An Empirical Study of Lead-Times in Incremental and Agile Software Development

Kai Petersen

Blekinge Institute of Technology Box 520,
SE-37225 Ronneby, Sweden
Ericsson AB,
Sweden

Abstract. Short lead-times are essential in order to have a first-move advantages and to be able to react on changes on a fast-paced market. Agile software development is a development paradigm that aims at being able to respond quickly to changes in customer needs. So far, to the best of our knowledge no empirical study has investigated lead-times with regard to different aspects (distribution between phases, difference of lead-time with regard to architecture dependencies, and size). However, in order to improve lead-times it is important to understand the behavior of lead-times. In this study the lead-times of a large-scale company employing incremental and agile practices are analyzed. The analysis focuses on 12 systems at Ericsson AB, Sweden.

1 Introduction

Lead-time (also referred to as cycle-times) is the time it takes to process an order from the request till the delivery [1]. An analysis and improvement of lead-time is highly relevant. Not being able to deliver in short lead-times leads to a number of disadvantages on the market, identified in the study of Bratthall et al. [2]: (1) The risk of market lock-out is reduced [3]. Bratthall et al. [2] provided a concrete example for that where one of the interviewees reported that they had to stall the introduction of a new product because the competitor was introducing a similar product one week earlier; (2) An early enrollment of a new product increase probability of market dominance [4]. One of the participants in the study of Bratthall et al. [2] reported that due to introducing a product three months after a competitor the company is holding 30 % less of the world market in comparison to the market leader; (3) Another benefit of being early on the market is that the product conforms more to the expectations of the market [5]. This is due to the market dynamics. Petersen et al. [6] found that a large portion (26 %) of gathered requirements are already discarded during development. Furthermore, the long lead-time provides a time-window for change requests and rework.

The review of literature revealed that, to the best of our knowledge, an empirical analysis of lead-times in incremental and agile development has not been conducted so far. However, as there is an increasing number of companies employing incremental and agile practices it is important to understand lead-time

J. Münch, Y. Yang, and W. Schäfer (Eds.): ICSP 2010, LNCS 6195, pp. 345–356, 2010.

behavior. The studied company intended to determine target levels for lead-times. The open question at the company was whether requirements should have different target levels depending on the following factors:

1. The distribution of lead-times between different phases.
2. The impact a requirement has on the systems. The impact is measured in terms of number of affected systems. Here we distinguish between single-system requirements (a requirement only affects one system) and multi-system requirements (a requirement affects at least two systems).
3. The size of the requirements.

This study investigated the effect of the three factors on lead-time. It is important to stress that existing work indicates what outcomes can be expected for the different factors, the expected results being presented in the related work. However, the outcome to be expected was not clear to the practitioners in the studied company. Hence, this study sets out with formulating a set of hypotheses related to the factors without assuming a specific outcome of the hypotheses prior to analyzing them.

The research method used was an industrial case study of a company developing large-scale systems in the telecommunication domain. The quantitative data was collected from a company proprietary system keeping track of the requirements flow throughout the software development lifecycle.

The remainder of the paper is structured as follows. Section 2 presents related work. The research method is explained in Section 3. The results of the study are shown in Section 4. Section 5 discusses the results. Section 6 concludes the paper.

2 Related Work

Petersen et al. [6] present lead-times for waterfall development, showing that the majority of the time (41 %) is spent on requirements engineering activities. The remaining time was distributed as follows: 17 % in design and implementation, 19 % on verification, and 23 % on the release project. As in agile software development the main activitis should be coding and testing [7] the literature would suggest that those are the most time consuming activities.

Petersen and Wohlin [8] investigated issues hindering the performance of incremental and agile development. When scaling agile the main issues are (1) complex decision making in the requirements phase; (2) dependencies of complex systems are not discovered early on; and (3) agile does not scale well as complex architecture requires up-front planning. Given this qualitative result the literature indicates that with increase of requirements impact the lead-time should increase. For example, if a requirement can only be deliverable when parts of it are implemented across several systems a delay in one system would lead to prolonged lead-times for this requirement.

Harter et al. [9] identified that lines of code (size) is a predictor for cycle time. This was confirmed by [10] who found that size was the only predictor for lead-time. Hence, from the related work point of view an increase of size should lead to an increase of lead-time.

Collier [11] summarizes a number of issues in cycle time reduction and states: (1) size prolongs lead-time, and (2) dependencies influence lead-time.

