UiO : Department of Informatics
University of Oslo

Does Limit on Work-In-Progress (WIP) in Software development matter?

Truls Skeie

Master's Thesis Spring 2014

Does Limit on Work-In-Progress (WIP) in Software development matter?

Truls Skeie

14th April 2014

Abstract

Background: In software engineering there are several principles depending the outcome of a software project. If one applies these principles the wrong way, or don't take it into account, it can starve a software project. WIP-limit is of one those principles. There is little evidence proving the impact of limiting work-in-progress for software development.
Aim: The aim for this thesis is to investigate the impact of WIP - limits in software development.
Methods: The methods used to investigate the research question were a case study of an in house software development company. The case study was based on a data set with recorded data from 2008 to 2013. The data set was interpreted with a program made for this thesis and the correlation and case summaries in SPSS.
Results:
The results show team size has an impact on both WIP and throughput. The results also show the impact both lead time and throughput has on a development process.
Conclusion:
The conclusion is WIP-limit don't matter in software development.

Contents

List of Figures

List of Tables

Listings

Preface

Chapter 1

Introduction

This master thesis focuses on limit Work In Progress (WIP), which is one of the principles in Kanban. The focus will be to see what impact WIP limit has in a development process. In order to do so, a data set gathered by an in house software company in Norway called Software Innovation (SI) was used. SI is a Scandinavian software company that delivers Enterprise Content Management applications.

The data set had already been interpreted by another study. That study investigated Scrum vs. Kanban for SI. For interested readers the case study can be found in the article "Quantifying the Effect of Using Kanban versus Scrum: A Case Study" (Sjøberg, Johnsen and Solberg, 2012).

1.1 Motivation

In software development, processes and methods are important in order to deliver the right product on time. In software development one rarely solves two identical problems for different stakeholders and the problems are getting bigger and more complex, which means that new processes and methods are introduced and the already existing processes and methods needs to be adapted to solve the complex problems in the most efficient ways. The number of popular software development methods (e.g. Extreme programming, Spiral, Scrum and Kanban) emerged in the recent years proves the assumption (Gandomani et al., 2013) (Marko Ikonen et al., 2010).

This is why this thesis will focus on software development methods, the method in each development project is such a key element. The main focus of this thesis will be the Kanban method and the principle to limit Work In Progress (WIP). In Kanban the WIP limit is used to limit the number of tasks each developer can work on at each workflow state to prevent bottlenecks and to ensure flow of tasks through the development cycle (Gandomani et al., 2013) (Marko Ikonen et al., 2010).

There are published various literature on Kanban in software development such as "Kanban: Successful Evolutionary Change for Your Technology Business" (D. J. Anderson, 2010), "Kanban and Scrum - making the most of both" (Kniberg, 2010) and "Lean Software Management: BBC Worldwide Case Study" (Middleton and Joyce, 2012). Although there is various literature, there is no information on how to apply WIP limit, even though most of the experience Kanban enthusiasts agree that WIP limit is an important principle. There is no research backing the statement. The literature states that one should experiment with WIP limits in order to find the best WIP limit for one's case (M. Ikonen et al., 2011) (Kniberg, 2010).

Since there is lack of available research on WIP limit my motivation is to investigate WIP limit in this thesis.

1.2 Research Question

In this thesis the overall research question will be to study the effects of WIP limits for an in house software company, in particular:

- Does WIP limit in software development matter?
- In case how to find the optimal WIP limit?
- Which parameters should be considered in order to optimize WIP?

1.3 Approach

This thesis will use case study as an approach to answer the research questions. To conduct the case study a data set from an in house software company will be used. The data set will be evaluated at team level. SI consist of ten teams, all of them will be investigated.

The program will take part of the analysis. The program was made for this thesis in order to transform the data set into more suited data for SPSS. The second part of the analysis will consist of statistical analysis to compute correlation and descriptive statistic.

1.4 Chapter overview

Chapter 2: Background:
Chapter 2 introduces background information and introduces relevant concepts and methods in software development as well as information about the in house software company, Software Innovation.

Chapter 3: Research Methods:
Chapter 3 introduces and explains the research methods used in the thesis as well as complementary information about Software Innovation and why the data set from Software Innovation is used in this thesis.

Chapter 4: Data collected and calculations:
Chapter 4 gives information about the data set and the calculations. Complementary information about how the program operates is given in as well as information about how the output data from the program is measured using statistical analysis.

Chapter 5: Results:
Chapter 5 presents the result produced by SPSS and the program, with descriptive statistics and correlation tables.

Chapter 6: Discussion:
Chapter 6 discuss the findings from the case study. There is also a discussion of the finding against other research.

Chapter 7: Conclusion:
Chapter 7 provides the answers to the research questions as well as recommending future work.

Chapter 2

Background

In this chapter there will be a brief introduction to Waterfall (Section 2.1), Scrum (Section 2.2), Lean (Section 2.3) and Kanban (Section 2.4) with affiliated tools. The software development company Software Innovation is briefly introduced (Section 2.9)

2.1 Waterfall

"The waterfall model is the classical model of software engineering. This model is one of the oldest models and is widely used in government projects and in many major companies" (Munassar and Govardhan, 2010). The main goal of the waterfall model is to plan in early stages to ensure design flaws before coding is started. Since planning is so critical in the waterfall method it fits project where quality control is a major concern (Munassar and Govardhan, 2010).

The waterfall method consist of several non-overlapping stages as shown in figure 2.1. The figure is an example of the waterfall model with a life cycle of establishing system requirements and software requirements and continues with architectural design, detailed design, coding, testing and maintenance (Munassar and Govardhan, 2010). One of the main principles of the waterfall method discourages return to an earlier phase. For example returning from detailed design to architectural design. However, if returning to an earlier phase is needed, it involves costly rework. When a phase is completed, the phase requires formal review and extensive documentation development. Therefore, if something is missed out an earlier phase it is expensive to correct it later (Munassar and Govardhan, 2010)

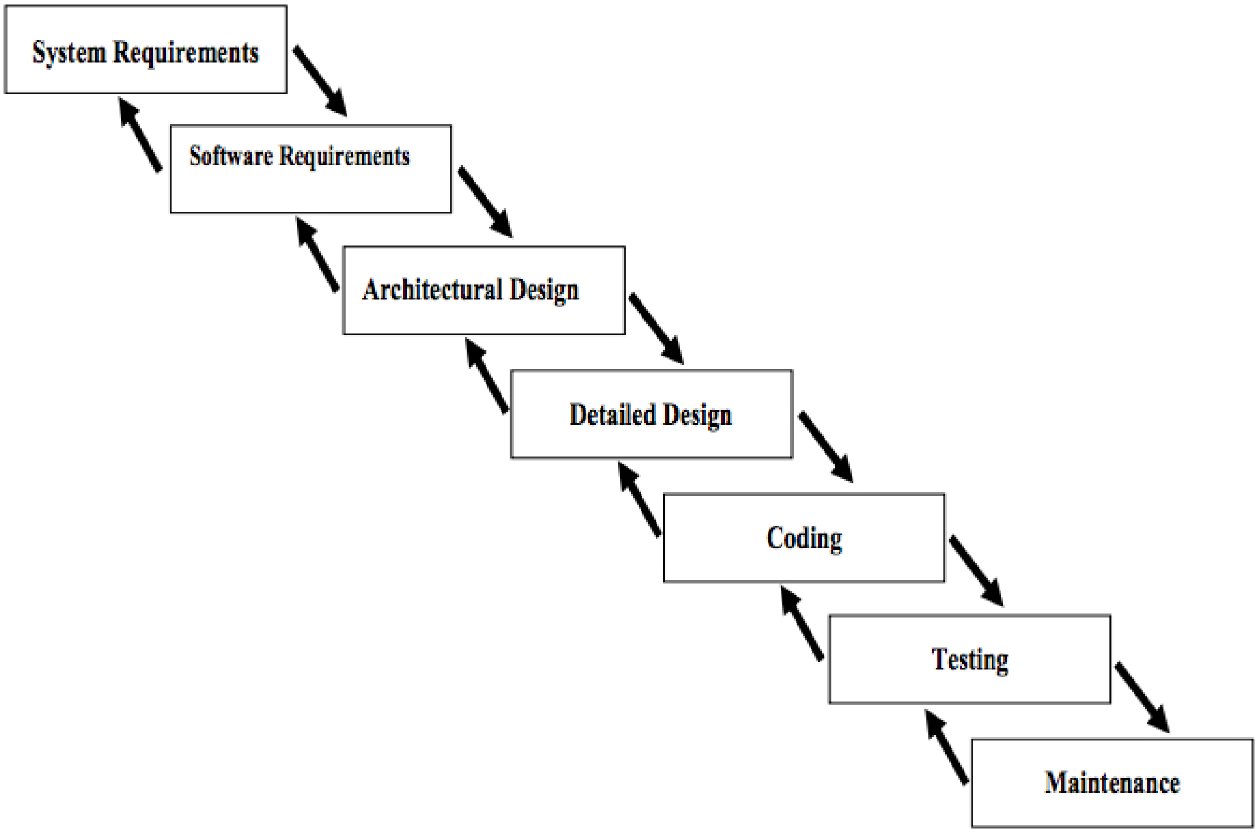


Figure 2.1: Waterfall model

2.2 Scrum

"Scrum is the best-known of the Agile frameworks. It is the source of much of the thinking behind the values and principles of the Agile Manifesto. These values are:

Individuals and interactions *over processes and tools*
Working software *over comprehensive documentation*
Customer collaboration *over contract negotiation*
Responding to change *over following a plan*
(Alliance, 2012).

These principles of Scrum and Agile manifesto are not so rigid as the principles of the Waterfall method. Some may say that Scrum is the opposite of the Waterfall method (Cocco et al., 2011).

Scrum have three main roles, the Product Owner, the Scrum Master and the members of the development team. The Product owner in collaboration with the Scrum Master decides which work to be prioritized in the backlog. The backlog represents the tasks to be done in order to complete the project. The Scrum Master acts like a team leader and helps the development team and the organization to take best advantages of Scrum. The development team works on tasks specific for the sprint there in (Alliance, 2012).

Sprint is a time-boxed interval over a given time. The Scrum framework suggests duration of sprints to be from one to four weeks. Before each sprint, a sprint planning meeting is conducted with all the team members attending. A Sprint planning meeting is held so the team can discuss tasks from the backlog and come to an agreement of which tasks to be put in the minimal backlog (Alliance, 2012).

In each sprint a minimal backlog is created so the developer knows which tasks to work on in the current sprint. The Product Owner and the team members discuss and decide which tasks from the backlog to be added to the minimal backlog. After the minimal backlog is complete, the Product Owner and the team members discuss each task in order to get a better and shared understanding of what is required to complete the tasks (Alliance, 2012).

One of the main principles in Scrum is that it requires that at least one new feature is ready for release after each sprint. The feature should be a visible part of the product in order to get feedback from end-users. So all the tasks in the minimal backlog combined should be a visible part of the product (Alliance, 2012).

2.3 Lean

"Lean is all about getting the right things to the right place at the right time the first time while minimizing waste and being open to change" (Raman, 1998). The Lean approach was introduced around 1948 in manufacturing in Japan. In 1975, Toyota was able to create almost 50 more production units per employee than in 1948 due to the Lean approach (Manning, 2013). Lean strives to maximize the value produced by an organization and delivered to costumer. This is done by finding and eliminate waste, controlling variability and maximizing the flow of delivered software all within the culture of continuous improvements (D. Anderson et al., 2011). In 2003 Mary and Tom Poppendieck first introduced Lean thinking to software development. Poppendieck published the book "Lean Software Development: An Agile Toolkit" (M. Poppendieck and T. Poppendieck, 2003). In the book, Poppendieck stated that an important tool to manage work flow is the concept of pull-systems, which means tasks are put in production only when a costumer asks for it (M. Poppendieck and T. Poppendieck, 2009). The pull based method Kanban has in recent years been introduced more and more to software development, and is becoming one of the keys to Lean practice in software development (D. Anderson et al., 2011). In Lean there are eight fundamental principles (M. Poppendieck, 2003).

1. **Start Early:** Don't wait for details. As soon as enough information is gathered start the development activity. Get everyone involved in figure out the details. Don't build any walls between people, make people collaborate and start a two-way communication as soon possible. This will start the learning cycle as well.

2. **Learn Constantly:** Start with a breadth-first approach, explore multiple options. The system is expected to change, so focus on creating simplicity code and robustness so the system is easy to change

3. **Delay Commitment:** In order to delay commitment, automated testing and refactoring are essential for keeping code changeable.

4. **Deliver Fast:** Deliver fast mark of excellent operational capability. The whole idea of **delaying commitment** is to make every decision as late as possible when one have the most knowledge.

5. **Eliminate Waste:** The only thing worth doing is deliver value to the costumer, anything else is waste. See waste and eliminate it is the first key of Lean. Lean suggests using a value stream map for removing waste. A Value Stream Map (VSM) is a map over the whole company chain. VSM helps visualize where waste are located within the company.

6. **Empower the team:** When one are going to deliver fast, there is no room for central control. The work environment should be structured so work and workers are self-directing.

7. **Build Integrity In:** Lean software is build with integrity. That's why one of the principles in Lean suggests that test are integrated into software development just as any code, so it becomes a part of the delivered product.

8. **Avoid Sub-optimization:** In software development it's normal to break down a complex problem into small parts of the problem in order to minimize the complexity. If some of the parts are sub-optimized, bottleneck can occur. For example, if ten developers are hired to work on tasks, but only three testers are hired. The development process is sub-optimized since the developers will likely produce more than the tester can test and that will cause bottleneck.

2.4 Kanban

Toyota production system introduced Kanban as a scheduling system for Lean and just-in-time (JIT) production during late 1940's and in the early 1950's in order to catch up with the American car industry. The Kanban method combined with the Lean approach was a success for Toyota. The success was noticed by the software development industry among others (Conboy, 2009), (Ohno, 2001). In the recent years, more software projects adapt to Kanban and Lean (D. Anderson et al., 2011), and this is one of the reasons why this thesis will focuses on Kanban and one of its principles; WIP limit.

"One can define Kanban software process as a WIP limited pull system visualized by the Kanban board" (D. Anderson et al., 2011). One of the most important people in Kanban software development, David Anderson also referred to as "father of Kanban in the software development industry" (Gupta, 2013) and author of the book "Kanban: Successful Evolutionary Change for Your Technology Business" stated "If you think that there was Capability Maturity Model Integration, there was Rational Unified Process, there was Extreme Programming and there was Scrum, Kanban is the next thing in that succession." (Leonardo Campos, 2013) .

In software development, Kanban splits the major problem into many small pieces of problems. When the small pieces are defined by the team, the problems are put up on the Kanban board to visualize the problems, track what others are working on and see potential bottlenecks during development. Shinkle stated that when people start to understand Kanban, they easily discover where the bottlenecks are (Shinkle, 2009). In short, Kanban systems focus on (D. Anderson et al., 2011):

- continuous flow of work,
- no fixed iterations or sprints,
- work is delivered when it's done,
- teams only work on few tasks at the time specified by the WIP limit and
- make constant flow of released tasks.

Contrary to Scrum, Kanban do not use the principles of sprints or estimations. In Kanban the tasks do not need to be estimated or finished within a certain time. In the article "Simulation of software maintenance process, with and without a work-in-process limit" (Concas et al., 2013) the authors found out that if they let the developers work with small tasks and not be interrupted, they will be more effective. They also found out that Scrum was too rigid for the development team because when the team had to estimate tasks, they felt interrupted. The estimation and sprint meetings worked counterproductive in their case. The authors made the developers change to Lean-Kanban. The change implied the removal of sprints and estimation. After removing sprints and estimation the teams increased the ability to perform work, lower the lead time and meet the production dates (Concas et al., 2013).

In the article "Quantifying the Effect of Using Kanban versus Scrum:" the company also felt that the Scrum approach was too rigid. The article also reported positive results when the team changed to Kanban. The company almost halved its lead time, reduced the number of weighted bugs by 10 percent, and improved productivity (Sjøberg, Johnsen and Solberg, 2012). Other articles also state that Scrum maybe too rigid and that's Kanbans advantages over Scrum (Beedle et al., 1999) (Brekkan and Mathisen, 2010) .

2.4.1 Kanban Board

"The Kanban board makes it clear to all the team members the exact status of progress, blockages, bottlenecks and they also signal possible future issues to prepare for"(Middleton and Joyce, 2012). The Kanban board is one of many tools in Kanban. It's used to control WIP, increase the information flow with visualization (Concas et al., 2013). A Kanban board is illustrated in figure 2.2. Each column in figure has 2.2 an intuitive name in order to describe itself so the developers easily can track where each task is.

The columns are named "Backlog", "In progress" and "Done". Each column can have a WIP limit to specify how many items in progress there are allowed in the column (Middleton and Joyce, 2012). In figure 2.2 the WIP limit is stated under the column name. The backlog column has a WIP limit of 4, In progress has 5 and Done doesn't need a WIP limit.

The yellow stickers represent the tasks. Some follow the path to mark stickers with different colors representing the severities or by marking if its a feature or a bug. In the article "Kanban Implementation in a Telecom Product Maintenance" for instance, the stickers has three different colors, green, yellow and red depending on how close to overdue the tasks are. If the sticker is red, the task is already overdue, if the tasks are soon-to-overdue its marked with yellow stickers (Seikola, Loisa and Jagos, 2011). In another project, they used yellow sticky notes for scenarios, blue for bugs, pink for issues (Shinkle, 2009).

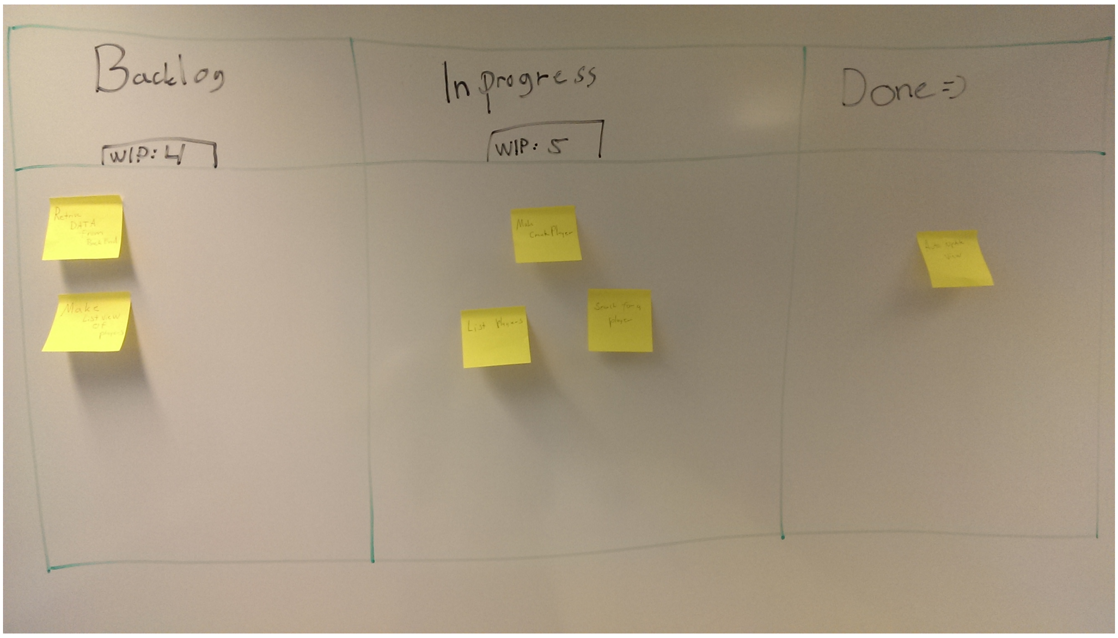


Figure 2.2: Example of a Kanban board

2.4.2 WIP limit

"WIP limits seem to be the worst understood part of the Kanban system. When used properly, it exposes bottlenecks and reduces lead time for individual work items. Used improperly, it can starve developers for work or result in too many people working on the same work items." (Shinkle, 2009)

WIP limit is one of the core principles in Kanban (Seikola, Loisa and Jagos, 2011). WIP limit helps to reduce overhead by limit task-switching for each developer and make constant flow of tasks throughout the development (D. Anderson et al., 2011). One way to explain WIP and the asserted impact of WIP limit is to use cars and roads as analogy. All roads have a maximum capacity of cars. When this limit is reached, traffic jam occurs and the throughput of cars decreases and lead time increases. The same can be said about software development teams. A software team has a maximum number of tasks they can perform, if the team is pushed over the maximum limit, the throughput of tasks may decreases and lead time may increases.

When first implementing Kanban, Shinkle explains that the users often don't care about WIP or setting a WIP limit, but rather the visibility of Kanban through the Kanban board. When users gain more experience with Kanban, they start to attempt the principles of WIP limit (Shinkle, 2009). Srinivasan, Ebbing and Swearing said that setting the WIP limit is not easy. They suggest that the WIP limit is set, and then observe throughput, and adjust after that (Srinivasan, Ebbing and Swearingen, 2003). In the book "Kanban and Scrum - making the most of both" suggests Kniberg that you start by limiting WIP, then experiment with it (Kniberg, 2010). The article "Lean Software Management" (Kniberg, 2010) and the "Impact of Kanban on Software Project Work" (M. Ikonen et al., 2011) both suggest that WIP should be minimized as well. The conclusion of present study is to keep the WIP limit low and experiment by slowly increase the WIP limit until the throughput decreased and lead time increased, then you know that the previous WIP limit was the perfect one.

The Section 2.4.2.1 shows a summary of the the articles by Giulio Concas, Hongyu Zhang (Concas et al., 2013) and David Anderson, Giulio Concas, Maria Ilaria Lunesu, and Michele Marchesi (D. Anderson et al., 2011). The articles researched the difference between limit WIP and unlimited WIP. Section 2.4.2.2 shows the importance of limit WIP, stated by various researches.

On how to determine WIP limit one article was found. If one implements Kanban with sprints or uses Scrum, Łukasz proposes to use the effectiveness metric to help determine the WIP limit.The effectiveness metric shown in formula 2.1, should be applied after end sprint according to Łukasz. After each sprint, one can apply the effectiveness metric and the result could be used as a guideline for WIP limit for the next sprint. The effectiveness metric takes the number of bugs found *(ai)* and the number of bugs found by external people (e.g. lawyers, accountants, coaches, consultants, translators, internal and external service providers etc.) *(ei)*, and minus *ai* and *ei*, then divide the result by *ai* and multiply it by 100% as shown in formula 2.1 (Sienkiewicz, 2012)

$$Ei = \frac{(ai - ei)}{ai} * 100\% \tag{2.1}$$

2.4.2.1 Limit WIP vs. Unlimited WIP

In the article by Giulio Concas and Hongyu Zhang, they simulated two different software maintenance processes. The first process where based on 4 years of experience with Microsoft maintenance team. The second process where from a Chinese software firm. The simulation executed 10 runs and one of the results were the average of closed tasks was 4145 when the WIP was limited and 3853 when the limit was not limited (about 7% less). The paper concludes their finds; developers are more focused on fixing few issues, because the number of issues they can work on is limited. The developers are more likely to continue on the issue from the day before, rather than starting on another issue, this reduces overhead. Because when developers start on a new issue, they need time to familiarize themselves with the code and the issue. That could create unnecessary overhead if some developer already has done it, but that developer is now working on another issue.

The study also showed that limit WIP can improve throughput and work efficiency. In the article, they authors also simulated a software process who was originally without WIP - limits, with WIP-limits. The original process was outperformed by the simulated process with WIP-limits. (Concas et al., 2013).

The case by David Anderson et al. (D. Anderson et al., 2011) did a simulation of lean-kanban approach with the impact of WIP limit vs. no WIP limit on developers with skills in different activities. The four skill activities from the article were design, development, testing and deployment.

The article did four different simulations. A simulation with WIP limits and seven developers with skill in two of the four activities. A simulation with no WIP limit and seven developers with skilled in two of the four activities. A simulation with WIP limits and seven developers with skill in all of the activities. A simulation with no WIP limits and seven developers with skill in two of the four activities.

The paper concluded that the last two is unlikely in the real world, since there is rarely a whole team with developers skilled in all activities. When the developers had skill in two out of four activities, the WIP limit simulation used 100 days, but the non WIP limit simulation used 120 days. The simulation with WIP limit shown an almost constant flow of features that completed, while in the same simulation with no WIP limit, the flow of feature was much more irregular (D. Anderson et al., 2011).

2.4.2.2 Benefits with setting WIP limit

This subsection contains excerpt from papers from various authors that have done study on WIP limit.

1. Lowering the WIP limit will help people avoid task switching. When one is task switching it's hard to be able to fully concentrate. (M. Ikonen et al., 2011).
2. There's stated when using short-cycle times and Kanban board to limit WIP, the software development team's learning is increased (Middleton and Joyce, 2012):
3. WIP - limit increases productivity (Middleton and Joyce, 2012).
4. WIP - limit reduce cycle time (Birkeland, 2010)
5. When WIP was too high, lead times grew and as a result so did the bugs and rework (Shinkle, 2009).

6. WIP-limits are important to reduce lead times (The-Kanban-Way, 2011)

Both the studies on WIP limit vs. No limit, the articles and research shows the importance of WIP limit. If Łukasz's effectiveness metric is isregarded , there is no clear rule on how to determine WIP limit even though WIP is suppose to be a crucial principle in order to take full advantages of Kanban.

2.5 Lead time

”Lead time is the total elapsed time from when a customer requests software to when the finished software is released to the customer” (Middleton and Joyce, 2012). Lead time is measured to track how quickly software is delivered to customers (Middleton and Joyce, 2012). Lead time could be an essential ingredient when you look for the optimal WIP, if there is one. Often in a project, lead time is split into pieces, so every task has its own lead time. This gives the development teams the advantages to experiment with different WIPs in order to see the different lead times and then measure which WIP that suits this project the best.

According to the paper "Quantifying the effect of Using Kanban versus Scrum" (Sjøberg, Johnsen and Solberg, 2012) the citation by Middleton and Joyce above is close to definition of what lead time is. They define lead time as the amount of time that passed from the moment that the development team receives a request to the moment that it completes the work item. The reason why the paper disapproves the definition by Middleton and Joyce is because: "The amount of time a work item remains in the backlog queue before it's put on the board is a function of priority, not whether the company uses Scrum, Kanban or other development methods. Furthermore, companies that develop and sell products to many customers might propose new features themselves and put them on the backlog before any customers request them. Second, given a policy of two or three releases a year, the result of a work item isn't delivered to the customer immediately after it's finished” (Sjøberg, Johnsen and Solberg, 2012).

2.6 Just-In-Time

"Just-In-Time is based on delivering only the necessary products, to the necessary time and the necessary quantity" (Lai, Lee and Ip, 2003). Just-In-Time (JIT) was introduced in the 1970s by Toyota in combination with Lean (Javadian Kootanaee, Babu and Talari, 2013). JIT has been introduced to increase productivity through waste reduction and increasing the value added in the production processes. To explain the JIT principle, Mary and Tom Poppendieck use the picture shown in Figure 2.3 (Lai, Lee and Ip, 2003) (M. Poppendieck and T. Poppendieck, 2006). The stream reflects the inventory. Under the stream, there are rocks located in different sizes. The rocks illustrates waste and problems that can occur. If the stream level is lowered, the rocks are more visualized. At this point you have to clear out rocks (remove waste and problems) in order to make the boat continue it's journey, or it will crash into the rocks. After the rocks are cleaned out, one can lower the stream level again and continue until there are pebbles left. Then the boat can float without problems.

