

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

Truls Skeie
Master's Thesis Spring 2014



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

Truls Skeie

2nd April 2014

Abstract

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Research Question	2
1.3	Approach	2
1.4	Chapter overview	3
2	Background	5
2.1	Waterfall	5
2.2	Scrum	6
2.3	Lean	7
2.4	Kanban	9
2.4.1	Kanban Board	10
2.4.2	WIP limit	11
2.4.3	Limit WIP vs. Unlimited WIP	12
2.4.4	Benefits with setting WIP limit	13
2.5	Lead time	14

2.6	Just-In-Time	14
2.7	Throughput	16
2.8	Code churn	16
2.9	Software Innovation	17
2.9.1	Teams	17
3	Research Methods	19
3.1	Case study	19
3.2	Choice of case	20
3.2.1	Software Innovation's development process	22
4	Data collected and calculations	25
4.1	SPSS	26
4.2	WIP per day	26
4.2.1	Step 1: Gather all unique dates into a ArrayList	26
4.2.2	Step 2: Gather the remaining dates	28
4.2.3	Step 3 Measure WIP	28
4.2.4	Example	29
4.3	Rest of the variables	33
4.3.1	Throughput	34
4.3.2	Churn	34
4.3.3	Lead time	35
4.3.4	Lead time and churn	36

4.3.5	Bug and feature	36
4.3.6	Bugs finished, quarter	37
4.3.7	Avg days backlog, bug	37
5	Results	39
5.1	Correlation - WIP	39
5.2	Correlation - Lead time	43
5.3	Correlation - Bugs	45
5.4	Correlation - Throughput	48
5.5	Correlation - Churn	50
5.6	Team size	51
6	Discussion	53
6.1	swag	53
Appendices		55
A	Descriptive statistics for the ten teams	57
A.1	Team 1 - Descriptive Statistics	57
A.2	Team 2 - Descriptive Statistics	60
A.3	Team 3 - Descriptive Statistics	62
A.4	Team 4 - Descriptive Statistics	65
A.5	Team 5 - Descriptive Statistics	67
A.6	Team 6 - Descriptive Statistics	70

A.7	Team 7 - Descriptive Statistics	72
A.8	Team 8 - Descriptive Statistics	75
A.9	Team 9 - Descriptive Statistics	77
A.10	Team 10 - Descriptive Statistics	80
I	Notes	83
6.10.1	WIP correlation	85
6.10.2	Throughput	90
6.10.3	Bugs	92
6.11	Team 2	93
6.11.1	WIP	93
6.11.2	Lead time	94
6.11.3	Analyzed data per quarter	95
6.12	360	101
6.13	Krypton	106
6.14	Neon	110
7	Conclusion	115

List of Figures

2.1	Waterfall model	6
2.2	Example of a Kanban board	11
2.3	JIT example	15
4.1	Illustrating the WIP timeline for example stated in section 4.2.4	30
5.1	Correlation graphs between WIP and the throughput variables.	41
5.2	Correlation graphs between WIP and the throughput variables for team ten.	42
5.3	Correlation graphs between lead time and churn variables for team ten.	45
5.4	Correlation graphs between bugs and the throughput variables.	46
5.5	Correlation graphs between bug and the churn variables for team nine. .	47
5.6	Correlation graphs between the throughput variables.	49
6.1	WIP, Throughput, Lead time and Bugs for 360 per quarter	95
6.2	WIP, Throughput, Lead time and Bugs for Frontend per quarter	96
6.3	WIP, Throughput, Lead time and Bugs for Krypton per quarter	97
6.4	WIP, Throughput, Lead time and Bugs for Neon per quarter	99

List of Tables

2.1	Throughput	16
2.2	Optional caption for list of figures	18
3.1	Excerpt from the data set	20
3.2	Variables from the SI dataset	21
3.3	Relationship between variable and columns from SI	22
4.1	The standard of the data set	25
4.2	Variables of the WIP objects	26
4.3	Showing Task ID, Date From and Date to	30
4.4	How churn is presented in the excel document	35
4.5	How lead time is recorded in the excel document	35
4.6	A excerpt from the result data produced by the program	37
5.1	Correlation - WIP	40
5.2	Descriptive Statistic - Correlation - WIP	43
5.3	Correlation - Lead time	44
5.4	Descriptive Statistic - Correlation - Lead time	45

5.5	Correlation - Bugs	46
5.6	Descriptive Statistic - Correlation - Bugs	48
5.7	Correlation - Throughput	48
5.8	Descriptive Statistic - Correlation - Throughput	50
5.9	Correlation - Churn	51
5.10	Descriptive Statistic - Correlation - Churn	51
5.11	Correlation - Leadtime	52
5.12	Descriptive Statistic - Correlation - Throughput	52
A.1	Optional caption for list of figures	57
A.2	Optional caption for list of figures	58
A.3	Optional caption for list of figures	58
A.4	Optional caption for list of figures	59
A.5	Optional caption for list of figures	59
A.6	Optional caption for list of figures	60
A.7	Optional caption for list of figures	60
A.8	Optional caption for list of figures	61
A.9	Optional caption for list of figures	61
A.10	Optional caption for list of figures	62
A.11	Optional caption for list of figures	62
A.12	Optional caption for list of figures	63
A.13	Optional caption for list of figures	63

A.14 Optional caption for list of figures	64
A.15 Optional caption for list of figures	64
A.16 Optional caption for list of figures	65
A.17 Optional caption for list of figures	65
A.18 Optional caption for list of figures	66
A.19 Optional caption for list of figures	66
A.20 Optional caption for list of figures	67
A.21 Optional caption for list of figures	67
A.22 Optional caption for list of figures	68
A.23 Optional caption for list of figures	68
A.24 Optional caption for list of figures	69
A.25 Optional caption for list of figures	69
A.26 Optional caption for list of figures	70
A.27 Optional caption for list of figures	70
A.28 Optional caption for list of figures	71
A.29 Optional caption for list of figures	71
A.30 Optional caption for list of figures	72
A.31 Optional caption for list of figures	72
A.32 Optional caption for list of figures	73
A.33 Optional caption for list of figures	73
A.34 Optional caption for list of figures	74
A.35 Optional caption for list of figures	74

A.36 Optional caption for list of figures	75
A.37 Optional caption for list of figures	75
A.38 Optional caption for list of figures	76
A.39 Optional caption for list of figures	76
A.40 Optional caption for list of figures	77
A.41 Optional caption for list of figures	77
A.42 Optional caption for list of figures	78
A.43 Optional caption for list of figures	78
A.44 Optional caption for list of figures	79
A.45 Optional caption for list of figures	79
A.46 Optional caption for list of figures	80
A.47 Optional caption for list of figures	80
A.48 Optional caption for list of figures	81
A.49 Optional caption for list of figures	81
A.50 Optional caption for list of figures	82
6.51 Team one - Correlation - WIP	85
6.52 Descriptive statistics for WIP - Team one	85
6.53 Descriptive statistic for throughput - team one	86
6.54 Descriptive statistic for throughput feature - team one	87
6.55 Descriptive statistic for throughput bugs - team one	87
6.56 Number of throughput divided into bugs and feature - Team one	87

6.57 The percent number of bugs finished in the same quarter as reported - Team one	88
6.58 Descriptive Statistic for churn feature - Team one	89
6.59 Descriptive Statistic for churn bugs - Team one	89
6.60 Team one - Correlation - Throughput	90
6.61 Descriptive statistic for throughput - team one	90
6.62 team one - Average churn for feature and bugs - Bugs	91
6.63 Lead time for throughput feature and bugs per quarter	91
6.64 Descriptive statistic for churn - team one	92
6.65 Team two - Correlation - Throughput	92
6.66 360 - Descriptive statistic - Bugs	93
6.67 Team two - Correlation - WIP	93
6.68 Descriptive statistic - WIP - Team two	94
6.69 Team one - Correlation - Throughput	94
6.70 Team one - Correlation - Lead time_union	94
6.71 Descriptive statistic - Throughput - Team two	94
6.72 360 - Correlation	101
6.73 Descriptive statistic - Lead-time and churn - 360	102
6.74 Frontend Correlation	103
6.75 Descriptive statistic - Lead-time and churn - Frontend	104
6.76 Frontend - Descriptive statistic - Bugs	105
6.77 Frontend - Descriptive statistic - TP feature	105

6.78	Frontend - Descriptive statistic - TP bugs	106
6.79	Krypton - correlation	106
6.80	Descriptive statistic - Lead-time and churn - Krypton	107
6.81	Frontend - Descriptive statistic - Throughput	108
6.82	Krypton - Descriptive statistic WIP	108
6.83	Krypton - Descriptive statistic - Bugs	109
6.84	Krypton - Descriptive statistic - TP feature	109
6.85	Krypton - Descriptive statistic - TP bugs	109
6.86	Neon - Correlation	110
6.87	Descriptive statistic - Lead-time and churn - Neon	111
6.88	Neon - Descriptive statistic - Throughput	112
6.89	Neon - Descriptive statistic - WIP	112
6.90	Neon - Descriptive statistic - Bugs	112
6.91	Neon - Descriptive statistic - TP feature	113
6.92	Neon - Descriptive statistic - TP bugs	113

Listings

4.1	Gather all unique dates into ArrayList	27
4.2	Gather WIP object to the right data structure	27
4.3	Gather the remaining dates.	28
4.4	WIP measurement	29
4.5	Creating WIP-object	31
4.6	Creating WIP-object	31
4.7	Pseudocode example of how throughput objects are added	33
4.8	Pseudocode example of how throughput is measured	34
4.9	Pseudocode example of how throughput is measured	34
4.10	Pseudocode example of lead time is measured	35

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 to answer the research questions.

The program will take the part of the analysis. The program was made for this thesis in order to transform the data set from SI into more suited. 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 ??: ??:

Chapter 7: Conclusion:

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)

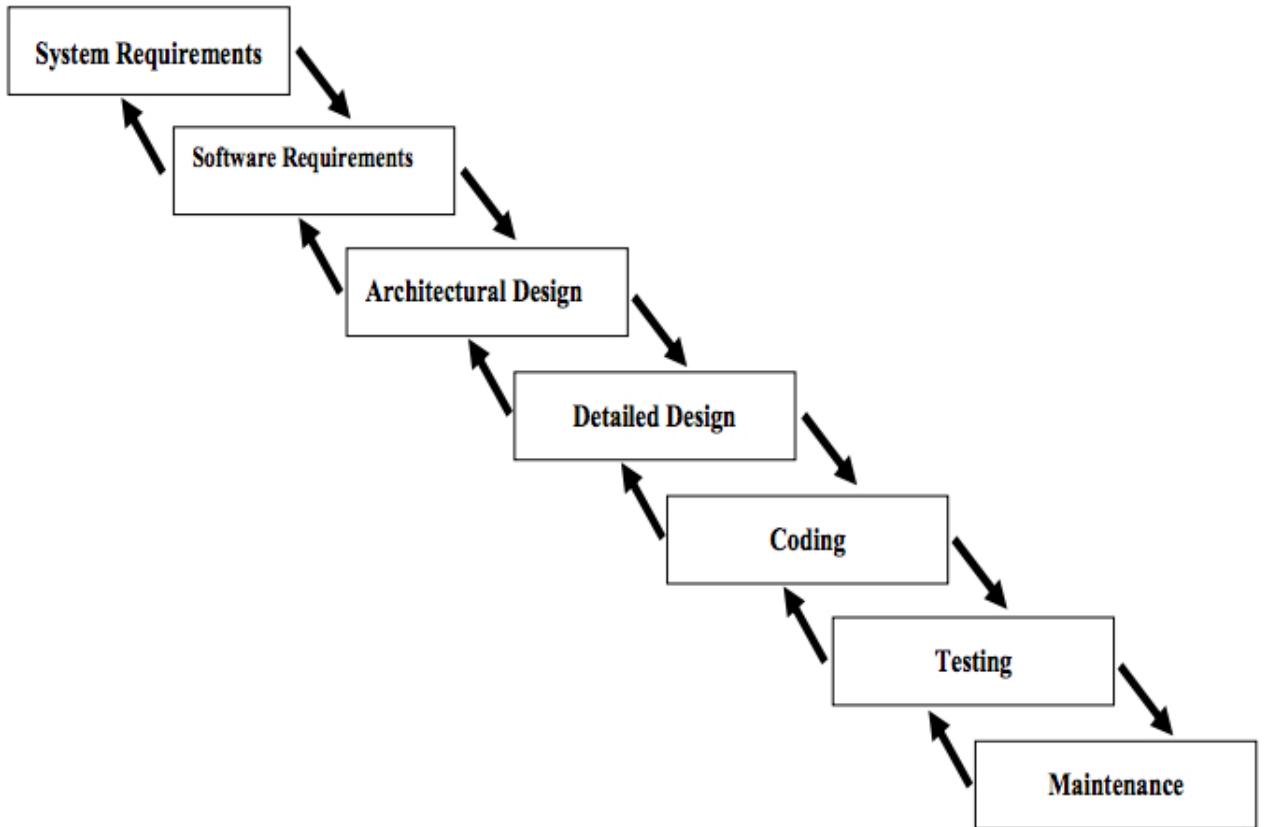


Figure 2.1: Waterfall model

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-system 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 focus 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.

Each column is 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).

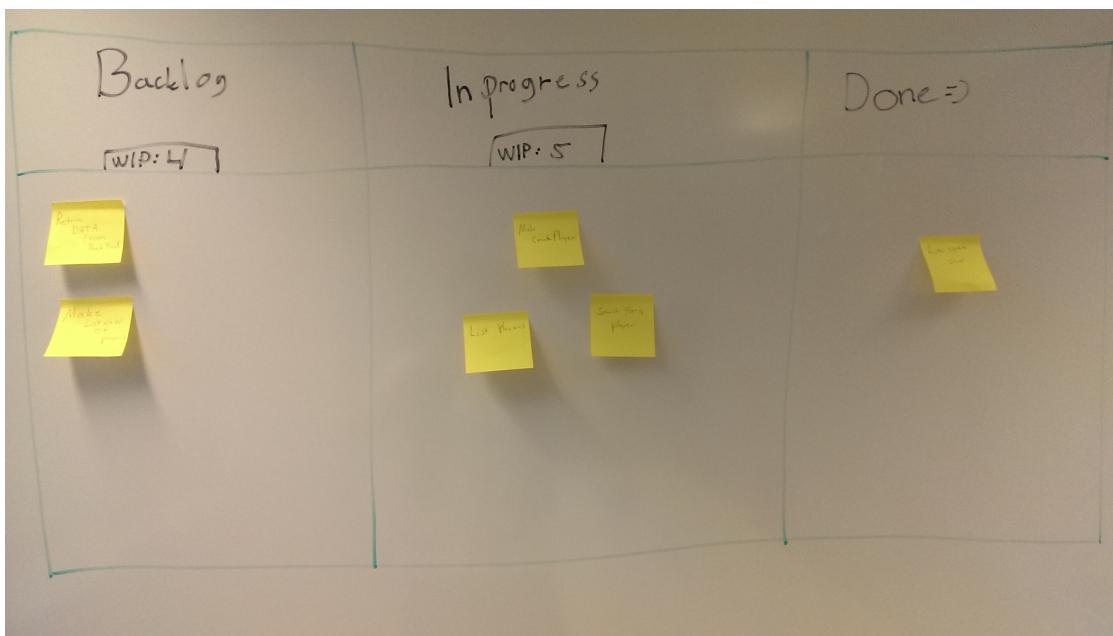


Figure 2.2: Example of a Kanban board

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 assertion 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 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 Swearning 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 importance of limit WIP has been stated by various researches. A summary of benefits with WIP limit is stated in Section 2.4.4 . 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) researched the difference between limit WIP and unlimited WIP. A summary of this two articles are stated in Section 2.4.3

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)

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

2.4.3 Limit WIP vs. Unlimited WIP

In the article by Giulio Concas and Hongyu Zhang, they compared the ability between WIP-limit and unlimited WIP by simulation a software maintenance process. The paper 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 showed that 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 (Concas et al., 2013).

The case by David Anderson et al. (D. Anderson et al., 2011) did a simulation of the impact of WIP limit vs. no WIP limit on developers with skills in different activities. The four skill activities from the article where 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 seldom 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).