Carmel [12] investigated key success factors for achieving short lead-times. The finding shows that team factors (small team size, cross-functional teams, motivation) are critical. Furthermore, an awareness of lead-times is important to choose actions specifically targeted towards lead-time reduction. However, it is important to take quality into consideration when taking actions towards lead-time reduction.

None of the lead-time studies focuses on agile development, and hence raising the need for empirical studies on lead-time in an incremental and agile development context.

3 Research Method

The research method used was a quantitative case study, the case being the telecommunication company Ericsson AB. The systems studied were developed in Sweden and India.

3.1 Research Context

The research context is important to describe in order to know to what degree the results of the study can be generalized [13]. Table 1 shows the context elements for this study. The analysis focused on in total 12 systems of which 3 systems are independent. The remaining nine systems belong to a very large communication system and are highly dependent on each other. Thus, all requirements belonging to the independent systems are treated as single-system requirements. The same applies to requirements only affecting one of the nine dependent systems.

The process of the company is shown in Figure 1. Requirements from the market are prioritized and described as high level requirements (HLR) in form

Table 1. Context Elements

Element	Description
Maturity	All systems older than 5 years
Size	Large-scale system with more than 5,000,000 LOC
Number of systems	9 dependent systems (indexed as A to I) and 3 independent (indexed as J to K)
Domain	Telecommunication
Market	Highly dynamic and customized market
Process	On the principle level incremental process with agile practices in development teams
Certification	ISO 9001:2000
Practices	Continuous integration; Internal and external releases; Time-boxing with sprints; Face-to-face interaction (stand-up meetings, co-located teams); Requirements prioritization with metaphors and Detailed requirements (Digital Product Backlog); Refactoring and system improvements

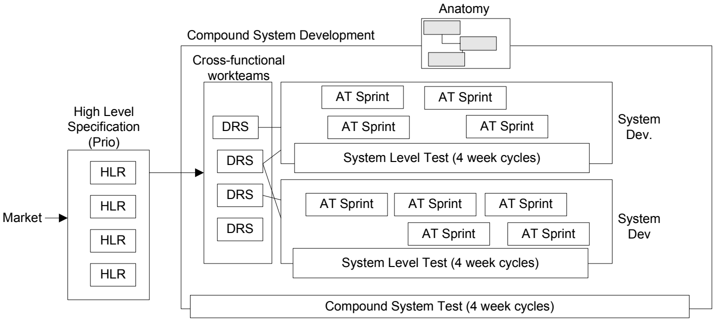
Anatomy
Compound System Development
Cross-functional workteams
High Level Specification (Prio)
HLR
HLR
HLR
HLR
Market
DRS
DRS
DRS
DRS
AT Sprint
AT Sprint
AT Sprint
AT Sprint
System Level Test (4 week cycles)
System Dev.
AT Sprint
AT Sprint
AT Sprint
AT Sprint
AT Sprint
System Level Test (4 week cycles)
System Dev
Compound System Test (4 week cycles)

Fig. 1. Development Process

of metaphors. These are further detailed by cross-functional work teams (people knowing the market and people knowing the technology) to detailed requirements specifications (DRS). The anatomy of the system is used to identify the impact of the HLR on the system. The impact determines how many systems are affected by the requirement. A requirement affecting one system is referred to as single-system requirement, while a requirement affecting multiple system is referred to as a multi-system requirement. Within system development agile teams (ATs) implement and unit test the requirements within four week sprints. The teams deliver the requirements to the system level test to continuously verify the integration on system level every four weeks. Furthermore, the system development delivers their increments to the compound system test, which is also integrating in four week cycles. Requirements having passed the test are handed over to the release projects to be packaged for the market.

3.2 Hypotheses

The hypotheses are related to differences between multi- and single-system requirements, the distribution of the lead-time between phases, and the difference between sizes of requirements. As mentioned earlier the goal is not to reject the null hypotheses, but to determine whether the different factors lead to differences in lead-time. In the case of not rejecting the null-hypotheses the factors do not affect the lead-time, while the rejection of the null-hypotheses implies that the factors effect lead-time. The following hypotheses were made:

- *Phases:* There is no difference in lead-time between phases ($H_{0,phaswe}$) opposed to there is a is a difference ($H_{a,phase}$).
- *Multi vs. Single:* There is no difference between multi- and single-system requirements ($H_{0,mult}$) opposed to there is a difference ($H_{a,mult}$).
- *Size:* There is no difference between sizes ($H_{0,size}$) opposed to there is a difference ($H_{a,size}$).

