The Impact of Lean Techniques on Factors Influencing Defect Injection in Software Development

Rob J. Kusters[1], Fabian M. Munneke[1] and Jos J. M. Trienekens[2]
[1]*Department MST, Open University The Netherlands, Heerlen, The Netherlands*
[2]*Department IE&IS, Eindhoven University of Technology, Eindhoven, The Netherlands*

Keywords: Lean Techniques, Defect Injection.

Abstract In this paper we will focus on the impact that lean may have in preventing the injection of defects. We will research the impact of a number of lean techniques on defect injection factors. Date have been obtained from a single large Dutch governmental organization which has been using lean techniques routinely for more than three years. To investigate the impact of lean on defect injection we developed a survey which focused on the perceptions of the software developers of this organisation. The results suggest that the link between lean techniques and factors influencing defect injection is real and they explain to a certain extent the positive impact of the usage of lean techniques on software productivity.

1 INTRODUCTION

The notion of 'lean' development, first developed by Toyota in a setting of complex repetitive production (Womack et. al., 2008), has been moving towards other types of production. Software engineering is one of those. We see a significant focus both in literature and practice on the application of lean concepts to the field of software engineering. The 'translation' of lean concepts developed in complex, but repetitive production of physical goods, to the admittedly complex non-repetitive and not physical production of software, is not immediately evident but has been made (Poppendieck and Poppendieck, 2003).

Suggestions in literature are, that this move will lead to improvements in both productivity and defect reduction. However, Jonsson (2012), in a structured review of empirical research, states that the actual evidence for this is sparse and unconvincing.

In this paper we will take a closer look at the impact that lean may have on preventing the injection of defects. More specifically, we will look at the perceived impact of a number of lean techniques on those factors that during development influence the injection of defects in software. Finding such relationships will provide a further step toward showing empirically the existence of effects of lean on software engineering. We choose to focus on defect injection since lean aims at reducing waste and one of the most obvious forms of waste in software development is the injection of, the searching for and the solution of defects. A recent study (Cambridge 2013) estimates that software bugs cost the global economy $312 billion per year.

To investigate the impact of lean on defect injection we developed a survey which was deployed in a large Dutch governmental organization which has been using lean routinely for more than three years. Although the research results on this impact are restricted to perceptions, they suggest that the link between lean techniques and factors influencing defect injection are real and they explain to a certain extent the positive impact of lean on software development productivity.

In the next section current work is discussed, followed by the research methodology. Since this methodology requires a set of defect injection factors, the literature survey executed to identify these is described next, followed by the design of the survey and its results. The paper finishes with a discussion of results and conclusions.

2 CURRENT WORK

The notion that lean will impact defect injection (positively) has been claimed since the transfer of the

J. Kusters R., M. Munneke F. and J. M. Trienekens J..
The Impact of Lean Techniques on Factors Influencing Defect Injection in Software Development.
DOI: 10.5220/0005400204120419
In *Proceedings of the 17th International Conference on Enterprise Information Systems* (ICEIS-2015), pages 412-419
ISBN: 978-989-758-097-0

notion of lean to software development. Schulmeyer (1990) reports on the application of ideas for achieving zero defects from manufacturing industry to software development. He focused on defect prevention, and estimates that savings of up to 40% are possible. Poppendieck and Poppendieck (2003) also named defects as one of the seven forms of waste that lean should attack. Middleton (2001) stated that lean will encourage looking at underlying problems behind the injection of defects, resulting in finding them before they become failures. Cook and Semouchtchak (2004) also state that implementing lean should reduce the number of defects, but also recognize it requires a long and complex transformation process.