2.4.4 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 increases productivity (Middleton and Joyce, 2012).
4. Limit WIP helps team to reduce overhead (Mark L. Spearman and J.Hopp, 1990).
5. Limit WIP decrease lead time (Mark L. Spearman and J.Hopp, 1990).
6. Limit WIP increase throughput (Mark L. Spearman and J.Hopp, 1990).
7. Limit WIP reduces flow times (Mark L. Spearman and J.Hopp, 1990).

8. Limit WIP reduces variation (Mark L. Spearman and J.Hopp, 1990).
9. Limit WIP improves quality (Mark L. Spearman and J.Hopp, 1990).

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 disregarded, 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 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 Talarai, 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.

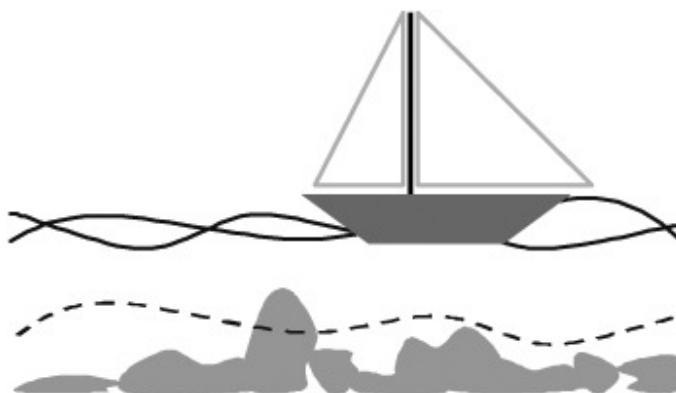


Figure 2.3: JIT example

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.

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 that's 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

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 2.2 shows the size of the ten teams vs. quarter. Team four shown in table 2.1(d) is the it-support team for SI.

(a) Team one			(b) Team two			(c) Team three			(d) Team four		
Year	Quarter	Team Size	Year	Quarter	Team Size	Year	Quarter	Team Size	Year	Quarter	Team Size
2010	3	6	2010	3	10	2010	3	6	2010	3	3
2010	4	3	2010	4	15	2010	4	9	2010	4	8
2011	1	16	2011	1	13	2011	1	7	2011	1	4
2011	2	28	2011	2	12	2011	2	10	2011	2	4
2011	3	2	2011	3	15	2011	3	9	2011	3	4
2011	4	38	2011	4	14	2011	4	10	2011	4	4
2012	1	35	2012	1	15	2012	1	11	2012	1	4
2012	2	34	2012	2	7	2012	2	11	2012	2	2
2012	3	32	2012	3	8	2012	3	13	2012	3	3
2012	4	29	2012	4	9	2012	4	13	2012	4	5
2013	1	24	2013	1	10	2013	1	13	2013	1	7
2013	2	37	2013	2	7	2013	2	7	2013	2	5
2013	3	23	2013	3	7	2013	3	8	2013	3	5
2013	4	23	2013	4	8	2013	4	8	2013	4	5
Total		330	Total		150	Total		135	Total		63

(e) Team five			(f) Team six			(g) Team Seven			(h) Team eighth		
Year	Quarter	Team Size	Year	Quarter	Team Size	Year	Quarter	Team Size	Year	Quarter	Team Size
2010	3	5	2010	3	5	2010	3	10	2010	4	2
2010	4	13	2010	4	6	2010	4	8	2011	1	8
2011	1	14	2011	1	6	2011	1	8	2011	2	8
2011	2	25	2011	2	6	2011	2	6	2011	3	13
2011	3	21	2011	3	5	2011	3	8	2011	4	9
2011	4	23	2011	4	5	2011	4	9	2012	1	10
2012	1	25	2012	1	4	2012	1	10	2012	2	2
2012	2	19	2012	2	6	2012	2	5	2012	3	25
2012	3	24	2012	3	6	2012	3	9	2012	4	11
2012	4	18	2012	4	9	2012	4	3	2013	1	22
2013	1	31	2013	1	9	2013	2	9	2013	2	21
2013	2	29	2013	3	9	2013	3	9	2013	3	23
2013	3	27	2013	4	11	2013	4	14	2013	4	8
Total		285	Total		99	Total		76	Total		162

(i) Team nine			(j) Team ten		
Year	Quarter	Team Size	Year	Quarter	Team Size
2010	4	5	2010	3	3
2011	1	8	2010	4	11
2011	2	7	2011	1	12
2011	3	7	2011	2	9
2011	4	9	2011	3	4
2012	1	10	2011	4	17
2012	2	8	2012	1	20
2012	3	10	2012	2	17
2012	4	12	2012	3	18
2013	1	8	2012	4	13
2013	2	9	2013	1	17
2013	3	8	2013	2	9
2013	4	8	2013	3	10
Total		109	Total		170

Table 2.2: 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 from Software Innovation where chosen for this master thesis will be stated. 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)(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 shown there is no clear rule what exactly a case study 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. In order to do so, the data set from SI with quantitative data was analyzed.

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 an 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 from 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 data from SI was analyzed on team level. The data from SI was analyzed using the program, which will compute 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

The **Date from** column consist of dates form when tasks where created. 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.

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. These two variables are used as an indicator of the workload together with churn.

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).

Chapter 4

Data collected and calculations

This chapter will introduce how the algorithms of the program works. The first 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) and lead time (Section 4.3.3). In the last section, a short introduction to the statistical analyze program SPSS is given (Section 4.1). For this thesis a program was made in order to measure the variables shown in Table 3.3.

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 WIP this programs 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 ArrayList¹ based on the team variable as shown in listing 4.2. 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

¹ArrayList 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).

```
1 While inputFile != EOF // EOF = End Of file
2     WIP = New WIP()
3     WIP.start = inputFile.start
4     WIP.end = inputFile.end
5     WIP.team = inputFile.team
6     WIP.processType = inputFile.processType
7     WIP.WIP = 1
8     FindTeam(WIP)
9
```

Listing 4.1: Gather all unique dates into ArrayList

```
1 void FindTeam (WIP w)
2     if w.team EQUALS "TeamOne"
3         TeamOne.add(w)
4     if w.team EQUALS "TeamTwo"
5         TeamTwo.add(w)
6     if w.team EQUALS "TeamThree"
7         TeamThree.add(w)
8 /* And so on for the rest if the seven teams */
9
```

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 shows. 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 six five to compute the current WIP. The if test in line four assures that there is only measured one WIP for each start date.

```

1 void measureWIP()
2     lastWIP = 0
3     for WIP Object IN ArrayList
4         if(DateNotMeasured(WIP.getStartDate()) == true)
5             WIP_for_this_date = get_current_WIP(WIP.getStartDate())
6             WIP_measured = WIP_for_this_date - Nr_of_finishedDates(WIP.getStartDate)
7             ) + lastWIP
8             WIP.setWIP(WIP_measured)
9             lastWIP = WIP_measured
10
11 int get_current_WIP(Date date)
12     current_WIP = 0
13     for WIP in ArrayList
14         if date EQUALS WIP.getStartDate()
15             Nr_of_dates_to_decrement++
16     return current_WIP
17
18 int Nr_of_finished_dates(Date date)
19     Nr_of_dates_to_decrement = 0
20     for WIP in ArrayList
21         if date AFTER WIP.getEndDate() DO
22             if date not picked
23                 Nr_of_dates_to_decrement++
24                 dateIsPicked(WIP)
25     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 2008-10-12, tasks 3, 5 and 6 are in progress, which means the WIP is 3 for 2008-10-12.

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

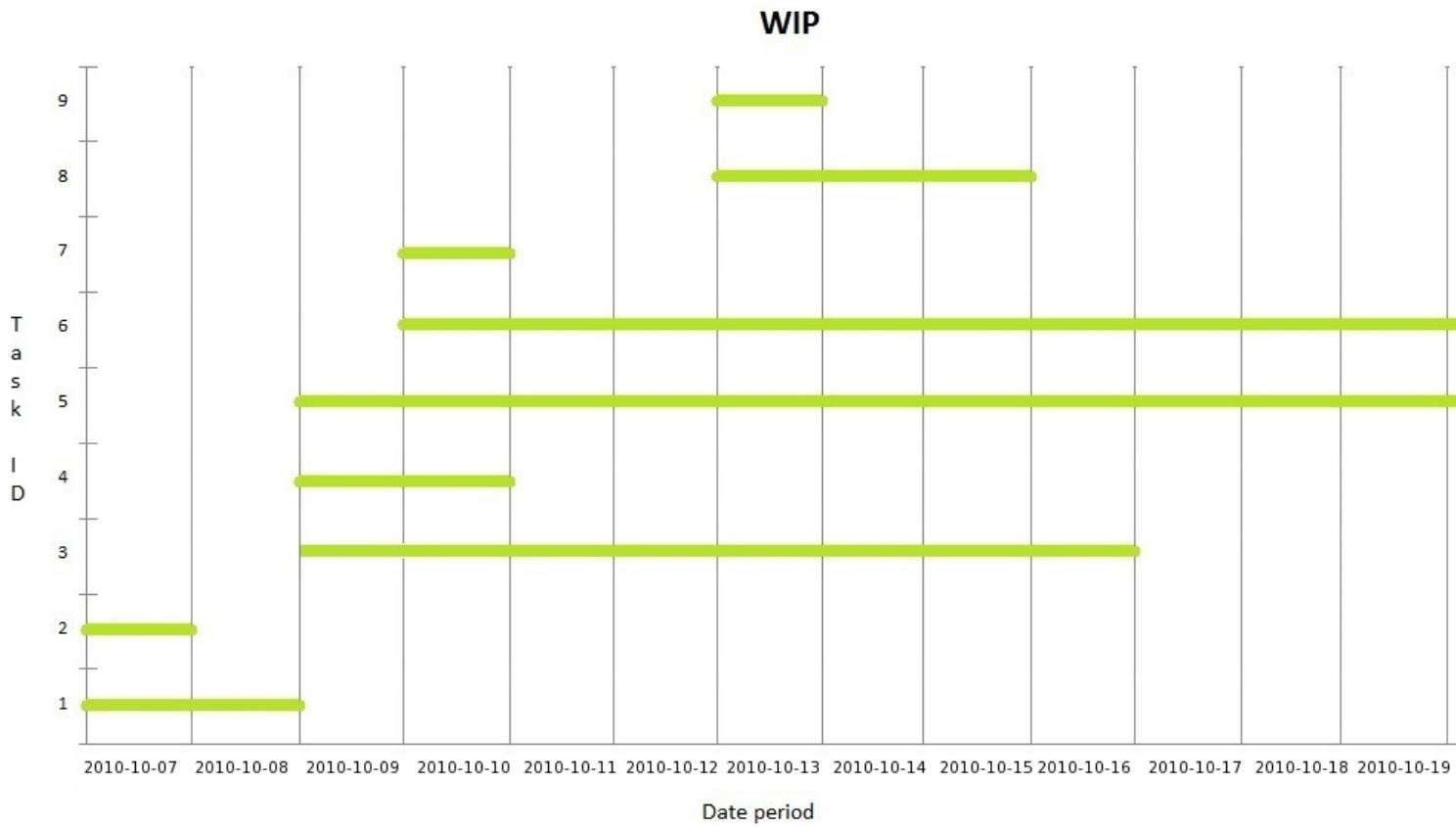


Figure 4.1: Illustrating the WIP timeline for example stated in section 4.2.4

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.

```
1 WIP = new WIP()
2 WIP.start = 2010-10-07
3 WIP.end = 2010-10-08
4 WIP.team = "Team One"
5 WIP.processType = "Kanban"
6 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 exists and will do so for all dates until the date 2010-10-13.

```
1 void createNewWIP(Date d, String team)
2     WIP.start = d
3     WIP.end = d
4     WIP.team = team
5     WIP.processType = "Unknown"
6     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 day 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 is almost identical. First the program read 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. In the following subsections the algorithm for each of the variables is presented. In subsection 4.3.1 the throughput algorithm is stated, in subsection 4.3.2 churn is stated, in subsection 4.3.3 lead time is stated and in subsection 4.3.5 a description of how bug and feature is measured

```
1 void addBug(Bug b)
2     if b.team EQUALS "TeamOne"
3         if dateExists(b.date, TeamOne) EQUALS false
4             // if date don't exists, then add the bug
5             TeamOne.add(b)
6
7     if b.team EQUALS "TeamTwo"
8         if dateExists(b.date, TeamTwo) EQUALS false
9             // if date don't exists, then add the bug
10            TeamTwo.add(b)
11
12    if b.team EQUALS "TeamThree"
13        if dateExists(b.date, TeamThree) EQUALS false
14            // if date don't exists, then add the bug
15            TeamThree.add(b)
16
17    if b.team EQUALS "TeamFour"
18        if dateExists(b.date, TeamFour) EQUALS false
19            // if date don't exists, then add the bug
20            TeamFour.add(b)
21
22 void dateExists(Date d, ArrayList list)
23     for Bug b in list
24         if b.date EQUALS d
25             b.counter++
26             return true
27
28 return false
29
```

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.

```
1 void dateExists(Throughput tp d, ArrayList list)
2   for Throughput t in list
3     if t.date EQUALS tp.date
4       t.counter++
5       return
6
7 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 algorithm is shown in listing 4.9

```
1 void updateChurn(Churn c, ArrayList list)
2   for Churn ch in list
3     if ch.date EQUALS c.date
4       ch.churn += c.churn
5       return
6   structure.add(c);
```