3.3 Data Collection

The lead-time is determined by keeping track of the duration the high level requirements reside in different states. When a certain activity related to the high-level requirement is executed (e.g. specification of the requirement) then the requirement is put into that state. For the tracking of lead-times a time-stamp was captured whenever a requirement enters a state, and leaves a state.

The lead-time data was collected from an electronic Kanban solution where the requirements can be moved between phases to change their state. The system can be edited over the web, showing the lists of the requirements and in which phase they are. Whenever a person is moving a requirement from one phase to another, a date is entered for this movement. The requirements go through the following states:

- *State Detailed Requirements Specification:* The state starts with the decision to hand over requirement to the cross-functional work-team for specification, and ends with the hand-over to the development organization.
- *State Implementation and Unit Test:* The state starts with the hand-over of the requirement to the development organization and ends with the delivery to the system test.
- *State Node/System Test:* The state starts with the hand-over of the requirement to the system/node test and ends with the successful completion of the compound system test. The time includes maintenance for fixing discovered defects.
- *State Ready for Release:* The state starts when the requirement has successfully completed the compound system test and thus is ready for release to the customer.

From the duration the requirements stay in the states the following lead-times were calculated:

- LT_a: Lead-time of a specific activity a based on the duration a requirement resided in the state related to the activity.
- LT_{n-a}: Lead-time starting with an activity a and ending with an activity n. In order to calculate this lead-time, the sum of the durations of all activities to work on a specific high-level requirement were calculated.

Waiting times are included in the lead-times. The accuracy of the measures is high as the data was under regular review due to that the electronic Kanban solution was used in daily work, and the data was subject to a monthly analysis and review.

3.4 Data Analysis

The descriptive statistics used were box-plots illustrating the spread of the data. The hypotheses were analyzed by identifying whether there is a relationship between the variable lead-time and the variables phases ($H_{0,phase}$) and system impact ($H_{0,mult}$). For the relationship between lead-time and system impact the Pearson correlation was used to determine whether there is a linear relationship, and the Spearman correlation to test whether there is a non-linear relationship. For the relationship between lead-time and phase ($H_{0,phase}$) no correlation was used as phase is a categorical variable. In order to capture whether phase leads to variance in lead-time we test whether specific phases lead to variance in lead-time, this is done by using stepwise regression analysis. Thereby, for each category a dummy variable is introduced.

If there is a relationship between the variables (e.g. between system impact and lead-time) this would mean that the system impact would be a variable explaining some of the variance in the lead-time. The hypotheses for size ($H_{0,size}$) was only evaluated using descriptive statistics due to the limited number of data points.

3.5 Threats to Validity

Four types of threats to validity are distinguished, namely conclusion validity (ability to draw conclusions about relationships based on statistical inference), internal validity (ability to isolate and identify factors affecting the studied variables without the researchers knowledge), construct validity (ability to measure the object being studied), and external validity (ability to generalize the results) [14].

Conclusion Validity: The statistical inferences that could be made from this study to a population are limited as this study investigates one particular case. In consequence, no inference statistics for comparing sample means and medians with regard to statistical significance are used. Instead correlation analysis was used, correlation analysis being much more common for observational studies such as case studies. For the test of hypotheses $H_{0,phase}$ we used stepwise regression analysis, regression analysis being a tool of statistical inference. Hence, the interpretation of regression analysis in observational studies has to be done with great care as for a single case study no random sampling with regard to a population has been conducted. The main purpose of the regression was to investigate whether the category leads to variance in lead-time at the specific company. That is, companies with similar contexts might make similar observations, but an inference to the population of companies using incremental and agile methods based on the regression would be misleading.

Internal Validity: One threat to internal validity is the objectivity of measurements, affected by the interpretation of the researcher. To reduce the risk the researcher presented the lead-time data during one meeting and discussed it with peers at the company.

Construct Validity: Reactive bias is a threat where the presence of the researcher influences the outcome of the study. The risk is low due that the researcher is employed at the company, and has been working with the company for a couple of years. Correct data is another threat to validity (in this case whether the lead-time data is accurately entered and up-to-date). As the system and the data entered support the practitioners in their daily work the data is continuously updated. Furthermore, the system provides an overview of missing values, aiding in keeping the data complete which reduces the risk.

External Validity: One risk to external validity is the focus on one company and thus the limitation in generalizing the result. To reduce the risk multiple systems were studied. Furthermore, the context of the case was described as this makes explicit to what degree the results are generalizable. In this case the results apply to large-scale software development with parallel system development and incremental deliveries to system testing. Agile practices were applied on team-level.