As to how lean will accomplish this effect, Mehta et al (2008) state that reducing system complexity where possible, and delivering in smaller increments (both possible effects of lean), should contribute. They also point out the positive effect of lean techniques, such as root cause analysis, which allow the identification and prevention of root causes. Similar claims are made by Card (2006) and Jalote and Agrawal (2005). Cottyn et al (2008) state that application of lean techniques such as “pull” and “standardization” will impact defect injection. A similar claim is made for the lean technique "kanban", which also should support early identification of defects (Ikonen, 2010), and Ikonen et al (2010). Petersen and Wohlin (2010) investigated a method SPI-LEAM that supports the identification of defects in development.

Finally, some studies indicate that the reduction of defect injection actually occurs. Middleton and Joyce (2012) show a decrease in the number of detected defects of 24%. McKinsey (2010) claim a decrease of 20 to 45% of defects detected.

All in all there appears to be substantial literature claiming an effect of lean on defect injection. However in a recent structural literature review (Jonsson, 2010) states that the actual evidence for the effect of lean is still sparse and not always convincing. Also, although in some papers the effect of lean is detailed to the effect of underlying techniques, the relationship between lean techniques and lean effectiveness is not well researched. Only two papers address this issue. McConnel (1997) shows that removal of non-required functionality reduces ‘complexity’, as is the case for simplification of functionality. Mehta et al (2008) show that frequent delivery of smaller functionality will impact both ‘complexity’ and ‘lack of traceability’. Both complexity (Fenton, 1999) and lack of traceability (O’Neill, 1997) are recognized as defect injector factors. Therefore these two papers provided limited support for explaining the impact of lean. We propose to contribute to the explanation of the impact of lean.

3 APPROACH

In principle for this type of research we can choose between measuring actual impact and capturing perceptions of an improvement by surveying developers. Obtaining accurate measurements is exceedingly difficult. The environment in which software development is being executed is continuously changing, e.g. by the introduction of new methods and techniques, by changes in staff and of course by the fact that every system being developed is by definition new. Measurement will therefore capture the compounded impact of all of these changes, and not just that of implementation of lean. Distinguishing the impact of a single action from that of a number of others is not feasible.

That brings us to surveys as an instrument of research. In principle the same problem is present. Asking if lean has had an impact will force developers / practitioners to try to identify and distinguish the impact of all of these changes. Reliable answers are not likely. However, if questions can be formulated that are sufficiently concrete and tie in to the personal experience of the respondent, more reliable information can be gathered (Lenzner, 2011). So, instead of asking questions as “did the introduction of lean impact of defects injection” we decided to be more specific.

First, instead of asking about the impact of lean as a whole, we decided to focus on a limited number of well-known and well used techniques. We appreciate that lean is a coherent approach, where the added value is achieved by combining a number of specific techniques and where the impact of the whole is meant to exceed the sum of the impact of the individual elements. However, that still allows us to look at the (perceived) impact of individual techniques for understanding the impact of lean (see e.g. Rivera and Chen (2007).

Secondly, on the other side of the question we did not look at the (perceived) impact on the injection of defects. This was seen as too abstract and difficult a question. Here we went one step deeper and looked in literature for factors that are seen to influence the injection of defects in software. Take e.g. the notion of complexity. There is evidence in literature (see e.g. Fenton and Neil, 1999) that the complexity of the problem solution will impact the injection of defects.

And the impact of a lean technique on complexity is easier to assess by persons involved in the process.

Therefore we decomposed the single question: “does usage of lean reduce the number of defects injected into the software” into a number of easier sub-questions: “will usage of technique X impact defect injection factor Y”. This results in a fairly large number of questions. The number of sub-questions will explode rapidly if the number of techniques and the number of factors increase. In a survey, an increase in the number of questions will both tend to reduce the response rate as well as reducing the quality of the answers that are delivered. This implies that in order to obtain a decent response, we limited the number of sub-questions. The factors used result from a literature search. When compiling the results from such a literature search there is always a choice in the level of detail used. For this research, it was decided to aim for a higher level of abstraction, thus reducing the number of factors, while still striving for factors that were sufficiently concrete to result in reliable answers.