Listing 4.9: Pseudocode example of how throughput is measured

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

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 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

```

1
2 addLeadTime(lead_time t, ArrayList list)
3   for lead_time in list
4     if lead_time.date EQUALS t.date
5       lead_time.lead_time+= t.lead_time
6       return
7
8 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 and in this thesis churn is used as surrogate effort to know the complexity of the task. 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

In order to measure feature and bug the program and SPSS where 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. 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.

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 recorded the program and SPSS where used. The program extracted the created date and date to values and checks if their quarters match. If they do, a boolean value is set to true, otherwise its set to false. The output is used in SPSS where the boolean value is grouped by 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 values. 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 results are presented with five correlation tables and five corresponding descriptive statistics tables. The content of the sections will consist of highlighting the variables with a significant correlation and variables that stands out. To back up any assumptions about the variables, descriptive statistic tables listed in Appendix A and graphs will be used . The correlation relationship between two variables will occur twice, in both the correlation tables, since correlation is symmetric. When the relationship is highlighted in one of the sections, the relationship will be omitted in the other section.

5.1 Correlation - WIP

Table 5.1 shows the correlation table for WIP. Vertically in the correlation table, the teams variables are stated. Horizontally is the corresponding teams stated. The team names are shortened. Team one is shortened to T1, team two is shortened to T2 and so on.

The 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, its natural that these variables also have a significant positive correlation to WIP. But, for team one, six, seven and ten that's not the case.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Throughput	.74**	0.21	.76**	.83**	0.52	.64*	.67*	0.47	.89**	.61*
Throughput Feature	.73**	-0.14	.83**	.82**	0.25	.68**	.63*	0.40	.82**	0.20
Throughput bug	0.02	0.25	.73**	.82**	.54*	0.07	0.55	0.15	.88**	.63*
Bugs	.73**	0.20	.60*	.56*	0.50	0.46	-0.07	0.04	.58*	0.18
Bugs finished, quarter	0.35	0.10	-0.07	.56*	0.19	0.19	0.50	0.23	0.52	0.35
Avg days in backlog, bugs	-0.03	0.44	0.42	.54*	-0.18	0.02	-0.64	0.14	-0.20	-0.18
Lead time	-0.21	0.23	-0.13	.69**	0.10	-0.18	0.39	0.11	-0.35	0.33
Churn	-0.18	-0.43	-0.47	.c	-0.24	-0.04	0.23	-0.42	-0.03	0.05
Churn feature	0.38	-0.39	-0.35	.c	-0.03	0.02	0.10	-0.26	-0.07	0.25
Churn bug	0.05	-0.04	-.57*	.c	-0.22	.80**	0.07	-0.50	-0.06	-0.09

Table 5.1: Correlation - WIP

* 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.

Both throughput and throughput feature in team one's case have significant positive relationship with WIP, throughput bug don't. Throughput and throughput feature have the correlation values .74 and .73 while throughput bug's value is 0.02. The reason throughput bug do not have a close relationship to throughput might be because throughput bug consist 37% (108/290) of throughput dates as shown in the total row in tables A.0(b) and A.1(b). But it's possible to have a close relationship even though, since the correlation is based on the mean values. But throughput feature has a total mean of 13.7, while throughput bug have 6.4 as shown in tables A.1(a) and A.1(b). The total mean of throughput is 11 as shown in table A.0(b). The mean values strengthens the assumption that throughput feature represents most of the throughput variable. The correlation graphs in Figure 5.1 and the throughput correlation table in Section 5.4 states the assumption. In the correlation graphs, the X-axis represents WIP and the Y-axis represents the different throughput variables. The pattern of the dots in the Figures 5.1(a) and 5.1(b) are similar. They both increase when X values do. While the dots in Figure 5.1(c) are more randomly gathered, which reflects the correlation values shown in Table 5.7 throughput and both throughput bug (0.03) and throughput feature (0.96).

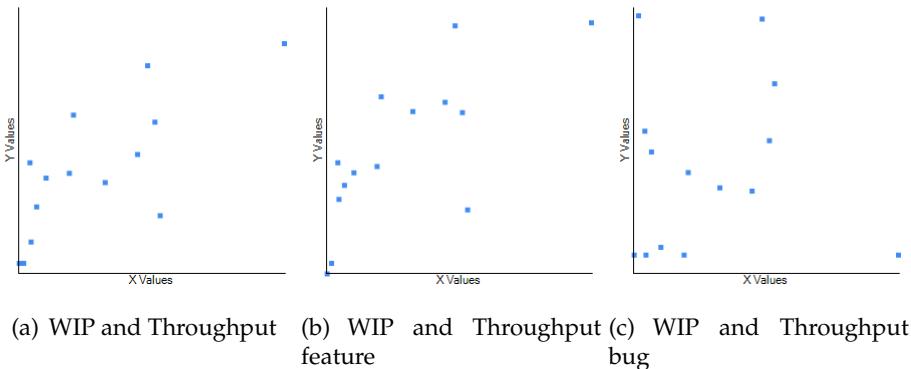


Figure 5.1: Correlation graphs between WIP and the throughput variables.

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 do not. Throughput, throughput feature and throughput bug have the correlation values .64, .68 and 0.07. The total row in tables A.25(b), A.26(a) and A.26(b) shows that throughput feature consist of 609 dates and a mean value of 4.8, while throughput bug consist of 82 dates and a mean of 3.3. The throughout variable consist of 691 dates and have a mean value of 4.58. The mean value and the number of dates strengthens the assumption that throughput feature represents more of throughput than throughput bug do. The assumption that total of dates and the total of mean can be used to indicate the correlation relationship is stated by the throughput correlation table in Section 5.4.

For team seven's case, the difference between throughput, throughput feature and throughput bug is so small, which also can be assumed by the total row in tables A.30(b), A.31(a) and A.31(b). In the total rows, throughput has a total mean of 2.7, while throughput feature have 2.8 and throughput bug have 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. Throughput for team ten consist of 404 tasks as show in the total row in Table A.30(b). Throughput bug represents 335 of these dates while throughput feature stands for 69 dates as shown in Tables A.31(a) and A.31(b). But the overall mean for throughput, throughput feature and throughput bug is 2.2, 2.3 and 2.2, which could reflect that these three variables have a close relationship. Both the throughput correlation table in Section 5.4 and the correlation graphs Figures in 5.2 disproves that assumption. The figure shows that the correlation dots between WIP and throughput and WIP and throughput bug is almost identical, which proves that throughput bug

have a close relationship to throughput while the correlation table shows that the correlation between throughput and throughput feature is .43 which is a medium strong correlation.

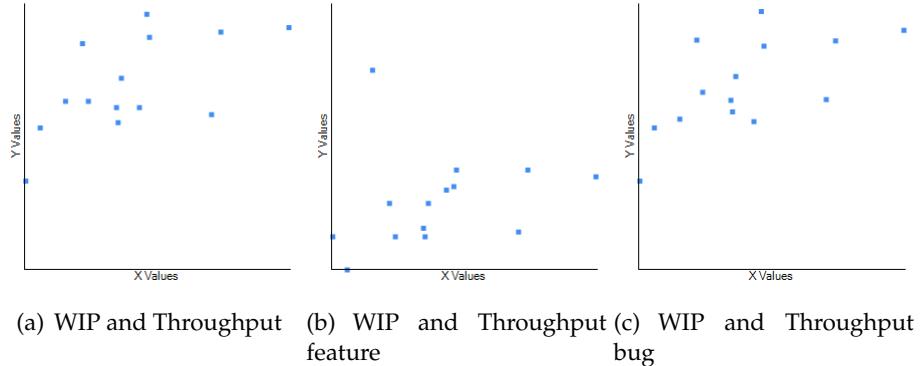


Figure 5.2: Correlation graphs between WIP and the throughput variables for team ten.

Team one, three, four and nine have a significant positive relationship between WIP and bugs as well. Team three and six have a significant positive relationship between WIP and churn bug. Why churn bug differ from churn for team six is stated in section 5.2. But for team three, churn bug do not differ much from churn which is shown in the correlation table for churn in Section 5.5. In WIP's correlation table has churn a correlation of -0.47, churn bug has -.57 and churn feature has -0.35. The tables A.12(b), A.13(a) and A.13(b) shows that churn has a total mean of 59.4, churn feature has 102.6 and churn bug 41.1. This shows that the churn feature variable has a small impact on the churn variable.

Team four is the it-support team for SI. IT-support do not have churn, since they don't code and that reflects why the churn variables is indicated with a .c in Table 5.1. However, the rest of the team four's variables have a significant positive relationship between WIP and the rest of the variables.

The descriptive statistics tables are based on summary of the correlation values. The tables shows number(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 positive correlation between WIP and throughput. The bug variables have a total mean of 0.4 with a median of 0.5, this indicates that WIP and bug have 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	0.4	0.4	0.3	0.9	0.0
Bugs	10	0.4	0.5	0.3	0.7	-0.1
Bugs finished, quarter	10	0.3	0.3	0.2	0.6	-0.1
Avg days backlog, bugs	10	0.0	-0.0	0.4	0.5	-0.6
Lead time	10	0.1	0.1	0.3	0.7	-0.3
Churn	10	-0.2	-0.1	0.2	0.2	-0.5
Churn ft	10	-0.0	-0.0	0.2	0.4	-0.4
Churn bug	10	-0.1	-0.1	0.4	0.8	-0.6

Table 5.2: Descriptive Statistic - Correlation - WIP

5.2 Correlation - Lead time

This section contains the correlation Table 5.3. The table shows the correlation between lead time and the variables. Team two, six and ten have significant negative relationship between lead time and churn bug. The churn and churn bug variables for team two have the correlation values of -0.45 and -.73. These values are relative close. However, churn feature have a value of 0.41. The total row in tables A.7(b), A.8(a) and A.8(b) shows that churn for team two consist of 1572 dates. Churn feature consist of 375 dates and churn bug of 1197 dates. Churn bug contribute 76% (1197/1572) of the total amount of dates in churn. The total mean of churn, churn feature and churn bug is 65.7, 111.4 and 51.3, which could reflect that churn bug contribute more to the churn variable than churn feature. The churn correlation table in Section 5.5 with the correlation value of .08 for churn feature and .728 for churn bug, states that churn bug represents the most of the churn variable. Team two also has significant negative relationship between lead time and the bugs finished, quarter variable.

Churn bug for team six have a negative correlation of -.55 while churn have a correlation of -0.08 and churn feature a correlation of -0.2. The reason why churn bug differ from churn is because churn bug consist of 17% (265/1533) of the churn dates as shown in tables A.27(b) and A.28(b). The two tables and Table A.28(a) also shows the total mean of churn is 146, while the total mean for churn feature is 156.5 and for churn bug it's 95.2. With the total mean, one can assume that churn bug don't contribute to much to the churn variable, hence big difference between the two mean values. The assumption is proved by the churn correlation table in Section 5.5 with the values .980 for churn feature and .018 for churn bug. Throughput bug also differ from throughput for team six. The reason why where explained in section 5.1.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
WIP	-0.21	0.23	-0.13	.69**	0.10	-0.18	0.39	0.11	-0.35	0.33
Throughput	-0.44	-0.22	-0.39	0.49	-0.19	-.55*	-0.03	0.05	-0.29	-0.13
Throughput Feature	-0.37	0.22	-0.38	0.47	-0.22	-.54*	0.05	-0.18	-0.19	-0.29
Throughput bug	-0.45	-0.31	-0.33	0.47	-0.14	-0.26	-0.58	-0.23	-0.37	-0.08
Bugs	-0.23	-0.23	-0.24	0.25	-0.25	-0.22	-0.01	-0.12	-0.06	-0.14
Bugs finished, quarter	0.38	-.55*	0.02	0.48	-0.47	-0.36	-0.49	0.28	-0.14	-0.36
Avg days in backlog, bugs	-0.27	0.44	-0.15	0.13	0.07	0.01	0.12	0.11	-0.02	-0.42
Churn	0.41	-0.45	-0.13	.c	0.24	-0.08	-0.07	-0.28	-0.32	-.68**
Churn feature	0.27	0.41	0.15	.c	-0.22	-0.20	0.04	-0.02	-0.32	0.31