4 Results

4.1 Time Distribution Phases

Figure 2 shows the box-plots for lead-times between phases P1 (requirements specification), P2 (implementation and unit test), P3 (node and system test), and P4 (release projects). The box-plots do not provide a clear indication of the differences of lead-time distribution between phases as the box-plots show high overlaps between the phases.

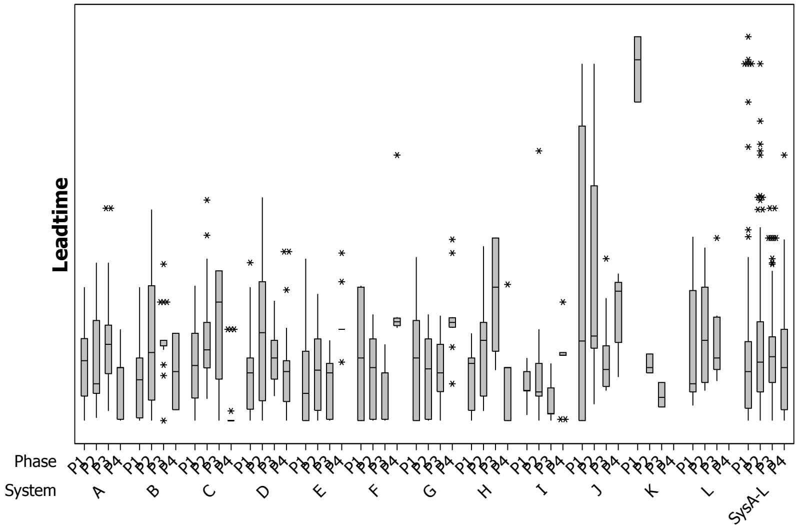

Fig. 2. Comparison of Lead-Times between Phases

Table 2 provides an overview of the statistical results of the correlations between phase and lead-time across systems. The stepwise regression shows that Dummy 4 was highly significant in the regression. Though, the overall explanatory power (which was slightly increased by the introduction of the Dummy 1) is still very low and accounts for 1.14 % of the variance in lead-time (R^2). Hence, $H_{0,phase}$ cannot be rejected with respect to this particular case.

Table 2. Results for Distribution of Lead-Time Phases, N=823

Step	One	Two
Constant	49.91	52.03
Dummy 4 (P4)	-8.9	-11.0
t-value	-2.5	-2.94
p-value	0.013	0.003
Dummy 1 (P1)	-	-6.4
t-value	-	-1.79
p-value	-	0.074
Expl. power R^2	0.0075	0.0114

4.2 Multi-System vs. Single-System Requirements

Figure 3 shows single system requirements (labeled as 0) in comparison to multi-system requirements (labeled as 1) for all phases and for the total lead-time. The box-plots indicate that there is no difference between single and multi-system requirements. In fact, it is clearly visible that the boxes and the median values are on the same level.

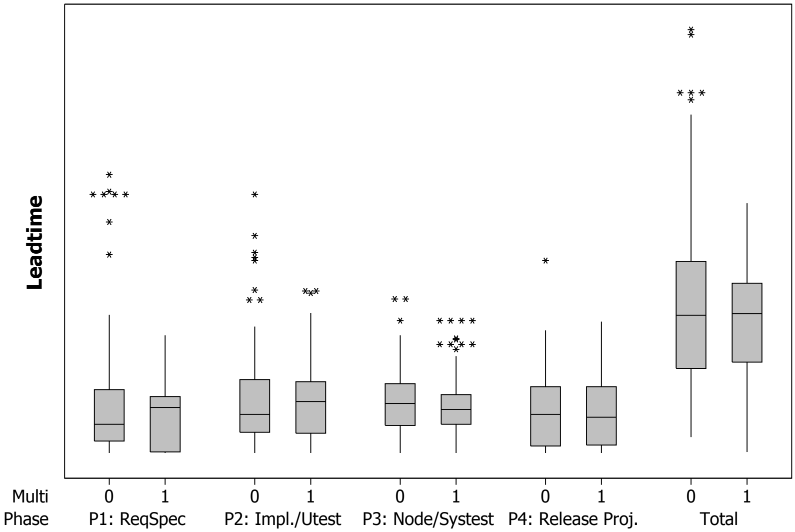

Fig. 3. Single-System (label=0) vs. Multi-System Requirements (label=1)

As there is no observable difference between system impacts we investigated whether the number of systems a requirement is dependent on has an influence on lead-time. As shown in the box-plots in Figure 4 the lead-time does not increase with a higher number of affected systems. On the single system side even more outliers to the top of the box-plot can be found for all phases.