Similarly we choose to reduce the number of techniques used in the survey. The number and type of technique used is of course closely linked to the type of organization that is included in the survey. Two extremes are possible. One is a survey sent out to different organizations. A way to reduce the number of techniques in such a context is to have respondents select a limited number of techniques that they are familiar with from a larger list and generate the sub-questions based on this selection. Another approach is to select a single organization, identify a limited number of techniques in use in this organization and use these in the survey.

A choice here would be impacted by possibilities for a wider response and more options for generalizability when taking the first approach and more limited but probably more reliable results when focusing on a single organization. More reliable since more information on the experience with lean is available and we can assume that the technique referred to in the survey is actually in use. There is less scope for misunderstanding. Given that, as far as we could ascertain, this is the first survey of this type, we opted for the more limited, but hopefully more reliable, option of obtaining data from a single organization.

As a result, the research follows the steps:

a) identify a list of relevant defect injection factors
b) select a relevant organization where lean is used
c) identify a number of lean techniques that are in use in this organization
d) conduct a survey and analyze the results.

4 IDENTIFYING DEFECT INJECTION FACTORS

A literature survey was executed in order to identify a list of defect injection factors. The search was based on key-words supported by a snowball approach (forward and backward). This resulted in 7 papers (see table 1). The results from these papers were merged in a single list, which should not be too large, in order not to have a negative impact on response rate. On the other hand the factor should be sufficiently clear to allow people answering the sub-questions with a degree of confidence. The final selection made was principally based on the set provided by Travassos et al (1999) ‘incompleteness’, ‘ambiguity’, ‘inconsistency’, ‘incorrect fact’, and ‘extraneous information’.

They used more global terms, which we still assessed as being sufficiently specific for answering the sub-questions. Results from other sources were classified to the Travassos terminology if possible. The remaining terms were used as the basis for additions to the list of Travassos resulting in the following additions:

- Lack of traceability (O’Neill, 1997);
- Complexity (Fenton and Neil, 1999; Bennett and Wennberg, 2005);
- Lack of skills (Fenton and Neil, 1999; Leszak et al, 2002);
- Lack of knowledge (Leszak et al, 2002);
- Time pressure (Leszak et al, 2002).

5 IDENTIFYING TECHNIQUES AND SURVEY DESIGN

The survey was done in the software group of a large Dutch government organization. It supports the primary processes of the organization. For this a staff of over 1600 developers is available. Lean as a core development approach was rolled out in 2010. The research was done in 2013, giving us an organization with three years of experience which we deemed this to be sufficient.

First we identified the techniques most commonly used. For this, four coaches with responsibility for the lean methodology were interviewed independent from each other. They all agreed as to the two most used techniques: “Day-start” and “Week-start”. After that, as a third most used technique various forms of "Coaching" were mentioned. Finally three out of four lean coaches mentioned structured improvement

("Kaizen"), while the fourth respondent did not name a fourth technique. Giving this almost complete agreement we decided to proceed with these four techniques. Table 2 gives a description of the techniques used.

Based on the results so far, i.e ten defect injection factors and four lean techniques, a survey was developed and sent out to all (approximately 1600) members of staff. For each of the four techniques they were asked if they use this technique on a regular basis to filter out staff members with no experience of this technique. Next (for this technique) for each of the defect injection factors the question was asked: “to what degree do you think this technique contributes to the basic causes of software defects”.

For the answer options we selected a five point labelled Likert scale starting with “not” indicating no effect and ending with “excellent”. A sixth item “I don’t know” was added to prevent people from being ‘forced’ to give an opinion without actually having one. This survey was tested on a lean-coach to check for comprehensibility and ease of use. After a positive response it was sent out to all members of staff.

Some limitations to the study are obvious. Data from a single organization limits the validity of the results. This is partly compensated for by the size of the organization, the response and the experience available within the organization with its way of working. A second limitation is the specific set of techniques that is in use. "Kaizen" is generally recognized as a ‘lean’ technique and is one of the original 14 principles of the Toyota production system. The others are less known. However, both "Daystart" and "Weekstart" enable team communication and leverage team knowledge, and they both further effective and efficient teamwork which is seen as an important effect of ‘lean’ techniques. "Coaching" supports staff development, one of the Toyota principles. This shows that the set of techniques discussed is relevant.