Churn bug	0.28	-.73**	-0.12	.c	0.25	-.55*	-0.34	-0.00	-0.21	-.73**

Table 5.3: Correlation - Lead time

* 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 is not the only churn variable that has a significant negative relationship between lead time for team ten, churn do as well, but churn feature don't. The Tables A.47(b), A.48(a) and A.48(b) shows the total mean of churn is 50.4, the total of churn feature its 66 and for churn bug its 47.7. This shows that churn and churn bug total mean is close, but the churn feature is not so far away. Churn bug also consist of 85% of the overall dates in churn (517/609) which supports the assumption that churn bug consist of much of the churn variable. The assumption is proven right by the churn correlation table in 5.5 and correlation graphs in Figure 5.3. The figure shows the lead time in the X-axis and the churn variable in the Y-axis. One can clearly see that Figures 5.3(a) and 5.3(c) have a significant negative correlation with lead time. The dot pattern in these two figures is also similar, which points to the case that churn and churn bug are alike and that churn feature don't represents of much of the churn nor have a close relationship to churn.

The descriptive statistic table for lead time shown in Table 5.4 shows that there is no value with overall significant relationship with lead time.

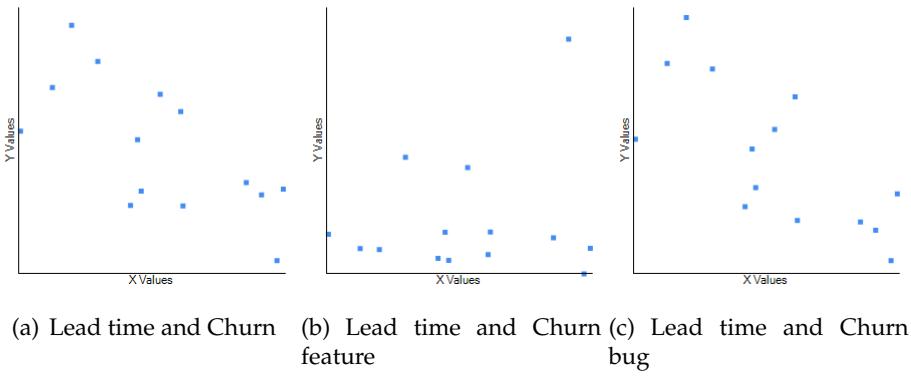


Figure 5.3: Correlation graphs between lead time and churn variables for team ten.

	N	Mean	Median	Std.Dev	Max	Min
WIP	10.0	0.1	0.1	0.3	0.7	-0.3
Throughput	10.0	-0.2	-0.2	0.3	0.5	-0.6
Throughput ft	10.0	-0.1	-0.2	0.3	0.5	-0.5
Throughput bug	10.0	-0.2	-0.3	0.3	0.5	-0.6
Bugs	10.0	-0.1	-0.2	0.2	0.2	-0.2
Bugs finished, quarter	10.0	-0.1	-0.2	0.4	0.5	-0.6
Avg days backlog, bugs	10.0	0.0	0.0	0.2	0.4	-0.4
Churn	10.0	-0.1	-0.1	0.3	0.4	-0.7
Churn ft	10.0	0.0	0.0	0.2	0.4	-0.3
Churn bug	10.0	-0.2	-0.2	0.4	0.3	-0.7

Table 5.4: Descriptive Statistic - Correlation - Lead time

5.3 Correlation - Bugs

This section contains information about the correlation table between the variables and bugs. The result is listed in Table 5.5. Teams one, two, three, four, five, nine and ten all have a significant positive correlation between bugs and throughput. In team three, four, five and nine's case, have all throughput variable a significant positive correlation with bugs. For team one and ten, the throughput relationship differ. The reason why where stated in Section 5.1.

Team two's Throughput feature differ from throughput. One could believe that the reason is that throughput bug consists of 2/3 of the throughput variable (460/690) as shown in the descriptive statistic Tables A.5(b) and A.6(b). But the two tables and table A.6(a) 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

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
WIP	.725**	0.20	.599*	.560*	0.50	0.46	0.62	0.04	.581*	0.18
Throughput	.687**	.810**	.880**	.586*	.975**	0.27	0.53	0.41	.699**	.557*
Throughput Feature	.738**	0.01	.822**	.583*	.881**	0.30	0.56	0.22	.597*	-0.14
Throughput bug	-0.17	.834**	.868**	.583*	.959**	.686**	0.50	.915**	.653*	.589*
Bugs finished, quarter	0.50	-0.18	0.12	0.40	0.17	.762**	.785**	0.18	.701**	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
Leadtime	-0.23	-0.23	-0.24	0.25	-0.25	-0.22	0.54	-0.12	-0.06	-0.14
Churn	0.23	-0.01	0.00	.c	-0.17	-0.10	0.19	-0.08	-0.34	0.31
Churn feature	0.30	0.07	-0.03	.c	0.28	-0.04	0.13	-0.15	-.618*	0.06
Churn bug	0.37	0.17	-0.08	.c	-0.19	0.43	0.14	0.04	0.10	0.28
Team size	.803**	0.26	0.27	0.17	.711**	0.41	0.41	0.42	0.41	0.16

Table 5.5: Correlation - Bugs

* 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.

reflect that these three variables should 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 Figure 5.4 shows the bugs variable in the X-axis and the throughput variable in the Y-axis. Both Figures 5.4(a) and 5.4(c) shows that when x increases, y also increases. But for Figure 5.4(b) one can see that the dots are more randomly placed. For example, when the Y value is at vertex, x value is low. When x is at vertex, y values are both low and high.

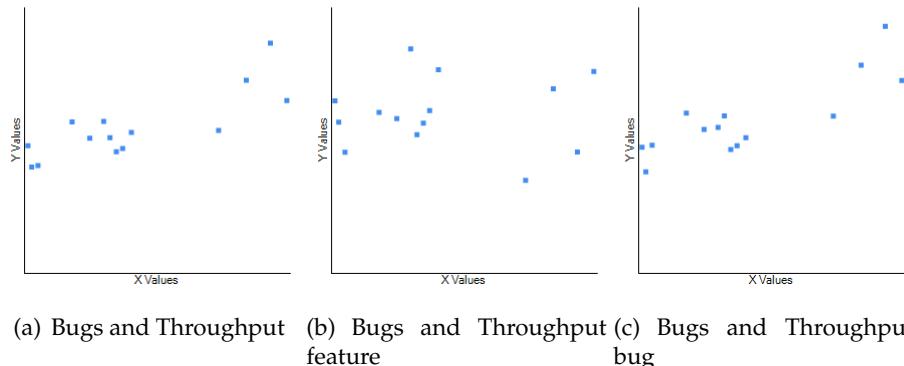


Figure 5.4: Correlation graphs between bugs and the throughput variables.

Team six and nine have a significant positive correlation between the variable Bugs finished, quarter. Team nine also has a significant negative correlation between bug and churn feature, but not churn. Churn feature consist of approximately 1/3 (295/923) of the churn variable as shown in Tables A.42(b) and A.43(a). The total mean of churn feature is 133.9, the total mean for churn is 81.9 and the total mean for churn bug is 57.5, as shown in the previous stated tables and Table A.48(b). The correlation graphs in Figure 5.5, show the bugs variable in the X-axis and churn variable in the Y-axis. The figure shows that there is neither a significant negative nor significant positive pattern between either bug and churn or bug and churn bug. The Figure 5.5(b) however, shows that there is a vaguely significant negative correlation between bug and churn. Even though churn feature have a significant correlation to a variable, and the churn do not, does not say that churn feature can't have a significant positive correlation to churn. The correlation table for churn in Section 5.5 shows that both churn feature and churn bug for team nine have a significant correlation with churn. Judging from the Figure 5.5 and the correlation table from Section 5.5 one can assume that correlation is not transitive. The paper "The Non-Transitivity of Pearson's Correlation Coefficient: An Educational Perspective" (Langford, Schwertman and Owens, 2001) proves the assumption.

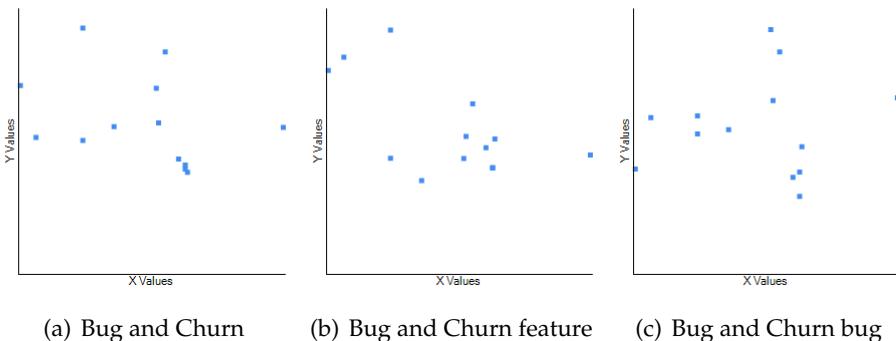


Figure 5.5: Correlation graphs between bug and the churn variables for team nine.

In Table 5.6, one can see that WIP with a mean value of 0.4 have a medium correlation with bugs as stated in Section 5.1. Throughput have 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	0.3	0.7	-0.1
Throughput	10	0.6	0.6	0.3	1.0	0.1
Throughput ft	10	0.4	0.4	0.4	0.9	-0.1
Throughput bug	10	0.6	0.7	0.4	1.0	-0.2
Bugs finished, quarter	10	0.3	0.2	0.3	0.8	-0.2
Avg days backlog, bugs	10	0.2	0.3	0.3	0.5	-0.4
Lead time	10	-0.1	-0.2	0.2	0.2	-0.2
Churn	10	-0.0	-0.0	0.2	0.3	-0.3
Churn ft	10	-0.1	-0.0	0.3	0.3	-0.6
Churn bug	10	0.1	0.1	0.2	0.4	-0.2

Table 5.6: Descriptive Statistic - Correlation - Bugs

5.4 Correlation - Throughput

This section shows the correlation table between throughput and the variables. The result is listed in Table 5.7. 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, why is explained in section 5.1. Why throughput bug differ from throughput for team ten is also stated in section 5.1. Throughput feature differ from throughput for team two and the reason for that is stated in section 5.3.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
WIP	.74**	0.21	.76**	.82**	0.52	.64*	.67*	0.47	.89**	.61*
Throughput Feature	.96**	0.09	.93**	.1**	.85**	.1**	.91**	.94**	.88**	0.43
Throughput bug	0.03	.97**	.99**	.1*	.99**	0.04	.91**	0.44	.96**	.98**
Bugs	.69**	.81**	.88**	.59*	.98**	0.27	0.07	0.41	.70**	.56*
Bugs finished, quarter	0.16	0.12	0.23	0.39	0.12	0.12	0.03	0.33	.70**	.60*
Avg days in backlog, bugs	0.16	0.12	0.45	0.37	0.14	-0.17	-0.17	-0.17	-0.41	-0.09
Lead time	-0.44	-0.22	-0.39	0.49	-0.19	-0.55*	-0.03	0.05	-0.29	-0.13
Churn	-0.27	-0.23	-0.26	.c	-0.20	-0.27	.76*	-0.64*	-0.01	0.26
Churn feature	0.11	-0.18	-0.25	.c	0.22	-0.23	0.54	-0.45	-0.27	-0.25
Churn bug	-0.11	0.06	-0.32	.c	-0.21	.59*	.69*	-0.28	0.04	0.21

Table 5.7: Correlation - Throughput

* 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.

Both team nine and ten have a positive significant correlation between bug and the Bugs finished, quarter variable. Team six have a significant correlation between bugs and churn bug. Why churn bug differ from churn and churn feature is stated in Section 5.2. Team seven has a significant positive correlation between throughput and the variables churn and churn bug, but not churn feature. Churn feature has almost a significant positive correlation with throughput. In the correlation table for churn in section 5.5, churn feature and churn bug have a positive correlation with churn. This is another example of transitivity and correlation as stated in Section 5.3.

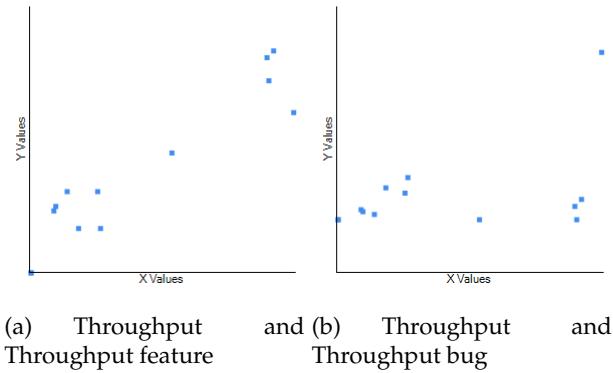


Figure 5.6: Correlation graphs between the throughput variables.

For team eight, throughput bug differ from throughput. Throughput bug consist of 99 dates and have total mean of 1.5. Throughput feature consist of 92 dates and a mean of 3.2. While throughput have a mean of 2.3. These numbers are gathered from the descriptive statistic Tables A.35(b), A.36(a) and A.36(b). On the basis 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. But, in Figure 5.6 the difference are stated. The Figure 5.6 shows the throughput variable in the X-axis and the sub variables in the Y-axis. In Figure 5.6(a) the dots are clearly moving in a upwards direction, while in Figure 5.6(b) almost all the dots are all gathered around the low values of Y.

Team eight also has a significant negative correlation between throughput and churn. Churn has a negative value of -.64, while churn feature has a value of -0.45 and churn bug has a value of -0.28. Churn feature consist of 246 of 329 tasks, as shown in Tables A.37(b) and A.38(a). The Table A.38(a) also shows that churn feature mostly consist of tasks that requires zero churn. The total mean of these three values are 0.9 for churn, 0.1 for churn feature and 3.6 for churn bug as shown in the Tables A.37(b), A.38(a) and A.38(b). The zeros from churn feature play a part in the mean value of churn. But, in the correlation for churn in section 5.5, one can see that churn bug have a significant correlation with churn, while churn feature don't. So the contribution of the churn feature's mean in the case of correlation between throughput and churn made enough impact for churn to have a significant correlation with throughput.

According to Table 5.12, bugs have a strong positive correlation with throughput as stated Section 5.3. WIP have a medium positive relationship with throughput which where stated in 5.1. The sub variables throughput feature (0.6) and throughput bug (0.4) have respectively a strong and medium relationship to throughput.

	N	Mean	Median	Std.Dev	Max	Min
WIP	10.0	0.4	0.5	0.3	0.7	-0.1
Throughput ft	10.0	0.6	0.6	0.3	1.0	0.1
Throughput bug	10.0	0.4	0.4	0.4	0.9	-0.1
Bugs	10.0	0.6	0.7	0.4	1.0	-0.2
Bugs finished, quarter	10.0	0.3	0.2	0.3	0.8	-0.2
Avg days backlog, bugs	10.0	0.2	0.3	0.3	0.5	-0.4
Lead time	10.0	-0.1	-0.2	0.2	0.2	-0.2
Churn	10.0	-0.0	-0.0	0.2	0.3	-0.3
Churn ft	10.0	-0.1	-0.0	0.3	0.3	-0.6
Churn bug	10.0	0.1	0.1	0.2	0.4	-0.2

Table 5.8: Descriptive Statistic - Correlation - Throughput

5.5 Correlation - Churn

This section contains information about the correlation table between the variables and churn. The result is listed in Table 5.9. All teams with valid churn have either one or both churn variable with significant positive correlation with churn. Teams two, five, six eight and ten doesn't have a positive correlation between both the churn variables and churn.

