods, but a shift pertaining to organizational concerns in software development. In other words, agile and lean are not different approaches that address the same kinds of issues encountered by software development practitioners. They address different organizational issues at different levels (strategic, tactical, and operational). However, as far as we are aware, there is no rigorous research that examines the driving factors behind this shift. That is, what are these organizational issues encountered by software companies that agile fails to answer but lean succeeds. This is the knowledge gap we intend to fill by our study.

3 Conceptual Framework

One promising avenue to explore the shift from agile to lean, we believe, is to understand the key differences between agile methods and lean approach. It serves as a conceptual framework to analyse the empirical data. The key differences will be drawn through a more comprehensive review of agile and lean literature. As a starting point, we will use the following list of key differences identified in the agile and lean literature (Highsmith 2002, Hibbs et al. 2009):

- The difference in the scope: Agile methods mostly concern themselves with the specific practice of developing software and the project management that surrounds software development. They

do not generally concern themselves with the surrounding business context in which the software development is taking place. Lean principles can be applied to any scope, from the specific practice of developing software to the entire enterprise where software development is just one small part. The larger the scope, the larger the potential benefits.

- The difference in the primary focus: The primary focus of agile methods is on close customer collaboration and the rapid delivery of working software as early as possible. The primary focus of lean software development is on the elimination of waste in the context of what the customer values.

- The difference in the formality of practice: Agile methods have a fair number of prescribed practices, such as pair programming, standup meetings, whereas lean software development has no formal practices. It has a toolkit of recommended practices from which to choose. It is quite common to pick agile practices as a starting point and begin applying other lean tools (such as value stream mapping) from there.

- The difference in the adoption style: Agile adoption generally takes a bottom-up approach whereas lean adoption is typically top-down.

We do not claim that this is a definitive and exclusive list. Actually it will be modified, refined and extended as we carry out more comprehensive literature review.

4   The Research Approach

The proposed study is exploratory in nature due to the fact that the topic we investigate is nascent. Our intention is to understand the perspectives of the agile community on the driving factors behind the focus shift from agile to lean. For this purpose we conduct a content analysis of two mailing lists of agile methods, which represent the online communities of interest of the two most popular agile methods: XP and Scrum. One assumption behind the choice of these two mailing lists is that people interested in lean related topics discuss them in these two agile specific mailing lists, rather than switching to lean specific mailing lists for these issues.

Content analysis of mailing lists is a commonly used research method (e.g., Lasker et al. 2005, Esquivel et al. 2006, Rdgby and Hassan 2007). Content analysis has been defined in many different ways, but one of the most commonly cited is "a research technique for the objective, systematic, and quantitative description of the manifest content of communication" (Berelson 1952). There are four main types of content analysis ranging from simple to complex, namely syntactical, referential, propositional, and thematic (see Cooper and Schindler 2003 for a detailed description of these). This study will adopt different types at the different stages of data analysis. The description information of the two mailing lists are summarised in Table 1:

Name of the mailing list	XP	Scrumdevelopment
Agile method representing	XP	Scrum
Number of members	9, 520	6, 766
Starting time	31/12/1999	01/02/2000
Messages posted within the period of analysis (Starting time – 30/11/2010)	156, 197	49, 463
Messages containing the keyword "lean"	2, 674	2, 811

Table 1.   The two agile mailing lists.

It can be seen from Table 1 that, in comparison, the XP mailing list is more active than its Scrum counterpart, both in terms of the number of subscribed members and the amount of posts. It is worth noting that the two online communities overlap to a certain extent, which can be identified by a

comparison of the member IDs. Due to the fact that the same person can have different identities in different forums, there is also a hidden overlap that is difficult to identify.

The data collection phase involves getting a local copy of all the messages posted to the two mailing lists within the period of analysis, including the meta information of the post, such as the sender, date sent, etc. The period of analysis is from the starting time of the mailing list to the 30th November 2010 (the latest possible date that is allowed by our study). We will also obtain a local copy of the member list of the two online communities, to understand to which extent the two online communities overlap. The data collection phase also involves eliminating the redundant data. It is typical that when a message is responded, the text or part of it from the original message is included. If the thread has many responses the same text would be repeated many times. If left un-dealt with, they would greatly affect the results of the data analysis. Therefore the data need to be cleansed to eliminate the redundancy.

The data analysis phase is composed of two steps: the analysis of each mailing list individually, and the comparison of the two mailing lists. For each mailing list, a simple type of content analysis of the messages posted within the specified period will be conducted, focusing on the frequency of the keywords related to LSD, to explore the existence of the shift from agile to lean. Then the more complex types of content analysis will be applied to conduct a more focused analysis of the discussion threads that contain posts with LSD related keywords. The purpose is to identify the perceived driving factors behind the shift from agile to lean. There is possibility that a post is LSD related but does not contain any LSD related keywords, such as a response to a post that contains these keywords. To take into account this factor, we will use a discussion thread as a unit of analysis rather than a single post and consider all the posts in the thread relevant data. To support the data analysis process, the software tools WordStat and QDA Miner from Provalis Research[3] will be used.