Table 1: the mapping to identified factors.

Found in literature	Factor identified
(Travassos, et al., 1999)	
Omission	Incompleteness
Ambiguity	Ambiguity
Inconsistency	Inconsistency
Incorrect fact	Incorrectness
Extraneous information	External factors
(O'Neill, 1997)	
Lack of traceability	Lack of traceability
(Chulani & Boehm, 1999)	
Incorrect requirements	Incorrectness

Table 1: The mapping to identified factors (cont.).

Found in literature	Factor identified
Incomplete requirements	Incompleteness
Design defects	Incorrectness
Programming defects	Incorrectness
(Fenton, 1999)	
Difficulty of the problem	Complexity
Complexity of designed solution	Complexity
Programmer / analyst skill	Lack of skills
Design methods and procedures	Incorrectness
(Leszak, et al., 2002)	
Incorrect documentation	Incorrectness
Incomplete documentation	Incompleteness
Unclear documentation	Ambiguity
Change in coordination	Lack of skills
Lack of domain knowledge	Lack of knowledge
Lack of system knowledge	Lack of knowledge
Lack of tool knowledge	Lack of knowledge
Lack of process knowledge	Lack of knowledge
Individual error	Incorrectness
Time pressure	Time pressure
Management error	Incorrectness
Error caused by other products	External factors
Insufficient preparation	Incompleteness
Insufficient participatin	Incompleteness
(Bennet & Wennberg, 2005)	
Assumptions/ambiguities affecting the interpretation of customer descriptions of desired system behavior	Ambiguity
The difficulty in fully understanding the real-world environment in which the system will interact	Complexity
The difficulty in anticipating all of the possible modes and states that the system may encounter	Complexity
The difficulty in thoroughly validating and verifying requirements	Complexity
Capturing accurate, unambiguous representations of requirements in a written document	Incompleteness
Misinterpretation of system-level requirements	Incorrectness
The difficulty in verifying that the design has correctly implemented the requirements.	Complexity
(D’Ambros, et al., 2010)	
Design flaws	Incorrectness
Violations of design principles and practices	Incompleteness

Table 2: description of the techniques used in the survey.

Technique	Description
Day-start	A daily session of approximately 15 minutes where staff share the status of their work with direct co-workers
Week-start	A weekly session of approximately one hour where staff look at the results obtained last week and set targets for the upcoming week

Table 2: description of the techniques used in the survey (cont.).

Technique	Description
Coaching	Regular interactions between managers and staff to develop insight into current performance and options for improvement
Kaizen	An approach to implement process improvement during a short period of time by going through the cycle: define – measure – analyze – generate improvements – implement – control – secure

6 RESULTS OF THE SURVEY

The response of approximately 34% is high for a survey and gives confidence in the results. Of the respondents 89% indicate they use lean in some form. Taking this as the basis for relevant response, we see differences in usage between the techniques. Day-start is commonly in use (95.3%) as is Week-start (80.7%). Both Coaching and Kaizen are used less. For Kaizen the result is reasonable. Not everyone will regularly be involved in Kaizen type improvement projects. With 43.3% of staff members involved in this activity it shows a decent result. That little usage was made of Coaching is surprising, since the lean-coaches of the department all agreed on its importance. However, there was already during the discussions with these lean-coaches some confusion in this area, since many different names for the technique were mentioned. In discussion with the lean-coaches a term (Coaching) was selected that they felt would be understood by most people. This might have been over optimistic. However, a response of 26.3% is still large enough to be able to draw conclusions.