For team two and ten's case, churn feature differ from churn, the reason why is stated in Section 5.2. The reason churn bug differ in the case of team six is also stated in Section 5.2. Team five on the other hand, has a significant positive correlation between churn and churn bug, but not for churn feature. Churn feature consist of approximately 10% of the total churn value as shown in Tables A.22(b) and A.23(a). The two previous stated tables and Table A.23(b) shows that churn has a mean value of 23.5, churn feature has a mean value of 43.1 and churn bug with a mean value of 21.4. These values show that churn feature don't represent much of the churn variable.

Team eight has a significant positive correlation between churn and churn bug and almost churn feature. Churn feature consist of approximately 75% (246/329) of all churn dates as shown in Tables A.37(b) and A.38(a). But as stated in in 5.4, almost each of the churn feature task requires zero churn, and with zero churn, its hard to make an impact on churn.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
WIP	-0.18	-0.43	-0.47	.c	-0.24	-0.04	0.23	-0.42	-0.03	0.05
Throughput	-0.27	-0.23	-0.26	.c	-0.20	-0.27	.76*	-.635*	-0.01	0.26
Throughput Feature	-0.24	-0.09	-0.28	.c	-0.10	-0.22	.633*	-.678*	0.20	0.05
Throughput bug	-.590*	-0.17	-0.28	.c	-0.21	-0.15	.866**	-0.10	-0.09	0.27
Bugs	0.23	-0.01	0.00	.c	-0.17	-0.10	-0.21	-0.08	-0.34	0.31
Bugs finished, quarter	0.48	0.15	-0.14	.c	-0.50	0.04	-0.53	0.06	-0.01	0.43
Avg days in backlog, bugs	0.36	-0.19	-0.01	.c	0.39	0.53	0.52	0.26	-0.15	0.04
Lead time	0.41	-0.45	-0.13	.c	0.24	-0.08	-0.07	-0.28	-0.32	-.681**
Churn feature	.570*	0.38	.790**	.c	0.27	.980**	.879**	0.55	.639*	-0.03
Churn bug	.828**	.728**	.955**	.c	.983**	0.18	.750*	.817**	.599*	.969**

Table 5.9: 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.

Table 5.10 shows both churn feature and churn have a close relationship with churn

	N	Mean	Median	Std.Dev	Max	Min
WIP	10.0	-0.2	-0.1	0.2	0.2	-0.5
Throughput	10.0	-0.1	-0.2	0.4	0.8	-0.6
Throughput ft	10.0	-0.1	-0.1	0.3	0.6	-0.7
Throughput bug	10.0	-0.0	-0.1	0.4	0.9	-0.6
Bugs	10.0	-0.0	-0.0	0.2	0.3	-0.3
Bugs finished, quarter	10.0	-0.0	0.0	0.3	0.5	-0.5
Avg days backlog, bugs	10.0	0.2	0.1	0.3	0.5	-0.2
Lead time	10.0	-0.1	-0.1	0.3	0.4	-0.7
Churn ft	10.0	0.5	0.6	0.3	1.0	-0.0
Churn bug	10.0	0.7	0.8	0.3	1.0	0

Table 5.10: Descriptive Statistic - Correlation - Churn

5.6 Team size

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
WIP	.684**	0.35	.778**	0.06	.575*	.767**	0.62	.651*	0.54	.759**
Throughput	.699**	0.05	0.52	0.16	.687**	.858**	0.62	.753**	0.53	.573*
Throughput Feature	.736**	-0.22	0.53	0.20	0.48	.893**	0.47	.745**	0.48	0.18
Throughput bug	-0.10	0.06	0.51	0.20	.673**	.001	.715*	0.40	0.48	.637*
Bugs	.803**	0.26	0.27	0.17	.711**	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	.711*	0.05	0.38	0.34
Avg days in backlog, bugs	0.48	.841**	0.04	0.44	0.03	-0.03	0.49	0.03	0.07	-0.03
Leadtime	-.542*	0.48	-0.01	-0.37	-0.33	-0.46	0.12	0	0.01	0.09
Churn	-0.06	-0.45	-.599*	.c	-.627*	0.01	0.51	-.635*	0.04	0.04
Churn feature	0.01	0.14	-0.41	.c	0.23	0.06	0.60	-0.41	-0.01	0.32
Churn bug	0.25	-0.41	-.593*	.c	-.646*	.758**	0.05	-.592*	-0.07	-0.07

Table 5.11: Correlation - Leadtime

* 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.

	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
Churn	10	-0.1	-0	0.3	0.5	-0.5
Churn ft	10	-0.2	-0	0.4	0.5	-0.6
Churn bug	10	0.1	0	0.3	0.6	-0.4

Table 5.12: Descriptive Statistic - Correlation - Throughput

Chapter 6

Discussion

6.1 swag

The research on kanban says you should limit WIP to increase throughput and decrease lead time. According to the Table 5.2, throughput increases when WIP increases for in average for all the teams. Six out of ten teams have a significant positive correlation between WIP and throughput. One could argue, when throughput decreases, bug also decreases according to Table 5.12. That may not be the case. When one produces more tasks, more bugs are produced. The Table 5.2 shows that WIP and bugs have a medium relationship. Out of the six teams with a positive correlation between WIP and throughput, four has a significant positive correlation between WIP and bugs. The correlation values are not transitive as said in Section 5.3, but that don't mean that there is no relationship between the WIP correlation and the throughput correlation.

Appendices

Appendix A

Descriptive statistics for the ten teams

A.1 Team 1 - Descriptive Statistics

(a) Descriptive Statistic -WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	25.0	3.6	4.0	0.6	5.0	3.0
2010-4	92.0	0.7	1.0	0.7	3.0	0
2011-1	90.0	3.4	1.0	6.9	30.0	0
2011-2	91.0	13.2	4.0	14.5	51.0	2.0
2011-3	92.0	1.8	2.0	0.6	3.0	1.0
2011-4	92.0	14.3	4.0	22.7	97.0	1.0
2012-1	91.0	22.2	21.0	14.5	67.0	4.0
2012-2	91.0	30.3	23.0	29.0	107.0	9.0
2012-3	92.0	36.0	38.5	13.6	65.0	18.0
2012-4	92.0	34.7	28.5	16.9	99.0	25.0
2013-1	90.0	32.8	25.0	13.7	85.0	25.0
2013-2	91.0	67.1	54.0	44.3	178.0	3.0
2013-3	92.0	7.4	3.0	8.8	31.0	1.0
2013-4	76.0	5.0	1.0	8.1	35.0	1.0
Total	1197.0	20.5	12.0	26.2	178.0	0

(b) Descriptive Statistic -Throughput

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	3.0	3.0	1.0	3.5	7.0	1.0
2010-4	3.0	1.0	1.0	0	1.0	1.0
2011-1	7.0	10.4	11.0	8.1	25.0	1.0
2011-2	32.0	9.4	10.0	6.7	26.0	1.0
2011-3	2.0	1.0	1.0	0	1.0	1.0
2011-4	25.0	14.9	10.0	14.6	49.0	1.0
2012-1	49.0	8.6	5.0	8.1	33.0	1.0
2012-2	45.0	11.2	3.0	16.0	56.0	1.0
2012-3	34.0	5.5	3.0	6.3	23.0	1.0
2012-4	17.0	14.2	14.0	13.7	44.0	1.0
2013-1	13.0	19.5	17.0	17.0	58.0	1.0
2013-2	26.0	21.6	18.0	16.9	60.0	1.0
2013-3	17.0	9.0	7.0	7.7	27.0	1.0
2013-4	17.0	6.3	3.0	7.5	24.0	1.0
Total	290	11.0	6.0	12.5	60	1

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

(a) Descriptive Statistic -Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1.0	7.0	7.0	-	7.0	7.0
2011-1	7.0	10.4	11.0	8.1	25.0	1.0
2011-2	24.0	10.1	10.0	5.6	24.0	1.0
2011-3	1.0	1.0	1.0	-	1.0	1.0
2011-4	11.0	16.6	13.0	11.0	35.0	4.0
2012-1	16.0	15.2	15.0	9.3	33.0	1.0
2012-2	26.0	16.1	5.0	17.7	56.0	1.0
2012-3	23.0	6.0	4.0	6.6	23.0	1.0
2012-4	14.0	15.1	14.5	14.3	44.0	1.0
2013-1	10.0	23.3	20.0	17.4	58.0	3.0
2013-2	21.0	23.6	24.0	18.1	60.0	1.0
2013-3	16.0	9.5	7.5	7.7	27.0	1.0
2013-4	12.0	8.3	7.5	8.1	24.0	1.0
Total	182.0	13.7	10.0	13.2	60.0	1.0

(b) Descriptive Statistic -Throughput bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	2.0	1.0	1.0	0	1.0	1.0
2010-4	3.0	1.0	1.0	0	1.0	1.0
2011-2	8.0	7.5	2.0	9.3	26.0	1.0
2011-3	1.0	1.0	1.0	-	1.0	1.0
2011-4	14.0	13.6	5.0	17.3	49.0	1.0
2012-1	33.0	5.3	5.0	5.0	21.0	1.0
2012-2	19.0	4.5	1.0	10.5	47.0	1.0
2012-3	11.0	4.4	3.0	5.8	21.0	1.0
2012-4	3.0	10.0	3.0	12.1	24.0	3.0
2013-1	3.0	7.0	3.0	8.7	17.0	1.0
2013-2	5.0	13.4	13.0	7.1	21.0	3.0
2013-3	1.0	1.0	1.0	-	1.0	1.0
2013-4	5.0	1.4	1.0	0.9	3.0	1.0
Total	108.0	6.4	3.0	9.6	49.0	1.0

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

(a) Descriptive Statistic - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1.0	13.0	13.0	.	13.0	13.0
2010-4	2.0	8.5	8.5	9.2	15.0	2.0
2011-2	94.0	3.9	3.0	3.2	14.0	1.0
2011-3	1.0	5.0	5.0	.	5.0	5.0
2011-4	161.0	2.7	2.0	2.7	15.0	1.0
2012-1	231.0	3.6	3.0	3.1	15.0	1.0
2012-2	283.0	3.1	3.0	2.2	13.0	1.0
2012-3	96.0	3.3	1.0	4.0	15.0	1.0
2012-4	136.0	3.5	2.0	2.7	13.0	1.0
2013-1	115.0	3.1	3.0	2.0	10.0	1.0
2013-2	253.0	5.3	4.0	4.4	15.0	1.0
2013-3	85.0	4.6	4.0	3.0	13.0	1.0
2013-4	63.0	3.7	3.0	2.6	10.0	1.0
Total	1521	3.7	3	3.3	15	1

(b) Descriptive Statistic - Churn

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1.0	13.0	13.0	.	13.0	13.0
2010-4	2.0	30.0	30.0	41.0	59.0	1.0
2011-2	94.0	21.5	10.0	22.9	83.0	1.0
2011-3	1.0	2.0	2.0	.	2.0	2.0
2011-4	161.0	17.5	8.0	23.1	94.0	0
2012-1	231.0	12.9	1.0	22.0	93.0	0
2012-2	283.0	12.9	1.0	22.2	92.0	0
2012-3	96.0	7.2	0	18.0	75.0	0
2012-4	136.0	4.1	0	12.2	67.0	0
2013-1	115.0	2.1	0	6.2	32.0	0
2013-2	253.0	7.6	0	15.7	92.0	0
2013-3	85.0	2.5	0	9.1	54.0	0
2013-4	63.0	4.3	0	16.2	90.0	0
Total	1521	10.2	.00	19.4	94	0

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

(a) Descriptive Statistic - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-2	8.0	23.2	22.0	21.2	49.0	1.0
2011-4	31.0	8.3	0	20.6	86.0	0
2012-1	106.0	4.2	0	11.8	49.0	0
2012-2	134.0	5.6	0	16.2	90.0	0
2012-3	65.0	3.1	0	12.8	75.0	0
2012-4	111.0	1.7	0	9.5	67.0	0
2013-1	89.0	0.0	0	0.1	1.0	0
2013-2	130.0	1.1	0	6.2	43.0	0
2013-3	75.0	1.0	0	6.5	54.0	0
2013-4	48.0	0	0	0	0	0
Total	797.0	2.8	0	11.3	90.0	0

(b) Descriptive Statistic - Churn bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1.0	13.0	13.0	-	13.0	13.0
2010-4	2.0	30.0	30.0	41.0	59.0	1.0
2011-2	86.0	21.3	10.0	23.2	83.0	1.0
2011-3	1.0	2.0	2.0	-	2.0	2.0
2011-4	130.0	19.7	11.0	23.1	94.0	0
2012-1	125.0	20.3	8.0	25.7	93.0	0
2012-2	149.0	19.5	9.0	24.7	92.0	0
2012-3	31.0	15.8	4.0	23.7	69.0	0
2012-4	25.0	14.6	8.0	16.9	64.0	0
2013-1	26.0	9.1	6.0	10.3	32.0	0
2013-2	123.0	14.4	6.0	19.4	92.0	0
2013-3	10.0	14.0	8.0	16.0	45.0	0
2013-4	15.0	18.1	3.0	29.8	90.0	0
Total	724.0	18.2	8.0	23.0	94.0	0

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

(a) Descriptive Statistic - Bugs

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1.0	1.0	1.0	.	1.0	1.0
2010-4	4.0	1.0	1.0	0	1.0	1.0
2011-2	32.0	4.2	3.5	3.6	14.0	1.0
2011-3	5.0	1.0	1.0	0	1.0	1.0
2011-4	36.0	4.9	2.5	5.4	22.0	1.0
2012-1	43.0	3.5	3.0	2.3	10.0	1.0
2012-2	33.0	5.4	3.0	5.5	21.0	1.0
2012-3	16.0	2.4	1.5	1.8	6.0	1.0
2012-4	13.0	2.8	2.0	1.8	6.0	1.0
2013-1	8.0	3.5	3.0	2.5	7.0	1.0
2013-2	27.0	5.8	4.0	4.8	17.0	1.0
2013-3	11.0	1.3	1.0	0.5	2.0	1.0
2013-4	10.0	1.7	1.0	1.9	7.0	1.0
Total	240	4.0	2.0	4.0	22	1

(b) Descriptive Statistic - Bugs finished within quarter

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

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

A.2 Team 2 - Descriptive Statistics

(a) Descriptive Statistic -WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	25.0	14.4	15.0	6.2	23.0	6.0
2010-4	92.0	21.4	20.0	7.2	41.0	9.0
2011-1	90.0	27.2	27.5	4.9	38.0	17.0
2011-2	91.0	29.7	27.0	14.4	62.0	12.0
2011-3	92.0	32.6	30.0	9.2	56.0	18.0
2011-4	92.0	30.1	30.0	10.1	46.0	13.0
2012-1	91.0	20.0	19.0	4.6	31.0	8.0
2012-2	91.0	25.3	26.0	10.3	51.0	6.0
2012-3	92.0	24.2	22.5	7.9	45.0	11.0
2012-4	92.0	21.6	23.0	10.5	47.0	3.0
2013-1	90.0	19.7	20.0	5.8	35.0	8.0
2013-2	91.0	28.0	27.0	4.4	37.0	15.0
2013-3	92.0	18.7	19.0	4.5	28.0	9.0
2013-4	87.0	13.3	14.0	6.6	29.0	2.0
Total	1208.0	23.8	23.0	9.8	62.0	2.0

(b) Descriptive Statistic -Throughput

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	16.0	4.2	3.0	4.0	16.0	1.0
2010-4	54.0	4.1	3.0	3.9	21.0	1.0
2011-1	57.0	4.6	4.0	3.6	17.0	1.0
2011-2	41.0	6.9	5.0	5.7	25.0	1.0
2011-3	52.0	3.8	2.0	3.6	15.0	1.0
2011-4	52.0	3.7	3.0	2.7	11.0	1.0
2012-1	55.0	4.3	3.0	3.4	12.0	1.0
2012-2	51.0	4.1	3.0	3.5	21.0	1.0
2012-3	57.0	5.8	5.0	4.3	18.0	1.0
2012-4	52.0	5.2	4.5	3.7	15.0	1.0
2013-1	51.0	4.6	3.0	3.6	16.0	1.0
2013-2	50.0	3.3	3.0	2.4	9.0	1.0
2013-3	55.0	3.9	4.0	2.9	16.0	1.0
2013-4	47.0	3.2	3.0	2.7	13.0	1.0
Total	690	4.39	3.00	3.7	25	1