If one lower the stream (inventory), problems and waste will become visible (visualized by rocks). Lean wants to lower inventory in order to make problems and waste occur, because when problems and waste occurs, you are able to fix the problems and remove the waste. Fixing the problem and removing the waste has several benefits such as, your process could be optimized and you are on step closer to have zero problems and zero waste. (Lai, Lee and Ip, 2003) (M. Poppendieck and T. Poppendieck, 2006).

In Software development the JIT principle means that one should not deliver anything before its demand. For example, if a development team adds two new features to a product without the stakeholders asking for it and use a work day on these features, and the stakeholders don't want the features. Then the development team has produced the number of team members persons days of waste.

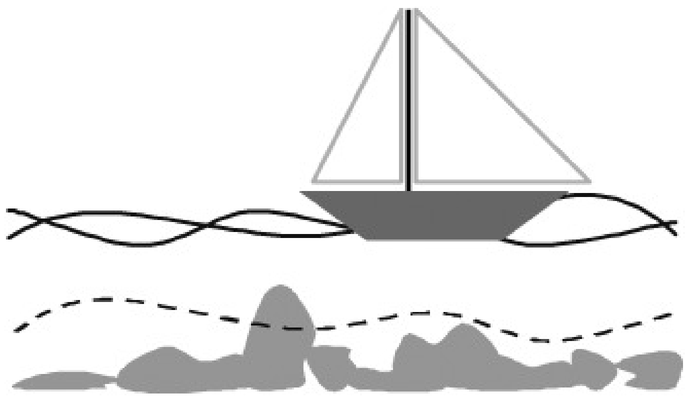


Figure 2.3: JIT example

2.7 Throughput

"The output of a production process (machine, workstation, line plant) per unit time (e.g., parts per hour) is defined as the systems throughput or sometimes throughput rate" (Adams and Smoak, 1990). The main concept of throughput is to measure how productive teams, people or companies are. Throughput is measured in number of finished delivered tasks or units per hour, day, week, month, quarter or year. A key factor in successfully measuring throughput in software development is to specify a standard size for each task. If the standard is not specified there is little use in throughput measurements. (Rouse, 2005). To illustrate throughput with different task sizes an example is provided:

Lets say Team x had a throughput of eighteen tasks after the first quarter, twenty after the second, fifteen after the third and twelve after the last quarter. Team x used Scrum the first two quarters and Kanban the last two as illustrated in table 2.1. It will look like team x benefits most from Scrum. But if the task during the Kanban time was twice the size of Scrum, Kanban would suite team x the best. So, to get valid result from throughput measurements, the size of tasks has to be agreed upon by the teams or company.

Quarter	Throughput	Method
1	18	Scrum
2	20	Scrum
3	15	Kanban
4	12	Kanban

Table 2.1: Throughput

2.8 Code churn

"Churn is defined as the sum of the number of lines added, deleted, and modified in the source code" (Sjøberg, Johnsen and Solberg, 2012). Churn is a measure that's not so known as lead time, throughput or WIP. Churn is a term used as surrogates for effort in software engineering. Many studies in software engineering use code churn or revisions as surrogate measure of effort (D. Sjøberg, Anda, Mockus et al., 2012). Emam stated that "analysts should be discouraged from using surrogate measures, such as code churn, unless there is evidence that they are indeed good surrogates" (El-Emam, 2000). The study by Sjøberg et al. showed that churn could be used as a surrogate for tasks size (D. Sjøberg, Anda, Mockus et al., 2012).

2.9 Software Innovation

Software Innovation is a Scandinavian software company. SI develops and delivers Enterprise Content Management applications that helps organizations improve and increase efficiency in document management, case handling and technical document control. SI builds products around the Microsoft Sharepoint platform. (Sjøberg, Johnsen and Solberg, 2012). (*Software Innovation* 2013).

SI has approximately 300 employees in Oslo, Copenhagen, Stockholm and Bangalore (*Software Innovation* 2013). From 2001 to 2006, SI used the Waterfall process. In 2007, SI changed to Scrum, and in 2010, SI went from Scrum to Kanban (Sjøberg, Johnsen and Solberg, 2012).

2.9.1 Teams

Table 2.4 shows the size of the ten teams vs. quarter. Team seven, shown in Table 2.4g contribute data from 2010 to 2012. After 2012, team seven was shut down.

Year	Quarter	Team Size
2010	3	6
2010	4	3
2011	1	16
2011	2	28
2011	3	2
2011	4	38
2012	1	35
2012	2	34
2012	3	32
2012	4	29
2013	1	24
2013	2	37
2013	3	23
2013	4	23
Total		330

(a) Team size - team one

Year	Quarter	Team Size
2010	3	10
2010	4	15
2011	1	13
2011	2	12
2011	3	15
2011	4	14
2012	1	15
2012	2	7
2012	3	8
2012	4	9
2013	1	10
2013	2	7
2013	3	7
2013	4	8
Total		150

(b) Team size - team two

Year	Quarter	Team Size
2010	3	6
2010	4	9
2011	1	7
2011	2	10
2011	3	9
2011	4	10
2012	1	11
2012	2	11
2012	3	13
2012	4	13
2013	1	13
2013	2	7
2013	3	8
2013	4	8
Total		135

(c) Team size - team three

Year	Quarter	Team Size
2010	3	3
2010	4	8
2011	1	4
2011	2	4
2011	3	4
2011	4	4
2012	1	4
2012	2	2
2012	3	3
2012	4	5
2013	1	7
2013	2	5
2013	3	5
2013	4	5
Total		63

(d) Team size - team four

Year	Quarter	Team Size
2010	3	5
2010	4	13
2011	1	14
2011	2	25
2011	3	21
2011	4	23
2012	1	25
2012	2	19
2012	3	24
2012	4	18
2013	1	31
2013	2	29
2013	3	27
2013	4	11
Total		285

(e) Team size - five

Year	Quarter	Team Size
2010	3	5
2010	4	6
2011	1	6
2011	2	6
2011	3	5
2011	4	5
2012	1	4
2012	2	6
2012	3	6
2012	4	9
2013	1	9
2013	2	9
2013	3	9
2013	4	14
Total		99

(f) Team size - team six

Year	Quarter	Team Size
2010	3	10
2010	4	8
2011	1	8
2011	2	6
2011	3	8
2011	4	9
2012	1	10
2012	2	5
2012	3	9
2012	4	3
Total		76

(g) Team size - team seven

Year	Quarter	Team Size
2010	4	2
2011	1	8
2011	2	8
2011	3	13
2011	4	9
2012	1	10
2012	2	2
2012	3	25
2012	4	11
2013	1	22
2013	2	21
2013	3	23
2013	4	8
Total		162

(h) Team size - team eight

Year	Quarter	Team Size
2010	4	5
2011	1	8
2011	2	7
2011	3	7
2011	4	9
2012	1	10
2012	2	8
2012	3	10
2012	4	12
2013	1	8
2013	2	9
2013	3	8
2013	4	8
Total		109

(i) Team size - team nine

Year	Quarter	Team Size
2010	3	3
2010	4	11
2011	1	12
2011	2	9
2011	3	4
2011	4	17
2012	1	20
2012	2	17
2012	3	18
2012	4	13
2013	1	17
2013	2	9
2013	3	10
2013	4	10
Total		170

(j) Team size - team ten

Figure 2.4: Caption of team size for teams in SI

Chapter 3

Research Methods

In this chapter the research methods will be introduced and the reason why the data set from Software Innovation were chosen. Section 3.1 gives a brief introduction to the research method "Case Study". Section 3.2 is about the choice of case and complementary information about SI.

3.1 Case study

To answer the research questions, a case study was conducted. A case study is used to explore causation in order to find underlying principles (Shepard and Greene, 2002)(R. K. Yin, 2008). But which methods one can use in a case study or how the case study is conducted is ambiguous. It might be that the case study maybe qualitative or quantitate. A case study might utilize a particular type of evidence (for example ethnographic, participant observation or field research). Jennifer Platt stated: "Much case study theorizing has been conceptually confused because too many different themes have been packed into the idea "case study" (Gerring, 2006). John Gerring stated: "A case study may be understood as the intensive study of a single case where the purpose of that study is – at least in part to shed light on a larger class of cases (Gerring, 2006). As one can see, there is no clear rule of how to conduct a case study or what it is.

In this thesis, the case study is used to explore WIP limit's effect in software development. The purpose is to shed light on WIP limit in software development and if it matters.

3.2 Choice of case

The data set from SI contains information about each task SI has worked from 2008 to 2013. The data set is represented in a excel document. An excerpt of some of the columns in the document is shown in table 3.1. Although the data set contains items from 2008-2013, data from year 2008, 2009 and the two first quarters of 2010 will be excluded. The dates will be excluded partially because the transition between processes and it was inaccurate measurements when SI first started with TFS.

The reason SI and the data set from SI is analyzed in this thesis is because the article "Quantifying the Effect of Using Kanban versus Scrum" (Sjøberg, Johnsen and Solberg, 2012) used the same data set. Since Dag is the supervisor of this thesis and he had access to the data set, to use the set from SI is a choice of convenience.

ID	Type	Created Date	From Day	Date To	Lead Time	Team
3027	Bug	2008-10-07	2008-10-09	2008-10-16	20	Team one
3028	Bug	2008-10-07	2008-10-07	2008-10-08	10	Team six
3029	Feature	2008-10-07	2008-12-30	2008-12-30	105	Team two
3030	Feature	2008-10-07	2008-10-07	2008-10-07	1	Team three
3035	Bug	2008-10-08	2008-11-20	2008-11-28	17	Team five
3037	Feature	2008-10-08	2008-10-19	2008-10-19	7	Team three
3040	Bug	2008-10-10	2008-11-19	2008-11-19	48	Team one

Table 3.1: Excerpt from the data set

The data set contains thirty columns with different data for each task, mostly of this columns are irrelevant for this study but the important columns is stated in table 3.2.

Variable	Description
Created Date	When a task is put in backlog
Date From	When a given task is pulled out from the backlog
Date to	When a task is finished and ready for release.
Lead Time	The amount of days elapsed from the date the task was created until the tasks has finished
Type	The type column is labeled as either bug or feature depending on the type of the task
Lines added	Number of lines added to a feature or bug
Lines modified	Number of lines modified when working on a feature or bug
Lines deleted	Number of lines deleted from a bug or feature
Team	States the team who has been working on the task.

Table 3.2: Variables from the SI dataset

The **Created date** column consist of dates form when tasks where created. The **Date from** column contains the date the tasks was pulled from the backlog. The **Date to** column consist of all of the dates when tasks where marked as finished. The **Lines added**, **Lines Modified** and **Lines Deleted** columns contains the amount of lines added, modified or deleted in order to finished the task. The **Type** column consist of a string that has the value as either "Bug" or "Feature". The **Lead time** column consist of the lead time value. The **Team** column consist of which team the task belongs to.

The data from SI was analyzed on team level. The data from SI was analyzed using the program, which computed the variables shown in table 3.3 for all of the teams.

Computed variable	Description	Columns from SI
WIP	Items in progress on the given day	Date From and Date To.
Throughput	Number of tasks finished on a given day	Date To
Churn	Lines added, lines modified and lines deleted added together	Lines Added, Lines Modified, Lines Deleted and Date To
Bugs	The number of tasks labeled as Bug and not feature	Type and Created Date
Lead time	The time used on a task, measured in days	Lead time and Date To
Bugs finished, quarter	Number of bugs finished, per quarter	Created date, Date to and Type
Avg days backlog, bug	Average days in backlog for bugs, per quarter	Created date, Date from and Type

Table 3.3: Relationship between variable and columns from SI

Both the variables churn and throughput is split up in two sub variables with suffix of *Feature* and *Bug*. The variable with suffix of *Feature* means tasks labeled with type *feature* are the only one who is counted. The same goes for variables with suffix *Bug*. The *Bugs finished, quarter* variable represents how many tasks labeled *Bug* that are finished within the same quarter as it was created. The *Avg days backlog, bug* variable represent the average number of days bugs are in backlog before it pulled out.

3.2.1 Software Innovation's development process

From 2001 to 2006 SI used the Waterfall process with a life cycle of:

1. Design
2. Implementation
3. Testing
4. Deployment for each new release

(Sjøberg, Johnsen and Solberg, 2012).

In 2007, SI examined their development process, which resulted in a decision to change to Scrum. Scrum was implemented with the standard elements of Scrum:

- Cross functional teams

- Sprint planning meetings
- Estimation of work items using planning poker
- Daily standup meetings
- Sprints

(Sjøberg, Johnsen and Solberg, 2012).

SI implemented three weeks sprint, after each sprint a fully tested shippable code was ready. In 2010, SI went from Scrum to Kanban. SI felt that Scrum was too rigid and didn't fit their purpose, they also feared that inaccurate estimation and time boxing gave them longer lead time. SI also saw Scrum planning meetings as waste which reduced productivity and quality (Sjøberg, Johnsen and Solberg, 2012).

SI decided to implement Kanban in the following manner. When a work item is pulled from the backlog, SI tries to make the item flow through all the stages until it's ready for release. This procedure happens as quickly as possible. In order for an item to be ready for release, it has to be at a satisfactory quality level, which is defined by SI. SI also implemented WIP limits. If the WIP limit is reached, no new tasks are started until another task is finished which is based on the principle of just-in-time (Sjøberg, Johnsen and Solberg, 2012).

3.3 Correlation

I want to look at the linear relationship between two variables. To do so, i have chosen to use Pearson correlation (L. Yin, Xiao and Xu, 2013).

Chapter 4

Data collected and calculations

This chapter will introduce how the algorithms of the program works as well as a brief introduction to SPSS. The first section gives a short introduction to the statistical analyze program SPSS (Section 4.1). The next section, Section 4.2 introduces the algorithm of how the program measure WIP for each day. The subsection 4.2.4 provides a comprehensive example of how the program measure WIP per day. The consecutively sections reveal the algorithm of how the program measure bugs (Section 4.3.5) throughput (Section 4.3.1), churn (Section 4.3.2), lead time (Section 4.3.3), number of bugs finished quarter (Section 4.3.6) and average days for bugs in backlog (Section 4.3.7).

The Table 4.1 shows how quarters, dates and days are represented in this thesis.

- The Date standard is specified as YYYY-MM-DD.
- All seven days in the week are taken into account when measuring, included Saturdays and Sundays
- Quarter of a year is defined as:
  - January, February and March (Q1),
  - April, May and June (Q2),
  - July, August and September (Q3),
  - October. November and December (Q4).

  (Investopedia, 2013)

Table 4.1: The standard of the data set

4.1 SPSS

"IBM®SPSS®Statistics is a comprehensive system for analyzing data. SPSS Statistics can take data from almost any type of file and use them to generate tabulated reports, charts and plots of distributions and trends, descriptive statistics, and complex statistical analyses." (IBM, 2014)

After the program has finished the measurements of the data, SPSS will be used to analyze the derived data. SPSS will help to answer the research question stated in chapter 1.2 with help of two statistics method: correlation and case summaries.

4.2 WIP per day

4.2.1 Step 1: Gather all unique dates into a Arraylist

The first step of this WIP algorithm is to create a WIP object with the attributes in table 4.2. The values that are assigned to the attributes are gathered from the data set file. The program creates a WIP object and assigned the data set values to the object a shown in listing 4.1 . After the values are assigned, the program puts the WIP object into the right Arraylist[1] based on the team variable as shown in listing 4.2.

Type	**Variable name**
Date	start
Date	end
String	team
String	processType
int	WIP

Table 4.2: Variables of the WIP objects

[1] Arraylist is a resizable array implementation of a list. The Arraylist class provides function for manipulate the size of the array, check the size of the list and convert the list to an array (Oracle, 2013).

```
While inputFile != EOF // EOF = End Of file
    WIP = New WIP()
    WIP.start = inputFile.start
    WIP.end = inputFile.end
    WIP.team = inputFile.team
    WIP.processType = inputFile.processType
    WIP.WIP = 1
    FindTeam(WIP)

```

Listing 4.1: Gather all unique dates into Arraylist

```
void FindTeam (WIP w)
    if w.team EQUALS "TeamOne"
      TeamOne.add(w)
    if w.team EQUALS "TeamTwo"
      TeamTwo.add(w)
    if w.team EQUALS "TeamThree"
      TeamThree.add(w)
/* And so on for the rest if the seven teams */

```

Listing 4.2: Gather WIP object to the right data structure

4.2.2 Step 2: Gather the remaining dates

There are some dates missing as shown in Table 4.3. For example, the date 2010-10-08 is missing. In order to generate WIP for each day, the program has to create the dates that are not in the set. In order to create the remaining dates, the program takes the first date and the last date from each of the teams's Arraylist created in previous section (4.2.1) as shown in line one and two of Listing 4.3. Then the program checks if all the dates between the first date and the last date are in the team's Arraylist. Each of the Arraylists are sorted on date. If the dates are not in the Arraylist, the program will put the date into the Arraylist as show in method addToArraylist on line ten to thirteen. In order to keep the pseudocode simple, the generateWIP method stated in line twelve was omitted. The generateWIP method creates a new WIP object and returns it.

```
1 WIP first = Arraylist.get(0)//points to the first WIP object in the Arraylist
2 WIP last = Arraylist.get(Arraylist.size()-1)//points to the last WIP object
    in the Arraylist
3 Next_date //points to the next date
4 Next_date = first.getDate() // Next_date assigned before iteration
5 while Next_date NOT EQUALS last.getDate()
6   New_date = Next_Date + 1 //Compute the next date
7   AddToArraylist(New_date, first.getTeam())
8   Next_date = New_date
9
10 void addToArraylist(Date d, String team)
11   if d NOT CONTAINS IN Arraylist
12     WIP = generateWIP(d, team)
13     Arraylist.add(WIP)
14
```

Listing 4.3: Gather the remaining dates.

4.2.3 Step 3 Measure WIP

The Arraylist from section 4.2.1 and 4.2.2 now contains a WIP object for each date from the third quarter of 2010 to 2013 for each team. In this step, the program will loop through each of the teams Arraylists. During the iteration each WIP object is extracted from the Arraylist and the WIP is measured. The two methods stated in line ten and eighteen respectively gather the current WIP (method in line ten) and finds out how many finished tasks (method in line eighteen) and returns the result. The result is used in line six to compute the current WIP. The if test in line four assures that only one instance of each date is measured.

```
void measureWIP()
  lastWIP =  0
  for WIP Object IN Arraylist
    if(DateNotMeasured(WIP.getStartDate()) == true)
      WIP_for_this_date = get_current_WIP(WIP.getStartDate())
      WIP_measured = WIP_for_this_date - Nr_of_finishedDates(WIP.getStartDate) + lastWIP
      WIP.setWIP(WIP_measured)
      lastWIP = WIP_measured

int get_current_WIP(Date date)
  current_WIP = 0
  for WIP in  Arraylist
    if date EQUALS WIP.getStartDate()
      Nr_of_dates_to_decrement++
return current_WIP

int Nr_of_finished_dates(Date date)
  Nr_of_dates_to_decrement = 0
  for WIP in Arraylist
    if date AFTER WIP.getEndDate() DO
      if date not picked
        Nr_of_dates_to_decrement++
        dateIsPicked(WIP)
return Nr_of_dates_to_decrement
```

Listing 4.4: WIP measurement

4.2.4 Example

This section will provide a comprehensive example of how the WIP algorithm works. Figure 4.1 shows tasks id's in the y-axis and dates in the x-axis. The green line indicates the duration of the task. The figure helps visualizes how many WIPs there are for a given date. For example on the date 2010-10-12, tasks 3, 5 and 6 are in progress, which means the WIP is 3 for 2010-10-12.

In this example, dates from table 4.3 will be used to illustrate how the algorithm measure WIP, while figure 4.1 visualize WIP for each date.

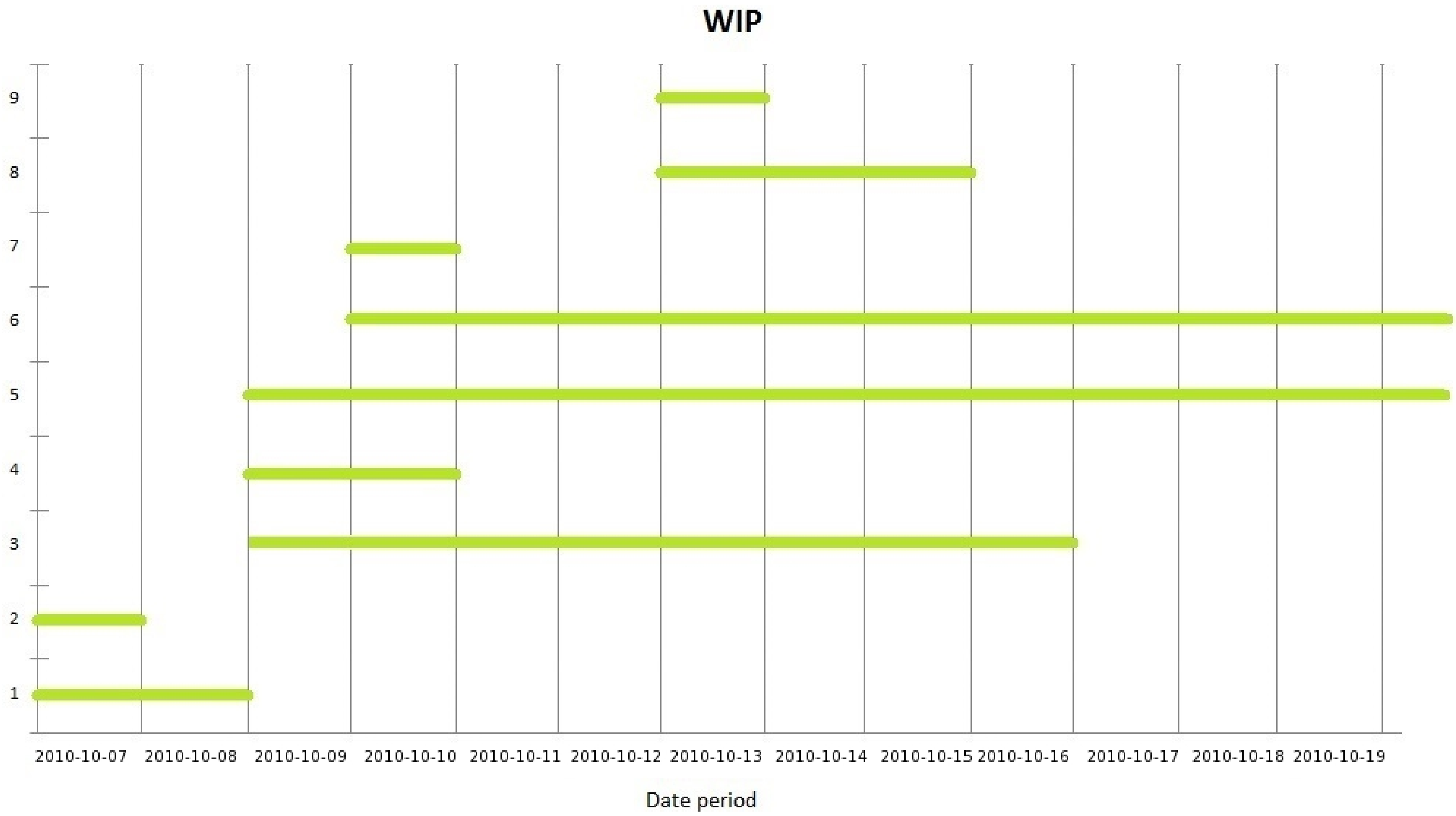


Figure 4.1: Illustrating the WIP timeline for example stated in section 4.2.4

Task ID	Date From	Date To	Team	Process Type
1	2010-10-07	2010-10-08	Team One	Kanban
2	2010-10-07	2010-10-07	Team One	Kanban
3	2010-10-09	2010-10-16	Team One	Kanban
4	2010-10-09	2010-10-10	Team One	Kanban
5	2010-10-09	2010-11-04	Team One	Kanban
6	2010-10-10	2010-11-05	Team One	Kanban
7	2010-10-10	2010-10-10	Team One	Kanban
8	2010-10-13	2010-10-15	Team One	Kanban
9	2010-10-13	2010-10-13	Team One	Kanban

Table 4.3: Showing Task ID, Date From and Date to

4.2.4.1 Step 1

The program will first read in the first line of table 4.3. The first line is the one labeled with task id one. The program creates the WIP-object for line one and it will look like the listing 4.5. The program will follow the exact same procedure until all the dates are read in.

```
WIP = new WIP()
WIP.start = 2010-10-07
WIP.end = 2010-10-08
WIP.team = "Team One"
WIP.processType = "Kanban"
WIP.WIP = 1
```

Listing 4.5: Creating WIP-object

4.2.4.2 Step 2

Now that the whole set has been read in and saved. The next thing to do is to create the remaining dates. The Arraylist contains all the dates from table 4.3. The program will now extract the first and the last date from the Arraylist. Before this step, the objects in the Arraylist are sorted by date. The first date is 2010-10-07 and the last date is 2010-10-13. The program will check if the date after 2010-10-07 contains in the set, which it don't. The program then generates a WIP object for the date 2010-10-08 and adds it to the Arraylist as shown in listing 4.6. After the date is created, the program will see if the date 2008-10-09 exits and will do so for all dates until the date 2010-10-13.

```
void createNewWIP(Date d, String team)
  WIP.start = d
  WIP.end = d
  WIP.team = team
  WIP.processType = "Uknown"
  WIP.WIP = 0
```

Listing 4.6: Creating WIP-object

4.2.4.3 Step 3

The Arraylist now contains the dates from 2010-10-08 to 2010-10-13. The next and last step is to measure WIP for each date. The program will now loop through the Arraylists. The first date is 2010-10-07. The get_current_wip method from line nine in listing 4.4 will be called with the date 2010-10-07 as parameter. The method will return two, since both task one and two where started at 2010-10-07 as shown by figure 4.1. The next thing to do is to find out how many tasks to decrement the current WIP with. The method Nr_of_fininshed_dates in line seventeen is called with the date 2010-10-07. As shown by the table 4.3 and figure 4.1 there was no task finished at the date 2010-10-07, so the method returns 0. The program then updates the WIP objects' counter to two and saves the WIP in the lastWIP object. The next date is 2010-10-08, who the program made in subsection 4.2.4.2. There is no task started at 2010-10-08, but task one is finished at the date. So the Nr_of_fininshed_dates returns one and flag the current date as shown in listing 4.4 by the line twenty-three. The result of WIP_measure in line five is 1 ($0 - 1 + 2 = 1$). WIP at date 2010-10-08 is one, as shown by figure 4.1. The program will continue this procedure until all the dates are measured. The reason why the date is flagged is to be sure that each date only evaluates ones. The if statement listed in listing 4.4 at line twenty-one assurance that.