If respondents indicate an effect is present they can indicate the size of this effect. Answer options range from “moderate” to “excellent”. To get a better show the results, we added two notions. The first is a “significant effect” that adds responses of “average” and higher. The second is a “major effect” that adds responses of “good” and “excellent”. More detailed results are presented in table 5, where we give results (in percentages) for each combination of technique and defect injection factor. To facilitate reading this table, highest scores within a technique are indicated in bold.

Results for the sub-questions are summarized in tables 3 and 4. In table 3 we provide a first overview of average results per technique across defect injection factors. The first block gives the answers to the question “is there an impact of this technique”. Answer options are “no”, “don’t know”, and “yes”). The basis of this percentage is the number of respondents, indicating they use the technique, minus those with a "no" answer on the sub-question (resulting in a missing value). If we deduct also the answer “don’t know” from the response we see the result of those respondents that felt sufficiently secure to give an answer. This is presented in a further column “yes-2”.

7 DISCUSSION

Across all techniques and defect injection factors, an average of 61.8 % of respondents perceive the existence of an impact of the identified lean techniques on the identified defect injection factors. If they see such an effect, its degree is estimated at “significant” by 66.4% of respondents and as “major” by 30.8%. These results seem to support the notion that ‘lean’ positively impacts defect injection. It goes some way toward explaining the reason for the existence of this effect of lean.

If we look across techniques we can see that the existence of an impact of Kaizen, a technique aimed at structural continuous improvement, is perceived to be (on average across defect injection factors) highest at 70.6%. Also the degree of this impact (significant: 74.1%; major: 39.6%) scores very high. This is followed by both Coaching (impact: 63.5%; significant: 72.2%; major: 31.8%) and Day-start (impact: 61.2%; significant: 62.2%; major: 30.8%). They are definitely trailed by Week-start (impact: 52.0%; significant: 57.2%; major: 20.9%), which shows a decidedly lower result. Similarly, if we look at the defect injection factors to see where the impact is seen most across techniques, “external factors” and “time pressure” score high, both with existence of impact and degree of impact, with “lack of skills” and “lack of knowledge” following close behind. Day-start is a daily session of about 15 minutes aimed at discussing the performance of the previous day and the goals of the current day, identification of improvement options and work allocation. Given this purpose, the major impact on ‘time pressure’ that is identified should not come as a surprise. Planning work based on the current situation is its main purpose and this should indeed isolate the team from undue expectations. The impact on ‘external factors’ is more surprising, since the technique is not specifically aimed at these. A possible explanation could be that this isolation also extends to other issues, which are handled of line. The higher impact on ‘lack of knowledge’, ‘lack of skills’ and ‘incompleteness’ could be explained by the

Table 3: Average results per technique across injection factors (in percentages).

Technique	**Is there an impact**				**Degree of impact if there is an impact**					
	no	**don't know**	**yes**	**yes-2**	**Moderate**	**Average**	**Good**	**Excellent**	**Significant**	**Major**
Daystart	32.1	17.1	50.8	61.2	37.8	31.4	25.5	5.3	62.2	30.8
Week-start	40.5	15.7	43.8	52.0	42.8	36.3	18.6	2.4	57.2	20.9
Coaching	32.5	11.0	56.5	63.5	27.8	40.4	21.9	10.0	72.2	31.8
Kaizen	23.1	21.6	55.3	70.6	25.9	34.5	27.2	12.4	74.1	39.6
Over all	32.1	16.3	51.6	61.8	33.6	35.6	23.3	7.5	66.4	30.8

Table 4: Average impact on defect injection factors across techniques (in percentages).