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

(a) Descriptive Statistic -Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	5.0	4.6	3.0	3.2	10.0	2.0
2010-4	22.0	3.1	2.5	2.8	11.0	1.0
2011-1	15.0	5.1	4.0	4.4	17.0	1.0
2011-2	5.0	3.2	3.0	1.3	5.0	2.0
2011-3	25.0	3.7	2.0	3.8	14.0	1.0
2011-4	10.0	3.4	3.0	2.9	11.0	1.0
2012-1	9.0	2.1	1.0	2.0	7.0	1.0
2012-2	16.0	3.5	3.5	2.3	8.0	1.0
2012-3	12.0	4.2	3.5	1.9	8.0	1.0
2012-4	25.0	4.6	3.0	3.9	13.0	1.0
2013-1	11.0	3.6	3.0	3.3	11.0	1.0
2013-2	27.0	2.7	2.0	2.0	9.0	1.0
2013-3	29.0	3.9	3.0	2.7	11.0	1.0
2013-4	19.0	3.4	2.0	3.0	10.0	1.0
Total	230.0	3.7	3.0	3.0	17.0	1.0

(b) Descriptive Statistic -Throughput bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	11.0	4.1	3.0	4.4	16.0	1.0
2010-4	32.0	4.8	4.0	4.4	21.0	1.0
2011-1	42.0	4.4	4.0	3.3	13.0	1.0
2011-2	36.0	7.4	5.5	5.9	25.0	1.0
2011-3	27.0	3.9	3.0	3.5	15.0	1.0
2011-4	42.0	3.7	3.0	2.7	11.0	1.0
2012-1	46.0	4.7	3.5	3.5	12.0	1.0
2012-2	35.0	4.3	3.0	4.0	21.0	1.0
2012-3	45.0	6.3	5.0	4.6	18.0	1.0
2012-4	27.0	5.8	5.0	3.3	15.0	1.0
2013-1	40.0	4.8	4.0	3.7	16.0	1.0
2013-2	23.0	3.9	3.0	2.8	9.0	1.0
2013-3	26.0	3.8	4.0	3.2	16.0	1.0
2013-4	28.0	3.1	3.0	2.5	13.0	1.0
Total	460.0	4.8	4.0	3.9	25.0	1.0

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

(a) Descriptive Statistic - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	33.0	8.6	5.0	8.1	24.0	1.0
2010-4	97.0	7.5	6.0	6.8	27.0	1.0
2011-1	133.0	7.2	7.0	5.3	27.0	1.0
2011-2	113.0	7.0	6.0	6.0	27.0	1.0
2011-3	98.0	8.8	6.0	6.8	27.0	1.0
2011-4	88.0	8.1	6.0	6.9	27.0	1.0
2012-1	128.0	5.6	4.0	5.0	27.0	1.0
2012-2	115.0	5.7	4.0	5.0	23.0	1.0
2012-3	212.0	6.0	5.0	4.7	22.0	1.0
2012-4	154.0	5.7	5.0	4.7	25.0	1.0
2013-1	136.0	6.5	5.0	5.5	24.0	1.0
2013-2	72.0	6.6	5.0	6.4	26.0	1.0
2013-3	100.0	6.7	5.0	6.0	27.0	1.0
2013-4	93.0	6.6	6.0	4.7	20.0	1.0
Total	1572	6.7	5	5.7	27	1

(b) Descriptive Statistic - Churn

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	33.0	56.7	10.0	97.9	352.0	1.0
2010-4	97.0	63.6	25.0	99.7	493.0	1.0
2011-1	133.0	41.6	18.0	66.2	407.0	1.0
2011-2	113.0	54.2	16.0	89.0	509.0	1.0
2011-3	98.0	53.6	25.0	78.2	379.0	1.0
2011-4	88.0	68.9	29.5	92.3	446.0	1.0
2012-1	128.0	68.8	23.0	96.0	521.0	0
2012-2	115.0	69.0	24.0	107.2	475.0	0
2012-3	212.0	63.3	24.0	92.0	442.0	0
2012-4	154.0	89.4	28.5	119.0	513.0	0
2013-1	136.0	54.0	12.0	92.5	467.0	0
2013-2	72.0	62.8	19.5	98.1	419.0	0
2013-3	100.0	88.2	33.5	120.5	514.0	0
2013-4	93.0	81.0	24.0	125.1	474.0	0
Total	1572	65.7	21.	99.3	521	0

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

(a) Descriptive Statistic - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	7.0	167.6	144.0	154.4	352.0	1.0
2010-4	24.0	155.8	87.0	155.8	493.0	6.0
2011-1	26.0	72.2	47.0	87.0	312.0	1.0
2011-2	6.0	117.3	70.5	137.7	383.0	2.0
2011-3	22.0	126.7	103.5	113.5	379.0	1.0
2011-4	12.0	116.3	120.0	96.0	294.0	12.0
2012-1	18.0	108.9	54.5	119.6	377.0	0
2012-2	29.0	97.7	35.0	123.8	432.0	0
2012-3	56.0	86.0	44.5	99.1	321.0	0
2012-4	50.0	140.4	48.0	157.7	513.0	0
2013-1	38.0	71.6	1.5	137.3	467.0	0
2013-2	16.0	100.1	39.0	135.9	419.0	0
2013-3	42.0	120.8	40.5	153.1	514.0	0
2013-4	29.0	140.0	59.0	161.5	465.0	0
Total	375.0	111.4	48.0	134.3	514.0	0

(b) Descriptive Statistic - Churn bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	26.0	26.8	8.0	46.6	193.0	1.0
2010-4	73.0	33.3	18.0	41.6	191.0	1.0
2011-1	107.0	34.1	16.0	58.2	407.0	1.0
2011-2	107.0	50.7	14.0	85.1	509.0	1.0
2011-3	76.0	32.5	13.5	47.8	321.0	1.0
2011-4	76.0	61.5	25.5	90.1	446.0	1.0
2012-1	110.0	62.2	18.0	90.5	521.0	0
2012-2	86.0	59.3	21.0	99.9	475.0	0
2012-3	156.0	55.2	19.5	88.3	442.0	0
2012-4	104.0	64.8	19.0	85.6	328.0	0
2013-1	98.0	47.2	14.0	67.5	318.0	0
2013-2	56.0	52.1	14.5	82.9	376.0	0
2013-3	58.0	64.6	32.5	83.8	345.0	0
2013-4	64.0	54.2	19.5	94.5	474.0	1.0
Total	1197.0	51.3	17.0	80.4	521.0	0

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

(a) Descriptive Statistic - Bugs

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	20.0	2.6	2.0	3.5	17.0	1.0
2010-4	40.0	2.5	2.0	1.7	9.0	1.0
2011-1	47.0	2.4	2.0	1.8	8.0	1.0
2011-2	40.0	3.8	3.0	2.5	13.0	1.0
2011-3	43.0	2.6	2.0	2.4	13.0	1.0
2011-4	47.0	2.5	2.0	1.6	8.0	1.0
2012-1	35.0	3.3	3.0	3.0	16.0	1.0
2012-2	34.0	2.3	2.0	1.5	7.0	1.0
2012-3	43.0	3.6	2.0	2.6	10.0	1.0
2012-4	33.0	3.9	3.0	3.1	14.0	1.0
2013-1	38.0	2.2	2.0	1.2	6.0	1.0
2013-2	32.0	1.9	1.5	1.2	5.0	1.0
2013-3	35.0	1.8	1.0	1.1	5.0	1.0
2013-4	37.0	1.9	1.0	1.3	7.0	1.0
Total	536	2.7	2.0	2.2	17	1

(b) Descriptive Statistic - Bugs finished within quarter

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

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

A.3 Team 3 - Descriptive Statistics

(a) Descriptive Statistic -WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	24.0	9.3	10.0	6.4	23.0	1.0
2010-4	92.0	13.9	13.0	3.9	25.0	5.0
2011-1	90.0	15.3	15.5	3.9	23.0	7.0
2011-2	91.0	23.5	24.0	4.2	37.0	13.0
2011-3	92.0	20.7	20.0	5.8	34.0	9.0
2011-4	92.0	23.3	23.0	6.9	36.0	9.0
2012-1	91.0	24.9	24.0	6.6	42.0	13.0
2012-2	91.0	23.9	23.0	3.4	34.0	19.0
2012-3	92.0	28.0	29.0	5.1	38.0	21.0
2012-4	92.0	29.6	28.5	5.3	44.0	22.0
2013-1	90.0	16.2	15.0	5.0	27.0	9.0
2013-2	91.0	7.0	6.0	3.2	13.0	2.0
2013-3	92.0	7.0	7.0	2.0	14.0	3.0
2013-4	67.0	5.6	5.0	2.6	13.0	2.0
Total	1187.0	18.5	19.0	9.1	44.0	1.0

(b) Descriptive Statistic -Throughput

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	16.0	3.3	3.0	2.9	12.0	1.0
2010-4	54.0	3.5	3.0	3.0	15.0	1.0
2011-1	42.0	2.2	2.0	1.4	7.0	1.0
2011-2	45.0	4.3	3.0	3.8	20.0	1.0
2011-3	51.0	4.0	3.0	3.4	15.0	1.0
2011-4	50.0	4.7	3.0	5.2	27.0	1.0
2012-1	46.0	6.5	5.0	5.7	20.0	1.0
2012-2	40.0	2.9	2.0	3.2	15.0	1.0
2012-3	36.0	3.4	2.5	3.1	13.0	1.0
2012-4	51.0	5.0	4.0	4.4	22.0	1.0
2013-1	42.0	3.2	2.0	2.9	10.0	1.0
2013-2	22.0	1.6	1.0	1.0	5.0	1.0
2013-3	29.0	2.0	1.0	2.0	11.0	1.0
2013-4	18.0	1.6	1.0	1.1	5.0	1.0
Total	542	3.7	3.0	3.8	27	1

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

(a) Descriptive Statistic -Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	3.0	2.3	3.0	1.2	3.0	1.0
2010-4	19.0	2.9	2.0	2.4	9.0	1.0
2011-1	29.0	2.3	2.0	1.5	7.0	1.0
2011-2	24.0	4.0	3.5	2.4	8.0	1.0
2011-3	23.0	4.3	3.0	3.5	13.0	1.0
2011-4	19.0	4.0	3.0	4.5	21.0	1.0
2012-1	10.0	5.4	2.0	5.6	16.0	1.0
2012-2	18.0	2.2	1.5	1.5	6.0	1.0
2012-3	12.0	3.9	1.5	4.3	13.0	1.0
2012-4	17.0	4.8	3.0	4.5	17.0	1.0
2013-1	8.0	2.1	1.5	1.7	6.0	1.0
2013-2	3.0	1.7	2.0	0.6	2.0	1.0
2013-3	8.0	1.8	1.5	0.9	3.0	1.0
2013-4	7.0	1.3	1.0	0.5	2.0	1.0
Total	200.0	3.3	2.0	3.2	21.0	1.0

(b) Descriptive Statistic -Throughput bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	13.0	3.5	3.0	3.2	12.0	1.0
2010-4	35.0	3.9	3.0	3.3	15.0	1.0
2011-1	13.0	2.1	2.0	1.0	3.0	1.0
2011-2	21.0	4.8	3.0	4.9	20.0	1.0
2011-3	28.0	3.8	3.0	3.5	15.0	1.0
2011-4	31.0	5.2	3.0	5.5	27.0	1.0
2012-1	36.0	6.8	5.0	5.8	20.0	1.0
2012-2	22.0	3.5	2.0	4.1	15.0	1.0
2012-3	24.0	3.2	3.0	2.4	9.0	1.0
2012-4	34.0	5.2	4.0	4.5	22.0	1.0
2013-1	34.0	3.4	2.0	3.0	10.0	1.0
2013-2	19.0	1.6	1.0	1.1	5.0	1.0
2013-3	21.0	2.1	1.0	2.2	11.0	1.0
2013-4	11.0	1.7	1.0	1.3	5.0	1.0
Total	342.0	4.0	3.0	4.1	27.0	1.0

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

(a) Descriptive Statistic - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	31.0	7.2	7.0	4.1	16.0	1.0
2010-4	79.0	8.7	6.0	7.6	30.0	1.0
2011-1	32.0	7.4	4.5	5.7	18.0	1.0
2011-2	95.0	7.0	5.0	6.9	29.0	1.0
2011-3	90.0	7.7	5.0	7.0	30.0	1.0
2011-4	141.0	7.8	6.0	7.0	25.0	1.0
2012-1	171.0	7.8	6.0	6.7	30.0	1.0
2012-2	64.0	11.9	11.0	8.8	30.0	1.0
2012-3	71.0	9.6	8.0	7.5	27.0	1.0
2012-4	139.0	5.9	5.0	5.6	29.0	1.0
2013-1	75.0	7.1	5.0	7.7	29.0	1.0
2013-2	27.0	8.2	7.0	6.7	27.0	1.0
2013-3	32.0	6.9	4.0	7.9	29.0	1.0
2013-4	18.0	12.0	12.5	4.5	21.0	1.0
Total	1065	7.9	6.0	7.0	30	1

(b) Descriptive Statistic - Churn

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	31.0	145.0	118.0	143.8	383.0	1.0
2010-4	79.0	76.7	40.0	91.2	471.0	2.0
2011-1	32.0	81.6	62.0	74.0	320.0	2.0
2011-2	95.0	68.5	29.0	96.7	426.0	1.0
2011-3	90.0	72.7	22.5	102.5	476.0	0
2011-4	141.0	28.4	2.0	64.0	393.0	0
2012-1	171.0	44.5	2.0	82.7	475.0	0
2012-2	64.0	42.5	1.0	84.1	365.0	0
2012-3	71.0	53.9	13.0	85.3	406.0	0
2012-4	139.0	73.4	20.0	106.1	494.0	0
2013-1	75.0	45.4	10.0	68.8	291.0	0
2013-2	27.0	56.8	30.0	62.1	216.0	0
2013-3	32.0	66.1	34.5	87.2	403.0	0
2013-4	18.0	89.0	52.5	92.2	266.0	0
Total	1065	59.4	16.0	91.3	494	0

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

(a) Descriptive Statistic - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	7.0	197.4	141.0	136.1	383.0	16.0
2010-4	33.0	105.4	59.0	116.5	471.0	2.0
2011-1	19.0	104.0	76.0	84.4	320.0	14.0
2011-2	36.0	119.7	84.0	127.9	426.0	1.0
2011-3	40.0	89.9	48.5	106.8	407.0	0
2011-4	44.0	64.8	33.5	99.3	393.0	0
2012-1	41.0	93.3	59.0	113.1	421.0	0
2012-2	12.0	132.1	70.5	150.7	365.0	0
2012-3	17.0	101.9	66.0	121.4	406.0	0
2012-4	44.0	122.8	67.5	127.5	494.0	0
2013-1	13.0	68.8	64.0	64.9	193.0	0
2013-2	2.0	76.5	76.5	65.8	123.0	30.0
2013-3	5.0	146.6	120.0	149.0	403.0	24.0
2013-4	4.0	151.5	167.5	92.0	237.0	34.0
Total	317.0	102.6	61.0	115.2	494.0	0

(b) Descriptive Statistic - Churn bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	24.0	129.8	64.5	145.2	377.0	1.0
2010-4	46.0	56.1	26.0	61.1	210.0	2.0
2011-1	13.0	48.9	41.0	39.3	121.0	2.0
2011-2	59.0	37.3	21.0	51.6	279.0	1.0
2011-3	50.0	58.9	13.0	97.8	476.0	0
2011-4	97.0	11.9	0	25.9	128.0	0
2012-1	130.0	29.1	0	63.6	475.0	0
2012-2	52.0	21.9	0	38.8	169.0	0
2012-3	54.0	38.8	8.5	64.8	302.0	0
2012-4	95.0	50.5	8.0	86.2	402.0	0
2013-1	62.0	40.5	7.0	69.1	291.0	0
2013-2	25.0	55.2	29.0	63.0	216.0	0
2013-3	27.0	51.1	15.0	64.6	223.0	0
2013-4	14.0	71.1	35.5	87.3	266.0	0
Total	748.0	41.1	9.0	71.7	476.0	0

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

(a) Descriptive Statistic - Bugs

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	14.0	2.8	2.0	2.9	12.0	1.0
2010-4	39.0	2.1	1.0	1.5	8.0	1.0
2011-1	22.0	1.8	1.0	1.3	5.0	1.0
2011-2	28.0	3.1	2.0	3.3	16.0	1.0
2011-3	38.0	2.4	2.0	1.9	10.0	1.0
2011-4	35.0	2.9	1.0	5.0	30.0	1.0
2012-1	39.0	3.7	2.0	4.0	23.0	1.0
2012-2	31.0	1.8	1.0	1.4	7.0	1.0
2012-3	28.0	2.5	2.0	1.8	8.0	1.0
2012-4	42.0	2.5	2.0	1.7	6.0	1.0
2013-1	30.0	1.8	1.5	1.0	4.0	1.0
2013-2	19.0	1.5	1.0	1.0	5.0	1.0
2013-3	26.0	1.3	1.0	0.5	2.0	1.0
2013-4	8.0	1.9	1.0	1.7	6.0	1.0
Total	399	2.7	1.0	2.6	30	1

(b) Descriptive Statistic - Finished bugs per quarter

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	.0	15.0	100.0	.0
Mean	58.5	8.7	67.2	86	14

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

A.4 Team 4 - Descriptive Statistics

(a) Descriptive Statistic -WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	9.0	2.7	2.0	2.5	7.0	1.0
2010-4	92.0	4.5	4.0	2.9	14.0	0
2011-1	90.0	10.4	10.0	3.1	18.0	4.0
2011-2	91.0	13.8	13.0	4.5	31.0	5.0
2011-3	92.0	14.1	13.0	4.6	28.0	6.0
2011-4	92.0	16.4	16.0	4.8	30.0	6.0
2012-1	91.0	16.0	15.0	3.9	25.0	9.0
2012-2	91.0	11.7	12.0	3.5	20.0	5.0
2012-3	92.0	14.0	14.0	4.2	26.0	7.0
2012-4	92.0	20.6	19.5	5.6	33.0	10.0
2013-1	90.0	19.5	19.0	7.2	37.0	5.0
2013-2	91.0	16.0	16.0	4.8	29.0	6.0
2013-3	92.0	15.5	15.0	5.9	29.0	6.0
2013-4	91.0	10.5	11.0	4.0	19.0	1.0
Total	1196.0	14.0	14.0	6.2	37.0	0

(b) Descriptive Statistic -Throughput

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	4.0	3.0	1.0	4.0	9.0	1.0
2010-4	39.0	2.9	1.0	2.5	11.0	1.0