4.3 Rest of the variables

To compute the remaining variables; churn, lead time and throughput a new algorithm is required. The algorithm for the three variables was almost identical. First the program reads in the data set from SI. For each of the lines in the data set, the program creates an object and saves the valuable information from the data set in the object. Then each object is saved in a data structure based on team association as showed in 4.7. When all the lines has been read in and all objects has been put in the right data structure the algorithm differs.

```
void addBug(Bug b)
  if b.team EQUALS "TeamOne"
    if dateExists(b.date, TeamOne) EQUALS false
      // if date don't exists, then add the bug
      TeamOne.add(b)

  if b.team EQUALS "TeamTwo"
    if dateExists(b.date, TeamTwo) EQUALS false
      // if date don't exists, then add the bug
      TeamTwo.add(b)

  if b.team EQUALS "TeamThree"
    if dateExists(b.date, TeamThree) EQUALS false
      // if date don't exists, then add the bug
      TeamThree.add(b)

  if b.team EQUALS "TeamFour"
    if dateExists(b.date, TeamFour) EQUALS false
      // if date don't exists, then add the bug
      TeamFour.add(b)

void dateExists(Date d, Arraylist list)
  for Bug b in list
    if b.date EQUALS d
      b.counter++
      return true

return false
```

Listing 4.7: Pseudocode example of how throughput objects are added

4.3.1 Throughput

When the steps described in section 4.3 are finished, the program takes each of the data structures and compute throughput. To compute throughput, a counter representing the throughput for each date is created. The method dateExists in listing 4.8 does the actual computation. The method starts of with a test. If the date of the throughput object is in the data structure, the corresponding counter is incremented. If the date is not in the data structure, the new throughput object is added to the data structure.

```
void dateExists(Throughput tp, Arraylist list)
  for Throughput t in list
    if t.date EQUALS tp.date
      t.counter++
      return

structure.add(tp);
```

Listing 4.8: Pseudocode example of how throughput is measured

4.3.2 Churn

As stated in section 2.8 in order to take churn into account one need to know it's good surrogates. SI has gathered churn with help of Microsoft's Team Foundation Server (TFS). The TFS system automatically records data such as churn and lead time. Based on TFS one can know that churn for SI is a good surrogate.

To measure churn the data set from SI contains three columns ("Lines added, "Lines modified" and "Lines deleted") shown in table 4.4. These three variables are summed together and saved in a variable called "churn" when the data set is read in. These three columns are automatically recorded by TFS. To complete the churn measurements the three columns are multiplied. For task id one, the churn is 2028 ($352 + 307 + 1369 = 2028$). Some tasks has zero churn, for example task with id six, these tasks don't need code in order to be finished such tasks needs technical support to be finished. The churn algorithm is shown in listing 4.9

Task id	Lines added	Lines modified	Lines deleted
1	352	307	1369
2	314	31	15
3	314	31	15
4	62	327	153
5	21	3	0
6	0	0	0

Table 4.4: How churn is presented in the excel document

```
void updateChurn(Churn c, Arraylist list)
  for Churn ch in list
    if ch.date EQUALS c.date
      ch.churn += c.churn
      return
structure.add(c);
```

Listing 4.9: Pseudocode example of how throughput is measured

4.3.3 Lead time

The program does not need to analyze the lead time for each task. The lead time for each task is recorded by TFS. The lead time is represented in the data set by a column as shown in table 4.5. The program will gather all the tasks that are started on the same day and belongs to the same team and add up their lead time together as shown in code listing 4.10.

ID	**Type**	**Lead time**
84096	Feature	1
84118	Bug	25
84096	Feature	7
84118	Bug	13

Table 4.5: How lead time is recorded in the excel document

```

addLeadTime(lead_time t, Arraylist list)
  for lead_time in list
    if lead_time.date EQUALS t.date
      lead_time.lead time+= t.lead time
      return

structure.add(t)
```

Listing 4.10: Pseudocode example of lead time is measured

4.3.4 Lead time and churn

As in the article "Quantifying the Effect of Using Kanban versus Scrum" (Sjøberg, Johnsen and Solberg, 2012) to prevent outliers from having a large effect on the results, the top and lowest ten percent of lead time and churn are removed from the data set. Churn is removed because a module or a feature, which consists of hundred or thousand lines of code could be removed without much work. Lead time is removed because some tasks could be given low priority due to lack of manpower in a given period or tasks could be labeled as not critical and the lead time of these tasks will effect the result.

4.3.5 Bug and feature

To measure feature and bug the program and SPSS was used. The program will generate throughput and churn as described in section 4.3.1 and 4.3.2. When the program has produced throughput and churn, SPSS will be used to produce throughput bug and feature as well as churn bug and churn feature. In order to do so a command called case summaries will be used. The case summaries groups variables based on a common values. In Table 4.6 a excerpt from the result data produced by the program. With the case summaries function the variable *Team name*, *Quarter* and *Type* will be grouped over *churn* in order to find churn feature and churn bug per quarter. Variables with prefix *bug* and *throughput* will be referred to as *sub variables*

Team name	**Churn**	**Date**	**Quarter**	**Type**
Team one	25	2011-12-20	2011-4	Feature
Team two	3	2012-04-19	2012-2	bug
Team one	7	2010-08-06	2010-3	Feature

Table 4.6: A excerpt from the result data produced by the program

4.3.6 Bugs finished, quarter

To get the statistics on number of bugs finished, the same quarter as it was recored the program and SPSS was used. The program extracted the created date and date to values from each tasks and checks if their quarters match. If they do, a boolean value is set to true, otherwise it's set to false. The output is used in SPSS where the boolean value is grouped with the same values as in Subsection 4.3.5. After the SPSS has measured, the number of finished bugs are divided by the total in order to find the percentages bugs finished within a quarter.

4.3.7 Avg days backlog, bug

To get the statistics on the average number of days bugs are in backlog per quarter, the program measure the number of days between the created date and the date from value. The number of days is saved together with the task. SPSS is used on the output of the program to measure the average days backlog for bugs variable.

Chapter 5

Results

The first correlation result showed a medium-high correlation relationship between *team size* and both *WIP* and *throughput*. Based on the correlation values, it was hard to find any evidence if WIP limit matter in software development. This resulted in a new analyze where each variable for each quarter was divided by their team size.

Each section except team size is presented with two correlation tables and two corresponding descriptive statistics tables, one correlation table and descriptive statistic table for the first analyze and the same two tables are presented for the second analyze. *Team size* is presented with one correlation table and one descriptive statistic table. The content of the sections will consist of highlighting the variables with a significant correlation and explain why sub variables differ. To back up any assumptions about the variables, descriptive statistic tables listed in Appendix A and correlation graphs will be used.

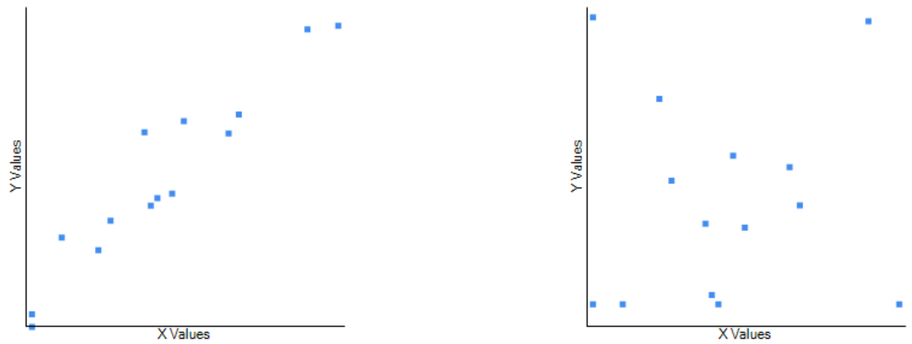
5.1 Correlation - WIP

Table 5.1 shows the correlation table for *WIP*. The variables are shown vertically in the correlation table. Horizontally are the corresponding teams. The team names are shortened. Team one is shortened to T1, team two is shortened to T2 and so on. The correlation table shows team one, three, four, six, seven, nine and ten have significant positive relationship between *WIP* and *throughput*. *Throughput feature* and *throughput bug* is subset of *throughput*. It's natural that these variables have a significant positive correlation to *WIP*, when *throughput* has. But, for team one, six, seven and ten that's not the case.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Throughput	0.74**	0.21	0.76**	0.83**	0.52	0.64*	0.67*	0.47	0.89**	0.61*
Throughput Feature	0.73**	-0.14	0.83**	0.82**	0.25	0.68**	0.63*	0.56*	0.82**	0.20
Throughput bug	0.02	0.25	0.73**	0.82**	0.54*	0.07	0.55	0.15	0.88**	0.63*
Bugs	0.72**	0.20	0.60*	0.56*	0.50	0.46	0.62	0.04	0.58*	0.18
Bugs finished, quarter	0.35	0.10	-0.07	0.56*	0.19	0.19	0.85**	0.23	0.52	0.35
Avg days in backlog, bugs	-0.03	0.44	0.42	0.54*	-0.18	0.02	0.10	0.14	-0.20	-0.18
Lead time	0.75**	0.46	0.49	0.70**	0.57*	0.29	0.68*	0.16	0.23	0.72**
Churn	0.47	-0.71**	-0.32	0.66*	0.03	-0.30	0.15	0.16	-0.09	0.16
Churn feature	0.72**	-0.25	-0.34	0.72**	0.06	-0.36	0.10	0.20	-0.12	0.32
Churn bug	0.15	-0.60*	-0.52	0.62*	0.11	0.77**	-0.05	-0.22	-0.30	-0.10
Team size	0.68**	0.35	0.78**	0.06	0.57*	0.77**	0.62	0.65*	0.54	0.76**

Table 5.1: Correlation - WIP - When team size is **not** taken into account
* Correlation is significant at the 0.05 level (2-tailed).
** Correlation is significant at the 0.01 level (2-tailed).

Both *throughput* and *throughput feature* in team one's case has significant positive relationship with *WIP*, while *throughput bug* don't. *Throughput* and *throughput feature* has the correlation values 0.74 and 0.73, while *throughput bug's* value is 0.02. The possible cause *throughput bug* don't has a significant correlation value with *WIP*, while *throughput* does, might be because *throughput bug* consist of 37% (108/290) of the *throughput* dates, as shown in the total rows in Tables A.1b and A.2b. But it's possible to have a close relationship although, since the correlation is based on the mean values. *Throughput feature* has a total mean of 13.7, *throughput bug* has the total mean of 6.4 and *throughput* has the total mean of 11 as shown in Tables A.1b, A.2a and A.2b. The mean values strengthens the assumption that *throughput feature* represents most of the *throughput* variable. The correlation graphs in Figure 5.1 and *throughput's* correlation table in Section 5.4 confirms the assumption. The pattern of dots in Figure 5.1a shows a significant positive correlation, while the dots in Figure 5.1b have no specific pattern. The throughput correlation table and the figure proves the assumption that *throughput feature* represents most of *throughput* for team one.



(a) Throughput and throughput feature (b) Throughput and throughput bugs

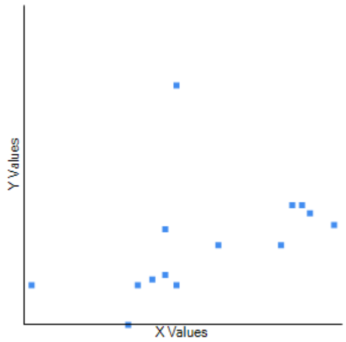
Figure 5.1: Correlation graphs between WIP and the throughput variables.

The same issue encounter for both team six and seven. In case of team six, both *throughput* and *throughput feature* have significant positive relationship with WIP, while *throughput bug* don't. *Throughput*, *throughput feature* and *throughput bug* has the correlation values 0.64, 0.68 and 0.07. The total row in Tables A.26b, A.27a and A.27b shows *throughput feature* consist of 609 dates and has a mean value of 4.8. While *throughput bug* consist of 82 dates and a mean of 3.3. *Throughout* consist of 691 dates and has a mean value of 4.58. The mean values and the number of dates strengthens the assumption that *throughput feature* represents more of *throughput* than *throughput bug* does. The throughput correlation Table 5.13, proves the assumption with *Throughput feature's* correlation value of 0.99 and *throughput bugs* value of 0.04.

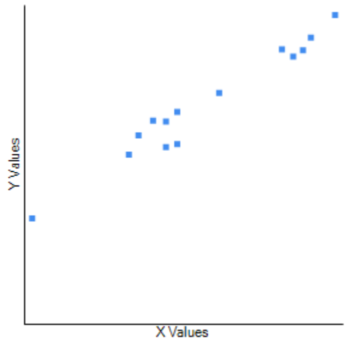
For team seven's case, the difference between *throughput*, *throughput feature* and *throughput bug* is very small, which also can be assumed by the total row in Tables A.31b, A.32a and A.32b. The total rows, *throughput* has a total mean of 2.7, while *throughput feature* has 2.8 and *throughput bug* has 2.6. *Throughput feature* contributes 156 dates to *throughput* and *throughput bug* contributes 172 dates. The *throughput* correlation table in Section 5.4 proves the assumption with values of .91 for both *throughput feature* and *bug*.

Team ten has significant positive relationship between *WIP* and both *throughput* and *throughput bug*, but not for *throughput feature*. *Throughput* for team ten consist of 404 dates as show in the total row in Table A.31b. *Throughput bug* represents 335 of these dates, while *throughput feature* stands for the remaining 69 dates as shown in Tables A.32a and A.32b. But the overall mean for *throughput*, *throughput feature* and *throughput bug* is 2.2, 2.3 and 2.2, which could reflect a close relationship between these three variables. Both the throughput correlation table in Section 5.4 and the correlation graphs in Figures 5.2 disproves that assumption. The Figures 5.2a shows a vague significant positive correlation, while Figure 5.2b shows a significant positive correlation. The correlation table shows *throughput bug* with a correlation of 0.98

and *throughput feature* with a correlation of 0.43. Which proves that *throughput bug* represents most of *throughput*.



(a) Throughput and throughput feature



(b) Throughput and throughput bug

Figure 5.2: Correlation graphs between throughput and the throughput sub variables for team ten.

Team one, three, four and nine has a significant positive relationship between WIP and bugs as well. For team one, differ all the three churn variable from each other, based on the correlation Table 5.1. *Churn* has a value of 0.47, *churn feature* has a value of 0.72 and *churn bug* of 0.15. Judging from these variables, it will look both *churn feature* and *churn bug* contribute to *churn*. The descriptive statistic Tables A.3b, A.4a and A.4b, empower the assumption. The total mean of *churn* is 20.2, while *churn feature* has total mean of 24.5 and *churn bug* has a total mean of 16.8. The descriptive statistic tables also shows that *churn feature* contribute 150 dates to *churn*, while *churn bug* contribute 189 dates. The churn correlation table in Section 5.5 proves the assumption. Both *churn feature* (.566) and *churn bug* (.798) have a significant positive correlation to *churn*. Judging from the correlation table from Section 5.4, one can assume that correlation is not transitive. The paper "The Non-Transitivity of Pearson's Correlation Coefficient: An Educational Perspective" (Vesaliusstraat, n.d.) proves the assumption.

Churn and *churn bug* have a significant negative correlation in team two's case. *Churn feature* on the other hand has a correlation value of -0.25. According to the correlation values, one can assume that *churn bug* represents most of *churn*. The Tables A.8b, A.9a and A.9b empower the assumption. *Churn bug* contribute 521 dates and has a total mean value of 36.3. *Churn feature* contribute 257 dates and has a mean value of 100.6. The total churn contains 778 dates and has a total mean of 57.6. The churn correlation table in Section 5.5 proves the hypothesis of *churn bug* contribute most. *churn bug* of has the value of 0.70 and *churn feature* has the value of 0.58. But both of them has a significant relationship with *churn*.

In team six's case, *churn bug* has a positive correlation of 0.77, while both *churn* and *churn feature* has the values of -0.30 and -0.36. Based on these values, one can assume *churn feature* represents most of *churn*. The Tables A.28b, A.29a and A.29b backs the theory. The tables shows the variable *churn feature* contribute 576 dates to *churn* and has a total mean of 105.9. *Churn bug* contains of 180 dates and has a total mean value of 73.8. *Churn* has 756 dates and a total mean value of 98.3. The churn correlation table in Section 5.5 proves the assumption of *churn feature* represent most of *churn*. One can see that *churn feature* has a correlation value of 0.98, while *churn bug* has a value of -0.02.

Throughput bug for team five has a significant value of 0.54, while *throughput* has a value of 0.52 and *throughput feature* a value of 0.25. Based on these values, one can assume that *throughput bug* represents most of *throughput* for team five. The descriptive statistic Tables A.21b, A.22a and A.22b shows the number of dates are 657 for *throughput*. Out of the 657 dates, represents *throughput feature* 108 dates, and *throughput bug* 556 dates. These values also points towards the fact that *throughput bug* represents most of *throughput*. The overall mean for *throughput* is 6.3, for *throughput feature* it's 5.7 and for *throughput bug* it is 6.4. Based on these values, it looks like both the sub variables contribute. The throughput correlation Table 5.13 proves with the values 0.85 for *throughput feature* and 0.99 for *throughput bug* that both the sub variables contribute.

Team one, four, five, seven and ten has a positive correlation *WIP* and *lead time*. Team four also has a positive correlation between each of the variables except *team size*. Team eight has a significant correlation for *throughput feature*, while *throughput* don't. The *throughput* relationship for team eight is explained in Section 5.2.

The descriptive statistics tables in each section are based on summary of the correlation values. The tables shows number of values measured(N), mean, median standard deviation(Std.dev), maximum(max) and minimum(min) values from the correlation table. In Table 5.2, the total mean of *throughput* is 0.6 for all the teams and the median is 0.7, which reflects the strong positive correlation between *WIP* and *throughput*. The overall correlation between *WIP* and *team size* is 0.6 as well. The *bug* variables has a total mean of 0.4 with a median of 0.5, this indicates that *WIP* and *bug* has a medium strong relationship.

	N	Mean	Median	Std.Dev	Max	Min
Throughput	10	0.6	0.7	0.2	0.9	0.2
Throughput ft	10	0.5	0.7	0.3	0.8	-0.1
Throughput bug	10	0.5	0.5	0.3	0.9	0
Bugs	10	0.4	0.5	0.2	0.7	0
Bugs finished, quarter	10	0.3	0.3	0.3	0.9	-0.1
Avg days backlog, bugs	10	0.1	0.1	0.3	0.5	-0.2
Lead time	10	0.5	0.5	0.2	0.7	0.2
Churn	10	0	0.1	0.4	0.7	-0.7
Churn ft	10	0.1	0.1	0.4	0.7	-0.4
Churn bug	10	-0	-0	0.4	0.8	-0.6
Team size	10	0.6	0.6	0.2	0.8	0.1

Table 5.2: Descriptive Statistic - Correlation - WIP - When team size is **not** taken into account

In Table 5.3 has team two a significant positive correlation between *WIP* and both *lead time* and *bugs finished, quarter*. Team three has a significant negative correlation between WIP and *bugs finished, quarter*. Team seven has significant a positive correlation for the three variables *bugs*, *bugs, finished quarter* and *lead time*. Team eight has a significant for *bugs finished, quarter*. Team nine has a significant negative correlation for both *bugs, finished quarter* and *churn*. Team ten has a significant negative correlation for *avg days in backlog, bugs*.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Throughput	0.37	0.59*	0.57*	0.86**	0.11	0.08	0.49	0.28	0.66*	-0.21
Throughput Feature	0.31	0.47	0.71**	0.85**	0	0.14	0.46	-0.26	0.60*	-0.16
Throughput bug	0.09	0.65*	0.52	0.85**	0.11	0.07	0.57	0.37	0.58*	-0.22
Bugs	0.10	0.49	0.25	0.27	0.11	0.25	0.75*	0.32	-0.05	-0.28
Bugs finished, quarter	-0.28	0.71**	-0.62*	0.74**	-0.24	-0.04	0.85**	0.82**	0.32	-0.28
Avg days in backlog, bugs	0.03	-0.31	0.51	0.10	-0.14	0	0.16	-0.17	-0.40	-0.61*
Leadtime	-0.09	0.67**	-0.03	0.87**	0.03	0.32	0.77*	-0.09	-0.18	-0.05
Churn	-0.27	0.16	-0.29	0.77**	-0.09	-0.35	-0.17	0.39	-0.34	-0.37
Churn feature	0.37	0.03	-0.38	0.78**	-0.15	-0.39	-0.17	-0.04	-0.17	0.08
Churn bug	-0.29	0.21	-0.39	0.26	-0.07	0.12	-0.17	0.66*	-0.49	-0.43

Table 5.3: Correlation - WIP - When team size is taken into account
* Correlation is significant at the 0.05 level (2-tailed).
** Correlation is significant at the 0.01 level (2-tailed).

For team three is *throughput bug* close to a significant positive correlation with a value of 0.52, while both *throughout* and *throughput feature* has the values 0.57 and 0.71. With these correlation values, it looks like *throughput bug* represents most of the *throughput* variable, even through *throughput bug's* value is not a significant. The descriptive statistic tables empower the assumption. *Throughput* contains 542 dates and has a total mean value of 3.7. *Throughput feature* represents 200 of these dates, and has a mean value of 3.3, while *throughput bug* represents the remaining 342 dates, and has a total mean value of 4.0, shown in tables A.11b, A.12a and A.12b. The *throughput* correlation Table 5.15 proves the assumption with a correlation value of 0.90 for *throughput feature* and a value of 0.98 for *throughput bug*

Team four has a signification positive correlation for all the *throughputs* variables, *bugs finished*, *quarter*, *lead time*, *churn* and *churn feature*. The reason *churn bug* differ is stated in section 5.2. For team two differ *throughput feature* based on the correlation values in Table 5.3. The reason for this is stated in Section 5.3. For team eight differ *churn feature* from *churn* based on the correlation value in Table 5.3. The reason for this is stated in Section 5.2. Team nine has a significant positive correlation for all *throughput* variables. Team nine also has a significant positive correlation for all the *throughput* variables. In Table 5.4, *throughput* has a overall correlation value of 0.4, while rest of the variables except *throughputs* sub variables has a correlation values 0.2 or less.

	N	Mean	Median	Std.Dev	Max	Min
Throughput	10	0.4	0.4	0.3	0.9	-0.2
Throughput ft	10	0.3	0.4	0.4	0.9	-0.3
Throughput bug	10	0.4	0.4	0.3	0.9	-0.2
Bugs	10	0.2	0.2	0.3	0.8	-0.3
Bugs finished, quarter	10	0.2	0.1	0.6	0.8	-0.6
Avg days backlog, bugs	10	-0.1	-0.1	0.3	0.5	-0.6
Lead time	10	0.2	0	0.4	0.9	-0.2
Churn	10	-0.1	-0.2	0.4	0.8	-0.4
Churn ft	10	0	-0.1	0.4	0.8	-0.4
Churn bug	10	-0.1	-0.1	0.4	0.7	-0.5

Table 5.4: Descriptive Statistic - Correlation - WIP - When team size is taken into account

5.2 Correlation - Lead time

The Table 5.5 shows the correlation between *lead time* and the variables. Team one, four, five, seven and ten has a significant positive correlation between *lead time* and *WIP*. Team one also has a significant positive correlation between *lead time* and all the variables except *throughput bug*, *avg days in backlog, bugs* and *churn bug*. Both the sub variables *throughput bug* and *churn bug* differ from respectively *throughput* and *churn*.

Both the *throughput* and *churn* relationship was stated in Sections 5.1 and 5.3.

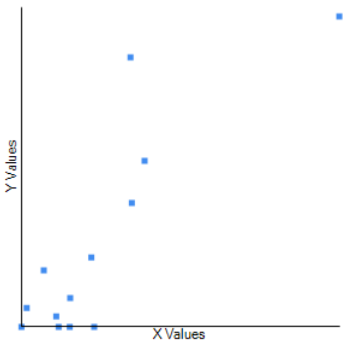
The three *churn* variables for team three has the correlation values of -0.45 for *churn*, -0.27 for *churn feature* and -0.64 for *churn bug*. These values indicates more contribution from *churn bug* than *churn feature*. The total dates and the mean churn in Tables A.13b, A.14a and A.14b empower the assumption. The total *churn* consist of 576 dates and a total mean of 61.8. *Churn feature* represents 205 of these dates and has a total mean of 98.9. *Churn bug* answers for the remaining 371 dates and has a total mean of 41.4. The *churn* correlation table in Section 5.5 proves the assumption. Still, *churn bug* has a correlation of 0.85, while the correlation value between *churn feature* and *churn* is 0.90. The strong relationship between *churn* and *churn bug* shows that total dates and total mean can be used as an indicator of the relationship between variables, but it can't state it.

	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**
WIP	0.75**	0.46	0.49	0.70**	0.57*	0.29	0.68*	0.16	0.23	0.72**
Throughput	0.70**	0.67**	0.49	0.68**	0.36	0.13	0.47	0.54	0.42	0.32
Throughput Feature	0.73**	0.09	0.44	0.64*	0.14	0.10	0.41	0.62*	0.41	-0.05
Throughput bug	-0.30	0.60*	0.52	0.64*	0.42	-0.01	0.61	-0.17	0.28	0.37
Bugs	0.77**	0.50	0.54*	0.31	0.32	-0.23	0.69*	-0.13	0.44	0.04
Bugs finished, quarter	0.70**	-0.14	0.20	0.23	-0.09	-0.27	0.73*	0.37	0.53	0.19
Avg days in backlog, bugs	0.06	0.40	0.07	0.07	-0.08	-0.03	0.57	-0.12	-0.48	-0.52
Churn	0.70**	-0.42	-0.45	0.97**	0.18	-0.34	0.37	0.91**	-0.37	-0.04
Churn feature	0.86**	0.20	-0.27	0.96**	0.11	-0.31	0.39	0.79**	-0.46	0.32
Churn bug	0.26	-0.39	-0.64*	0.20	0.24	0.28	0.16	-0.12	-0.08	-0.27
Team size	0.61*	0.38	0.44	-0.30	0.36	-0.11	0.59	0.22	0.38	0.53

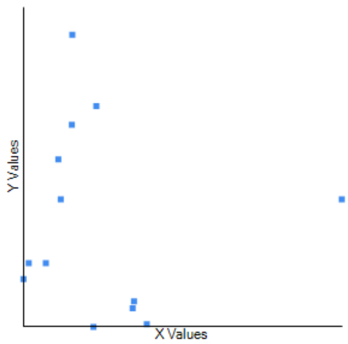
Table 5.5: Correlation - Lead time - Without taken time size into account
* Correlation is significant at the 0.05 level (2-tailed).
** Correlation is significant at the 0.01 level (2-tailed).

In both team four and eight's case, the variable *churn bug* differ from *churn* according to Table 5.5. The *churn* variable representing team four has the correlation value of 0.97 and *churn bug* has the correlation value of 0.2. The Tables A.18b, A.19a and A.19b shows that *churn* consist of 574 dates. *Churn bug* represents 78 of these dates while *churn feature* represents the remaining 496 dates. *Churn features* has the total mean of 8.4, *churn bug's* mean is 1 and *churn's* mean is 7.4. These variables clearly indicates the strong relationship between *churn feature* and *churn*. The churn correlation Table 5.17 verifies the theory with *Churn feature* has the correlation value of 0.99 and *churn bug* has the correlation value 0.13.