Factor	**Is there an impact**				**Degree of impact**					
	no	**don't know**	**yes**	**yes-2**	**Moderate**	**Average**	**Good**	**Excellent**	**Significant**	**Major**
Incompleteness	33.4	15.6	51.0	60.5	35.1	33.8	**26.2**	4.9	64.9	31.1
Incorrectness	34.8	15.6	49.6	58.9	36.3	38.0	21.4	4.2	63.7	25.6
Inconsistency	34.1	15.9	50.0	59.7	39.2	35.7	21.0	4.1	60.8	25.1
Ambiguity	34.0	16.6	49.4	59.4	**39.5**	38.0	17.7	4.8	60.5	22.5
Complexity	32.3	16.2	51.6	61.7	34.3	33.5	25.0	7.2	65.7	32.3
Time pressure	28.9	15.6	**55.6**	65.8	28.6	33.0	26.0	**12.4**	71.4	**38.4**
Lack of traceability	**35.5**	17.6	46.9	57.1	36.6	36.9	21.6	4.9	63.4	26.5
Lack of skills	31.8	15.6	52.7	62.4	29.8	35.1	23.8	11.3	70.2	35.1
Lack of knowledge	29.6	15.3	55.1	65.1	29.4	33.8	24.6	12.1	70.6	36.8
External factors	26.3	**19.2**	54.5	**67.6**	26.9	**38.5**	25.7	8.9	**73.1**	34.6

communication among the team that is fostered by this frequent contact. The purpose of a Day-start is also asking or offering knowledge or availability.

Week-start is a weekly session of about an hour with similar purposes to the Day-start, now only with a time frame of one week. Given the same approximate purpose we see a similar result, with ‘time pressure’ and ‘external factors’ scoring highest, again followed by ‘lack of knowledge’, ‘lack of skills’ and ‘incompleteness’. However, we also see that the impact of this technique is rated decidedly lower when compare to that of day-start. This is surprising. A possible explanation could be that a time frame of one week is too much to provide the ‘isolation’ effects postulated above. However, we also see that participation for the Day-start (95.3%) is higher than that for the Week-start (80.7%) which is still high but might indicate a lower acceptance of the practice and its results.

Coaching and performance dialogues are aimed at coaching the employee of discussing his – hers performance. Each employee is supposed to have such a talk at least once per week. Indicated participation for this technique is low with 26.3%. As stated a possible explanation might be that some confusion exists about the name of the activity. Another reason might be that the recipients of the regular coaching discussions no longer see it as a separate technique but as a part of the common daily activities. The technique is then so embedded in the way of working of the organization that only (team-) managers are likely to notice it. However as stated, the perceived impact of this technique, aimed at continuous improvement, is high. Impact and degree of impact score high for both by ‘lack of knowledge’, ‘lack of skills’, as is to be expected with a technique aimed at improving the human capability. Somewhat more surprising, ‘time pressure’ also received a high score, maybe because of time management issues being taken along with the coaching. The factor ‘external factors’ scores very high in input, but fairly low in its degree of impact. Since coaching is aimed at the improvement of the human resource, and not specifically at the management of work processes this might explain the lower degree of impact.

Kaizen, a technique aimed at structural continuous improvement, is aimed at developing and implementing concrete improvements. In a (limited) number of sessions a small team guided by a kaizen facilitator follows a structured process aimed at developing and implementing an improvement for an identified issue. Since this is usually more product and process oriented, factors associated with this are more likely to be impacted as opposed to the more

Table 5: results / technique / defect injection factor (%).