The correlation analysis between system impact and lead-time across systems is shown in Table 3. The table shows that the correlations are neither close to

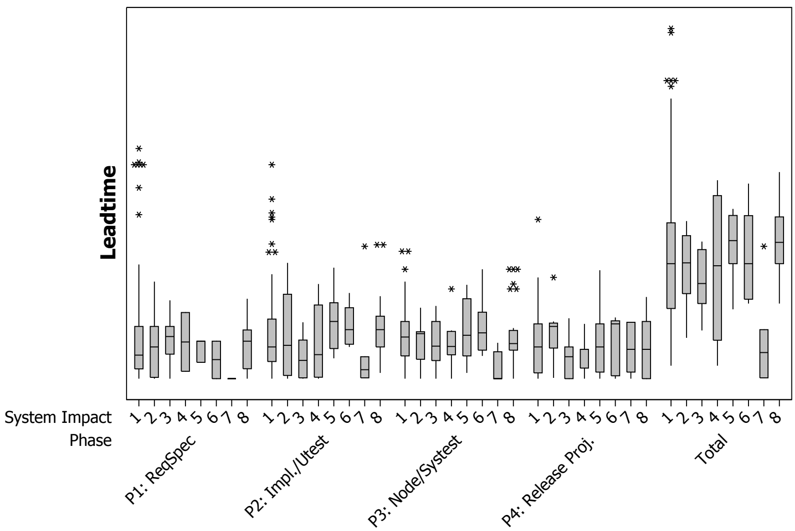

Fig. 4. Difference for System Impact (Number of systems a requirement is affecting, ranging from 1-8)

Table 3. Test Results for $H_{0,multi}$, N=823

Statistic	Value	p
Pearson (φ)	-0.029	0.41
Spearman (ρ)	0.074	0.57
Expl. power (R^2)	0.001	

$+/-1$, and are not significant. Hence, this indicates that the two variables do not seem to be related in the case of this company, leading to a rejection of $H_{0,multi}$.

4.3 Difference between Small / Medium / Large

Figure 5 shows the difference of lead-time between phases grouped by size, the size being an expert estimate by requirements and system analysts. The sizes are defined as intervals in person days, where $Small(S) := [0; 300]$, $Medium(M) := [301; 699]$, and $Large(L) := [700; \infty]$. No correlation analysis was used for analyzing this data as three groups only have two values, namely P2-Large, P3-Medium, and P3-Large. The reason for the limitation was that only recently the requirements were attributed with the size.

However, the data already shows that the difference for size seems to be small in the phases requirements specification and node as well as compound system testing. However, the size of requirements in the design phase shows a trend of increased lead-time with increased size.

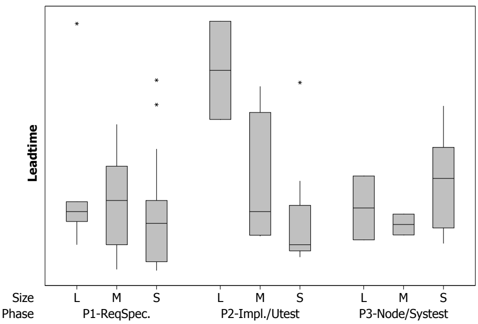

Fig. 5. Difference for Small, Medium, and Large Requirements

5 Discussion

This section presents the practical and research implications. Furthermore, the reasons for the results seen in the hypotheses tests are provided. The explanations have been discussed within an analysis team at the studied companies and the team agreed on the explanations given.

5.1 Practical Implications

No Difference in Lead-Times Between Phases: One explanation for the similarities of lead-times between phases is that large-scale system requires more specification and testing, and that system integration is more challenging when having systems of very large scale. Thus, systems documentation and management activities should only be removed with care in this context as otherwise there is a risk of breaking the consistency of the large system. Furthermore, there is no single phase that requires a specific focus on shortening the lead-time due to that there is no particularly time-consuming activity. Hence, in the studied context the result contradicts what would be expected from the assumptions made in literature. A consequence for practice is that one should investigate which are the value-adding activities in the development life-cycle, and reduce the non-value adding activities. An approach for that is lean software development [15].