Team eight's *churn bug* differ from *churn* in respect of Table 5.5. The correlation values between *lead time* and *churn* is 0.91 and between *lead time* and *churn bug* it's -0.12. Based on these variables, it will look like *churn feature* represents a greater part of the *churn*. The Tables A.38b, A.39a and A.39b indicates otherwise. *Churn* is composed of 137 dates, and has a total mean of 13.4. *Churn feature* represents 79 of these tasks, and has a total mean of 17.4. *Churn bug* represents the remaining 58 tasks and has a total mean of 8. The tables indicates both *Churn feature* and *Churn bug* contribute to *churn*. The Figure 5.3 and correlation Table 5.17 disapproves the assumption. The *churn* correlation Table 5.17 shows that *churn feature* has the correlation of .84, while *churn bug* has the value of -0.1. This is the same as shown in the correlation graphs. The Figure 5.3a shows a clear positive correlation, while Figure 5.3b shows no pattern and shows a correlation close to 0.



(a) Churn and churn feature



(b) Churn and churn bug

Figure 5.3: Correlation graphs between the churn variables for team eight.

Throughput feature has a significant correlation for team eight as well. The *throughput* relationship is explained in Section 5.3. Team three and seven has a significant positive correlation between *lead time* and *bugs*. Team seven also has a significant positive correlation between *lead time* and *bugs finished, quarter*. Team two has a significant positive correlation for *throughput* and *throughput bug*. The *throughput* relationship is explained in Section 5.2. The descriptive statistic table for lead time shown in Table 5.6 shows that both *WIP* and *throughput* has a medium to strong correlation with *lead time*.

	N	Mean	Median	Std.Dev	Max	Min
WIP	10	0.5	0.5	0.2	0.7	0.2
Throughput	10	0.5	0.5	0.2	0.7	0.1
Throughput ft	10	0.4	0.4	0.3	0.7	-0.1
Throughput bug	10	0.3	0.4	0.3	0.6	-0.3
Bugs	10	0.3	0.4	0.3	0.8	-0.2
Bugs finished, quarter	10	0.2	0.2	0.3	0.7	-0.3
Avg days backlog, bugs	10	-0	0	0.3	0.6	-0.5
Churn	10	0.2	0.1	0.6	1	-0.5
Churn ft	10	0.3	0.3	0.5	1	-0.5
Churn bug	10	-0.1	-0.1	0.3	0.3	-0.6
Team size	10	0.3	0.4	0.3	0.6	-0.3

Table 5.6: Descriptive Statistic - Correlation - Lead time- Without taken time size into account

In Table 5.7 has Team one a significant positive correlation for *bugs*, *bugs finished, quarter*, *churn* and *churn bug*. The *churn* relationship were explained in Section 5.1. Team two has a significant positive correlation for *WIP*, the *throughput* variables, *bugs*, *bugs finished, quarter* and *churn bug*, but not *churn feature*. The relationship between the *churn* variables were explained in section 5.1. Team four has a significant positive correlation between *WIP*, the *throughput* variables, *churn* and *churn feature*. The *churn* relationship for team four is explained in Section 5.2.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
WIP	-0.09	0.67**	-0.03	0.87**	0.03	0.32	0.77*	-0.09	-0.18	-0.05
Throughput	0.06	0.88**	0.26	0.90**	0.80**	0.69**	0.33	0.24	0.32	0.90**
Throughput Feature	0.01	0.75**	0.16	0.89**	-0.05	0.61*	0.34	0.87**	0.20	-0.26
Throughput bug	0.51	0.84**	0.29	0.89**	0.84**	0.41	0.56	-0.27	0.25	0.91**
Bugs	0.83**	0.72**	0.48	0.22	0.72**	0.18	0.85**	-0.23	0.54	0.88**
Bugs finished, quarter	0.88**	0.79**	0.32	0.52	0.80**	0.36	0.77**	-0.04	0.45	0.57*
Avg days in backlog, bugs	0.41	-0.52	-0.04	-0.07	0.85**	-0.09	0.51	-0.20	-0.17	0.10
Churn	0.72**	0.49	0.16	0.96**	0.94**	0.04	0.08	0.71**	-0.18	0.01
Churn feature	-0.52	0.38	0.22	0.96**	-0.21	0.11	0.05	0.86**	-0.28	-0.24
Churn bug	0.73**	0.55*	0.07	0.13	0.95**	-0.17	-0.06	-0.12	0.24	0.03

Table 5.7: Correlation - Lead time - With team size taken into account
* Correlation is significant at the 0.05 level (2-tailed).
** Correlation is significant at the 0.01 level (2-tailed).

Team five has significant correlation value ff *bugs*, *bugs, finished, quarter*, *avg days in* *backlog, bugs*, *churn*, *churn bug*, *throughput* and *throughput bug*, but not *throughput bug*. The relationship between the *throughput* variables were stated in Section 5.1, while the *churn* relationship is explained in Section 5.5. Team six has a significant correlation for *throughput* and *throughput feature*. The *throughput* relationship were explained in Section 5.1. Team seven has a significant correlation for *WIP*, *bugs* and *bugs, finished, quarter*. Team eight has a significant correlation for *throughput feature*, *churn* and *churn feature*, but not *throughput* or *churn bug*. The *throughput* relationship is explained in Section 5.4, while the *churn* relationship is explained in Section 5.5. Team ten has a significant positive correlation for *throughput*, *throughput bug* and *bugs*, but not *throughput feature*. The *throughput* relationship were explained in Section 5.1. The Table 5.8 Shows a medium or strong relationship for *throughput* variables, *bugs*, *bugs finished, quarter* and *churn*.

	N	Mean	Median	Std.Dev	Max	Min
WIP	10	0.2	0	0.4	0.9	-0.2
Throughput	10	0.5	0.5	0.3	0.9	0.1
Throughput ft	10	0.4	0.3	0.4	0.9	-0.3
Throughput bug	10	0.5	0.5	0.4	0.9	-0.3
Bugs	10	0.5	0.6	0.4	0.9	-0.2
Bugs finished, quarter	10	0.5	0.5	0.3	0.9	-0
Avg days backlog, bugs	10	0.1	-0.1	0.4	0.8	-0.5
Churn	10	0.4	0.3	0.4	1	-0.2
Churn ft	10	0.1	0.1	0.5	1	-0.5
Churn bug	10	0.2	0.1	0.4	1	-0.2

Table 5.8: Descriptive Statistic - Correlation - Lead time - With team size taken into account

5.3 Correlation - Bugs

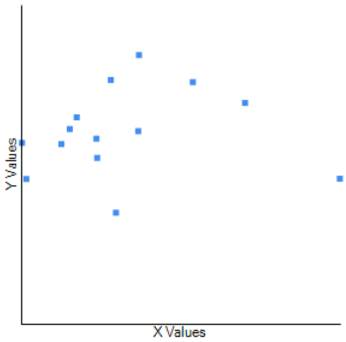
This section contains information about the correlation table between the variables and *bugs*. In Table 5.9 Teams one, two, three, four, five, nine and ten all have a significant positive correlation between *bugs* and *throughput*. For team three, four, five and nine's case, all *throughput* variables have a significant positive correlation with *bugs*. For team one and ten, the *throughput* variables differ. The *throughput* relationships were stated in Section 5.1.

Team one, three, four and nine has a significant positive correlation for *WIP*. Team one also has a significant positive correlation for *lead time*, *churn*, *churn feature* and *team size*. The *churn* relationship for team one were stated in Section 5.1. Team five has a significant positive correlation for *team size* while *team eight* has a significant positive correlation for *churn bug*. The *churn* relationship were explained in Section 5.2.

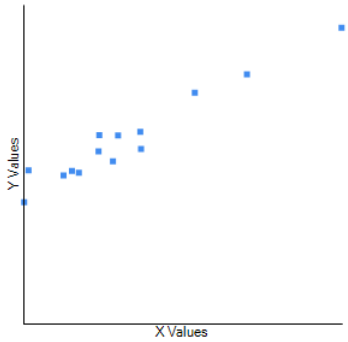
	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
WIP	0.72**	0.20	0.60*	0.56*	0.50	0.46	0.62	0.04	0.58*	0.18
Throughput	0.69**	0.81**	0.88**	0.59*	0.97**	0.27	0.53	0.41	0.70**	0.56*
Throughput Feature	0.74**	0.01	0.82**	0.58*	0.88**	0.30	0.56	0.22	0.60*	-0.14
Throughput bug	-0.17	0.83**	0.87**	0.58*	0.96**	0.69**	0.50	0.92**	0.65*	0.59*
Bugs finished, quarter	0.50	-0.18	0.12	0.40	0.17	0.76**	0.79**	0.18	0.70**	0.05
Avg days in backlog, bugs	0.52	0.38	0.43	0.30	0.18	0.24	0.23	0.28	0.21	0.13
Lead time	0.77**	0.50	0.54*	0.31	0.32	-0.23	0.69*	-0.13	0.44	0.04
Churn	0.62*	-0.27	0.10	0.27	-0.06	-0.12	0.11	-0.16	-0.48	0.04
Churn feature	0.77**	0.01	0.09	0.35	0.43	-0.10	0.11	-0.25	-0.62*	0.07
Churn bug	0.42	-0.19	-0.19	0.53	-0.11	0.42	0.12	0.65*	-0.04	0
Team size	0.80**	0.26	0.27	0.17	0.71**	0.41	0.41	0.42	0.41	0.16

Table 5.9: Correlation - Bugs - Without team size is taken into account
* Correlation is significant at the 0.05 level (2-tailed).
** Correlation is significant at the 0.01 level (2-tailed).

Team two's *throughput feature* differ from *throughput* based on the correlation values form Table 5.9. One could believe the reason is because *throughput bug* consists of 2/3 of *throughput* (460/690) dates as shown in the descriptive statistic Tables A.6b and A.7b. But the two tables and Table A.7a shows that the total mean of *throughput* is 4.39. While for *throughput bug* it's 4.8 and 3.7 for *throughput feature*. These three variables are quiet close, which could reflect that these three variables could be close, based on correlation measurement. But, the Figures in 5.4 and the throughput correlation table in 5.4 shows that they are not. The Figures 5.4b shows a significant positive correlation, while Figure 5.4a shows dots that are more randomly placed. The throughput correlation Table 5.13 shows a correlation relationship with a value of 0.97 for *throughput bug* and 0.1 for *throughput feature*.



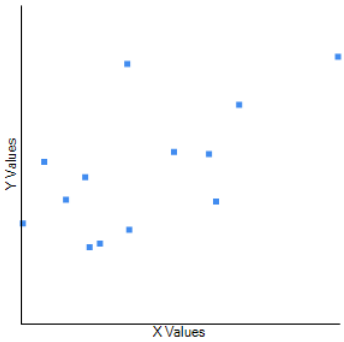
(a) Throughput and throughput feature



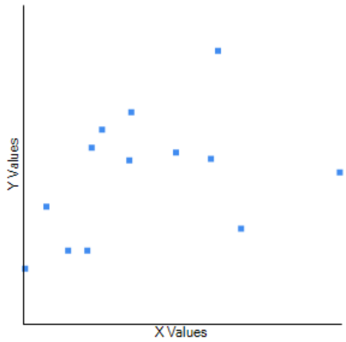
(b) Throughput and throughput bug

Figure 5.4: Correlation graphs between the throughput variables for team two.

Team six, seven and nine has a significant positive correlation between the variable *Bugs finished*, *quarter* and *lead time*. Team nine also has a significant negative correlation between *bug* and *churn feature*, but not *churn*. *Churn feature* represents 201 of 538 dates as shown in Tables A.43b and A.44a. The total mean of *churn feature* is 115.9, the total mean for *churn* is 72.2 and the total mean for *churn bug* is 46.1, as shown in the previous stated tables and Table A.49b. Judging from these numbers, both *churn feature* and *churn bug* contribute to *churn*. The Figure 5.5 shows the correlation values in a graph. There is hard to find any pattern or evidence that *churn feature* and *churn bug* contribute to *churn*. But, The churn correlation Table 5.17 shows *churn feature* correlation values is 0.60 and *churn bug* correlation value is 0.39. This proves that both the sub variables of *churn* contribute, with *churn feature* contributing slightly more .



(a) Churn and churn feature



(b) Churn and churn bug

Figure 5.5: Correlation graphs for the churn variables for team nine.

Team three and seven has a significant positive correlation between *bugs* and *lead time*. Team six has a significant positive correlation for *throughput bug*. The *throughput* relationship were explained in Section 5.1. Team eight has a significant positive correlation for *throughput bug*. The *throughput* relationship is explained in Section 5.4. In Table 5.10, one can see that *WIP* with a mean value of 0.4 has a medium correlation with *bugs* as stated in Section 5.1. *Throughput* has a value of 0.6. This shows that *throughput* have a high correlation with bugs.

	N	Mean	Median	Std.Dev	Max	Min
WIP	10	0.4	0.5	0.2	0.7	0
Throughput	10	0.6	0.6	0.2	1	0.3
Throughput ft	10	0.5	0.6	0.3	0.9	-0.1
Throughput bug	10	0.6	0.7	0.3	1	-0.2
Bugs finished, quarter	10	0.3	0.3	0.3	0.8	-0.2
Avg days backlog, bugs	10	0.3	0.3	0.1	0.5	0.1
Lead time	10	0.3	0.4	0.3	0.8	-0.2
Churn	10	0	-0	0.3	0.6	-0.5
Churn ft	10	0.1	0.1	0.4	0.8	-0.6
Churn bug	10	0	0	0	0.6	-0
Team size	10	0.3	0.4	0.3	0.6	-0.3

Table 5.10: Descriptive Statistic - Correlation - Bugs - Without team size is taken into account

In Table 5.11 has Team one a significant positive correlation for *throughput bug*, *bugs finished, quarter avg days in backlog, bugs* and *lead time*, but not *throughput*. The *throughput* relationship were stated in Section 5.1. Team two has a positive correlation for the *throughput* variables and *leadtime*. Team three has a positive correlation for *throughput* variables and *churn* variables. Team five has a significant positive correlation between *avg days in backlog, bugs*, *lead time*, *churn*, *churn bug*, *throughput* and *throughput bug*. The *churn* relationship is stated in Section 5.2, while the *throughput* relationship were stated in Section 5.1.

	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**
WIP	0.10	0.49	0.25	0.27	0.11	0.25	0.75*	0.32	-0.05	-0.28
Throughput	0.05	0.90**	0.81**	0.32	0.97**	-0.02	0.46	0.82**	0.57*	0.96**
Throughput Feature	-0.12	0.62*	0.66**	0.33	0.50	-0.10	0.48	-0.52	0.36	-0.25
Throughput bug	0.79**	0.92**	0.79**	0.33	0.95**	0.80**	0.60	0.98**	0.64*	0.96**
Bugs finished, quarter	0.59*	0.50	0.44	0.17	0.52	0.77**	0.86**	0.70**	0.19	0.59*
Avg days in backlog, bugs	0.76**	-0.07	0.38	0.18	0.75**	0.30	0.37	0.38	0.37	0.39
Leadtime	0.83**	0.72**	0.48	0.22	0.72**	0.18	0.85**	-0.23	0.54	0.88**
Churn	0.47	0.29	0.67**	0.22	0.73**	0.20	-0.05	0.34	-0.02	0.19
Churn feature	-0.40	0.12	0.62*	0.27	-0.10	0.27	-0.07	-0.26	-0.24	-0.19
Churn bug	0.48	0.35	0.55*	0.44	0.73**	-0.19	-0.08	0.78**	0.43	0.23

Table 5.11: Correlation - Bugs - When team size is taken into account
* Correlation is significant at the 0.05 level (2-tailed).
** Correlation is significant at the 0.01 level (2-tailed).

Team nine has a significant positive correlation for *throughput* and *throughput bug*. The reason *throughput feature* don't has a significant correlation while *throughput* has it, in respect of the Table 5.11, could be because *throughput bug* represents most of the *throughput* variable, as shown in Tables A.41b, A.42a and A.42b. The *throughput* variable contains 521 dates and has a mean value of 2.6. *Throughput feature* represents 214 of these dates and has a total mean of 2.4, while *throughput bug* represents the remaining 307 dates and has a total mean of 2.8. These variables also shows that both the *throughput* sub variables contributes. The throughput correlation Table 5.15 proves that both the attributes contributes, but *throughput bug* contributes the most with a correlation value of 0.95, while *throughput feature* has a correlation value of 0.82.

Team six has a significant positive correlation for *throughput bug* and *bugs finished quarter*. The *throughput* relationship were explained in Section 5.1. Team seven has a significant positive correlation for *WIP*, *bugs finished, quarter* and *lead time*. Team eight has a significant positive correlation for *throughput*, *throughput bug*, *bugs finished, quarter* and *churn bug*, but not *throughput feature*. The *throughput* relationship is explained in Section 5.4 while the *churn* relationship were explained in Section 5.2. Team ten has a significant positive correlation for *throughput*, *throughput bug*, *bugs finished, quarter* and *lead time*, but not *throughput feature*. The *throughput* relationship were stated in Section 5.1. Table 5.12 shows a medium or strong relationship for *throughput*, *throughput bug*, *bugs finished, quarter*, *avg days backlog, days*, *lead time* and *churn bug*.

	N	Mean	Median	Std.Dev	Max	Min
WIP	10	0.2	0.2	0.3	0.8	-0.3
Throughput	10	0.6	0.7	0.4	1	0
Throughput ft	10	0.2	0.3	0.4	0.7	-0.5
Throughput bug	10	0.8	0.8	0.2	1	0.3
Bugs finished, quarter	10	0.5	0.6	0.2	0.9	0.2
Avg days backlog, bugs	10	0.4	0.4	0.2	0.8	-0.1
Lead time	10	0.5	0.6	0.4	0.9	-0.2
Churn	10	0.3	0.3	0.3	0.7	-0.1
Churn ft	10	0	-0.1	0.3	0.6	-0.4
Churn bug	10	0.4	0.4	0.3	0.8	-0.2

Table 5.12: Descriptive Statistic - Correlation - Bugs - When team size is taken into account

5.4 Correlation - Throughput

This section shows the correlation table between *throughput* and the variables. Table 5.13 shows that *Throughput* has a significant correlation to either *throughput feature* or *throughput bug* for each of the teams. For team three, four, five, seven and nine, both the sub variables of *throughput* have significant positive correlation to *throughput*. For team one and six, *throughput feature* differ from *throughput*, the reason were explained in Section 5.1. The Section 5.1 also explain the *throughput* relationship for team ten. Team two's *throughput* relationship is explained in Section 5.3. Team one, three, four, six, seven, nine and ten has a positive correlation for *WIP*. Team one, two, three, four and five has a significant correlation for *bugs*.

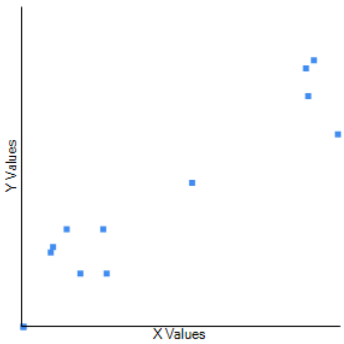
Both team nine and ten has a positive significant correlation for *bugs* and the *bugs finished, quarter*. Team six has a significant correlation between for *churn bug*. The *churn* relationship were stated in Section 5.1. The three *Churn* variables differ in respect of Table 5.13 for team one. The reason for this were stated in Section 5.1

For team eight differ *throughput bug* from *throughput*. *Throughput bug* consist of 99 dates and the has total mean of 1.5. *Throughput feature* consist of 92 dates and a total mean of 3.2. While *throughput* has a total mean of 2.3 and a total of 191 dates. These numbers are gathered from the descriptive statistic Tables A.36b, A.37a and A.37b. On the basis

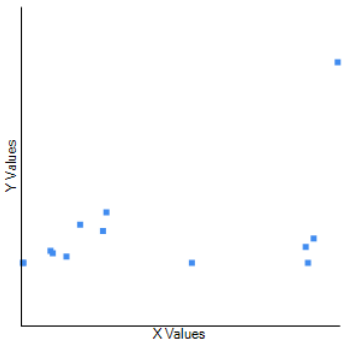
	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
WIP	0.74**	0.21	0.76**	0.83**	0.52	0.64*	0.67*	0.47	0.89**	0.61*
Throughput Feature	0.96**	0.09	0.93**	1	0.85**	0.99**	0.91**	0.94**	0.88**	0.43
Throughput bug	0.03	0.97**	0.99**	1	0.99**	0.04	0.91**	0.44	0.96**	0.98**
Bugs	0.69**	0.81**	0.88**	0.59*	0.97**	0.27	0.53	0.41	0.70**	0.56*
Bugs finished, quarter	0.16	0.12	0.23	0.39	0.12	0.12	0.58	0.33	0.70**	0.59*
Avg days in backlog, bugs	0.16	0.12	0.45	0.37	0.14	-0.17	0.21	-0.17	-0.41	-0.09
Lead time	0.70**	0.67**	0.49	0.68**	0.36	0.13	0.47	0.54	0.42	0.32
Churn	0.37	-0.43	-0.18	0.72**	-0.06	-0.40	0.60	0.59*	-0.14	0.02
Churn feature	0.78**	-0.10	-0.20	0.81**	0.41	-0.40	0.43	0.43	-0.29	-0.20
Churn bug	-0.06	-0.21	-0.33	0.49	-0.10	0.57*	-0.10	0.03	-0.29	-0.06
Team size	0.70**	0.05	0.52	0.16	0.69**	0.86**	0.62	0.75**	0.53	0.57*

Table 5.13: Correlation - Throughput - Without time size is taken into account
* Correlation is significant at the 0.05 level (2-tailed).
** Correlation is significant at the 0.01 level (2-tailed).

of these numbers it will look like both of the sub variables of *throughput* contributes and both of them should have close relationship to *throughput*. The Figure 5.6 shows otherwise. In Figure 5.6a, the dots are clearly moving in a upwards direction, hence positive correlation, while in Figure 5.6b almost all the dots are all gathered around the low values of Y, which obviously reflects the correlation values.



(a) Throughput and throughput feature



(b) Throughput and throughput bug

Figure 5.6: Correlation graphs between the throughput variables.

Team eight also has a significant positive correlation between *throughput* and *churn*, but the sub variables don't has a significant relationship. The *churn* relationship were stated in Section 5.2. Team four's *churn* variables also differ. The reason for this were stated in section 5.2. Team one, two and four has a significante relationship between *throughput* and *lead time*. According to Table 5.14, *bugs* have a strong positive correlation with *throughput* as stated Section 5.3. *WIP* has a strong positive relationship with *throughput* which where stated in Section 5.1. *Team size* and *lead time* has a medium to strong relationship to *throughput*. The sub variables *throughput feature* (0.8) and *throughput bug* (0.7) has a strong and *throughput*.

	N	Mean	Median	Std.Dev	Max	Min
WIP	10	0.6	0.7	0.2	0.9	0.2
Throughput ft	10	0.8	0.9	0.3	1	0.1
Throughput bug	10	0.7	1	0.4	1	0
Bugs	10	0.6	0.6	0.2	1	0.3
Bugs finished, quarter	10	0.3	0.3	0.2	0.7	0.1
Avg days backlog, bugs	10	0.1	0.1	0.3	0.5	-0.4
Lead time	10	0.5	0.5	0.2	0.7	0.1
Churn	10	0.2	-0	0.4	0.7	-0.4
Churn ft	10	0.2	0.2	0.5	0.8	-0.4
Churn bug	10	-0	-0.1	0.3	0.6	-0.4
Team size	10	0.5	0.6	0.3	0.9	0.1

Table 5.14: Descriptive Statistic - Correlation - Throughput - Without time size is taken into account

The Table 5.15 shows that team one has a significant positive correlation for *throughput feature*, but not *throughput*. The reason *throughput* relationship were stated in Section 5.1.Team two has a significant positive correlation for *WIP*, both *throughputs* sub variable, *bugs*, *bugs finished, quarter* and *lead time*. Team three has a positive correlation for *WIP*, both *throughputs* sub variables and *bugs*. Team four has a significant relationship for *WIP*, both *throughputs* variable, *bugs finished, quarter*, *lead time*, *churn* and *churn feature*, but not *churn bug*. The *churn* relationship were described in Section 5.2. For team five has *throughput bug*, *bugs*, *bugs finished, quarter*, *avg, days in backlog, bugs*, *lead time*, *churn* and *churn feature* a positive correlation, but not *churn bug*. The *churn* relationship is explained in Section 5.5. The *throughput* relationship were described in 5.1.

Team six has a significant positive correlation for *throughput feature* and *lead time*. The *throughput* relationship were stated in Section 5.1. Team seven has a significant positive correlation value for *throughput feature*. The relationship for *throughput* were explained 5.1. Team eight has a significant positive correlation value for *throughput bugs*, *bugs*, *bugs finished, quarter*, *churn* and *churn bug*. The *throughput* relationship were stated in Section 5.3. The *churn* relationship is explained in Section 5.2. The conclusion from

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
WIP	0.37	0.59*	0.57*	0.86**	0.11	0.08	0.49	0.28	0.66*	-0.21
Throughput Feature	0.71**	0.72**	0.90**	1**	0.40	0.98**	0.90**	-0.02	0.82**	-0.11
Throughput bug	0.16	0.98**	0.98**	1**	0.99**	0.31	0.56	0.82**	0.95**	1**
Bugs	0.05	0.90**	0.81**	0.32	0.97**	-0.02	0.46	0.82**	0.57*	0.96**
Bugs finished, quarter	-0.11	0.69**	0.04	0.56*	0.57*	0.07	0.31	0.71**	0.55	0.67**
Avg days in backlog, bugs	0.02	-0.33	0.44	-0.12	0.80**	-0.35	-0.06	0.09	-0.27	0.27
Leadtime	0.06	0.88**	0.26	0.90**	0.80**	0.69**	0.33	0.24	0.32	0.90**
Churn	-0.14	0.37	0.23	0.88**	0.79**	-0.28	0.46	0.75**	-0.14	0.06
Churn feature	0.05	0.20	0.18	0.90**	-0.11	-0.16	0.12	0.18	-0.29	-0.35
Churn bug	-0.17	0.47	0.14	0.18	0.80**	-0.17	0.01	0.70**	-0.11	0.11

Table 5.15: Correlation - Throughput - When time size is taken into account.
* Correlation is significant at the 0.05 level (2-tailed).
** Correlation is significant at the 0.01 level (2-tailed).

the *churn* relationship was that *churn feature* represents most of the *churn* variable. The values from Table 5.15 and *churn* correlation Table 5.20 proves the first assumption from Section 5.2, which states that both the *churn* sub variables contributes to *churn*.

Team nine has a significant correlation value for *WIP*, both sub variables of *throughput* and *bugs*. Team ten has a significant positive correlation for *throughput bug*, *bugs*, *bugs finished, quarter* and *lead time*. The *throughput* relationship were explained in Section 5.1. Based on Table 5.16, *throughput* has either medium or a strong relationship between *WIP*, both *throughput* sub variables, *bugs*, *bugs finished, quarter* and *lead time*.