Technique / Defect injection factor	Is there an impact?	Degree of impact	
		Signifi-cant	Major
DAY START			
Incompleteness	61.7	62.1	33.2
Incorrectness	58.1	57.6	26.3
Inconsistency	60.2	55.4	26.0
Ambiguity	56.4	56.3	20.5
Complexity	57.0	61.0	30.8
Time pressure	**71.6**	71.7	**45.1**
Lack of traceability	52.6	55.5	23.1
Lack of skills	60.3	65.4	32.7
Lack of knowledge	67.4	64.2	33.6
External factors	67.2	**72.4**	36.7
WEEK START			
Incompleteness	49.2	53.5	20.8
Incorrectness	47.2	55.3	18.4
Inconsistency	47.7	49.0	15.7
Ambiguity	48.1	50.6	13.0
Complexity	53.0	52.7	18.3
Time pressure	60.0	64.1	**31.3**
Lack of traceability	45.4	55.6	18.8
Lack of skills	53.1	60.5	22.1
Lack of knowledge	55.9	63.0	23.2
External factors	**60.3**	**67.9**	27.8
COACHING			
Incompleteness	62.7	65.2	21.7
Incorrectness	60.9	62.7	17.9
Inconsistency	58.2	67.2	17.2
Ambiguity	61.1	62.1	16.7
Complexity	62.4	70.6	32.4
Time pressure	66.7	81.1	**50.0**
Lack of traceability	58.3	68.3	22.2
Lack of skills	**68.5**	82.9	51.3
Lack of knowledge	67.6	**84.0**	53.3
External factors	68.2	78.1	35.6
KAIZEN			
Incompleteness	68.5	78.8	**48.7**
Incorrectness	69.3	**79.1**	40.0
Inconsistency	72.7	71.7	41.7
Ambiguity	72.0	72.9	39.8
Complexity	74.4	78.7	47.5
Time pressure	65.0	68.9	27.4
Lack of traceability	72.2	74.4	41.9
Lack of skills	67.7	72.1	34.2
Lack of knowledge	69.5	71.1	36.8
External factors	**74.7**	73.9	38.3

human oriented factors targeted by coaching. This is indeed confirmed by the results. The factors ‘incompleteness’, ‘incorrectness’, ’inconsistency’, ‘complexity’, and ‘lack of traceability’ are the ones that are identified by the respondents as having a high degree of impact (‘major’ above 40%) with the other factors scoring (sometimes well) below that.

The one defect injection factor that had not surfaced in any of the techniques as having a major degree of impact is ‘ambiguity’. It scored not only lowest overall, but also lowest for three out of four of the individual techniques (based on the ‘major’ degree of impact). Apparently this is too complex a concept and cannot be easily dealt with by these fairly straightforward techniques. Also, ‘ambiguity’ is likely to be caused to a large degree by the interaction with the user community, which is not involved in either of these techniques, suggesting the need for other means to deal with this issue.

8 CONCLUSIONS

Notions of ‘lean’ and associated techniques are being used more and more in practice in software engineering. Literature suggests that adoption of these practices could lead to improved productivity and a reduction of the number of defects injected. Literature also suggests that the actual proof for this is still not very convincing. In this paper we added to the body of knowledge by looking at a detailed level at the possible impact of a number of lean techniques on a number of defect injection factors. A survey, conducted within a single large software development organization, shows that respondents indeed perceive that such an impact is present. Different techniques have to a different degree an expected impact on the defect injection factors. This provides additional proof for the effectiveness of lean in software engineering and also gives some more detailed explanation for this effect.

Having said this, the research has a number of limitations. The context is a single (albeit large) organization. Also, the research does not look at the comprehensive concept of lean, but only at a limited number of techniques. Both issues limit the generalizability of the results, suggesting immediate areas for further research.

Another limitation is due to the fact that the results are based on subjective assessments of the respondents taking part. They are in principle knowledgeable, and the questions were tailored to be answerable. Also the ‘don’t know’ option, which was used on average by 16% of respondents should weed out uninformed answers. But still people have to assess the impact of techniques they use on factors that are fairly abstract. We do indeed see a fairly strong respondent bias. People tend to be more optimistic / pessimistic across the board, resulting in a high correlation between the individual answers. However, even when taking into account this effect, differences between different techniques are considerable and reasonable explanations for these differences can be provided. All in all the main direction of the results is in our mind sufficiently reliable and therefore relevant and warrants further research.

REFERENCES

Bennett, T. L., and Wennberg, P.W. "Eliminating embedded software defects prior to integration test." *Crosstalk, Dec., pp. 13-18* (2005).

Cambridge. http://www.prweb.com/releases/2013/1/prweb1029
8185.htm (2013).