2011-1	48.0	3.8	3.0	2.8	15.0	1.0
2011-2	48.0	6.3	5.0	5.6	31.0	1.0
2011-3	54.0	6.3	5.0	5.0	31.0	1.0
2011-4	52.0	7.5	5.0	5.9	23.0	1.0
2012-1	60.0	6.7	5.0	4.4	17.0	1.0
2012-2	57.0	3.9	3.0	2.8	15.0	1.0
2012-3	31.0	6.2	5.0	4.4	15.0	1.0
2012-4	48.0	5.8	5.0	4.8	21.0	1.0
2013-1	51.0	9.2	8.0	7.3	34.0	1.0
2013-2	50.0	8.5	7.0	4.8	19.0	1.0
2013-3	58.0	8.1	7.5	5.1	26.0	1.0
2013-4	44.0	5.1	5.0	3.8	15.0	1.0
Total	644.0	6.2	5.0	5.1	34.0	1.0

Table A.16: Caption of Descriptive Statistic for WIP and Throughput (a) (b)

(a) Descriptive Statistic -Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	4.0	3.0	1.0	4.0	9.0	1.0
2010-4	39.0	2.9	1.0	2.5	11.0	1.0
2011-1	48.0	3.8	3.0	2.8	15.0	1.0
2011-2	48.0	6.3	5.0	5.6	31.0	1.0
2011-3	54.0	6.3	5.0	5.0	31.0	1.0
2011-4	52.0	7.5	5.0	5.9	23.0	1.0
2012-1	60.0	6.7	5.0	4.4	17.0	1.0
2012-2	57.0	3.9	3.0	2.8	15.0	1.0
2012-3	31.0	6.2	5.0	4.4	15.0	1.0
2012-4	48.0	5.8	5.0	4.8	21.0	1.0
2013-1	51.0	9.2	8.0	7.3	34.0	1.0
2013-2	50.0	8.5	7.0	4.8	19.0	1.0
2013-3	58.0	8.1	7.5	5.1	26.0	1.0
2013-4	44.0	5.1	5.0	3.8	15.0	1.0
Total	644.0	6.2	5.0	5.1	34.0	1.0

(b) Descriptive Statistic -Throughput bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2012-1	1.0	11.0	11.0	-	11.0	11.0
2012-3	2.0	3.5	3.5	3.5	6.0	1.0
2012-4	4.0	4.8	5.0	3.0	8.0	1.0
2013-1	10.0	5.5	6.0	2.5	9.0	1.0
2013-2	9.0	7.2	7.0	4.7	15.0	2.0
2013-3	2.0	4.5	4.5	2.1	6.0	3.0
2013-4	2.0	1.0	1.0	0	1.0	1.0
Total	30.0	5.6	5.5	3.7	15.0	1.0

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

(a) Descriptive Statistic - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-2	117.0	3.9	3.0	2.7	12.0	1.0
2011-3	184.0	4.0	3.0	3.0	12.0	1.0
2011-4	204.0	4.3	4.0	2.9	12.0	1.0
2012-1	230.0	3.8	3.0	3.0	12.0	1.0
2012-2	117.0	4.5	3.0	3.2	12.0	1.0
2012-3	89.0	5.5	6.0	2.7	12.0	1.0
2012-4	134.0	4.8	5.0	3.2	12.0	1.0
2013-1	316.0	2.7	2.0	2.4	11.0	1.0
2013-2	281.0	2.7	1.0	2.5	12.0	1.0
2013-4	157.0	2.8	2.0	2.5	12.0	1.0
Total	1829	3.6	3.0	2.9	12	1

(b) Descriptive Statistic - Churn

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-2	117.0	0	0	0	0	0
2011-3	184.0	0	0	0	0	0
2011-4	204.0	0	0	0	0	0
2012-1	230.0	0	0	0	0	0
2012-2	117.0	0	0	0	0	0
2012-3	89.0	0	0	0	0	0
2012-4	134.0	0	0	0	0	0
2013-1	316.0	0	0	0	0	0
2013-2	281.0	0	0	0	0	0
2013-3	157.0	0	0	0	0	0
Total	1829	.00	.00	.000	0	0

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

(a) Descriptive Statistic - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-2	116.0	0	0	0	0	0
2011-3	183.0	0	0	0	0	0
2011-4	204.0	0	0	0	0	0
2012-1	228.0	0	0	0	0	0
2012-2	116.0	0	0	0	0	0
2012-3	85.0	0	0	0	0	0
2012-4	114.0	0	0	0	0	0
2013-1	266.0	0	0	0	0	0
2013-2	254.0	0	0	0	0	0
2013-3	149.0	0	0	0	0	0
Total	1715.0	0	0	0	0	0

(b) Descriptive Statistic - Churn bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-2	1.0	0	0	-	0	0
2011-3	1.0	0	0	-	0	0
2012-1	2.0	0	0	0	0	0
2012-2	1.0	0	0	-	0	0
2012-3	4.0	0	0	0	0	0
2012-4	20.0	0	0	0	0	0
2013-1	50.0	0	0	0	0	0
2013-2	27.0	0	0	0	0	0
2013-3	8.0	0	0	0	0	0
Total	114.0	0	0	0	0	0

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

(a) Descriptive Statistic - Bugs

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-1	1.0	1.0	1.0	.	1.0	1.0
2011-2	2.0	1.0	1.0	0	1.0	1.0
2011-3	1.0	1.0	1.0	.	1.0	1.0
2012-1	2.0	1.0	1.0	0	1.0	1.0
2012-2	1.0	1.0	1.0	.	1.0	1.0
2012-3	4.0	1.0	1.0	0	1.0	1.0
2012-4	12.0	1.8	1.5	0.9	3.0	1.0
2013-1	32.0	1.6	1.0	0.9	4.0	1.0
2013-2	19.0	1.5	1.0	0.6	3.0	1.0
2013-3	12.0	1.2	1.0	0.4	2.0	1.0
2013-4	2.0	1.0	1.0	0	1.0	1.0
Total	88	1.4	1	0.8	4	1

(b) Descriptive Statistic - Bugs finished within quarter

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

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

A.5 Team 5 - Descriptive Statistics

(a) Descriptive Statistic -WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	24.0	8.4	8.0	3.3	15.0	2.0
2010-4	92.0	18.7	18.0	6.2	40.0	8.0
2011-1	90.0	7.8	8.5	6.2	20.0	0
2011-2	91.0	21.3	18.0	12.9	58.0	0
2011-3	92.0	26.8	27.0	9.3	45.0	8.0
2011-4	92.0	27.8	27.0	9.9	46.0	10.0
2012-1	91.0	44.5	47.0	9.5	65.0	24.0
2012-2	91.0	51.3	51.0	7.5	74.0	38.0
2012-3	92.0	19.6	19.0	11.2	50.0	4.0
2012-4	92.0	20.0	19.0	9.0	38.0	7.0
2013-1	90.0	124.8	126.0	94.7	270.0	9.0
2013-2	91.0	231.1	266.0	85.9	286.0	12.0
2013-3	92.0	21.2	19.0	7.4	43.0	11.0
2013-4	51.0	9.6	10.0	4.3	19.0	1.0
Total	1171.0	48.4	24.0	70.5	286.0	0

(b) Descriptive Statistic -Throughput

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	12.0	3.6	3.0	2.2	7.0	1.0
2010-4	49.0	4.2	3.0	3.5	15.0	1.0
2011-1	34.0	3.6	3.0	2.7	12.0	1.0
2011-2	51.0	7.2	7.0	5.4	19.0	1.0
2011-3	63.0	5.6	4.0	5.0	24.0	1.0
2011-4	58.0	5.0	5.0	3.7	17.0	1.0
2012-1	59.0	6.2	5.0	4.5	17.0	1.0
2012-2	59.0	5.3	4.0	3.8	15.0	1.0
2012-3	49.0	6.4	5.0	5.0	27.0	1.0
2012-4	50.0	4.7	3.0	4.1	17.0	1.0
2013-1	60.0	15.8	9.5	15.1	59.0	1.0
2013-2	58.0	6.7	7.0	4.6	22.0	1.0
2013-3	53.0	4.6	3.0	4.3	17.0	1.0
2013-4	19.0	2.0	1.0	1.7	7.0	1.0
Total	674	6.3	5.0	6.9	59	1

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

(a) Descriptive Statistic -Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1.0	1.0	1.0	-	1.0	1.0
2010-4	8.0	2.9	1.0	3.2	10.0	1.0
2011-1	7.0	3.4	4.0	2.2	7.0	1.0
2011-2	19.0	9.1	7.0	5.1	17.0	1.0
2011-3	23.0	6.4	4.0	5.7	24.0	1.0
2011-4	11.0	3.7	3.0	2.7	8.0	1.0
2012-1	6.0	4.7	3.0	4.2	10.0	1.0
2012-2	8.0	4.4	3.0	4.6	15.0	1.0
2012-4	10.0	5.1	3.5	4.2	14.0	1.0
2013-1	4.0	16.8	16.5	11.9	30.0	4.0
2013-2	1.0	2.0	2.0	-	2.0	2.0
2013-3	4.0	2.2	2.0	1.3	4.0	1.0
2013-4	6.0	1.7	1.0	1.0	3.0	1.0
Total	108.0	5.7	4.0	5.5	30.0	1.0

(b) Descriptive Statistic -Throughput bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	11.0	3.8	3.0	2.1	7.0	1.0
2010-4	41.0	4.5	3.0	3.5	15.0	1.0
2011-1	27.0	3.6	3.0	2.8	12.0	1.0
2011-2	32.0	6.2	5.5	5.4	19.0	1.0
2011-3	40.0	5.2	4.5	4.6	22.0	1.0
2011-4	47.0	5.2	5.0	3.9	17.0	1.0
2012-1	53.0	6.4	6.0	4.6	17.0	1.0
2012-2	51.0	5.5	5.0	3.7	15.0	1.0
2012-3	49.0	6.4	5.0	5.0	27.0	1.0
2012-4	40.0	4.6	3.0	4.1	17.0	1.0
2013-1	56.0	15.7	9.0	15.4	59.0	1.0
2013-2	57.0	6.8	7.0	4.6	22.0	1.0
2013-3	49.0	4.8	3.0	4.5	17.0	1.0
2013-4	13.0	2.1	1.0	1.9	7.0	1.0
Total	566.0	6.4	5.0	7.0	59.0	1.0

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

(a) Descriptive Statistic - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	16.0	15.1	9.0	16.8	58.0	1.0
2010-4	87.0	10.5	7.0	11.3	59.0	1.0
2011-1	33.0	6.5	5.0	4.3	17.0	1.0
2011-2	132.0	5.4	3.0	5.5	30.0	1.0
2011-3	169.0	8.0	4.0	9.5	56.0	1.0
2011-4	171.0	9.9	6.0	11.4	55.0	1.0
2012-1	181.0	11.6	6.0	13.6	66.0	1.0
2012-2	170.0	15.4	9.0	14.9	66.0	1.0
2012-3	167.0	5.9	5.0	5.5	37.0	1.0
2012-4	131.0	11.3	5.0	14.5	60.0	1.0
2013-1	292.0	7.6	4.0	9.8	65.0	1.0
2013-2	197.0	10.5	8.0	10.1	60.0	1.0
2013-3	139.0	8.7	5.0	10.5	59.0	1.0
2013-4	16.0	6.7	4.0	5.9	17.0	1.0
Total	1901	9.4	6.0	11.2	66	1

(b) Descriptive Statistic - Churn

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	16.0	65.8	49.5	60.1	195.0	6.0
2010-4	87.0	51.6	31.0	54.8	205.0	1.0
2011-1	33.0	43.3	14.0	58.9	201.0	1.0
2011-2	132.0	41.9	18.0	52.9	207.0	1.0
2011-3	169.0	19.8	0	39.4	198.0	0
2011-4	171.0	19.8	0	40.6	207.0	0
2012-1	181.0	17.0	0	35.2	200.0	0
2012-2	170.0	24.7	2.0	44.1	211.0	0
2012-3	167.0	15.1	0	32.3	149.0	0
2012-4	131.0	13.3	0	28.2	167.0	0
2013-1	292.0	22.9	2.5	40.2	206.0	0
2013-2	197.0	21.6	1.0	40.8	209.0	0
2013-3	139.0	18.3	0	40.0	175.0	0
2013-4	16.0	19.4	2.0	31.9	115.0	0
Total	1901	23.5	2.0	42.3	211	0

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

(a) Descriptive Statistic - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	11.0	98.5	78.0	79.4	205.0	6.0
2011-1	4.0	88.5	74.0	88.3	201.0	5.0
2011-2	11.0	94.8	59.0	75.0	205.0	2.0
2011-3	60.0	13.4	0	37.2	151.0	0
2011-4	25.0	47.9	13.0	59.4	172.0	0
2012-1	12.0	20.7	0	49.2	170.0	0
2012-2	18.0	69.8	61.0	66.7	192.0	0
2012-3	2.0	27.5	27.5	9.2	34.0	21.0
2012-4	12.0	24.5	0	38.5	100.0	0
2013-1	15.0	64.8	0	81.2	206.0	0
2013-2	8.0	31.5	4.0	40.5	91.0	0
2013-3	4.0	73.5	65.0	86.0	164.0	0
Total	182.0	43.1	2.0	62.8	206.0	0

(b) Descriptive Statistic - Churn bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	16.0	65.8	49.5	60.1	195.0	6.0
2010-4	76.0	44.8	25.0	47.2	176.0	1.0
2011-1	29.0	37.1	14.0	52.9	200.0	1.0
2011-2	121.0	37.1	16.0	48.0	207.0	1.0
2011-3	109.0	23.3	6.0	40.4	198.0	0
2011-4	146.0	14.9	0	34.5	207.0	0
2012-1	169.0	16.8	0	34.2	200.0	0
2012-2	152.0	19.4	1.0	37.5	211.0	0
2012-3	165.0	14.9	0	32.4	149.0	0
2012-4	119.0	12.2	0	26.9	167.0	0
2013-1	277.0	20.6	3.0	35.7	204.0	0
2013-2	189.0	21.1	0	40.8	209.0	0
2013-3	135.0	16.7	0	37.2	175.0	0
2013-4	16.0	19.4	2.0	31.9	115.0	0
Total	1719.0	21.4	2.0	38.9	211.0	0

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

(a) Descriptive Statistic - Bugs

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	19.0	1.9	1.0	1.8	7.0	1.0
2010-4	46.0	2.6	2.0	1.7	8.0	1.0
2011-1	36.0	1.9	1.5	1.3	7.0	1.0
2011-2	45.0	5.5	4.0	4.7	19.0	1.0
2011-3	51.0	3.0	2.0	2.6	15.0	1.0
2011-4	53.0	3.5	3.0	2.4	11.0	1.0
2012-1	52.0	3.7	3.0	2.6	10.0	1.0
2012-2	56.0	2.7	2.0	1.9	7.0	1.0
2012-3	49.0	3.2	2.0	2.7	13.0	1.0
2012-4	35.0	3.0	2.0	2.7	15.0	1.0
2013-1	56.0	9.6	7.0	8.3	38.0	1.0
2013-2	49.0	4.2	4.0	2.7	12.0	1.0
2013-3	41.0	3.1	2.0	2.4	10.0	1.0
2013-4	11.0	1.4	1.0	0.9	4.0	1.0
Total	604	3.8	3.0	4.1	38	1

(b) Descriptive Statistic - Bugs finished within quarter

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

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

A.6 Team 6 - Descriptive Statistics

(a) Descriptive Statistic -WIP							(b) Descriptive Statistic - Throughput						
Quarter	N	Mean	Median	Std.Dev	Max	Min	Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	24.0	9.5	9.0	3.6	16.0	4.0	2010-3	17.0	4.5	3.0	3.1	10.0	1.0
2010-4	92.0	10.3	10.0	2.6	16.0	6.0	2010-4	51.0	3.3	3.0	2.6	10.0	1.0
2011-1	90.0	9.8	10.0	2.0	17.0	7.0	2011-1	45.0	2.3	1.0	1.9	8.0	1.0
2011-2	91.0	10.4	11.0	2.4	16.0	4.0	2011-2	37.0	2.8	3.0	1.9	8.0	1.0
2011-3	92.0	19.5	20.5	7.3	34.0	6.0	2011-3	49.0	2.7	1.0	2.1	7.0	1.0
2011-4	92.0	22.9	22.0	9.3	44.0	9.0	2011-4	40.0	3.2	3.0	2.3	9.0	1.0
2012-1	91.0	15.6	16.0	3.7	27.0	6.0	2012-1	54.0	3.3	3.0	2.4	9.0	1.0
2012-2	91.0	17.5	18.0	6.1	42.0	8.0	2012-2	51.0	5.2	3.0	5.8	37.0	1.0
2012-3	92.0	15.2	15.0	4.5	26.0	6.0	2012-3	45.0	4.0	3.0	3.6	21.0	1.0
2012-4	92.0	26.3	25.5	10.6	50.0	11.0	2012-4	63.0	6.0	5.0	4.5	23.0	1.0
2013-1	90.0	32.6	31.0	8.4	51.0	15.0	2013-1	59.0	6.3	5.0	4.2	16.0	1.0
2013-2	91.0	43.7	43.0	5.0	60.0	36.0	2013-2	61.0	4.4	3.0	3.7	15.0	1.0
2013-3	92.0	30.6	29.5	8.0	61.0	17.0	2013-3	61.0	4.7	4.0	3.6	15.0	1.0
2013-4	85.0	37.4	39.0	20.8	125.0	10.0	2013-4	58.0	9.1	5.0	23.8	181.0	1.0
Total	1205.0	22.1	18.0	13.4	125.0	4.0	Total	691	4.58	3	181	1	7.8

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

(a) Descriptive Statistic -Throughput feature							(b) Descriptive Statistic -Throughput bug						
Quarter	N	Mean	Median	Std.Dev	Max	Min	Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	14.0	4.3	3.0	3.4	10.0	1.0	2010-3	3.0	5.3	6.0	2.1	7.0	3.0
2010-4	47.0	3.5	3.0	2.7	10.0	1.0	2010-4	4.0	1.0	1.0	0	1.0	1.0
2011-1	42.0	2.3	1.0	1.9	8.0	1.0	2011-1	3.0	2.3	1.0	2.3	5.0	1.0
2011-2	33.0	2.7	3.0	1.9	8.0	1.0	2011-3	4.0	3.5	3.0	2.5	7.0	1.0
2011-3	45.0	2.7	1.0	2.2	7.0	1.0	2011-4	4.0	2.5	2.0	1.9	5.0	1.0
2011-4	38.0	3.2	3.0	2.4	9.0	1.0	2012-1	2.0	3.5	3.5	0.7	4.0	3.0
2012-1	51.0	3.3	3.0	2.4	9.0	1.0	2012-2	3.0	3.7	2.0	2.9	7.0	2.0