	N	Mean	Median	Std.Dev	Max	Min
WIP	10	0.4	0.4	0.3	0.9	-0.2
Throughput ft	10	0.6	0.8	0.4	1	-0.1
Throughput bug	10	0.8	1	0.3	1	0.2
Bugs	10	0.6	0.7	0.4	1	0
Bugs finished, quarter	10	0.4	0.6	0.3	0.7	-0.1
Avg days backlog, bugs	10	0	0	0.4	0.8	-0.3
Lead time	10	0.5	0.5	0.3	0.9	0.1
Churn	10	0.3	0.3	0.4	0.9	-0.3
Churn ft	10	0.1	0.1	0.4	0.9	-0.3
Churn bug	10	0.2	0.1	0.3	0.8	-0.2

Table 5.16: Descriptive Statistic - Correlation - Throughput - When time size is taken into account

5.5 Correlation - Churn

This section contains information about the correlation table between the variables and churn. Table 5.17 shows all teams have either one or both sub variables with significant positive correlation with *churn*. Team four, five, six, seven, eight, nine and ten don't has a positive correlation between both the *churn* sub variables. The relationship between the *churn* variables for team four, six, eight and nine were stated in Sections 5.1, 5.2 and 5.4.

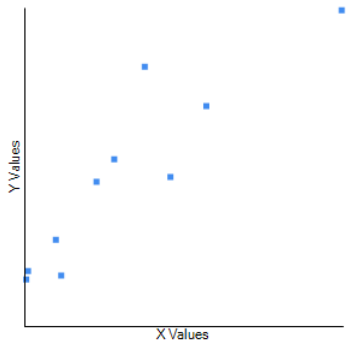
	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
WIP	0.47	-0.71**	-0.32	0.66*	0.03	-0.30	0.10	0.16	-0.09	0.16
Throughput	0.37	-0.43	-0.18	0.72**	-0.06	-0.40	0.43	0.59*	-0.14	0.02
Throughput Feature	0.36	0	-0.12	0.69**	-0.03	-0.37	0.45	0.63*	0.02	-0.17
Throughput bug	-0.52	-0.42	-0.22	0.69**	-0.03	-0.03	0.54	-0.17	-0.20	0.07
Bugs	0.62*	-0.27	0.10	0.27	-0.06	-0.12	0.11	-0.16	-0.48	0.04
Bugs finished, quarter	0.80**	-0.22	-0.11	0.15	-0.31	0.04	0.17	0.49	-0.05	0.31
Avg days in backlog, bugs	0.19	-0.12	-0.06	0	0.15	0.56*	0.60	-0.17	-0.01	-0.11
Lead time	0.70**	-0.42	-0.45	0.97**	0.18	-0.34	0.39	0.91**	-0.37	-0.04
Churn feature	0.57*	0.58*	0.90**	0.99**	0.22	0.98**	0.91**	0.84**	0.62*	0.14
Churn bug	0.80**	0.70**	0.85**	0.13	0.94**	-0.02	0.13	-0.07	0.39	0.94**
Team size	0.42	-0.16	-0.51	-0.24	-0.54*	-0.18	0.36	0.14	0.11	0.12

Table 5.17: Correlation - Churn
* Correlation is significant at the 0.05 level (2-tailed).
** Correlation is significant at the 0.01 level (2-tailed).
c. Cannot be computed because at least one of the variables is constant.

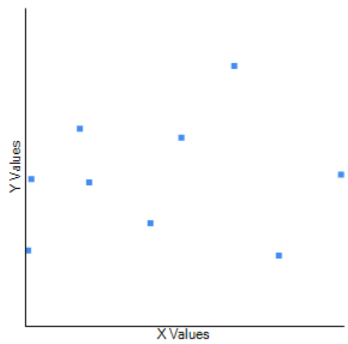
Churn bug for team five has a significant positive correlation of .94, while *churn feature* has the correlation value of 0.2. This proves that *churn bug* represents most of *churn*. The Tables A.23b, A.24a and A.24b shows the same result. *Churn* consist of 698 dates and has a total mean of 33.4, while *churn feature* and *churn bug* represents has 123 dates and 575 dates. The total mean of *churn feature* is 52.1 and for *churn bug* it's 29.4.

For team ten has *churn bug* the correlation value of .94 while *churn feature* has the value .14. This shows that churn bug represents most of *churn*. Based on the Tables A.48b, A.49a and A.49b, one can see that *churn* contains 361 dates and has a total mean value of 45.4. *Churn feature* represents 69 of these dates and has a total mean of 75.5, while *churn bug* represents 292 of *churn's* dates and has a total mean of 38.3. These data also shows a close indication of the relationship between *churn* and *churn bug*.

Churn feature for team seven has the correlation value of .91 while *churn bug* has the correlation value 0.13. The descriptive statistic Tables A.33b, A.34a and A.34b shows that *churn* contains 359 dates and has a total mean of 77.3. *Churn feature* represents 141 of these dates, and has a total mean of 121.2 . *Churn bug* represents the 218 remaining dates and has a total mean of 48.9. Based on these values, one could assume that both *churn feature* and *churn bug* contribute to *churn*, but the churn correlation table disapproves that. Which once again shows that the descriptive statistic tables can be used as indication of relationship, but can't prove them. The Figure 5.7a shows clearly the significant positive relationship between *churn* and *churn feature*.



(a) Churn and churn feature



(b) Churn and churn bug

Figure 5.7: Correlation graphs between the throughput variables.

The variables *bugs, bugs finished, quarter* and *lead time* including the previous stated *churn feature* and *churn bug* has a significant positive relationship with *churn* for team one. Both team two and four has a significant relationship between *churn* and *WIP*. Team four has a significant positive relationship between all the *throughputs* variable and *churn*. Team eight has a close relationship between *throughput* and *throughput feature*, but not *throughput bug*. The reason *throughput bug* differ from *throughput* were stated in Section 5.13. Team six has a positive correlation between *Avg days in backlog*, *bugs* and *churn*. Team four and eight has a significant positive correlation between *churn* and *lead time* and team five has a significant negative correlation between *churn* and *team size*. The Table 5.18 shows that there is no variables with neither a medium nor strong relationship to *churn* with out the *churn's* sub variables

	N	Mean	Median	Std.Dev	Max	Min
WIP	10	0	0.1	0.4	0.7	-0.7
Throughput	10	0.1	0	0.4	0.7	-0.4
Throughput ft	10	0.1	0	0.4	0.7	-0.4
Throughput bug	10	0	-0.1	0.4	0.7	-0.5
Bugs	10	0	0	0.3	0.6	-0.5
Bugs finished, quarter	10	0.1	0.1	0.3	0.8	-0.3
Avg days backlog, bugs	10	0.1	-0	0.3	0.6	-0.2
Lead time	10	0.2	0.1	0.6	1	-0.5
Churn ft	10	0.7	0.7	0.3	1	0.1
Churn bug	10	0.5	0.5	0.4	0.9	-0.1
Team size	10	0	0	0.3	0.4	-0.5

Table 5.18: Descriptive Statistic - Correlation - Churn

Team one has a significant positive correlation for *bugs finished, quarter*, *lead time* and *churn bug*, but not *churn*. The *churn* relationship were stated in Section 5.1. Team two has a significant positive correlation for *throughput feature*, *bugs finished, quarter* and both *churn* sub variables. The *throughput* relation were stated in Section 5.3. Team two also has a significant negative correlation for *avg days in backlog, bugs*. Team three has significant positive correlation values for *bugs*, *bugs finished, quarter* and both *churn* sub variables. Team four has a significant positive correlation for *WIP*, all *throughput* variables, *lead time* and *churn feature*. The *churn* relationship were stated in Section 5.2. Team five has a significant correlation value for *throughput*, *throughput bug*, *bugs*, *bugs finished, quarter*, *avg days in backlog, bugs*, *lead time* and *churn bug*. The *throughput* relationship were stated in Section 5.1.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
WIP	-0.27	0.16	-0.29	0.77**	-0.09	-0.35	-0.17	0.39	-0.34	-0.37
Throughput	-0.14	0.37	0.23	0.88**	0.79**	-0.28	0.46	0.75**	-0.14	0.06
Throughput Feature	-0.39	0.68**	0.11	0.87**	-0.02	-0.39	0.56	0.48	0.04	0.08
Throughput bug	-0.03	0.30	0.20	0.87**	0.83**	0.28	0.08	0.34	-0.10	0.07
Bugs	0.47	0.29	0.67**	0.22	0.73**	0.20	-0.05	0.34	-0.02	0.19
Bugs finished, quarter	0.95**	0.68**	0.76**	0.34	0.90**	0.33	-0.16	0.61*	-0.12	0.44
Avg days in backlog, bugs	-0.11	-0.55*	0.15	-0.12	0.88**	0.56*	0.39	-0.18	0.21	0.76**
Leadtime	0.72**	0.49	0.16	0.96**	0.94**	0.04	0.08	0.71**	-0.18	0.01
Churn feature	-0.40	0.79**	0.95**	0.99**	-0.10	0.98**	0.78**	0.66*	0.56*	0.33
Churn bug	1**	0.93**	0.95**	0.09	1**	-0.19	0.25	0.51	0.56*	0.99**

Table 5.19: Correlation - Churn - With size taken into account
* Correlation is significant at the 0.05 level (2-tailed).
** Correlation is significant at the 0.01 level (2-tailed).

Team six has a significant positive correlation of *avg days in backlog, bugs* and *churn feature*, The *churn* relationship were explained in Section 5.1. Team seven has a significant positive correlation to *churn feature*. The *churn* relationship were stated above in this section. Team eight has a significant positive correlation for *throughput*, *bugs finished, quarter*, *lead time* and *churn feature*. The *churn* relationship were described in Section 5.2, while the *throughput* relations were described in Section 5.3. Team nine has a significant positive correlation for both the *churn* sub variables. Team ten's has a significant positive correlation for *avg days in backlog, bugs* and *churn bug*. The *churn* relationship were explained above in this section. The Table 5.20 shows medium or strong overall linear relationship for *throughput*, *throughput bug*, *bugs*, *bugs finished, quarter*, *lead time* and both *churn* variables.

	N	Mean	Median	Std.Dev	Max	Min
WIP	10	-0.1	-0.2	0.4	0.8	-0.4
Throughput	10	0.3	0.3	0.4	0.9	-0.3
Throughput ft	10	0.2	0.1	0.4	0.9	-0.4
Throughput bug	10	0.3	0.2	0.3	0.9	-0.1
Bugs	10	0.3	0.3	0.3	0.7	-0.1
Bugs finished, quarter	10	0.5	0.5	0.4	0.9	-0.2
Avg days backlog, bugs	10	0.2	0.2	0.5	0.9	-0.6
Lead time	10	0.4	0.3	0.4	1	-0.2
Churn ft	10	0.6	0.7	0.5	1	-0.4
Churn bug	10	0.6	0.7	0.4	1	-0.2

Table 5.20: Descriptive Statistic - Correlation - Churn - With size taken into account

5.6 Correlation - Team size

The team size correlation table is shown in 5.21. Team one, three, five, six, eight and ten have a significant positive correlation between *team size* and *WIP*. Team one, five, six, eight and ten has a significant positive correlation between *team size* and *throughput*. For team one, six and eight's case, don't *throughput bug* has a significant positive correlation, the reason *throughput bug* differ for team one, six and eight were stated in Section 5.1 and 5.4 . For team five and ten differ *throughput feature*. The reason for that were stated in Section 5.1

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
WIP	0.68**	0.35	0.78**	0.06	0.57*	0.77**	0.62	0.65*	0.54	0.76**
Throughput	0.70**	0.05	0.52	0.16	0.69**	0.86**	0.62	0.75**	0.53	0.57*
Throughput Feature	0.74**	-0.22	0.53	0.20	0.48	0.89**	0.47	0.74**	0.48	0.18
Throughput bug	-0.10	0.06	0.51	0.20	0.67**	0	0.71*	0.40	0.48	0.64*
Bugs	0.80**	0.26	0.27	0.17	0.71**	0.41	0.41	0.42	0.41	0.16
Bugs finished, quarter	0.42	-0.53	0.25	-0.19	0.28	0.30	0.71*	0.05	0.38	0.34
Avg days in backlog, bugs	0.48	0.84**	0.04	0.44	0.03	-0.03	0.49	0.03	0.07	-0.03
Lead time	0.61*	0.38	0.44	-0.30	0.36	-0.11	0.59	0.22	0.38	0.53
Churn	0.42	-0.16	-0.51	-0.24	-0.54*	-0.18	0.33	0.14	0.11	0.12
Churn feature	0.79**	0.41	-0.42	-0.17	0.32	-0.23	0.36	0.07	0.01	0.36
Churn bug	0.26	-0.44	-0.61*	0.27	-0.55*	0.74**	-0.32	-0.14	-0.16	-0.10

Table 5.21: Correlation - Team size
* Correlation is significant at the 0.05 level (2-tailed).
** Correlation is significant at the 0.01 level (2-tailed).

Team one and five has a significant positive correlation between *bugs* and *team size*. Team two has a significant positive correlation between *avg days in backlog, bugs* and *team size*. Team one also has a positive correlation between *team size* and *lead time*, as well as *churn feature*. The *churn* relationship were stated in Section 5.1.

	N	Mean	Median	Std.Dev	Max	Min
WIP	10	0.6	0.6	0.2	0.8	0.1
Throughput	10	0.5	0.6	0.3	0.9	0.1
Throughput ft	10	0.4	0.5	0.3	0.9	-0.2
Throughput bug	10	0.4	0.4	0.3	0.7	-0.1
Bugs	10	0.4	0.4	0.2	0.8	0.2
Bugs finished, quarter	10	0.2	0.3	0.3	0.7	-0.5
Avg days backlog, bugs	10	0.2	0.1	0.3	0.8	-0
Lead time	10	0.3	0.4	0.3	0.6	-0.3
Churn	10	-0	-0	0.3	0.4	-0.5
Churn ft	10	0.1	0.2	0.4	0.8	-0.4
Churn bug	10	-0.1	-0.1	0.4	0.7	-0.6

Table 5.22: Descriptive Statistic - Correlation - Team size

Team three and six has a significant values between *team size* and *churn bug*. For team three's case, the *churn* relationship were stated in Section 5.2 and for team six the reason were stated in Section 5.1. *Churn feature* in case of team five has a significant value, the reason *churn feature* and *churn* don't has a close relationship were stated in Section 5.1. *Churn* and *churn bug* has negative significant correlation to *team size*, while *churn feature* has a medium positive relationship for team five. The *churn* relationship were stated in Section 5.5. The Table 5.22 shows a strong correlation between *WIP*, *throughput* and *team size*, as well as a medium correlation for *bugs*.

Chapter 6

Discussion

In this chapter a discussion around the research questions, the different results and what literature says are presented.

6.1 WIP and lead time

As stated in Section 2.5, lead time could be important to measure productivity. Each development process would like to get their lead time as low as possible. In Section 2.4.2, there is stated that WIP-limits are important to reduce lead times. The same is showed by the results here. The result from when team size was **not** taken into account shows a linear relationship of 0.5, as shown in Table 5.2, which means when WIP increases, so does lead time. Team one, four, five, seven and ten has significant positive linear relationship and both team two and three has a medium correlation relationship between *WIP* and *lead time* as shown in Table 5.2.

The result from when team size was taken into account is a linear value 0.2, whichs shows a weaker linear relationship than when team size was **not** taken into account. Three of the ten teams have a significant positive correlation. While the rest of teams correlation values varies from -0.18 to 0.32, as shown by Tables 5.3 and 5.4.

The standard deviation shows a variation of 0.2, and a median of 0.5 when team is **not** taken into account. The standard deviation when team size is taken into account is 0.4 and the median is 0 as shown in Tables 5.2 and 5.4. The standard deviation shows a high proliferation of values, which shows evidence of spread relationship between *WIP* and *lead time* across the teams, when team size is taken into account. The standard deviation value of 0.2 shows that each of the ten *lead time* values are closer when team size is **not** taken into account. The median of the two results is 0.5 and 0. These values shows how the overall values when team size is **not** taken into account is greater. But, the max value for when team size is taken into account is 0.9 while 0.7 otherwise. This shows, isolated for that team what an impact increased WIP has.

Despite *WIP* and *lead time* don't has a close relationship, *Lead time* has a close positive relationship to four of the five individual variables, as shown in table 5.8. *Throughput*, *Bugs* and *Bugs finished, quarter* has the total correlation value 0.5, while *churn* has the value of 0.4, when time size is taken into account. When team size is **not** taken into account, *WIP* and *throughput* has the overall correlation value of 0.5 and *throughput feature* has the value 0.4, while the rest variables has a value of 0.3 or less. Especially, when team size is taken into account, the importance of decreasing *lead time* is shown.

6.2 WIP and bugs

To minimize bugs should be a goal independent of software process and methods. If WIP-limit could reduce bugs without compromising, one could conclude that WIP-limit matter in software development. In the article by Shinkle (Shinkle, 2009), when WIP was too high, the number of bugs increased. The same can be said about the values from this study. When team size is **not** taken into account, the correlation value between *WIP* and *bugs* is 0.4 and four of the teams have a significant positive correlation value between *bugs* and *WIP*. All ten teams has a positive correlation value, as shown in table 5.1. The two variable *bug finished, quarter* and *avg days backlog bugs* is used as a side measure to how long bugs are in backlog and the amount of bugs fixed vs. the amount of bugs recorded in system per quarter. The correlation value for *bug finished, quarter* is 0.3 and *avg days backlog bugs* is 0.1. The value *bug finished, quarter* shows a low-medium linear relationship who indicates that if WIP increases, SI is able to finished more bugs. One can argue that when one produces more, more bugs are produced, but the value of *bug finished, quarter* is measured in percentages. The 0.1 correlation value of *avg days backlog bugs* shows that *WIP* has no influence on the number of days bugs are in backlog. But both *bugs* and *bug finished, quarter* shows a medium linear relationship, which shows moderate linear relationship between *WIP* and these variables.

When team size is taken into account, the correlation value is 0.2, as shown in Table 5.4. Eight out of ten teams have a positive correlation relationship, but only one team has a significant value as shown in Table 5.3. The values *bug finished, quarter* and *avg days backlog bugs* has the correlation values 0.2 and -0.1 This shows that there is small linear relationship between *WIP* and any of the *bugs* variables, when team size is taken into account.

The standard deviation when team size is **not** taken into account is 0.2 and the median is 0.5. The median when team size is taken into account is 0.2, while the standard deviation is increased by 0.1. The median numbers shows significant impact of team size between the relationship of *WIP* and *bug*. The std.dev value shows a minor variation between the two results variables. Shown in 5.2 and 5.4

6.3 WIP and throughput

In this thesis, throughput is a measure to see how productive teams are. In Section 2.4.2, there was stated by various people, when lowering WIP-limit, throughput increases. The data from this study shows otherwise. Both Tables 5.14 and 5.16 proves that if *WIP* increase, so does *throughput* with the correlation value of 0.6 and 0.4. In Table 5.14, one can argue that the throughput correlation value is biased, based on the strong team size correlation. But the Table 5.16, where team size is taken into account shows a overall correlation value of 0.4 between *throughput* and *WIP*. This shows that still when team size is taken into account, *WIP* and *throughput* still has a positive correlation.

When team size is **not** taken into account, All of teams have a positive correlation between *WIP* and *throughput*, and seven of the teams have a significant correlation value, as shown in table 5.13. When team size is taken into account, the Table 5.4 shows when *WIP* increases so does *throughput* for nine out of ten teams. Team ten is the only team with a negative correlation. Team, five, six, eight and ten has low small correlation between *throughput* and *WIP*, Team one and seven has a low-medium relationship, team two, three, four and nine has a significant linear relationship. The *throughput's* sub variables both have a overall correlation of 0.5 when team size is **not** taken into account. When team size is taken into account *throughput feature* has a value of 0.3 and *throughput bug* has the value 0.4. This shows that both *throughput's* sub variables have a positive linear relationship with *WIP*.

The correlation between *throughput* and *WIP* when team size is **not** taken into account is a strong linear relationship, the correlation value when team size is taken into account represents a medium linear relationship. The strong relationship means when *WIP* increases, *throughput* increases. The medium relationship means that when *WIP* increases, *throughput* increase as well to an extent. The throughput descriptive statistic Tables 5.14 and 5.16 shows *Bugs* has the overall correlation value of 0.6 and *lead time* has the overall value of 0.5 with *throughput*. One could see that if *throughput* increases, so does both *lead time* and *bugs*. Both these values can't be used to see the impact of WIP-limits. These variables are isolated and can only be used to see what impact *throughput* has on a software process.

The median when team size is **not** taking into account is 0.7 and 0.4 otherwise. The std.dev is 0.2 when team size is **not** taking into account and 0.3 otherwise. A median of

0.7 is high, while 0.4 also relative high and when the std.dev is at the lowest shown in Tables 5.2 and 5.4, the relationship between *WIP* and *throughput* is significant.

6.4 WIP and Churn

Churn is used as surrogates for effort in this thesis. It turns out based on the results, independently of team size, there is no relationship between *WIP* and *churn* or the sub variables of churn. Both the results shows respectively a overall correlation of -0.1 in Table 5.4 and 0 in Table 5.2. When team size is **not** taken into account *Churn* has no linear relationship between any of the other variables, if one disregards *churn's* sub variable shown in Table 5.18. When team size is taken into account, *bugs finished, quarter* has a strong relationship with a value of 0.6, while *lead time* has a value of 0.4 which is a medium-strong relationship, shown in Table 5.20.

The std.dev for both the results are 0.4, while the median is 0.1 when team size is **not** taken into account and -0.1 for the other case. The std.dev empower the assumption of a relationship between *churn* and *WIP*, since the values are spread and their is no pattern.

6.5 Team size

As the two different results sections in Chapter 5 and the sections above shows, it's important to take team size into account when looking for relationship between variables such as *WIP*, *throughput*, *lead time* and *bugs*. If team size were left out as a variable in this case study, *WIP* would had a linear relationship of 0.6 with *throughput*, 0,4 with *bugs* and 0.4 with *lead time*. The three correlation values increased by 0.2 compared to when team size was taken into account. These variables would have showed stronger results towards the fact that WIP-limit matter in software development.

For *throughput* when time size was **not** taken into account, *bugs* has a value of 0.6 and *lead time* has a value of 0.5, which indicates a medium - strong relationship. The median in *bugs* case is 0.8 and for *lead time* it's 0.5 as shown in Tables 5.14 and 5.16. This values shows that when throughout increases, do does *bugs* and *lead time*. When team size is taken into account, the correlation values between *throughput* and both *bugs* and *lead time* is the same , the std.dev and median for *bugs* is different. If *WIP* omitted, the correlation relationship between the other variables and *throughput* don't differ to much. This show evidence that team size don't has a big impact on the correlation relationship between variables and *throughput* and team size.

The bugs correlation tables 5.10 and 5.12 shows a difference in the correlation value between *bugs* and *WIP*. The correlation value is increased by 0.2 when team size is **not** taken into account. Both the *throughput* correlation values are 0.6. The correlation value for *lead time* is also increased by 0.2 when team size is **not** taken into account, while churn increased by 0.3. These values show an impact of team size on the correlation relationship for 3 of 4 variables.

For *lead time* increases the churn value by 0.2 when time size is **not** taken into account, otherwise are the correlation values for *lead time* stated preceding paragraphs. The values show the importance of taking team size into account when a case study like this is conducted.

6.6 WIP

The article "Studying Lean-Kanban Approach Using Software Process Simulation" (D. Anderson et al., 2011) simulated a lean-kanban process as stated in 2.4.2. The simulation showed when WIP was limit, the software process used 100 days and 120 otherwise. The article "Simulation of software maintenance process, with and without a work-in-process limit" (Concas et al., 2013) showed that WIP-limit can lead to improvement in throughput. The article also showed that when taking a process without WIP-limit, and simulate it using WIP-limit it outperformed the original process. The result from this thesis shows that *throughput* increases when *WIP* does. *Throughput* is also the only variable with a correlation relationship of interest. The other correlation relationship is equal or lower than 0.2.

Chapter 7

Conclusion

In this thesis the main goal was to investigate the research question "Does WIP limit in software development matter?". If so, further goals were to see "How to find the optimal WIP limit" and "Which parameters should be considered in order to optimize WIP". To answer the research questions a data set from a company called Software Innovation were interpreted. The data set is based on metadata about tasks from 2010 to 2013.

If WIP limit matters in software development some of its benefits should be to reduce bugs, increase throughput, decrease lead time etc. Based on the results from the case study, *WIP* has an impact on *throughput*. The results shows a positive linear relationship between *WIP* and *throughput*. The previous research has stated to lower WIP to decrease throughput. This research shows otherwise. This study shows *throughput* decreases when *WIP* does. It is also not evidence after this case study that lowering WIP increases *throughput* or decreases *lead time* or *bugs*. Since WIP - limits don't matter in software development based on this study, there is no need to find an optimal WIP-limit or know which parameters to take into account in order to optimize WIP.

However, the results from this paper show some other findings. The importance of taking team size into account when measuring a similar research, the churn variable has no or little impact on the software process.

7.1 Future work

The conclusion from this thesis is made on one case study. I would recommend doing the same calculation as in this thesis with another data set and comparing the outcome.

I would also suggest an different approach as looking more deeply into the relationship between *WIP* and *team size*. I suggest one measure the number of employees working on each task instead of take the number of employees per quarter and divide on the mean of each variables value.

I will also suggest looking at the values up against release dates. The number of WIP's, throughput, bugs et cetera usually decrease around release date, which could cause bias in the data. There is also done another research on the data from SI, I would suggest comparing the results.