No difference in Multi vs. Single System Requirements: The number of dependencies a requirement has does not increase the lead-time. An explanation is that with requirements affected by multiple systems the systems drive each other to be fast as they can only deliver value together. The same driving force is not found on single system requirements. However, we can hypothesize that single system lead-time can be shortened more easily, the reason being that waiting due to dependencies in a compound system requires the resolution of these dependencies to reduce the lead-time. On the other hand, no technical dependencies have to be resolved to remove lead-time in single systems.

Difference in Size: No major difference can be observed between small, medium and large requirements, except for large requirements in implementation and unit test. That is, at a specific size the lead-time for implementation and test increases drastically. In consequence, there seems to be a limit that the size should have to avoid longer lead-times. This result is well in line with the findings presented in literature (cf [9,10]).

5.2 Research Implications

The study investigated a very large system, hence research should focus on investigating time consumption for different contexts (e.g. small systems). This helps to understand the scalability of agile in relation to lead-times and with that the ability to respond quickly to market needs. Furthermore, the impact of size on lead-time is interesting to understand in order to right-size requirements.

In this study we have shown the absence of explanatory power for the variance in lead-time for phases and system impact. As time is such an important outcome variable research should focus on investigating the impact of other variables (e.g. experience, schedule pressure, team and organizational size, distribution, etc.) in a broader scale. Broader scale means that a sample of projects should be selected, e.g. by using publicly available repository with project data.

6 Conclusion

This paper evaluates software development lead-time in the context of a large-scale organization using incremental and agile practices. The following observations were made regarding lead-time:

- Phases do not explain much of the variance in lead-time. From literature one would expect that implementation and testing are the most time-consuming activities in agile development. However, due to the context (large-scale) other phases are equally time-consuming.
- There is no difference in lead-time for singe-system and multi-system requirements. This finding also contradicts literature. An explanation is that if a requirement has impact on multiple systems these systems drive each other in implementing the requirements quickly.
- With increasing size the lead-time within the implementation phase increases. This finding is in agreement with the related work.

In future work lead-times should be investigated in different contexts to provide further understanding of the behavior of lead-times in incremental and agile development.

References

1. Carreira, B.: Lean manufacturing that works: powerful tools for dramatically reducing waste and maximizing profits. American Management Association, New York (2005)

2. Bratthall, L., Runeson, P., Adelswärd, K., Eriksson, W.: A survey of lead-time challenges in the development and evolution of distributed real-time systems. Information & Software Technology 42(13), 947–958 (2000)
3. Schilling, M.A.: Technological lockout: an integrative model of the economic and strategic factors driving technology success and failure. Academy of Management Review 23(2), 267–284 (1998)
4. Urban, G.L., Carter, T., Gaskin, S., Mucha, Z.: Market share rewards to pioneering brands: an empirical analysis and strategic implications. Management Science 32(6), 645–659 (1986)
5. Stalk, G.: Time - the next source of competitive advantage. Harvard Business Review 66(4) (1988)
6. Petersen, K., Wohlin, C., Baca, D.: The waterfall model in large-scale development. In: Proceedings of the 10th International Conference on Product-Focused Software Process Improvement (PROFES 2009), pp. 386–400 (2009)
7. Beck, K.: Embracing change with extreme programming. IEEE Computer 32(10), 70–77 (1999)
8. Petersen, K., Wohlin, C.: A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. Journal of Systems and Software 82(9) (2009)
9. Harter, D.E., Krishnan, M.S., Slaughter, S.A.: Effects of process maturity on quality, cycle time, and effort in software product development. Management Science 46(4) (2000)
10. Agrawal, M., Chari, K.: Software effort, quality, and cycle time: A study of cmm level 5 projects. IEEE Trans. Software Eng. 33(3), 145–156 (2007)
11. Aoyama, M.: Issues in software cycle time reduction. In: Proceedings of the 1995 IEEE Fourteenth Annual International Phoenix Conference on Computers and Communications, pp. 302–309 (1995)
12. Carmel, E.: Cycle time in packaged software firms. Journal of Product Innovation Management 12(2), 110–123 (1995)
13. Petersen, K., Wohlin, C.: Context in industrial software engineering research. In: Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement, pp. 401–404 (2010)
14. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslen, A.: Experimentation in Software Engineering: An Introduction (International Series in Software Engineering). Springer, Heidelberg (2000)
15. Poppendieck, M., Poppendieck, T.: Lean software development: an agile toolkit. Addison-Wesley, Boston (2003)