Card, D. N. "Myths and strategies of defect causal analysis." *Proceedings of Pacific Northwest Software Quality Conference.* 2006.

Chulani, S., and Boehm, B. Modeling software defect introduction and removal: COQUALMO (COnstructive QUALity MOdel). *Technical Report USC-CSE-99-510*, University of Southern California, Center for Software Engineering, 1999.

Cook, J., & Semouchtchak, V. Lean object-oriented software development. *SAM Advanced Management Journal, 69(2),* 2004.

Cottyn, J., Stockman, K., and Landeghem, H. van. "The Complementarity of Lean Thinking and the ISA 95 standard." *WBF 2008: Bridging the Divide between IT and Manufacturing*. 2008.

D'Ambros, M., Bacchelli, A., and Lanza, M. "On the impact of design flaws on software defects." Quality Software (QSIC), *2010 10th International Conference on. IEEE*, 2010.

Fenton, N. E., and Neil, M. "A critique of software defect prediction models." *Software Engineering, IEEE Transactions on 25.5* (1999): 675-689.

Ikonen, M., Kettunen, P., Oza, N., and Abrahamsson, P. "Exploring the sources of waste in kanban software development projects." Software Engineering and Advanced Applications (SEAA), *2010 36th EUROMICRO Conference on. IEEE*, 2010.

Ikonen, M. "Leadership in Kanban software development projects: A quasi-controlled experiment." Lean Enterprise *Software and Systems. Springer Berlin Heidelberg, 2010*. 85-98.

Jalote, P., and Agrawal, N. "Using defect analysis feedback for improving quality and productivity in iterative software development." Information and Communications Technology, 2005. Enabling Technologies for the New Knowledge Society: *ITI 3rd International Conference on. Ieee,* 2005.

Jonsson, H. "Lean Software Development: A Systematic Review." *IDT Miniconference on Interesting Results in Computer Science and Engineering* (IRCSE) (2010).

Lenzner, T. A Psycholinguistic Look at Survey Question Design and Response Quality. *PhD-thesis* Universität Mannheim, 2011.

Leszak, M, Perry, D. E., and Stoll, D. "Classification and evaluation of defects in a project retrospective." *J. of Systems and Software 61.3, pp. 173-187*, 2002.

McConnell, S. (1997). Archieving Leaner Software. IEEE Software(November/December).

McKinsey & Company. (2010). Ten Core Principles for Applying Lean ADM: McKinsey & Company, 2010.

Mehta, M., Anderson, D., and Raffo, D. "Providing value to customers in software development through lean principles." *Software Process: Improvement and Practice 13.1* (2008): 101-109.

Middleton, P., and Joyce, D. "Lean software management: BBC Worldwide case study." Engineering Management, *IEEE Trans. on 59.1* (2012): 20-32.

Middleton, P. "Lean software development: two case studies." Software Quality J. 9.4 (2001) 241-252.

O'Neill, D. "Issues in software inspection." Software, IEEE 14.1 (1997): 18-19.

Petersen, K., and Wohlin, C. "Software process improvement through the Lean Measurement (SPI-LEAM) method." *J. of systems and software 83.7* (2010) 1275-1287.

Poppendieck, M, and Poppendieck, T. Lean software development: *an agile toolkit. Addison-Wesley Professional*, 2003.

Rivera, L., and Chen, F.F. "Measuring the impact of Lean tools on the cost–time investment of a product using cost–time profiles." *Robotics and Computer-Integrated Manufacturing 23.6 (2007)*: 684-689.

Schulmeyer, G. G. Zero defect software. McGraw-Hill, Inc., 1990.

Travassos, G., Shull, F., Fredericks, M., & Basili, V. R. "Detecting defects in object-oriented designs: using reading techniques to increase software quality." *ACM Sigplan Notices. Vol. 34. No. 10. ACM, (1999)*: 47-56.

Womack, J. P., Jones, D. T, and Roos D. *The machine that changed the world*. Simon and Schuster, 2008.