Appendices

Appendix A

Descriptive statistics (DS) for the ten teams

A.1 Team 1 - Descriptive Statistics

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	25	3.6	4	0.6	5	3
2010-4	92	0.7	1	0.7	3	0
2011-1	90	3.4	1	6.9	30	0
2011-2	91	13.2	4	14.5	51	2
2011-3	92	1.8	2	0.6	3	1
2011-4	92	14.3	4	22.7	97	1
2012-1	91	22.2	21	14.5	67	4
2012-2	91	30.3	23	29	107	9
2012-3	92	36	38.5	13.6	65	18
2012-4	92	34.7	28.5	16.9	99	25
2013-1	90	32.8	25	13.7	85	25
2013-2	91	67.1	54	44.3	178	3
2013-3	92	7.4	3	8.8	31	1
2013-4	76	5	1	8.1	35	1
Total	1197	20.5	12	26.2	178	0

(a) DS - WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	3	3	1	3.5	7	1
2010-4	3	1	1	0	1	1
2011-1	7	10.4	11	8.1	25	1
2011-2	32	9.4	10	6.7	26	1
2011-3	2	1	1	0	1	1
2011-4	25	14.9	10	14.6	49	1
2012-1	49	8.6	5	8.1	33	1
2012-2	45	11.2	3	16	56	1
2012-3	34	5.5	3	6.3	23	1
2012-4	17	14.2	14	13.7	44	1
2013-1	13	19.5	17	17	58	1
2013-2	26	21.6	18	16.9	60	1
2013-3	17	9	7	7.7	27	1
2013-4	17	6.3	3	7.5	24	1
Total	290	11	6	12.5	60	1

(b) DS - Throughput

Table A.1: Caption of Descriptive Statistic for WIP and Throughput a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1	7	7	-	7	7
2011-1	7	10.4	11	8.1	25	1
2011-2	24	10.1	10	5.6	24	1
2011-3	1	1	1	-	1	1
2011-4	11	16.6	13	11	35	4
2012-1	16	15.2	15	9.3	33	1
2012-2	26	16.1	5	17.7	56	1
2012-3	23	6	4	6.6	23	1
2012-4	14	15.1	14.5	14.3	44	1
2013-1	10	23.3	20	17.4	58	3
2013-2	21	23.6	24	18.1	60	1
2013-3	16	9.5	7.5	7.7	27	1
2013-4	12	8.3	7.5	8.1	24	1
Total	182	13.7	10	13.2	60	1

(a) DS - Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	2	1	1	0	1	1
2010-4	3	1	1	0	1	1
2011-2	8	7.5	2	9.3	26	1
2011-3	1	1	1	-	1	1
2011-4	14	13.6	5	17.3	49	1
2012-1	33	5.3	5	5	21	1
2012-2	19	4.5	1	10.5	47	1
2012-3	11	4.4	3	5.8	21	1
2012-4	3	10	3	12.1	24	3
2013-1	3	7	3	8.7	17	1
2013-2	5	13.4	13	7.1	21	3
2013-3	1	1	1	-	1	1
2013-4	5	1.4	1	0.9	3	1
Total	108	6.4	3	9.6	49	1

(b) DS - Throughput bug

Table A.2: Caption of Descriptive Statistic for Throughput feature and Throughput bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1	13	13	-	13	13
2010-4	2	8.5	8.5	9.2	15	2
2011-2	28	13.1	7.5	16.5	78	1
2011-3	1	5	5	-	5	5
2011-4	28	15.7	14.5	11.2	45	1
2012-1	66	12.5	9	11.5	49	1
2012-2	47	18.7	12	19	107	1
2012-3	32	9.9	7	11.3	49	1
2012-4	26	18.1	5.5	58.3	303	1
2013-1	19	18.7	6	27	103	2
2013-2	48	27.9	8.5	75.8	508	1
2013-3	25	15.6	5	25.1	110	1
2013-4	16	14.5	4.5	24	76	1
Total	339	16.7	8	36.2	508	1

(a) DS - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1	13	13	-	13	13
2010-4	2	30	30	41	59	1
2011-2	28	20.1	13	18	74	1
2011-3	1	2	2	-	2	2
2011-4	28	22.9	17.5	19.9	86	0
2012-1	66	18.6	12.5	19.8	97	0
2012-2	47	20.9	17	20	103	0
2012-3	32	13.9	5	20.6	75	0
2012-4	26	24	9	58.8	302	0
2013-1	19	17.8	9	25.3	99	0
2013-2	48	27.9	9.5	73.5	495	0
2013-3	25	14.7	5	23	99	0
2013-4	16	15.1	4.5	23.2	72	0
Total	339	20.2	10	36.8	495	0

(b) DS - Churn

Table A.3: Caption of Descriptive Statistic for Lead time and Churn a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-2	8	23.2	22	21.2	49	1
2011-4	8	24.5	14.5	28	86	4
2012-1	20	17.9	17	12.4	48	0
2012-2	21	23.8	16	25.6	103	0
2012-3	20	11.2	3	19.4	75	0
2012-4	16	30.9	9.5	74.1	302	0
2013-1	11	24.7	9	31.8	99	0
2013-2	23	42.6	7	104.9	495	0
2013-3	17	16.6	4	26.7	99	0
2013-4	6	30.3	16	31.3	72	0
Total	150	24.5	10	51.6	495	0

(a) DS - Churn bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1	13	13	-	13	13
2010-4	2	30	30	41	59	1
2011-2	20	18.9	13	17	74	2
2011-3	1	2	2	-	2	2
2011-4	20	22.2	19	16.5	65	0
2012-1	46	18.9	10.5	22.4	97	0
2012-2	26	18.6	18	14.2	43	0
2012-3	12	18.4	6.5	22.7	63	1
2012-4	10	12.9	8.5	15.2	52	0
2013-1	8	8.2	8.5	4.4	16	1
2013-2	25	14.4	13	9.6	34	2
2013-3	8	10.8	5.5	12.6	38	0
2013-4	10	5.9	3	10.1	33	0
Total	189	16.8	11	17.2	97	0

(b) DS - Churn bug

Table A.4: Caption of Descriptive Statistic for Churn feature and Churn bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1	1	1	-	1	1
2010-4	4	1	1	0	1	1
2011-2	32	4.2	3.5	3.6	14	1
2011-3	5	1	1	0	1	1
2011-4	36	4.9	2.5	5.4	22	1
2012-1	43	3.5	3	2.3	10	1
2012-2	33	5.4	3	5.5	21	1
2012-3	16	2.4	1.5	1.8	6	1
2012-4	13	2.8	2	1.8	6	1
2013-1	8	3.5	3	2.5	7	1
2013-2	27	5.8	4	4.8	17	1
2013-3	11	1.3	1	0.5	2	1
2013-4	10	1.7	1	1.9	7	1
Total	240	4	2	4	22	1

(a) DS - Bugs

Quarter	Finished	Not finished	Total	Finished	Not finished
2010-3	1	0	1	100	0
2010-4	4	0	4	100	0
2011-2	130	3	133	97.7	2.3
2011-3	1	4	5	20	80
2011-4	156	22	178	87.6	12.3
2012-1	146	4	150	97.3	2.7
2012-2	176	3	179	98.3	1.7
2012-3	37	2	39	94.9	5.1
2012-4	33	3	36	91.7	8.3
2013-1	24	4	28	85.7	14.3
2013-2	157	0	157	100	0
2013-3	13	1	14	92.9	7.1
2013-4	17	0	17	100	0
Mean	63.9	3.3	67.3	83.3	16.7

(b) DS - Bugs per quarter

Table A.5: Caption of Descriptive Statistic for Bugs and Bugs finished within quarter a, b

A.2 Team 2 - Descriptive Statistics

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	25	14.4	15	6.2	23	6
2010-4	92	21.4	20	7.2	41	9
2011-1	90	27.2	27.5	4.9	38	17
2011-2	91	29.7	27	14.4	62	12
2011-3	92	32.6	30	9.2	56	18
2011-4	92	30.1	30	10.1	46	13
2012-1	91	20	19	4.6	31	8
2012-2	91	25.3	26	10.3	51	6
2012-3	92	24.2	22.5	7.9	45	11
2012-4	92	21.6	23	10.5	47	3
2013-1	90	19.7	20	5.8	35	8
2013-2	91	28	27	4.4	37	15
2013-3	92	18.7	19	4.5	28	9
2013-4	87	13.3	14	6.6	29	2
Total	1208	23.8	23	9.8	62	2

(a) DS - WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	16	4.2	3	4	16	1
2010-4	54	4.1	3	3.9	21	1
2011-1	57	4.6	4	3.6	17	1
2011-2	41	6.9	5	5.7	25	1
2011-3	52	3.8	2	3.6	15	1
2011-4	52	3.7	3	2.7	11	1
2012-1	55	4.3	3	3.4	12	1
2012-2	51	4.1	3	3.5	21	1
2012-3	57	5.8	5	4.3	18	1
2012-4	52	5.2	4.5	3.7	15	1
2013-1	51	4.6	3	3.6	16	1
2013-2	50	3.3	3	2.4	9	1
2013-3	55	3.9	4	2.9	16	1
2013-4	47	3.2	3	2.7	13	1
Total	690	4.4	3	3.7	25	1

(b) DS - Throughput

Table A.6: Caption of Descriptive Statistic for WIP and Throughput a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	5	4.6	3	3.2	10	2
2010-4	22	3.1	2.5	2.8	11	1
2011-1	15	5.1	4	4.4	17	1
2011-2	5	3.2	3	1.3	5	2
2011-3	25	3.7	2	3.8	14	1
2011-4	10	3.4	3	2.9	11	1
2012-1	9	2.1	1	2	7	1
2012-2	16	3.5	3.5	2.3	8	1
2012-3	12	4.2	3.5	1.9	8	1
2012-4	25	4.6	3	3.9	13	1
2013-1	11	3.6	3	3.3	11	1
2013-2	27	2.7	2	2	9	1
2013-3	29	3.9	3	2.7	11	1
2013-4	19	3.4	2	3	10	1
Total	230	3.7	3	3	17	1

(a) DS - Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	11	4.1	3	4.4	16	1
2010-4	32	4.8	4	4.4	21	1
2011-1	42	4.4	4	3.3	13	1
2011-2	36	7.4	5.5	5.9	25	1
2011-3	27	3.9	3	3.5	15	1
2011-4	42	3.7	3	2.7	11	1
2012-1	46	4.7	3.5	3.5	12	1
2012-2	35	4.3	3	4	21	1
2012-3	45	6.3	5	4.6	18	1
2012-4	27	5.8	5	3.3	15	1
2013-1	40	4.8	4	3.7	16	1
2013-2	23	3.9	3	2.8	9	1
2013-3	26	3.8	4	3.2	16	1
2013-4	28	3.1	3	2.5	13	1
Total	460	4.8	4	3.9	25	1

(b) DS - Throughput bug

Table A.7: Caption of Descriptive Statistic for Throughput feature and Throughput bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	19	15	9	14.2	55	1
2010-4	53	13.7	9	13.2	55	1
2011-1	67	14.4	11	11.3	67	2
2011-2	41	19.4	13	17.5	79	2
2011-3	55	15.6	11	14	55	1
2011-4	49	14.5	10	13.9	61	1
2012-1	63	11.4	8	10.3	41	1
2012-2	58	11.2	10	8.8	38	1
2012-3	83	15.4	13	12	66	1
2012-4	70	12.5	9	12.1	68	1
2013-1	70	12.6	9.5	10.7	44	1
2013-2	40	11.8	7.5	11.1	44	1
2013-3	59	11.3	6	12.1	49	1
2013-4	51	12	10	12.3	71	1
Total	778	13.5	10	12.3	79	1

(a) DS - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	19	69.6	14	106.2	352	3
2010-4	53	78.6	26	120.7	493	1
2011-1	67	35.1	20	57.3	407	1
2011-2	41	40.5	21	64.5	383	2
2011-3	55	57.8	30	86.3	379	1
2011-4	49	46.9	28	55.7	294	2
2012-1	63	58.4	23	81.3	377	0
2012-2	58	58.1	19	99.3	408	0
2012-3	83	43.4	20	68.6	433	0
2012-4	70	69.8	20	112.9	513	0
2013-1	70	47.4	14.5	94.1	467	0
2013-2	40	43.1	11.5	76.3	310	0
2013-3	59	77.7	26	114.5	459	0
2013-4	51	91.3	32	138.8	474	0
Total	778	57.6	22	94.2	513	0

(b) DS - Churn

Table A.8: Caption of Descriptive Statistic for Lead time and Churn a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	6	148	93.5	152.8	352	12
2010-4	18	178.4	118	164.1	493	10
2011-1	19	48.9	37	53.2	214	1
2011-2	4	134.5	70.5	168.8	383	14
2011-3	16	128.1	91	128.1	379	1
2011-4	11	106.3	120	87.6	294	12
2012-1	16	112.8	54.5	125.1	377	0
2012-2	21	90.1	34	122.4	408	0
2012-3	29	52.7	19	67.2	226	0
2012-4	32	103.7	27	151.4	513	0
2013-1	23	93.7	32	150.3	467	0
2013-2	14	82.1	25	117.4	310	0
2013-3	28	97	20	142	459	0
2013-4	20	125.1	37.5	163.5	463	0
Total	257	100.6	38	131.3	513	0

(a) DS - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	13	33.5	12	52	193	3
2010-4	35	27.2	19	28.8	153	1
2011-1	48	29.6	18	58.5	407	1
2011-2	37	30.3	19	34	152	2
2011-3	39	29	20	34.3	196	1
2011-4	38	29.7	19.5	24.6	95	2
2012-1	47	39.9	20	49.3	237	1
2012-2	37	39.9	15	79.6	380	0
2012-3	54	38.4	22.5	69.5	433	0
2012-4	38	41.2	19.5	52.3	226	0
2013-1	47	24.7	12	29.5	127	0
2013-2	26	22.1	11	24.5	91	0
2013-3	31	60.2	26	80.9	296	0
2013-4	31	69.5	29	118.1	474	4
Total	521	36.3	19	58.4	474	0

(b) DS - Churn feature

Table A.9: Caption of Descriptive Statistic for Churn feature and Churn bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	20	2.6	2	3.5	17	1
2010-4	40	2.5	2	1.7	9	1
2011-1	47	2.4	2	1.8	8	1
2011-2	40	3.8	3	2.5	13	1
2011-3	43	2.6	2	2.4	13	1
2011-4	47	2.5	2	1.6	8	1
2012-1	35	3.3	3	3	16	1
2012-2	34	2.3	2	1.5	7	1
2012-3	43	3.6	2	2.6	10	1
2012-4	33	3.9	3	3.1	14	1
2013-1	38	2.2	2	1.2	6	1
2013-2	32	1.9	1.5	1.2	5	1
2013-3	35	1.8	1	1.1	5	1
2013-4	37	1.9	1	1.3	7	1
Total	536	2.7	2	2.2	17	1

(a) DS - Bugs

Quarter	Finished	Not finished	Total	Finished	Not finished
2010-3	30	23	53	56.6	43.4
2010-4	65	34	99	65.7	34.3
2011-1	101	13	114	88.6	11.4
2011-2	142	8	150	94.7	5.3
2011-3	87	24	111	78.4	21.6
2011-4	90	29	119	75.6	24.4
2012-1	94	23	117	80.3	19.7
2012-2	70	9	79	88.6	11.4
2012-3	146	7	153	95.4	4.6
2012-4	101	27	128	78.9	21.1
2013-1	78	5	83	94.0	6.0
2013-2	58	3	61	95.1	4.9
2013-3	62	2	64	96.9	3.1
2013-4	69	0	69	100	0
Mean	66.4	12.3	78.8	74.4	25.6

(b) DS - Bugs per quarter

Table A.10: Caption of Descriptive Statistic for Bugs and Bugs finished within quarter a, b

A.3 Team 3 - Descriptive Statistics

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	24	9.3	10	6.4	23	1
2010-4	92	13.9	13	3.9	25	5
2011-1	90	15.3	15.5	3.9	23	7
2011-2	91	23.5	24	4.2	37	13
2011-3	92	20.7	20	5.8	34	9
2011-4	92	23.3	23	6.9	36	9
2012-1	91	24.9	24	6.6	42	13
2012-2	91	23.9	23	3.4	34	19
2012-3	92	28	29	5.1	38	21
2012-4	92	29.6	28.5	5.3	44	22
2013-1	90	16.2	15	5	27	9
2013-2	91	7	6	3.2	13	2
2013-3	92	7	7	2	14	3
2013-4	67	5.6	5	2.6	13	2
Total	1187	18.5	19	9.1	44	1

(a) DS - WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	16	3.3	3	2.9	12	1
2010-4	54	3.5	3	3	15	1
2011-1	42	2.2	2	1.4	7	1
2011-2	45	4.3	3	3.8	20	1
2011-3	51	4	3	3.4	15	1
2011-4	50	4.7	3	5.2	27	1
2012-1	46	6.5	5	5.7	20	1
2012-2	40	2.9	2	3.2	15	1
2012-3	36	3.4	2.5	3.1	13	1
2012-4	51	5	4	4.4	22	1
2013-1	42	3.2	2	2.9	10	1
2013-2	22	1.6	1	1	5	1
2013-3	29	2	1	2	11	1
2013-4	18	1.6	1	1.1	5	1
Total	542	3.7	3	3.8	27	1

(b) DS - Throughput

Table A.11: Caption of Descriptive Statistic for WIP and Throughput a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	3	2.3	3	1.2	3	1
2010-4	19	2.9	2	2.4	9	1
2011-1	29	2.3	2	1.5	7	1
2011-2	24	4	3.5	2.4	8	1
2011-3	23	4.3	3	3.5	13	1
2011-4	19	4	3	4.5	21	1
2012-1	10	5.4	2	5.6	16	1
2012-2	18	2.2	1.5	1.5	6	1
2012-3	12	3.9	1.5	4.3	13	1
2012-4	17	4.8	3	4.5	17	1
2013-1	8	2.1	1.5	1.7	6	1
2013-2	3	1.7	2	0.6	2	1
2013-3	8	1.8	1.5	0.9	3	1
2013-4	7	1.3	1	0.5	2	1
Total	200	3.3	2	3.2	21	1

(a) DS - Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	13	3.5	3	3.2	12	1
2010-4	35	3.9	3	3.3	15	1
2011-1	13	2.1	2	1	3	1
2011-2	21	4.8	3	4.9	20	1
2011-3	28	3.8	3	3.5	15	1
2011-4	31	5.2	3	5.5	27	1
2012-1	36	6.8	5	5.8	20	1
2012-2	22	3.5	2	4.1	15	1
2012-3	24	3.2	3	2.4	9	1
2012-4	34	5.2	4	4.5	22	1
2013-1	34	3.4	2	3	10	1
2013-2	19	1.6	1	1.1	5	1
2013-3	21	2.1	1	2.2	11	1
2013-4	11	1.7	1	1.3	5	1
Total	342	4	3	4.1	27	1

(b) DS - Throughput bug

Table A.12: Caption of Descriptive Statistic for Throughput feature and Throughput bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	21	10.6	11	6.4	24	1
2010-4	59	11.6	9	8.9	34	1
2011-1	27	8.7	8	5.9	18	1
2011-2	51	13	11	9.2	34	1
2011-3	48	14.4	10.5	11.3	49	2
2011-4	62	17.8	15	11.7	46	1
2012-1	59	22.6	18	16.6	76	1
2012-2	39	19.5	16	15.2	54	1
2012-3	40	17.1	12.5	17.3	72	1
2012-4	66	12.5	8	12.5	58	1
2013-1	44	12.1	6.5	12.7	60	1
2013-2	20	11	10	9.7	34	1
2013-3	28	7.9	4.5	8.1	29	1
2013-4	12	18	14	18.7	75	1
Total	576	14.6	11	12.9	76	1

(a) DS - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	21	120.1	58	138.4	383	3
2010-4	59	60.2	28	67.2	295	2
2011-1	27	73.4	65	62.6	320	2
2011-2	51	79.7	36	104.8	423	1
2011-3	48	67.5	31.5	91	407	0
2011-4	62	37.4	18	66.3	343	0
2012-1	59	47.3	27	55.3	286	0
2012-2	39	38.2	20	66.4	365	0
2012-3	40	66.7	23.5	99.3	406	0
2012-4	66	79.7	28.5	114.5	494	0
2013-1	44	36.7	22	42.6	174	0
2013-2	20	59.5	40	60.9	216	0
2013-3	28	70.6	48.5	90.1	403	0
2013-4	12	79.2	49.5	80.5	237	0
Total	576	61.8	29	85.4	494	0

(b) DS - Churn

Table A.13: Caption of Descriptive Statistic for Lead time and Churn a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	7	197.4	141	136.1	383	16
2010-4	24	78.7	55.5	77	295	2
2011-1	15	89.9	75	73.8	320	14
2011-2	24	128.6	87.5	129.2	423	4
2011-3	24	88.4	40.5	108.8	407	0
2011-4	23	69.9	35	94.7	343	0
2012-1	17	80	59	76.2	286	0
2012-2	9	74.2	33	117	365	0
2012-3	15	111	66	126.1	406	0
2012-4	25	122.8	57	142.6	494	0
2013-1	11	53	65	52.7	174	0
2013-2	2	76.5	76.5	65.8	123	30
2013-3	5	146.6	120	149	403	24
2013-4	4	151.5	167.5	92	237	34
Total	205	98.9	62	109.2	494	0

(a) DS - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	14	81.5	20	126.8	383	3
2010-4	35	47.5	23	57.3	210	3
2011-1	12	52.8	41.5	38.5	121	2
2011-2	27	36.3	31	46.7	245	1
2011-3	24	46.5	13.5	64.5	222	0
2011-4	39	18.2	7	29	132	0
2012-1	42	34	21.5	37.9	157	0
2012-2	30	27.4	18	38.5	169	0
2012-3	25	40.1	15	69.2	302	0
2012-4	41	53.4	19	85	402	0
2013-1	33	31.3	18	38.1	146	0
2013-2	18	57.7	40	62.1	216	0
2013-3	23	54	18	65.8	223	0
2013-4	8	43.1	29.5	45.6	123	0
Total	371	41.4	20	59.8	402	0

(b) DS - Churn bug

Table A.14: Caption of Descriptive Statistic for Churn feature and Churn bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	14	2.8	2	2.9	12	1
2010-4	39	2.1	1	1.5	8	1
2011-1	22	1.8	1	1.3	5	1
2011-2	28	3.1	2	3.3	16	1
2011-3	38	2.4	2	1.9	10	1
2011-4	35	2.9	1	5	30	1
2012-1	39	3.7	2	4	23	1
2012-2	31	1.8	1	1.4	7	1
2012-3	28	2.5	2	1.8	8	1
2012-4	42	2.5	2	1.7	6	1
2013-1	30	1.8	1.5	1	4	1
2013-2	19	1.5	1	1	5	1
2013-3	26	1.3	1	0.5	2	1
2013-4	8	1.9	1	1.7	6	1
Total	399	2.7	1	2.6	30	1

(a) DS - Bugs

Quarter	Finished	Not finished	Total	Finished	Not finished
2010-3	30	9	39	76.9	23.1
2010-4	75	6	81	92.6	7.4
2011-1	27	13	40	67.5	32.5
2011-2	79	7	86	91.9	8.1
2011-3	77	13	90	85.6	14.4
2011-4	88	13	101	87.1	12.9
2012-1	132	11	143	92.3	7.7
2012-2	44	12	56	78.6	21.4
2012-3	54	15	69	78.3	21.7
2012-4	97	10	107	90.7	9.3
2013-1	48	5	53	90.6	9.4
2013-2	21	7	28	75	25
2013-3	32	1	33	97	3
2013-4	15	0	15	100	0
Mean	58.5	8.7	67.2	86	14

(b) DS - Bugs per quarter

Table A.15: Caption of Descriptive Statistic for Bugs and Finished bugs per quarter a, b

A.4 Team 4 - Descriptive Statistics

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	9	2.7	2	2.5	7	1
2010-4	92	4.5	4	2.9	14	0
2011-1	90	10.4	10	3.1	18	4
2011-2	91	13.8	13	4.5	31	5
2011-3	92	14.1	13	4.6	28	6
2011-4	92	16.4	16	4.8	30	6
2012-1	91	16	15	3.9	25	9
2012-2	91	11.7	12	3.5	20	5
2012-3	92	14	14	4.2	26	7
2012-4	92	20.6	19.5	5.6	33	10
2013-1	90	19.5	19	7.2	37	5
2013-2	91	16	16	4.8	29	6
2013-3	92	15.5	15	5.9	29	6
2013-4	91	10.5	11	4	19	1
Total	1196	14	14	6.2	37	0

(a) DS - WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	4	3	1	4	9	1
2010-4	39	2.9	1	2.5	11	1
2011-1	48	3.8	3	2.8	15	1
2011-2	48	6.3	5	5.6	31	1
2011-3	54	6.3	5	5	31	1
2011-4	52	7.5	5	5.9	23	1
2012-1	61	6.8	5	4.4	17	1
2012-2	57	3.9	3	2.8	15	1
2012-3	33	6	5	4.4	15	1
2012-4	52	5.8	5	4.7	21	1
2013-1	61	8.6	7	6.9	34	1
2013-2	59	8.3	7	4.7	19	1
2013-3	60	8	7	5.1	26	1
2013-4	46	5	4	3.8	15	1
Total	674	6.2	5	5	34	1

(b) DS - Throughput

Table A.16: Caption of Descriptive Statistic for WIP and Throughput a, a

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	4	3	1	4	9	1
2010-4	39	2.9	1	2.5	11	1
2011-1	48	3.8	3	2.8	15	1
2011-2	48	6.3	5	5.6	31	1
2011-3	54	6.3	5	5	31	1
2011-4	52	7.5	5	5.9	23	1
2012-1	60	6.7	5	4.4	17	1
2012-2	57	3.9	3	2.8	15	1
2012-3	31	6.2	5	4.4	15	1
2012-4	48	5.8	5	4.8	21	1
2013-1	51	9.2	8	7.3	34	1
2013-2	50	8.5	7	4.8	19	1
2013-3	58	8.1	7.5	5.1	26	1
2013-4	44	5.1	5	3.8	15	1
Total	644	6.2	5	5.1	34	1

(a) DS - Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2012-1	1	11	11	-	11	11
2012-3	2	3.5	3.5	3.5	6	1
2012-4	4	4.8	5	3	8	1
2013-1	10	5.5	6	2.5	9	1
2013-2	9	7.2	7	4.7	15	2
2013-3	2	4.5	4.5	2.1	6	3
2013-4	2	1	1	0	1	1
Total	30	5.6	5.5	3.7	15	1

(b) DS - Throughput bug

Table A.17: Caption of Descriptive Statistic for Throughput feature and Throughput bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-2	34	13.2	10.5	10.7	50	1
2011-3	54	13.5	12.5	8.7	34	1
2011-4	49	17.9	14	14.4	61	1
2012-1	65	13.4	13	9.2	46	1
2012-2	56	9.4	8	7.2	33	1
2012-3	32	15.3	11	11.5	43	2
2012-4	63	10.2	8	10	66	1
2013-1	97	8.9	7	7.8	40	1
2013-2	80	9.4	8	8.6	48	1
2013-3	44	10.1	5.5	11.1	53	1
Total	574	11.6	9	10	66	1

(a) DS - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-2	34	9.8	8	10.1	43	0
2011-3	54	8.4	7	6.9	26	0
2011-4	49	13.2	10	12.8	53	0
2012-1	65	8.4	6	8.7	41	0
2012-2	56	4.2	2	6.1	27	0
2012-3	32	9.2	5	10.7	37	0
2012-4	63	5.1	2	9.4	59	0
2013-1	97	5.5	3	7.3	33	0
2013-2	80	6.4	3.5	8.3	47	0
2013-3	44	7.9	3	11	52	0
Total	574	7.4	5	9.2	59	0

(b) DS - Churn

Table A.18: Caption of Descriptive Statistic for Lead time and Churn a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-2	33	10.2	8	10.1	43	0
2011-3	53	8.6	7	6.9	26	0
2011-4	49	13.2	10	12.8	53	0
2012-1	63	8.7	7	8.8	41	0
2012-2	55	4.2	2	6.2	27	0
2012-3	29	10.1	6	10.9	37	0
2012-4	50	6.1	3	10.3	59	0
2013-1	65	7.8	7	7.9	33	0
2013-2	62	7.9	5.5	8.9	47	0
2013-3	37	9.3	6	11.4	52	0
Total	496	8.4	6	9.5	59	0

(a) DS - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-2	1	0	0	-1	0	0
2011-3	1	0	0	-1	0	0
2012-1	2	0	0	0	0	0
2012-2	1	0	0	-1	0	0
2012-3	3	0.7	0	1.2	2	0
2012-4	13	1.5	0	2.5	7	0
2013-1	32	0.9	0	1.6	5	0
2013-2	18	1.2	0	2.3	9	0
2013-3	7	0.4	0	1.1	3	0
Total	78	1	0	1.8	9	0