The comparison of the two mailing lists intends to search for common patterns that emerge from the first step of analysis, in order to deepen the understanding and enhance the generalisability of the findings. It should be noted that although the authors propose that this approach is a useful mechanism to explore the research question at a high-level, they also acknowledge that a number of issues existing in relation to content analysis techniques. A number of researchers have assessed the validity of content research, and have documented the various limitations and potential pitfalls of using the approach. The first possible pitfall occurs if the researcher is guilty of faulty definition and structure of content categories (Weber 1990). This, however, is not very relevant in this study as content analysis is used in a very simple manner. For example, one use is to assess the frequency with which lean related keywords were referred to. This does not involve the development of categories, but just requires the development of a list of lean related keywords. The second pitfall is analysing content under categories that are not mutually exclusive (Weber 1990; Roberts 1997). To resolve this, overlaps are identified. For example, in searching for the number of references to lean software development, a search is done for the terms lean, lean development, as well as all references to the lean principles. The final limitation of content analysis and especially simple word frequency counts is the failure to account for the context in which the words are used. Researchers such as Roberts (1997) and Carley (1990) suggest that once the researcher has identified the number of citations of a keyword, they should then double check each citation to ensure the word is being used in the right context, thus strengthening the validity of any inferences drawn from the analysis. Reliability of the data is a concern given that keywords found in different postings may not have been intended to convey the same meaning. This is particularly true given that both 'agile' and 'lean' are concepts shown to be highly polymorphous, multi-dimensional, vague and ambiguous (Conboy, 2009). To achieve this, NVivo software is used which allows the researchers to extract not just the number of times each keyword was cited, but also the sentence surrounding each citation, thus making this task relatively straight-forward.

---

[3] http://www.provalisresearch.com/

5 Summary

The focus shift of agile methods from XP and Scrum to lean software development has been noticeable in the last several years. However our understanding of the driving factors behind this shift is very limited. The proposed study attempts to investigate this phenomenon through a content analysis of two major agile mailing lists. We intend to identify a set of driving factors behind the shift from agile to lean, which can help both agile practitioners and researchers better understand this relatively new evolution of agile movement.

A quite interesting observation at this early stage is that it seems that the Scrum online community is displaying a higher interest in lean than the XP one. Although we may not be able to explain this different level of interest, it presents an intriguing direction for future study. It also has implications that, there might be a different set of driving factors behind a shift from a specific agile method to LSD. We may need to conduct separate investigations on the shift to lean software development depending upon the method in use, i.e. XP to Lean or Scrum to lean.

Acknowledgement

References

Abrahamsson, P., O. Salo, J. Ronkainen and J. Warsta (2002). Agile Software Development Methods: Review and Analysis. VTT Electronics, Espoo VTT Publications 478.

Abrahamsson, P., Conboy, K. and Wang, X.: 'Lots Done, More To Do': the Current State of Agile Systems Development Research, European Journal of Information Systems, (2009), 18(4): 281-284.

Ambler, S. W. (2009). Scaling agile software development through lean governance. SDG'09, Vancouver, Canada, May 17, 2009.

Anderson, David J. (2010). Business Drivers for Kanban Adoption. In Proceedings of Lean Software & Systems Conference, 7-14.

Beck, K. (1999). Extreme Programming Explained. Addison Wesley, Reading, MA.

Berelson, B. (1952). Content Analysis in Communication Research. Glencoe, Ill: Free Press 1971 (first edition from 1952).

Birkeland, O. (2010). From a Timebox Tangle to a More Flexible Flow. In Proceedings of the 11th International Conference on Agile Processes in Software Engineering and Extreme Programming, 325-334. Trondheim.

Carley, K. (1990). Content Analysis. The Encyclopedia of Language and Linguistics. R. Asher. Elmsford, NY, Pergamon: 725-730.

Charette, R. N. (2002). Foundations of Lean Development: The Lean Development Manager's Guide. Vol. 2, The Foundations Series on Risk Management (CD). Spotsylvania, Va.: ITABHI Corporation, 2002.

Coad, P. and S. Palmer (2002). Feature-Driven Development. Prentice Hall, Englewood Cliffs, NJ.

Cockburn, A. (2001). Crystal Clear: A Human-Powered Software Development Methodology for Small Teams. Addison-Wesley, Reading, MA.