(b) DS - Churn feature

Table A.19: Caption of Descriptive Statistic for Churn feature and Churn bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-1	1	1	1	-	1	1
2011-2	2	1	1	0	1	1
2011-3	1	1	1	-	1	1
2012-1	2	1	1	0	1	1
2012-2	1	1	1	-	1	1
2012-3	4	1	1	0	1	1
2012-4	12	1.8	1.5	0.9	3	1
2013-1	32	1.6	1	0.9	4	1
2013-2	19	1.5	1	0.6	3	1
2013-3	12	1.2	1	0.4	2	1
2013-4	2	1	1	0	1	1
Total	88	1.4	1	0.8	4	1

(a) DS - Bugs

Quarter	Finished	Not finished	Total	Finished	Not finished
2011-1	1	0	1	100	0
2011-2	2	0	2	100	0
2011-3	1	0	1	100	0
2012-1	2	0	2	100	0
2012-2	1	0	1	100	0
2012-3	4	0	4	100	0
2012-4	21	1	22	95.5	4.5
2013-1	49	2	51	96.1	3.9
2013-2	27	1	28	96.4	3.6
2013-3	14	0	14	100	0
2013-4	2	0	2	100	0
Mean	11.3	.4	11.6	98.9	1.1

(b) DS - Bugs per quarter

Table A.20: Caption of Descriptive Statistic for Bugs and Bugs finished within quarter a, b

A.5 Team 5 - Descriptive Statistics

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	24	8.4	8	3.3	15	2
2010-4	92	18.7	18	6.2	40	8
2011-1	90	7.8	8.5	6.2	20	0
2011-2	91	21.3	18	12.9	58	0
2011-3	92	26.8	27	9.3	45	8
2011-4	92	27.8	27	9.9	46	10
2012-1	91	44.5	47	9.5	65	24
2012-2	91	51.3	51	7.5	74	38
2012-3	92	19.6	19	11.2	50	4
2012-4	92	20	19	9	38	7
2013-1	90	124.8	126	94.7	270	9
2013-2	91	231.1	266	85.9	286	12
2013-3	92	21.2	19	7.4	43	11
2013-4	51	9.6	10	4.3	19	1
Total	1171	48.4	24	70.5	286	0

(a) DS - WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	12	3.6	3	2.2	7	1
2010-4	49	4.2	3	3.5	15	1
2011-1	34	3.6	3	2.7	12	1
2011-2	51	7.2	7	5.4	19	1
2011-3	63	5.6	4	5	24	1
2011-4	58	5	5	3.7	17	1
2012-1	59	6.2	5	4.5	17	1
2012-2	59	5.3	4	3.8	15	1
2012-3	49	6.4	5	5	27	1
2012-4	50	4.7	3	4.1	17	1
2013-1	60	15.8	9.5	15.1	59	1
2013-2	58	6.7	7	4.6	22	1
2013-3	53	4.6	3	4.3	17	1
2013-4	19	2	1	1.7	7	1
Total	674	6.3	5	6.9	59	1

(b) DS - Throughput

Table A.21: Caption of Descriptive Statistic for WIP and Throughput a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1	1	1	-	1	1
2010-4	8	2.9	1	3.2	10	1
2011-1	7	3.4	4	2.2	7	1
2011-2	19	9.1	7	5.1	17	1
2011-3	23	6.4	4	5.7	24	1
2011-4	11	3.7	3	2.7	8	1
2012-1	6	4.7	3	4.2	10	1
2012-2	8	4.4	3	4.6	15	1
2012-4	10	5.1	3.5	4.2	14	1
2013-1	4	16.8	16.5	11.9	30	4
2013-2	1	2	2	-	2	2
2013-3	4	2.2	2	1.3	4	1
2013-4	6	1.7	1	1	3	1
Total	108	5.7	4	5.5	30	1

(a) DS - Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	11	3.8	3	2.1	7	1
2010-4	41	4.5	3	3.5	15	1
2011-1	27	3.6	3	2.8	12	1
2011-2	32	6.2	5.5	5.4	19	1
2011-3	40	5.2	4.5	4.6	22	1
2011-4	47	5.2	5	3.9	17	1
2012-1	53	6.4	6	4.6	17	1
2012-2	51	5.5	5	3.7	15	1
2012-3	49	6.4	5	5	27	1
2012-4	40	4.6	3	4.1	17	1
2013-1	56	15.7	9	15.4	59	1
2013-2	57	6.8	7	4.6	22	1
2013-3	49	4.8	3	4.5	17	1
2013-4	13	2.1	1	1.9	7	1
Total	566	6.4	5	7	59	1

(b) DS - Throughput bug

Table A.22: Caption of Descriptive Statistic for Throughput feature and Throughput bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	9	26.9	22	30	91	1
2010-4	37	24.6	20	19.3	71	2
2011-1	21	10.1	8	7.8	29	1
2011-2	47	15.2	14	10.2	41	1
2011-3	84	16.1	10	17.9	105	1
2011-4	69	24.5	15	25.8	153	1
2012-1	68	30.7	22	27.9	148	1
2012-2	72	36.3	26	30.3	138	1
2012-3	53	18.6	16	15.5	80	1
2012-4	54	27.3	14.5	39.7	259	1
2013-1	71	31.4	24	29.9	161	1
2013-2	60	34.5	21.5	37.5	178	2
2013-3	44	27.6	19	27	118	1
2013-4	9	11.9	10	9	31	1
Total	698	25.7	17	27.5	259	1

(a) DS - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	9	63.2	70	51.5	168	6
2010-4	37	59.6	44	58.2	205	1
2011-1	21	41.1	17	60.2	201	1
2011-2	47	35	20	45.1	185	1
2011-3	84	24.1	8	37.2	151	0
2011-4	69	29	15	37.8	172	0
2012-1	68	27.4	14.5	37	170	0
2012-2	72	40	22	48.7	192	0
2012-3	53	20.8	17	24.8	110	0
2012-4	54	24.4	6.5	40.6	244	0
2013-1	71	41	27	45	206	0
2013-2	60	37.8	24	39.8	161	0
2013-3	44	30.6	13.5	41.6	164	0
2013-4	9	32	27	36.6	115	0
Total	698	33.4	17	43	244	0

(b) DS - Churn

Table A.23: Caption of Descriptive Statistic for Lead time and Churn a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	9	93.6	78	79.8	205	8
2011-1	4	88.5	74	88.3	201	5
2011-2	8	69	45.5	57.4	182	23
2011-3	30	29.4	8	47.5	151	0
2011-4	18	46.3	13	58.2	172	0
2012-1	10	26.3	0	52.9	170	0
2012-2	13	75.7	83	65.9	192	0
2012-3	2	27.5	27.5	9.2	34	21
2012-4	9	38.1	41	38	100	0
2013-1	8	94.2	71	75.1	206	7
2013-2	8	31.5	4	40.5	91	0
2013-3	4	73.5	65	86	164	0
Total	123	52.1	27	61.2	206	0

(a) DS - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	9	63.2	70	51.5	168	6
2010-4	28	48.7	38	46	157	1
2011-1	17	30	15	48.7	187	1
2011-2	39	28.1	17	39.6	185	1
2011-3	54	21.1	9	30.2	149	0
2011-4	51	22.9	15	25.4	107	0
2012-1	58	27.6	15	34.1	152	0
2012-2	59	32.2	21	40.7	187	0
2012-3	51	20.5	17	25.2	110	0
2012-4	45	21.6	6	41	244	0
2013-1	63	34.3	23	35.1	153	0
2013-2	52	38.8	25	40	161	0
2013-3	40	26.4	13.5	33.7	113	0
2013-4	9	32	27	36.6	115	0
Total	575	29.4	17	36.8	244	0

(b) DS - Churn bug

Table A.24: Caption of Descriptive Statistic for Churn feature and Churn bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	19	1.9	1	1.8	7	1
2010-4	46	2.6	2	1.7	8	1
2011-1	36	1.9	1.5	1.3	7	1
2011-2	45	5.5	4	4.7	19	1
2011-3	51	3	2	2.6	15	1
2011-4	53	3.5	3	2.4	11	1
2012-1	52	3.7	3	2.6	10	1
2012-2	56	2.7	2	1.9	7	1
2012-3	49	3.2	2	2.7	13	1
2012-4	35	3	2	2.7	15	1
2013-1	56	9.6	7	8.3	38	1
2013-2	49	4.2	4	2.7	12	1
2013-3	41	3.1	2	2.4	10	1
2013-4	11	1.4	1	0.9	4	1
Total	604	3.8	3	4.1	38	1

(a) DS - Bugs

Quarter	Finished	Not finished	Total	Finished	Not finished
2010-3	24	13	37	64.9	35.1
2010-4	108	13	121	89.3	10.7
2011-1	57	12	69	82.6	17.4
2011-2	202	47	249	81.1	18.9
2011-3	119	33	152	78.3	21.7
2011-4	147	37	184	79.9	20.1
2012-1	149	45	194	76.8	23.2
2012-2	116	35	151	76.8	23.2
2012-3	133	25	158	84.2	15.8
2012-4	99	5	104	95.2	4.8
2013-1	502	37	539	93.1	6.9
2013-2	183	21	204	89.7	10.3
2013-3	123	5	128	96.1	3.9
2013-4	15	0	15	100	0
Mean	109.9	18.6	128.5	74.3	25.7

(b) DS - Bugs per quarter

Table A.25: Caption of Descriptive Statistic for Bugs and Bugs finished within quarter a, b

A.6 Team 6 - Descriptive Statistics

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	24	9.5	9	3.6	16	4
2010-4	92	10.3	10	2.6	16	6
2011-1	90	9.8	10	2	17	7
2011-2	91	10.4	11	2.4	16	4
2011-3	92	19.5	20.5	7.3	34	6
2011-4	92	22.9	22	9.3	44	9
2012-1	91	15.6	16	3.7	27	6
2012-2	91	17.5	18	6.1	42	8
2012-3	92	15.2	15	4.5	26	6
2012-4	92	26.3	25.5	10.6	50	11
2013-1	90	32.6	31	8.4	51	15
2013-2	91	43.7	43	5	60	36
2013-3	92	30.6	29.5	8	61	17
2013-4	85	37.4	39	20.8	125	10
Total	1205	22.1	18	13.4	125	4

(a) DS - WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	17	4.5	3	3.1	10	1
2010-4	51	3.3	3	2.6	10	1
2011-1	45	2.3	1	1.9	8	1
2011-2	37	2.8	3	1.9	8	1
2011-3	49	2.7	1	2.1	7	1
2011-4	40	3.2	3	2.3	9	1
2012-1	54	3.3	3	2.4	9	1
2012-2	51	5.2	3	5.8	37	1
2012-3	45	4	3	3.6	21	1
2012-4	63	6	5	4.5	23	1
2013-1	59	6.3	5	4.2	16	1
2013-2	61	4.4	3	3.7	15	1
2013-3	61	4.7	4	3.6	15	1
2013-4	58	9.1	5	23.8	181	1
Total	691	4.6	3	181	1	7.8

(b) DS - Throughput

Table A.26: Caption of Descriptive Statistic for WIP and Throughput a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	14	4.3	3	3.4	10	1
2010-4	47	3.5	3	2.7	10	1
2011-1	42	2.3	1	1.9	8	1
2011-2	33	2.7	3	1.9	8	1
2011-3	45	2.7	1	2.2	7	1
2011-4	38	3.2	3	2.4	9	1
2012-1	51	3.3	3	2.4	9	1
2012-2	51	5.2	3	5.8	37	1
2012-3	43	4	3	3.7	21	1
2012-4	55	6.4	5	4.5	23	1
2013-1	49	6.7	6	4.3	16	1
2013-2	47	5	3	3.8	15	1
2013-3	44	4.8	4	3.8	15	1
2013-4	50	10.1	5	25.5	181	1
Total	609	4.8	3	8.3	181	1

(a) DS - Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	3	5.3	6	2.1	7	3
2010-4	4	1	1	0	1	1
2011-1	3	2.3	1	2.3	5	1
2011-3	4	3.5	3	2.5	7	1
2011-4	4	2.5	2	1.9	5	1
2012-1	2	3.5	3.5	0.7	4	3
2012-2	3	3.7	2	2.9	7	2
2012-3	2	3	3	1.4	4	2
2012-4	8	3.5	2	3.3	11	1
2013-1	10	4.4	4.5	2.7	9	1
2013-2	14	2.4	1.5	2.9	12	1
2013-3	17	4.2	4	3.2	13	1
2013-4	8	2.5	3	1.1	4	1
Total	82	3.3	2.5	2.6	13	1

(b) DS - Throughput bug

Table A.27: Caption of Descriptive Statistic for Throughput feature and Throughput bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	19	4.7	2	7.9	34	1
2010-4	34	7.7	3.5	10.7	48	1
2011-1	35	10.9	8	9.1	33	1
2011-2	21	9.9	6	10.2	44	1
2011-3	20	15.9	15.5	10.7	46	3
2011-4	33	17.4	15	11.9	52	1
2012-1	59	16.1	14	13.2	70	1
2012-2	53	22.6	18	17.8	77	1
2012-3	55	15.4	13	12	53	1
2012-4	88	17.3	11	19.6	120	1
2013-1	109	12.9	8	11.8	54	1
2013-2	67	12	8	11.6	73	1
2013-3	84	13.5	10.5	13.4	94	1
2013-4	79	13.9	8	18.1	93	1
Total	756	14.3	10	14.5	120	1

(a) DS - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	19	139.4	31	224.3	812	2
2010-4	34	185.4	67.5	255.5	1030	1
2011-1	35	110.5	31	214.6	901	1
2011-2	21	266.6	140	321.4	1187	1
2011-3	20	175.4	146	159.7	496	8
2011-4	33	68.5	7	149.4	596	0
2012-1	59	72.2	9	213.6	1191	0
2012-2	53	59.9	16	149.5	769	0
2012-3	55	60	8	196.9	1207	0
2012-4	88	75.7	16	160.1	658	0
2013-1	109	91.1	19	202.4	937	0
2013-2	67	144.3	40	213.5	766	0
2013-3	84	69.4	19	128.6	739	0
2013-4	79	92.2	19	198.5	1127	0
Total	756	98.3	19	197.3	1207	0

(b) DS - Churn

Table A.28: Caption of Descriptive Statistic for Lead time and Churn a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	13	191.1	37	255.8	812	2
2010-4	33	189	67	258.6	1030	1
2011-1	32	119.4	32.5	222.5	901	1
2011-2	21	266.6	140	321.4	1187	1
2011-3	20	175.4	146	159.7	496	8
2011-4	31	72.9	7	153.2	596	0
2012-1	52	81.2	14.5	226.3	1191	0
2012-2	53	59.9	16	149.5	769	0
2012-3	46	70.7	12.5	214	1207	0
2012-4	64	85.2	16	167.8	655	0
2013-1	62	87.8	20	199.3	937	0
2013-2	44	154.7	28	232.9	766	0
2013-3	54	72.4	24.5	128.2	739	0
2013-4	51	94.8	30	187.7	994	0
Total	576	105.9	21	204.9	1207	0

(a) DS - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	6	27.5	9	45.3	119	2
2010-4	1	68	68	-	68	68
2011-1	3	15.7	11	13.6	31	5
2011-4	2	0.5	0.5	0.7	1	0
2012-1	7	5.6	0	12.7	34	0
2012-3	9	4.8	1	8	24	0
2012-4	24	50.4	11.5	137.7	658	0
2013-1	47	95.4	12	208.4	934	0
2013-2	23	124.3	54	173.4	694	0
2013-3	30	64	13	131.2	574	0
2013-4	28	87.5	10.5	220.3	1127	0
Total	180	73.8	12	169.3	1127	0

(b) DS - Churn bug

Table A.29: Caption of Descriptive Statistic for Churn feature and Churn bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	10	1.5	1.5	0.5	2	1
2010-4	7	1	1	0	1	1
2011-1	5	1.4	1	0.9	3	1
2011-2	8	1.1	1	0.4	2	1
2011-3	4	1.2	1	0.5	2	1
2011-4	2	2	2	0	2	2
2012-1	7	1.1	1	0.4	2	1
2012-3	11	1.3	1	0.5	2	1
2012-4	24	1.8	1.5	1	4	1
2013-1	39	1.9	2	1.2	7	1
2013-2	33	1.5	1	0.7	3	1
2013-3	34	1.6	1	0.8	4	1
2013-4	27	1.8	2	0.9	4	1
Total	211	1.6	1	0.9	7	1

(a) DS - Bugs

Quarter	Finished	Not finished	Total	Finished	Not finished
2010-3	14	1	15	93.3	6.7
2010-4	6	1	7	85.7	14.3
2011-1	6	1	7	85.7	14.3
2011-2	7	2	9	77.8	22.2
2011-3	3	2	5	60	40
2011-4	3	1	4	75	25
2012-1	8	0	8	100	0
2012-3	12	2	14	85.7	14.3
2012-4	41	2	43	95.3	4.7
2013-1	66	9	75	88	12
2013-2	43	7	50	86	14
2013-3	52	1	53	98.1	1.9
2013-4	49	0	49	100	0
Mean	23.9	2.2	26.1	87.0	13.0

(b) DS - Bugs

Table A.30: Caption of Descriptive Statistic for Bugs and Bugs finished within quarter a, b

A.7 Team 7 - Descriptive Statistics

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	12	17.7	8.5	19.5	54	1
2010-4	64	13	8	11.9	50	1
2011-1	57	12.8	8	15.1	89	1
2011-2	37	14.3	9	13	51	1
2011-3	36	17.8	11.5	18.1	79	1
2011-4	51	15	9	14.9	63	1
2012-1	35	14.9	11	18.3	86	1
2012-2	23	18.8	9	27.8	124	1
2012-3	42	15.1	7	18.6	81	1
2012-4	2	1.5	1.5	0.7	2	1
Total	359	14.8	8	16.6	124	1

(a) DS - WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	11	3.9	2	4	14	1
2010-4	53	3.7	3	2.5	13	1
2011-1	54	3.2	3	2.3	13	1
2011-2	33	2.3	2	1.2	5	1
2011-3	36	2	2	1.1	4	1
2011-4	44	2.2	2	1.5	6	1
2012-1	37	2	1	1.5	7	1
2012-2	25	2.2	2	1.4	6	1
2012-3	32	3.4	3	2.5	13	1
2012-4	3	1	1	0	1	1
Total	328	2.7	2	2.1	14	1

(b) DS - Throughput

Table A.31: Caption of Descriptive Statistic for WIP and Throughput a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	8	3.2	2.5	2.4	7	1
2010-4	37	4	3	2.7	13	1
2011-1	22	3.4	3	2.4	10	1
2011-2	10	2.7	2.5	1.7	5	1
2011-3	17	2.1	2	1.2	4	1
2011-4	26	2	1.5	1.3	5	1
2012-1	12	2	2	1.3	5	1
2012-2	9	2.2	2	0.8	3	1
2012-3	12	2.7	3	1.5	5	1
2012-4	3	1	1	0	1	1
Total	156	2.8	2	2.1	13	1

(a) DS - Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	3	5.7	2	7.2	14	1
2010-4	16	2.9	2.5	2	9	1
2011-1	32	3	3	2.3	13	1
2011-2	23	2.1	2	0.9	4	1
2011-3	19	1.9	2	1	4	1
2011-4	18	2.4	2	1.7	6	1
2012-1	25	2.1	1	1.6	7	1
2012-2	16	2.2	2	1.6	6	1
2012-3	20	3.8	3	2.9	13	1
Total	172	2.6	2	2.1	14	1

(b) DS - Throughput bug

Table A.32: Caption of Descriptive Statistic for Throughput feature and Throughput bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	12	17.7	8.5	19.5	54	1
2010-4	64	13	8	11.9	50	1
2011-1	57	12.8	8	15.1	89	1
2011-2	37	14.3	9	13	51	1
2011-3	36	17.8	11.5	18.1	79	1
2011-4	51	15	9	14.9	63	1
2012-1	35	14.9	11	18.3	86	1
2012-2	23	18.8	9	27.8	124	1
2012-3	42	15.1	7	18.6	81	1
2012-4	2	1.5	1.5	0.7	2	1
Total	359	14.8	8	16.6	124	1

(a) DS - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	12	154.2	49.5	189	647	4
2010-4	64	94.1	30	137.1	662	1
2011-1	57	74.3	26	108.4	479	1
2011-2	37	106.7	29	183.6	726	0
2011-3	36	85.1	21.5	143.6	577	0
2011-4	51	68.1	23	112.8	458	0
2012-1	35	43.4	15	70.1	367	0
2012-2	23	55.7	33	73.6	302	0
2012-3	42	53.8	28	82.5	424	3
2012-4	2	44	44	1.4	45	43
Total	359	77.3	26	124.8	726	0

(b) DS - Churn

Table A.33: Caption of Descriptive Statistic for Lead time and Churn a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	7	248.4	248	201	647	41
2010-4	38	117.8	77.5	130.8	585	6
2011-1	18	131.7	67.5	153.5	479	4
2011-2	10	173.3	84.5	231.5	726	0
2011-3	11	204.3	115	208.8	577	0
2011-4	25	114	55	141.8	458	0
2012-1	9	37.6	31	31.6	82	1
2012-2	7	40.6	34	47	140	0
2012-3	14	68.6	43	87.7	318	3
2012-4	2	44	44	1.4	45	43
Total	141	121.2	57	150.7	726	0

(a) DS - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	26	59.5	11.5	141.2	662	1
2011-1	39	47.9	20	67	276	1
2011-2	27	82	21	160.6	719	0
2011-3	25	32.6	17	50.1	226	1
2011-4	26	24	14.5	44.9	234	0
2012-1	26	45.4	11	79.7	367	0
2012-2	16	62.3	28	83.1	302	0
2012-3	28	46.4	22	80.4	424	4
Total	218	48.9	18	94.8	719	0

(b) DS - Churn bug

Table A.34: Caption of Descriptive Statistic for Churn feature and Churn bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	17	1.8	2	0.8	3	1
2010-4	28	1.7	1.5	0.9	4	1
2011-1	39	3.1	2	2.5	12	1
2011-2	26	2	2	1.3	5	1
2011-3	26	1.6	1	1.1	5	1
2011-4	24	2.2	2	1.5	7	1
2012-1	29	1.7	1	1.4	8	1
2012-2	18	2.7	2	2.6	11	1
2012-3	29	2.3	2	1.3	5	1
Total	240	2.1	2	1.7	12	1

(a) DS - Bugs

Quarter	Finished	Not finished	Total	Finished	Not finished
2010-3	20	10	30	66.7	33.3
2010-4	47	1	48	97.9	2.1
2011-1	119	2	121	98.3	1.7
2011-2	45	8	53	84.9	15.1
2011-3	35	6	41	85.4	14.6
2011-4	45	7	52	86.5	13.5
2012-1	46	2	48	95.8	4.2
2012-2	36	12	48	75	25
2012-3	67	0	67	100	0
Mean	38.3	4.3	42.7	65.9	34.1

(b) DS - Bugs per quarter

Table A.35: Caption of Descriptive Statistic for Bugs and Bugs finished within quarter a, b

A.8 Team 8 - Descriptive Statistics

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	19	0.4	0	0.5	1	0
2011-1	90	3.7	3	2.4	9	0
2011-2	91	5.9	6	1.9	11	1
2011-3	92	11.2	12	2.6	16	7
2011-4	92	7.9	7	3.3	14	3
2012-1	91	9.7	9	3.7	16	3
2012-2	91	4.3	2	4.6	12	1
2012-3	92	9.1	9	7.3	32	1
2012-4	92	5.6	7	4.6	18	1
2013-1	90	8.4	4	9.1	30	1
2013-2	91	19.7	18	11.2	55	2
2013-3	92	8.2	4	8.5	29	0
2013-4	77	4	5	2.8	11	0
Total	1100	8.1	6	7.3	55	0

(a) DS - WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	2	1	1	0	1	1
2011-1	12	1.5	1	0.7	3	1
2011-2	21	1.2	1	0.5	3	1
2011-3	15	1.7	1	1.1	4	1
2011-4	19	1.3	1	0.6	3	1
2012-1	16	1.4	1	1	5	1
2012-2	3	1	1	0	1	1
2012-3	23	2.5	2	2.5	12	1
2012-4	10	1.7	2	0.7	3	1
2013-1	25	3.8	3	3.4	14	1
2013-2	20	3.5	2.5	2.9	9	1
2013-3	21	3.5	2	3.5	13	1
2013-4	4	3.5	3	2.6	7	1
Total	191	2.3	1	2.4	14	1

(b) DS - Throughput

Table A.36: Caption of Descriptive Statistic for WIP and Throughput a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-1	2	1	1	0	1	1
2011-2	5	1.4	1	0.9	3	1
2011-3	1	1	1	-	1	1
2011-4	6	1.5	1	0.8	3	1
2012-1	6	1.8	1	1.6	5	1
2012-3	20	2.7	2	2.7	12	1
2012-4	6	1.8	2	0.8	3	1
2013-1	18	3.6	3	2.6	9	1
2013-2	12	5	5	2.8	9	1
2013-3	13	4.8	4	3.9	13	1
2013-4	3	4.3	4	2.5	7	2
Total	92	3.2	2	2.8	13	1

(a) DS - Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	2	1	1	0	1	1
2011-1	10	1.6	1.5	0.7	3	1
2011-2	16	1.2	1	0.4	2	1
2011-3	14	1.8	1	1.1	4	1
2011-4	13	1.1	1	0.4	2	1
2012-1	10	1.1	1	0.3	2	1
2012-2	3	1	1	0	1	1
2012-3	3	1	1	0	1	1
2012-4	4	1.5	1.5	0.6	2	1
2013-1	7	4.1	1	5	14	1
2013-2	8	1.4	1	0.7	3	1
2013-3	8	1.2	1	0.5	2	1
2013-4	1	1	1	-	1	1
Total	99	1.5	1	1.6	14	1

(b) DS - Throughput bug

Table A.37: Caption of Descriptive Statistic for Throughput feature and Throughput bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	1	1	1	-1	1	1
2011-1	3	3	3	2	5	1
2011-2	8	19.6	9.5	25.8	71	1
2011-3	13	20.5	15	18.9	69	1
2011-4	10	21.4	18.5	16	43	2
2012-1	9	27	8	61.3	190	1
2012-2	1	1	1	-1	1	1
2012-3	20	28.6	28.5	26.3	89	1
2012-4	10	17	15	13.3	45	3
2013-1	22	14.6	9.5	17.2	75	1
2013-2	16	23.1	5	41.4	150	1
2013-3	20	24.9	13.5	39.2	161	1
2013-4	4	70.2	75.5	56.3	129	1
Total	137	22.6	11	32.2	190	1

(a) DS - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	1	3	3	-	3	3
2011-1	3	7.7	3	9	18	2
2011-2	8	9.5	1.5	12.3	26	0
2011-3	13	12.8	3	16.4	51	0
2011-4	10	6	1.5	10.2	32	0
2012-1	9	17.7	0	50.4	152	0
2012-2	1	8	8	-	8	8
2012-3	20	12.4	2	20.5	84	0
2012-4	10	3.7	0.5	5.2	13	0
2013-1	22	9.6	5	15.5	73	0
2013-2	16	19.6	1.5	40.6	149	0
2013-3	20	17.9	4.5	35.9	145	0
2013-4	4	45.8	29	57.2	125	0
Total	137	13.4	3	27.9	152	0