Conboy, K. (2006). A Framework of Method Agility in Information Systems Development, PhD thesis, University of Limerick.

Conboy, K. (2009) Agility from first principles: Reconstructing the concept of agility in information systems development, Information Systems Research, 20, 3, 329-354.

Cooper, D. R. and Schindler P. (2003). Business Research Methods, McGraw Hill.

Dingsøyr, T., T. Dybå and P. Abrahamsson. (2008). A Preliminary Roadmap for Empirical Research on Agile Software Development. Agile 2008 Conference, Toronto, 4-8 August.

Dyba, T. and T. Dingsoyr (2009). What Do We Know about Agile Software Development? IEEE Software, Sept./Oct. 2009: 6-9.

Erlandson, D., E. Harris, et al. (1993). Doing Naturalistic Inquiry: A Guide to Methods. Newbury Park, CA, Sage.

Esquivel, A., F. Meric-Bernstam and E. V. Bernstam (2006). Accuracy and self correction of information received from an internet breast cancer list: content analysis. BMJ (Clinical research ed.) 332, no. 7547 (April): 939-42.

Hibbs, C., S. Jewett and M. Sullivan (2009). The Art of Lean Software Development: A Practical and Incremental Approach, 1st edition, O'Reilly Media, Inc., CA.

Highsmith, J. (2002). Agile software development ecosystems. Addison-Wesley, Boston.

Kerievsky, J. (2010). The Limited Red Society. http://xp2010.org/content.ap?thisId=84, last accessed on 07/12/2010.

Ladas, C. (2009). Scrumban - Essays on Kanban Systems for Lean Software Development. Modus Cooperandi Press , USA

Lasker, J., E. Sogolow and R. Sharim (2005). The role of an online community for people with a rare disease: content analysis of messages posted on a primary biliary cirrhosis mailinglist. Journal of medical Internet research 7, no. 1 (January): e10.

Long, K. and D. Starr (2008). Agile Supports Improved Culture and Quality for Healthwise. Agile 2008 Conference, Toronto, 4-8 August.

Miller, R. (2006). Second Generation Agile Software Development, http://blogs.msdn.com/b/randymiller/archive/2006/03/23/559229.aspx, last accessed on 07/12/2010.

Parnell-Klabo, E. (2006). Introducing Lean Principles with Agile Practices at a Fortune 500 Company. Agile 2006 Conference. Minneapolis, Minnesota, USA, 23-28 July 2006.

Poppendieck, M. (2001). Lean Programming. Software Development Magazine, 9(5): 71-75.

Poppendieck, M. (2007). Lean Software Development. In the Proceedings of the 29th International Conference on Software Engineering. Minneapolis, Minnesota, USA. 20 - 26 May 2007.

Poppendieck, M. and T. Poppendieck (2003). Lean Software Development: An Agile Toolkit Addison-Wesley Professional.

Rigby, P. and A. Hassan (2007). What Can OSS Mailing Lists Tell Us? A Preliminary Psychometric Text Analysis of the Apache Developer Mailing List (May): 23.

Roberts, C. E. (1997). Text Analysis for the Social Sciences: Methods for Drawing Statistical Inferences from Texts and Transcripts. Mahwah, NJ, Lawrence Erlbaum.

Schwaber, K and A. Beedle (2002). Agile Software Development with SCRUM. Prentice-Hall, Upper Saddle River, NJ.

Shapiro, G. and J. Markoff (1997). A Matter of Definition. Text Analysis for the Social Sciences: Methods for Drawing Statistical Inferences from Texts and Transcripts. C. Roberts. Mahwah, NJ, Lawrence Erlbaum: 35-54.

Shinkle, C. M. (2009). Applying the Dreyfus Model of Skill Acquisition to the Adoption of Kanban Systems at Software Engineering Professionals (SEP). Agile 2009 Conference, Chicago, USA. 24-28 August 2009.

Smits, H. (2007). The Impact of Scaling on Planning Activities in an Agile Software Development Center. In the Proceedings of the 40th Hawaii International Conference on System Sciences, Waikoloa, Big Island, Hawaii, 3-6 January 2007.

Sutherland, J. and J. Tabaka (2007). Incorporating Lean Development Practices into Agile Software Development. In the Proceedings of the 40th Hawaii International Conference on System Sciences, Waikoloa, Big Island, Hawaii, 3-6 January 2007.

Weber, R. (1990). Basic Content Analysis. Newbury Park, CA. , Sage.

Willeke, E. R. (2009). The Inkubook Experience: A Tale of Five Processes. Agile 2009 Conference, Chicago, USA. 24-28 August 2009.

Womack, J. P., D. T. Jones and D. Roos (2007). The Machine that Changed the World: How lean production revolutionized the global car wars, Simon & Schuster Ltd.