(b) DS - Churn

Table A.38: Caption of Descriptive Statistic for Lead time and Churn a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-1	1	2	2	-	2	2
2011-2	2	0	0	0	0	0
2011-3	1	0	0	-	0	0
2011-4	3	10.7	0	18.5	32	0
2012-1	3	50.7	0	87.8	152	0
2012-3	19	13.1	3	20.9	84	0
2012-4	7	3.6	1	4.9	13	0
2013-1	15	5.5	4	5.7	17	0
2013-2	10	31.2	6.5	48.4	149	0
2013-3	15	23.3	7	40.2	145	0
2013-4	3	58.3	50	62.9	125	0
Total	79	17.4	4	34.4	152	0

(a) DS - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	1	3	3	-	3	3
2011-1	2	10.5	10.5	10.6	18	3
2011-2	6	12.7	12.5	12.8	26	0
2011-3	12	13.8	6.5	16.7	51	0
2011-4	7	4	3	5.3	14	0
2012-1	6	1.2	0	2.9	7	0
2012-2	1	8	8	-	8	8
2012-3	1	0	0	-	0	0
2012-4	3	4	0	6.9	12	0
2013-1	7	18.3	12	25.1	73	0
2013-2	6	0.2	0	0.4	1	0
2013-3	5	1.6	0	3.6	8	0
2013-4	1	8	8	-1	8	8
Total	58	8	1.5	13.6	73	0

(b) DS - Churn bug

Table A.39: Caption of Descriptive Statistic for Churn feature and Churn bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1	1	1	-1	1	1
2010-4	7	1.1	1	0.4	2	1
2011-1	9	1.2	1	0.4	2	1
2011-2	16	1.6	1	1.5	7	1
2011-3	15	1.3	1	0.6	3	1
2011-4	13	1.2	1	0.6	3	1
2012-1	9	1.2	1	0.4	2	1
2012-2	2	1	1	0	1	1
2012-3	4	1.2	1	0.5	2	1
2012-4	2	1	1	0	1	1
2013-1	10	3.3	2.5	2.8	10	1
2013-2	4	1	1	0	1	1
2013-3	4	1.5	1.5	0.6	2	1
2013-4	1	1	1	-1	1	1
Total	100	1.5	1	1.3	10	1

(a) DS - Bugs

Quarter	Finished	Not finished	Total	Finished	Not finished
2010-3	0	1	1	0	100
2010-4	2	6	8	25	75
2011-1	7	4	11	63.6	36.4
2011-2	16	10	26	61.5	38.5
2011-3	16	4	20	80	20
2011-4	15	1	16	93.8	6.3
2012-1	9	2	11	81.8	18.2
2012-2	2	0	2	100	0
2012-3	2	3	5	40	60
2012-4	1	1	2	50	50
2013-1	27	6	33	81.8	18.2
2013-2	3	1	4	75	25
2013-3	6	0	6	100	0
2013-4	1	0	1	100	0
Mean	6.7	2.6	9.3	59.5	40.5

(b) DS - Bugs per quarter

Table A.40: Caption of Descriptive Statistic for Bugs and Bugs finished within quarter a, b

A.9 Team 9 - Descriptive Statistics

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	52	4.5	4.5	3.4	10	0
2011-1	90	11.8	12.5	3.8	19	5
2011-2	91	11.2	8	6.9	34	3
2011-3	92	12.8	12	4	24	6
2011-4	92	16	17	5.1	25	5
2012-1	91	16.2	15	4.7	30	8
2012-2	91	35.4	33	16.4	67	8
2012-3	92	32.6	33.5	7.9	51	15
2012-4	92	21.8	23.5	10.4	39	3
2013-1	90	21.4	20.5	8	38	7
2013-2	91	26.6	21	12.7	47	11
2013-3	92	15.9	14	6.3	35	6
2013-4	84	17.1	17	4.5	29	7
Total	1140	19.2	16	11.6	67	0

(a) DS - WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	15.0	1.7	1.0	1.0	4.0	1.0
2011-1	30.0	1.8	1.0	1.0	4.0	1.0
2011-2	31.0	2.2	2.0	1.8	9.0	1.0
2011-3	27.0	1.6	1.0	0.9	5.0	1.0
2011-4	33.0	2.0	2.0	1.3	6.0	1.0
2012-1	41.0	2.4	2.0	1.4	5.0	1.0
2012-2	48.0	3.4	3.0	2.0	9.0	1.0
2012-3	53.0	3.3	3.0	2.2	9.0	1.0
2012-4	43.0	2.9	2.0	2.0	10.0	1.0
2013-1	51.0	2.9	2.0	1.8	9.0	1.0
2013-2	46.0	3.5	3.0	2.5	12.0	1.0
2013-3	50.0	2.6	2.0	1.8	9.0	1.0
2013-4	53.0	2.4	2.0	1.5	7.0	1.0
Total	521.0	2.6	2.0	1.9	12.0	1.0

(b) DS - Throughput

Table A.41: Caption of Descriptive Statistic for WIP and Throughput a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	12	1.7	1	1	4	1
2011-1	13	1.6	1	0.8	3	1
2011-2	5	1.4	1	0.5	2	1
2011-3	9	1.2	1	0.4	2	1
2011-4	17	1.7	1	1.4	6	1
2012-1	11	1.6	1	0.8	3	1
2012-2	23	2.9	3	1.6	6	1
2012-3	12	3.8	3.5	2.8	9	1
2012-4	20	2.7	2	1.7	6	1
2013-1	15	2.1	2	1	4	1
2013-2	24	3.6	3	2.8	12	1
2013-3	22	2.5	2	1.7	7	1
2013-4	31	2.4	2	1.5	7	1
Total	214	2.4	2	1.8	12	1

(a) DS - Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	3	2	2	1	3	1
2011-1	17	1.9	1	1.1	4	1
2011-2	26	2.4	2	1.9	9	1
2011-3	18	1.7	1.5	1	5	1
2011-4	16	2.3	2	1.1	5	1
2012-1	30	2.7	2.5	1.5	5	1
2012-2	25	3.8	3	2.3	9	1
2012-3	41	3.1	3	2	9	1
2012-4	23	3	3	2.3	10	1
2013-1	36	3.2	3	1.9	9	1
2013-2	22	3.4	3	2.2	8	1
2013-3	28	2.6	2	1.9	9	1
2013-4	22	2.5	2	1.6	7	1
Total	307	2.8	2	1.9	10	1

(b) DS - Throughput bug

Table A.42: Caption of Descriptive Statistic for Throughput feature and Throughput bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	14	11.4	8	10.2	33	2
2011-1	23	20.3	14	15.2	56	3
2011-2	24	19.7	17	12.8	54	4
2011-3	18	10.8	9	7.6	32	2
2011-4	27	10.5	6	7.9	30	2
2012-1	44	12.1	10.5	9.3	46	2
2012-2	58	12.8	10.5	10.2	59	2
2012-3	62	17.6	14	14.2	62	2
2012-4	37	20.5	13	26.1	140	2
2013-1	63	16.8	16	11.6	48	2
2013-2	56	20.8	15	22	128	2
2013-3	60	15	12.5	12	62	2
2013-4	52	14.4	10.5	11.4	48	2
Total	538	15.9	12	14.7	140	2

(a) DS - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	14	59.4	39.5	62	212	4
2011-1	23	69	56	63.8	204	1
2011-2	24	48.3	24.5	50.4	171	1
2011-3	18	68.6	17	104.3	309	1
2011-4	27	119.9	70	126.5	401	2
2012-1	44	79.9	35	102.8	426	1
2012-2	58	58.3	25	85.2	423	1
2012-3	62	61.9	31	93.6	472	1
2012-4	37	53.6	21	86.5	367	0
2013-1	63	43.1	21	62	218	0
2013-2	56	88.5	35	115.5	445	0
2013-3	60	90.2	31.5	113.3	432	0
2013-4	52	95.8	40	114.9	382	0
Total	538	72.2	30	97.5	472	0

(b) DS - Churn

Table A.43: Caption of Descriptive Statistic for Lead time and Churn a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	10	60.9	35.5	68.4	212	5
2011-1	7	74.6	61	44	154	19
2011-2	3	127.7	162	67.4	171	50
2011-3	2	204.5	204.5	130.8	297	112
2011-4	12	210.1	182.5	128.7	401	70
2012-1	14	135.5	72.5	133	426	12
2012-2	22	115.6	76.5	113.4	423	13
2012-3	17	63.8	41	82.1	310	3
2012-4	15	98.1	53	119.6	367	0
2013-1	26	79.3	44	82.8	218	0
2013-2	25	133.7	91	149.5	445	0
2013-3	24	96.7	29	114.1	354	0
2013-4	24	172.4	178.5	130.9	382	0
Total	201	115.9	72	118.8	445	0

(a) DS - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	4	55.5	49	50.7	120	4
2011-1	16	66.6	35.5	71.9	204	1
2011-2	21	37	20	37.3	136	1
2011-3	16	51.6	10.5	91.7	309	1
2011-4	15	47.7	28	64.3	242	2
2012-1	30	54	24	74.6	312	1
2012-2	36	23.3	13.5	27.8	134	1
2012-3	45	61.2	30	98.4	472	1
2012-4	22	23.3	12	30	110	0
2013-1	37	17.7	16	15.6	52	0
2013-2	31	52	31	59.3	244	0
2013-3	36	85.9	35	114.2	432	0
2013-4	28	30.2	21.5	25.9	102	0
Total	337	46.1	23	70.4	472	0

(b) DS - Churn bug

Table A.44: Caption of Descriptive Statistic for Churn feature and Churn bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	2	1	1	0	1	1
2010-4	13	1.9	2	1.1	4	1
2011-1	18	1.8	1	0.9	3	1
2011-2	20	2.2	1	2.6	12	1
2011-3	14	1.6	1.5	0.6	3	1
2011-4	23	1.8	1	1.2	5	1
2012-1	33	2.1	2	1.5	7	1
2012-2	43	2.2	2	1.5	7	1
2012-3	40	2.7	2	1.9	9	1
2012-4	33	2.2	2	1.7	8	1
2013-1	34	2.2	1	1.7	8	1
2013-2	39	2.1	2	1.6	6	1
2013-3	38	2.1	1.5	1.5	7	1
2013-4	42	1.5	1	0.8	4	1
Total	403	2.1	1	1.5	12	1

(a) DS - Bugs

Quarter	Finished	Not finished	Total	Finished	Not finished
2010-3	0	2	2	0	100
2010-4	8	17	25	32	68
2011-1	19	13	32	59.4	40.6
2011-2	42	3	45	93.3	6.7
2011-3	11	11	22	50	50
2011-4	22	19	41	53.7	46.3
2012-1	52	18	70	74.3	25.7
2012-2	73	22	95	76.8	23.2
2012-3	100	7	107	93.5	6.5
2012-4	58	16	74	78.4	21.6
2013-1	73	3	76	96.1	3.9
2013-2	80	4	84	95.2	4.8
2013-3	79	1	80	98.8	1.3
2013-4	63	0	63	100	0
Mean	37.9	8.3	46.17	58.4	41.6

(b) DS - Bugs per quarter

Table A.45: Caption of Descriptive Statistic for Bugs and Bugs finished within quarter a, b

A.10 Team 10 - Descriptive Statistics

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	24	2.7	3	0.9	5	1
2010-4	92	13.1	13.5	6	26	2
2011-1	90	8.1	8	6.2	22	0
2011-2	91	6	4	4.8	17	0
2011-3	92	0.9	1	0.8	3	0
2011-4	92	16.7	17.5	13.7	40	1
2012-1	91	24.6	24	3.8	36	17
2012-2	91	34.5	35	8.4	51	18
2012-3	92	12.7	10	8.7	44	4
2012-4	92	25.8	19.5	13.6	59	10
2013-1	90	16.3	6	14.5	49	5
2013-2	91	8.9	8	4.5	21	5
2013-3	92	12.5	12	5.8	29	3
2013-4	57	15.4	15	4.8	26	7
Total	1177	14.8	12	12.2	59	0

(a) DS - WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	5	1.6	1	0.9	3	1
2010-4	44	2.2	2	1.6	7	1
2011-1	29	2.5	2	1.8	7	1
2011-2	21	1.9	1	1.4	6	1
2011-3	8	1	1	0	1	1
2011-4	34	2.6	2	1.7	7	1
2012-1	32	1.8	1	1.2	6	1
2012-2	52	2.7	2	1.6	7	1
2012-3	38	1.7	1	1.1	6	1
2012-4	47	2.7	2	3	16	1
2013-1	25	2.9	2	1.8	8	1
2013-2	10	1.9	1.5	1.1	4	1
2013-3	36	1.8	1.5	1.2	5	1
2013-4	23	1.8	1	1.2	5	1
Total	404	2.2	2	1.7	16	1

(b) DS - Throughput

Table A.46: Caption of Descriptive Statistic for WIP and Throughput a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	5	2	1	2.2	6	1
2011-1	2	2	2	0	2	2
2011-2	1	6	6	-	6	6
2011-3	3	1	1	0	1	1
2011-4	7	3	2	2.2	7	1
2012-1	7	1.1	1	0.4	2	1
2012-2	15	2.8	2	1.8	7	1
2012-3	6	1	1	0	1	1
2012-4	11	3	1	4.4	16	1
2013-1	2	2.5	2.5	0.7	3	2
2013-2	1	1	1	-	1	1
2013-3	4	1.2	1	0.5	2	1
2013-4	5	2.4	2	1.7	5	1
Total	69	2.3	1	2.3	16	1

(a) DS Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	5	1.6	1	0.9	3	1
2010-4	39	2.2	2	1.5	7	1
2011-1	27	2.6	2	1.9	7	1
2011-2	20	1.7	1	1.1	5	1
2011-3	5	1	1	0	1	1
2011-4	27	2.5	2	1.5	6	1
2012-1	25	1.9	1	1.3	6	1
2012-2	37	2.7	3	1.5	6	1
2012-3	32	1.8	1	1.2	6	1
2012-4	36	2.6	2	2.6	13	1
2013-1	23	2.9	2	1.9	8	1
2013-2	9	2	2	1.1	4	1
2013-3	32	1.9	2	1.2	5	1
2013-4	18	1.7	1	1	4	1
Total	335	2.2	2	1.6	13	1

(b) DS Throughput bug

Table A.47: Caption of Descriptive Statistic for Throughput feature and Throughput bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1	18	18	-1	18	18
2010-4	30	18	13.5	12.9	45	2
2011-1	26	11.6	8	9.6	41	3
2011-2	18	16.2	6	18.2	60	2
2011-3	7	9	6	7.2	21	3
2011-4	37	21.1	13	17.9	56	2
2012-1	20	27.8	27	21.6	78	2
2012-2	69	27	22	21.3	106	2
2012-3	27	22.7	17	23.2	97	2
2012-4	46	29.8	17.5	48.3	313	3
2013-1	26	19.5	10.5	19.1	67	2
2013-2	13	28.1	31	14.1	52	11
2013-3	24	24.4	19.5	18	62	2
2013-4	17	19.7	15	21.6	96	2
Total	361	22.7	15	24.8	313	2

(a) DS - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1	5	5	-	5	5
2010-4	30	58	15	106.2	469	1
2011-1	26	41	17.5	56.8	266	0
2011-2	18	14	3.5	19.1	59	0
2011-3	7	70	1	131.8	358	0
2011-4	37	24.2	7	53.9	309	0
2012-1	20	43.6	13.5	103	441	0
2012-2	69	39.3	11	85	438	0
2012-3	27	38.3	12	60.7	267	0
2012-4	46	52.2	17	88.1	373	0
2013-1	26	71.7	26	110.7	406	0
2013-2	13	37.1	24	45.9	123	0
2013-3	24	64.2	16.5	113.9	469	0
2013-4	17	61.6	10	112.8	321	0
Total	361	45.4	14	86.2	469	0

(b) DS - Churn

Table A.48: Caption of Descriptive Statistic for Lead time and Churn a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	1	219	219	-	219	219
2011-1	4	77	21	126.5	266	0
2011-2	3	29.3	25	27.8	59	4
2011-3	2	47.5	47.5	67.2	95	0
2011-4	10	35.3	0	97	309	0
2012-1	1	441	441	-	441	441
2012-2	25	66	5	131.9	438	0
2012-3	4	79	24.5	127.4	267	0
2012-4	8	95.5	16	133.9	310	0
2013-1	3	45.7	13	68.1	124	0
2013-2	2	48	48	67.9	96	0
2013-3	3	168.3	219	149.6	286	0
2013-4	3	78.3	0	135.7	235	0
Total	69	75.5	5	123.6	441	0

(a) DS - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1	5	5	-	5	5
2010-4	29	52.5	15	103.6	469	1
2011-1	22	34.4	17.5	35.6	125	0
2011-2	15	10.9	1	16.5	54	0
2011-3	5	79	1	156.7	358	0
2011-4	27	20.1	14	26.4	128	0
2012-1	19	22.7	13	44.3	195	0
2012-2	44	24.1	14.5	32.5	141	0
2012-3	23	31.2	12	42.3	151	0
2012-4	38	43.1	17	74.6	373	0
2013-1	23	75.1	27	115.8	406	0
2013-2	11	35.1	24	45.2	123	0
2013-3	21	49.3	14	104	469	0
2013-4	14	58	11	113	321	0
Total	292	38.3	15	73.2	469	0

(b) DS - Churn bug

Table A.49: Caption of Descriptive Statistic for Churn feature and Churn bug a, b

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	11	2.3	1	3	11	1
2010-4	32	2.6	2	1.7	8	1
2011-1	29	2.1	2	1.3	6	1
2011-2	24	1.3	1	0.7	4	1
2011-3	15	1.5	1	0.6	3	1
2011-4	37	2.5	2	2.1	9	1
2012-1	26	1.6	1	0.9	4	1
2012-2	34	2	2	1.5	8	1
2012-3	29	1.6	1	0.9	4	1
2012-4	35	2	1	1.5	7	1
2013-1	29	2.3	1	2.7	13	1
2013-2	16	1.5	1	0.6	3	1
2013-3	22	2.3	2	1.8	7	1
2013-4	19	1.4	1	0.6	3	1
Total	370	1.9	1	1.6	13	1

(a) DS - Bugs

Quarter	Finished	Not finished	Total	Finished	Not finished
2010-3	8	17	25	32	68
2010-4	65	17	82	79.3	20.7
2011-1	49	11	60	81.7	18.3
2011-2	29	2	31	93.5	6.5
2011-3	9	13	22	40.9	59.1
2011-4	72	22	94	76.6	23.4
2012-1	23	19	42	54.8	45.2
2012-2	53	16	69	76.8	23.2
2012-3	30	15	45	66.7	33.3
2012-4	65	6	71	91.5	8.5
2013-1	62	6	68	91.2	8.8
2013-2	16	8	24	66.7	33.3
2013-3	45	5	50	90	10
2013-4	26	0	26	100	0
Mean	30.8	9.3	40.13	67.0	33.0

(b) DS - Bugs per quarter

Table A.50: Caption of Descriptive Statistic for Bugs and Bugs finished within quarter a, b

Bibliography

Adams, M. and B. Smoak (1990). 'Managing manufacturing improvement using computer integrated manufacturing methods'. In: *Semiconductor Manufacturing Science Symposium, 1990. ISMSS 1990., IEEE/SEMI International*, pp. 9–13. DOI: 10.1109/ISMSS.1990.66111.

Alliance, Scrum (2012). 'Scrum, A description'. In:

Anderson, David J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press. ISBN: 0984521402.

Anderson, David et al. (2011). 'Studying Lean-Kanban Approach Using Software Process Simulation'. In: *Agile Processes in Software Engineering and Extreme Programming*. Ed. by Alberto Sillitti et al. Vol. 77. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, pp. 12–26. ISBN: 978-3-642-20676-4. DOI: 10.1007/978-3-642-20677-1_2.

Beedle, Mike et al. (1999). 'SCRUM: An extension pattern language for hyperproductive software development'. In: *Pattern Languages of Program Design* 4, pp. 637–651.

Birkeland, JørnOla (2010). 'From a Timebox Tangle to a More Flexible Flow'. In: *Agile Processes in Software Engineering and Extreme Programming*. Ed. by Alberto Sillitti et al. Vol. 48. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, pp. 325–334. ISBN: 978-3-642-13053-3. DOI: 10.1007/978-3-642-13054-0_35.

Brekkan, Elin and Eystein Mathisen (2010). 'Introducing Scrum in Companies in Norway: A Case Study'. In:

Cocco, Luisanna et al. (2011). 'Simulating Kanban and Scrum vs. Waterfall with System Dynamics'. In: *Agile Processes in Software Engineering and Extreme Programming*. Springer, pp. 117–131.

Conboy, Kieran (2009). 'Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development'. In: *Information Systems Research* 20.3, pp. 329–354. DOI: 10.1287/isre.1090.0236.

Concas, Giulio et al. (2013). 'Simulation of software maintenance process, with and without a work-in-process limit'. In: *Journal of Software: Evolution and Process* 25.12, pp. 1225–1248. ISSN: 2047-7481. DOI: 10.1002/smr.1599.

D. Sjøberg Yamashita, A, B Anda, A Mockus et al. (2012). 'Quantifying the effect of code smells on maintenance effort'. In:

El-Emam, Khaled (2000). 'A methodology for validating software product metrics'. In:

Gandomani, TAGHI JAVDANI et al. (2013). 'Important considerations for agile software development methods governance.' In: *Journal of Theoretical & Applied Information Technology* 55.3.

Gerring, John (2006). *Case Study Research: Principles and Practices*. Cambridge University Press. ISBN: 0521676568.

Gupta, Vikram (May 2013). *InfoQ Interviews David J. Anderson at Lean Kanban 2013 Conference*. URL: http://www.infoq.com/articles/David_Anderson_Lean_Kanban_2013_Conference_Intervew (visited on 01/10/2013).

IBM (Jan. 2014). *IBM SPSS Statistics 21 Core System User's Guide*. URL: http://www.sussex.ac.uk/its/pdfs/SPSS_Core_System_Users_Guide_21.pdf (visited on 30/01/2014).

Ikonen, Marko et al. (2010). 'Exploring the sources of waste in Kanban software development projects'. In: *Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference on*. IEEE, pp. 376–381.

Ikonen, M. et al. (2011). 'On the Impact of Kanban on Software Project Work: An Empirical Case Study Investigation'. In: *Engineering of Complex Computer Systems (ICECCS), 2011 16th IEEE International Conference on*, pp. 305–314. DOI: 10.1109/ICECCS.2011.37.

Investopedia (30th Nov. 2013). *Quarter - Q1, Q2, Q3, Q4*. URL: http://www.investopedia.com/terms/q/quarter.asp (visited on 30/11/2013).

Javadian Kootanaee, Akbar, K Babu and Hamid Talari (2013). 'Just-in-Time Manufacturing System: From Introduction to Implement'. In: *Available at SSRN 2253243*.

Kniberg, Henrik (2010). *Kanban and Scrum - making the most of both*. lulu.com. ISBN: 0557138329.

Lai, C.L., W.B. Lee and W.H. Ip (2003). 'A study of system dynamics in just-in-time logistics'. In: 138, pp. 265–269.

Leonardo Campos Rafael Buzon, Eric Fer (Mar. 2013). *Kanban Pioneer: Interview with David J. Anderson*. URL: http://www.infoq.com/articles/David-Anderson-Kanban/ (visited on 30/09/2013).

Manning, James (2013). 'Lean Software Development'. In:

Middleton, P. and D. Joyce (2012). 'Lean Software Management: BBC Worldwide Case Study'. In: *Engineering Management, IEEE Transactions on* 59.1, pp. 20–32. ISSN: 0018-9391. DOI: 10.1109/TEM.2010.2081675.

Munassar, Nabil Mohammed Ali and A Govardhan (2010). 'A Comparison Between Five Models Of Software Engineering.' In: *International Journal of Computer Science Issues (IJCSI)* 7.5.

Ohno, Taiichi (2001). *Toyota Production System on Compact Disc: Beyond Large-Scale Production*. Productivity Press. ISBN: 1563272679.

Oracle (2013). 'Arraylist'. In:

Poppendieck, Mary (2003). 'Lean Development and the Predictability Paradox'. In:

Poppendieck, Mary and Tom Poppendieck (2003). *Lean Software Development: An Agile Toolkit*. Addison-Wesley Professional. ISBN: 0321150783.
– (2006). *Implementing Lean Software Development: From Concept to Cash*. Addison-Wesley Professional. ISBN: 0321437381.
– (2009). *Leading Lean Software Development: Results Are not the Point*. Addison-Wesley Professional. ISBN: 0321620704.
Raman, S. (Oct. 1998). 'Lean software development: is it feasible?' In: *Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE*. Vol. 1, C13/1–C13/8 vol.1. DOI: 10.1109/DASC.1998.741480.
Rouse, Margaret (2005). *Throughput*. URL: http://searchnetworking.techtarget.com/definition/throughput (visited on 04/03/2014).
Seikola, M., H. Loisa and A. Jagos (Aug. 2011). 'Kanban Implementation in a Telecom Product Maintenance'. In: *Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on*, pp. 321–329. DOI: 10.1109/SEAA.2011.56.
Shepard, Jon M. and Robert W. Greene (2002). *Sociology and You*. Glencoe/McGraw-Hill. ISBN: 0078285763.
Shinkle, C.M. (2009). 'Applying the Dreyfus Model of Skill Acquisition to the Adoption of Kanban Systems at Software Engineering Professionals (SEP)'. In: *Agile Conference, 2009. AGILE '09*. Pp. 186–191. DOI: 10.1109/AGILE.2009.25.
Sienkiewicz, Lukasz (2012). 'Scrumban - the Kanban as an addition to Scrum software development method in a Network Organization'. In:
Sjøberg, D.I.K., A. Johnsen and J. Solberg (2012). 'Quantifying the Effect of Using Kanban versus Scrum: A Case Study'. In: *Software, IEEE* 29.5, pp. 47–53. ISSN: 0740-7459. DOI: 10.1109/MS.2012.110.
Software Innovation (Dec. 2013). URL: http://www.software-innovation.com/EN/COMPANY/pages/default_.aspx (visited on 12/12/2013).
Srinivasan, Mandyam M., Steven J. Ebbing and Alan T. Swearingen (2003). 'Woodward Aircraft Engine Systems Sets Work-in-Process Levels for High-Variety, Low-Volume Products'. In: *Interfaces* 33.4, pp. 61–69. DOI: 10.1287/inte.33.4.61.16377.
The-Kanban-Way (2011). *Importance of Kanban work-in-progress (WIP) limits*. URL: http://www.kanbanway.com/importance-of-kanban-work-in-progress-wip-limits#.U0Pmzq2Swkc (visited on 14/10/2013).
Vesaliusstraat, Andreas (n.d.). 'The Non-Transitivity of Pearson's Correlation Coefficient: An Educational Perspective'. In:
Yin, Liangfang, Yuanqing Xiao and Zhenyu Xu (June 2013). 'A new principle based on Pearson correlation coefficient to avoid mal-operation of the restricted earth fault protection'. In: *Electricity Distribution (CIRED 2013), 22nd International Conference and Exhibition on*, pp. 1–4. DOI: 10.1049/cp.2013.0799.
Yin, Robert K. (2008). *Case Study Research: Design and Methods (Applied Social Research Methods)*. SAGE Publications, Inc. ISBN: 1412960991.