2012-2	51.0	5.2	3.0	5.8	37.0	1.0	2012-3	2.0	3.0	3.0	1.4	4.0	2.0
2012-3	43.0	4.0	3.0	3.7	21.0	1.0	2012-4	8.0	3.5	2.0	3.3	11.0	1.0
2012-4	55.0	6.4	5.0	4.5	23.0	1.0	2013-1	10.0	4.4	4.5	2.7	9.0	1.0
2013-1	49.0	6.7	6.0	4.3	16.0	1.0	2013-2	14.0	2.4	1.5	2.9	12.0	1.0
2013-2	47.0	5.0	3.0	3.8	15.0	1.0	2013-3	17.0	4.2	4.0	3.2	13.0	1.0
2013-3	44.0	4.8	4.0	3.8	15.0	1.0	2013-4	8.0	2.5	3.0	1.1	4.0	1.0
2013-4	50.0	10.1	5.0	25.5	181.0	1.0	Total	82.0	3.3	2.5	2.6	13.0	1.0
Total	609.0	4.8	3.0	8.3	181.0	1.0							

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

(a) Descriptive Statistic - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	29.0	3.1	2.0	3.9	20.0	1.0
2010-4	46.0	5.7	3.0	6.9	28.0	1.0
2011-1	43.0	8.9	5.0	8.0	28.0	1.0
2011-2	26.0	8.0	6.0	7.4	28.0	1.0
2011-3	28.0	11.4	9.0	7.8	27.0	1.0
2011-4	63.0	9.1	8.0	6.9	25.0	1.0
2012-1	107.0	8.9	8.0	6.3	28.0	1.0
2012-2	145.0	8.2	7.0	5.7	26.0	1.0
2012-3	101.0	8.4	7.0	5.9	27.0	1.0
2012-4	211.0	7.2	6.0	6.1	28.0	1.0
2013-1	218.0	6.5	5.0	5.3	28.0	1.0
2013-2	128.0	6.3	6.0	4.8	24.0	1.0
2013-3	155.0	7.3	5.0	6.5	28.0	1.0
2013-4	233.0	4.7	2.0	5.5	25.0	1.0
Total	1533	7.0	6.0	6.2	28	1

(b) Descriptive Statistic - Churn

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	29.0	154.2	30.0	273.2	1122.0	1.0
2010-4	46.0	240.6	92.0	315.2	1078.0	1.0
2011-1	43.0	134.9	31.0	252.2	1112.0	1.0
2011-2	26.0	263.1	191.0	293.3	1187.0	1.0
2011-3	28.0	226.2	154.5	231.2	952.0	4.0
2011-4	63.0	102.2	0	214.0	1002.0	0
2012-1	107.0	109.6	0	260.7	1234.0	0
2012-2	145.0	113.0	0	244.6	1197.0	0
2012-3	101.0	96.5	0	238.8	1225.0	0
2012-4	211.0	137.3	16.0	237.8	1144.0	0
2013-1	218.0	140.2	19.0	246.6	1210.0	0
2013-2	128.0	221.8	91.5	301.9	1174.0	0
2013-3	155.0	145.5	31.0	244.7	1164.0	0
2013-4	233.0	147.7	14.0	260.1	1199.0	0
Total	1533	146	17.0	257	1234	0

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

(a) Descriptive Statistic - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	23.0	187.3	37.0	298.3	1122.0	1.0
2010-4	45.0	244.5	106.0	317.7	1078.0	1.0
2011-1	40.0	143.9	40.0	259.4	1112.0	1.0
2011-2	26.0	263.1	191.0	293.3	1187.0	1.0
2011-3	28.0	226.2	154.5	231.2	952.0	4.0
2011-4	61.0	105.6	0	216.7	1002.0	0
2012-1	99.0	118.1	0	269.4	1234.0	0
2012-2	145.0	113.0	0	244.6	1197.0	0
2012-3	89.0	107.0	0	252.4	1225.0	0
2012-4	171.0	157.8	25.0	253.0	1144.0	0
2013-1	148.0	144.6	19.5	239.6	1166.0	0
2013-2	94.0	247.4	99.0	324.2	1174.0	0
2013-3	109.0	172.9	50.0	272.3	1164.0	0
2013-4	190.0	156.3	16.0	269.2	1199.0	0
Total	1268.0	156.5	17.0	266.5	1234.0	0

(b) Descriptive Statistic - Churn bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	6.0	27.5	9.0	45.3	119.0	2.0
2010-4	1.0	68.0	68.0	.	68.0	68.0
2011-1	3.0	15.7	11.0	13.6	31.0	5.0
2011-2	2.0	0.5	0.5	0.7	1.0	0
2012-1	8.0	4.2	0	12.0	34.0	0
2012-3	12.0	18.8	0	39.5	114.0	0
2012-4	40.0	49.6	10.5	125.3	658.0	0
2013-1	70.0	130.8	18.5	262.3	1210.0	0
2013-2	34.0	151.0	65.0	218.2	947.0	0
2013-3	46.0	80.7	15.0	144.2	574.0	0
2013-4	43.0	109.6	8.0	214.4	1127.0	0
Total	265.0	95.2	12.0	197.8	1210.0	0

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

(a) Descriptive Statistic - Bugs							(b) Descriptive Statistic - Bugs finished within quarter					
Quarter	N	Mean	Median	Std.Dev	Max	Min	Quarter	Finished	Not finished	Total	Finished	Not finished
2010-3	10.0	1.5	1.5	0.5	2.0	1.0	2010-3	14	1	15	93.3	6.7
2010-4	7.0	1.0	1.0	0	1.0	1.0	2010-4	6	1	7	85.7	14.3
2011-1	5.0	1.4	1.0	0.9	3.0	1.0	2011-1	6	1	7	85.7	14.3
2011-2	8.0	1.1	1.0	0.4	2.0	1.0	2011-2	7	2	9	77.8	22.2
2011-3	4.0	1.2	1.0	0.5	2.0	1.0	2011-3	3	2	5	60	40
2011-4	2.0	2.0	2.0	0	2.0	2.0	2011-4	3	1	4	75	25
2012-1	7.0	1.1	1.0	0.4	2.0	1.0	2012-1	8	0	8	100	0
2012-2	11.0	1.3	1.0	0.5	2.0	1.0	2012-2	12	2	14	85.7	14.3
2012-3	24.0	1.8	1.5	1.0	4.0	1.0	2012-3	41	2	43	95.3	4.7
2012-4	39.0	1.9	2.0	1.2	7.0	1.0	2012-4	66	9	75	88	12
2013-1	33.0	1.5	1.0	0.7	3.0	1.0	2013-1	43	7	50	86	14
2013-2	34.0	1.6	1.0	0.8	4.0	1.0	2013-2	52	1	53	98.1	1.9
2013-3	27.0	1.8	2.0	0.9	4.0	1.0	2013-3	49	0	49	100	0
Total	211	1.6	1.0	.868	7	1	Mean	23.9	2.2	26.1	87.0	13.0

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

A.7 Team 7 - Descriptive Statistics

(a) Descriptive Statistic -WIP							(b) Descriptive Statistic -Throughput						
Quarter	N	Mean	Median	Std.Dev	Max	Min	Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	25.0	14.1	14.0	5.1	27.0	9.0	2010-3	11.0	3.9	2.0	4.0	14.0	1.0
2010-4	92.0	23.0	23.5	7.8	48.0	12.0	2010-4	53.0	3.7	3.0	2.5	13.0	1.0
2011-1	90.0	14.8	13.5	5.0	26.0	6.0	2011-1	54.0	3.2	3.0	2.3	13.0	1.0
2011-2	91.0	12.8	12.0	4.6	22.0	3.0	2011-2	33.0	2.3	2.0	1.2	5.0	1.0
2011-3	92.0	17.2	16.0	6.5	32.0	8.0	2011-3	36.0	2.0	2.0	1.1	4.0	1.0
2011-4	92.0	17.5	17.0	6.3	30.0	4.0	2011-4	44.0	2.2	2.0	1.5	6.0	1.0
2012-1	91.0	14.3	15.0	3.9	24.0	7.0	2012-1	37.0	2.0	1.0	1.5	7.0	1.0
2012-2	91.0	16.5	14.0	6.0	34.0	10.0	2012-2	25.0	2.2	2.0	1.4	6.0	1.0
2012-3	92.0	23.8	20.0	10.4	43.0	7.0	2012-3	32.0	3.4	3.0	2.5	13.0	1.0
2012-4	67.0	1.7	0	3.0	11.0	0	2012-4	3.0	1.0	1.0	0	1.0	1.0
Total	823.0	16.1	15.0	8.4	48.0	0	Total	328.0	2.7	2.0	2.1	14.0	1.0

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

(a) Descriptive Statistic -Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	8.0	3.2	2.5	2.4	7.0	1.0
2010-4	37.0	4.0	3.0	2.7	13.0	1.0
2011-1	22.0	3.4	3.0	2.4	10.0	1.0
2011-2	10.0	2.7	2.5	1.7	5.0	1.0
2011-3	17.0	2.1	2.0	1.2	4.0	1.0
2011-4	26.0	2.0	1.5	1.3	5.0	1.0
2012-1	12.0	2.0	2.0	1.3	5.0	1.0
2012-2	9.0	2.2	2.0	0.8	3.0	1.0
2012-3	12.0	2.7	3.0	1.5	5.0	1.0
2012-4	3.0	1.0	1.0	0	1.0	1.0
Total	156.0	2.8	2.0	2.1	13.0	1.0

(b) Descriptive Statistic -Throughput bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	3.0	5.7	2.0	7.2	14.0	1.0
2010-4	16.0	2.9	2.5	2.0	9.0	1.0
2011-1	32.0	3.0	3.0	2.3	13.0	1.0
2011-2	23.0	2.1	2.0	0.9	4.0	1.0
2011-3	19.0	1.9	2.0	1.0	4.0	1.0
2011-4	18.0	2.4	2.0	1.7	6.0	1.0
2012-1	25.0	2.1	1.0	1.6	7.0	1.0
2012-2	16.0	2.2	2.0	1.6	6.0	1.0
2012-3	20.0	3.8	3.0	2.9	13.0	1.0
Total	172.0	2.6	2.0	2.1	14.0	1.0

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

(a) Descriptive Statistic - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	30.0	7.1	4.0	6.9	23.0	1.0
2010-4	126.0	6.6	3.0	7.9	35.0	1.0
2011-1	116.0	6.3	3.0	7.4	32.0	1.0
2011-2	51.0	10.4	8.0	8.7	34.0	1.0
2011-3	54.0	11.8	7.0	10.6	35.0	1.0
2011-4	85.0	9.0	6.0	8.8	35.0	1.0
2012-1	59.0	8.9	6.0	7.7	25.0	1.0
2012-2	30.0	14.4	8.0	13.1	34.0	1.0
2012-3	99.0	6.4	3.0	8.5	35.0	1.0
2012-4	2.0	1.5	1.5	0.7	2.0	1.0
Total	652.0	8.1	4.0	8.8	35.0	1.0

(b) Descriptive Statistic - Churn

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	30.0	212.8	214.0	217.9	647.0	1.0
2010-4	126.0	118.1	47.5	152.7	700.0	1.0
2011-1	116.0	91.9	34.0	132.9	589.0	1.0
2011-2	51.0	99.3	21.0	171.1	726.0	0
2011-3	54.0	65.4	15.0	124.9	577.0	0
2011-4	85.0	105.2	20.0	168.1	675.0	0
2012-1	59.0	58.6	17.0	110.3	636.0	0
2012-2	30.0	64.0	33.5	91.6	379.0	0
2012-3	99.0	85.0	34.0	128.0	713.0	0
2012-4	2.0	44.0	44.0	1.4	45.0	43.0
Total	652.0	97.1	28.5	147.7	726.0	0

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

(a) Descriptive Statistic - Churn feature							(b) Descriptive Statistic - Churn bug						
Quarter	N	Mean	Median	Std.Dev	Max	Min	Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	13.0	335.7	258.0	215.1	647.0	5.0	2010-3	17.0	118.8	11.0	172.4	448.0	1.0
2010-4	89.0	131.9	86.0	150.7	700.0	1.0	2010-4	37.0	84.9	17.0	154.4	662.0	1.0
2011-1	24.0	148.2	87.5	160.2	523.0	3.0	2011-1	92.0	77.2	27.0	121.5	589.0	1.0
2011-2	12.0	190.7	84.5	234.7	726.0	0	2011-2	39.0	71.2	18.0	138.3	719.0	0
2011-3	15.0	165.1	115.0	195.4	577.0	0	2011-3	39.0	27.0	12.0	47.4	226.0	0
2011-4	37.0	169.3	57.0	208.0	675.0	0	2011-4	48.0	55.7	12.0	107.9	480.0	0
2012-1	15.0	121.5	45.0	177.7	636.0	0	2012-1	44.0	37.2	15.5	65.3	367.0	0
2012-2	8.0	40.1	33.5	43.6	140.0	0	2012-2	22.0	72.6	30.5	103.3	379.0	0
2012-3	20.0	94.0	56.0	110.7	344.0	0	2012-3	79.0	82.7	28.0	132.6	713.0	0
2012-4	2.0	44.0	44.0	1.4	45.0	43.0	Total	417.0	68.4	18.0	120.2	719.0	0
Total	235.0	148.1	78.0	175.9	726.0	0							

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

(a) Descriptive Statistic - Bugs							(b) Descriptive Statistic - Bugs finished within quarter					
Quarter	N	Mean	Median	Std.Dev	Max	Min	Quarter	Finished	Not finished	Total	Finished	Not finished
2010-3	17.0	1.8	2.0	0.8	3.0	1.0	2010-3	20	10	30	66.7	33.3
2010-4	28.0	1.7	1.5	0.9	4.0	1.0	2010-4	47	1	48	97.9	2.1
2011-1	39.0	3.1	2.0	2.5	12.0	1.0	2011-1	119	2	121	98.3	1.7
2011-2	26.0	2.0	2.0	1.3	5.0	1.0	2011-2	45	8	53	84.9	15.1
2011-3	26.0	1.6	1.0	1.1	5.0	1.0	2011-3	35	6	41	85.4	14.6
2011-4	24.0	2.2	2.0	1.5	7.0	1.0	2011-4	45	7	52	86.5	13.5
2012-1	29.0	1.7	1.0	1.4	8.0	1.0	2012-1	46	2	48	95.8	4.2
2012-2	18.0	2.7	2.0	2.6	11.0	1.0	2012-2	36	12	48	75	25
2012-3	29.0	2.3	2.0	1.3	5.0	1.0	2012-3	67	0	67	100	0
Total	240.0	2.1	2.0	1.7	12.0	1.0	Mean	38.3	4.3	42.7	65.9	34.1

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

A.8 Team 8 - Descriptive Statistics

(a) Descriptive Statistic -WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	19.0	0.4	0	0.5	1.0	0
2011-1	90.0	3.7	3.0	2.4	9.0	0
2011-2	91.0	5.9	6.0	1.9	11.0	1.0
2011-3	92.0	11.2	12.0	2.6	16.0	7.0
2011-4	92.0	7.9	7.0	3.3	14.0	3.0
2012-1	91.0	9.7	9.0	3.7	16.0	3.0
2012-2	91.0	4.3	2.0	4.6	12.0	1.0
2012-3	92.0	9.1	9.0	7.3	32.0	1.0
2012-4	92.0	5.6	7.0	4.6	18.0	1.0
2013-1	90.0	8.4	4.0	9.1	30.0	1.0
2013-2	91.0	19.7	18.0	11.2	55.0	2.0
2013-3	92.0	8.2	4.0	8.5	29.0	0
2013-4	77.0	4.0	5.0	2.8	11.0	0
Total	1100.0	8.1	6.0	7.3	55.0	0

(b) Descriptive Statistic -Throughput

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	2.0	1.0	1.0	0	1.0	1.0
2011-1	12.0	1.5	1.0	0.7	3.0	1.0
2011-2	21.0	1.2	1.0	0.5	3.0	1.0
2011-3	15.0	1.7	1.0	1.1	4.0	1.0
2011-4	19.0	1.3	1.0	0.6	3.0	1.0
2012-1	16.0	1.4	1.0	1.0	5.0	1.0
2012-2	3.0	1.0	1.0	0	1.0	1.0
2012-3	23.0	2.5	2.0	2.5	12.0	1.0
2012-4	10.0	1.7	2.0	0.7	3.0	1.0
2013-1	25.0	3.8	3.0	3.4	14.0	1.0
2013-2	20.0	3.5	2.5	2.9	9.0	1.0
2013-3	21.0	3.5	2.0	3.5	13.0	1.0
2013-4	4.0	3.5	3.0	2.6	7.0	1.0
Total	191.0	2.3	1.0	2.4	14.0	1.0

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

(a) Descriptive Statistic -Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-1	2.0	1.0	1.0	0	1.0	1.0
2011-2	5.0	1.4	1.0	0.9	3.0	1.0
2011-3	1.0	1.0	1.0	-	1.0	1.0
2011-4	6.0	1.5	1.0	0.8	3.0	1.0
2012-1	6.0	1.8	1.0	1.6	5.0	1.0
2012-2	20.0	2.7	2.0	2.7	12.0	1.0
2012-4	6.0	1.8	2.0	0.8	3.0	1.0
2013-1	18.0	3.6	3.0	2.6	9.0	1.0
2013-2	12.0	5.0	5.0	2.8	9.0	1.0
2013-3	13.0	4.8	4.0	3.9	13.0	1.0
2013-4	3.0	4.3	4.0	2.5	7.0	2.0
Total	92.0	3.2	2.0	2.8	13.0	1.0

(b) Descriptive Statistic -Throughput bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	2.0	1.0	1.0	0	1.0	1.0
2011-1	10.0	1.6	1.5	0.7	3.0	1.0
2011-2	16.0	1.2	1.0	0.4	2.0	1.0
2011-3	14.0	1.8	1.0	1.1	4.0	1.0
2011-4	13.0	1.1	1.0	0.4	2.0	1.0
2012-1	10.0	1.1	1.0	0.3	2.0	1.0
2012-2	3.0	1.0	1.0	0	1.0	1.0
2012-3	3.0	1.0	1.0	0	1.0	1.0
2012-4	4.0	1.5	1.5	0.6	2.0	1.0
2013-1	7.0	4.1	1.0	5.0	14.0	1.0
2013-2	8.0	1.4	1.0	0.7	3.0	1.0
2013-3	8.0	1.2	1.0	0.5	2.0	1.0
2013-4	1.0	1.0	1.0	-	1.0	1.0
Total	99.0	1.5	1.0	1.6	14.0	1.0

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

(a) Descriptive Statistic - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	1.0	1.0	1.0	-	1.0	1.0
2011-1	3.0	3.0	3.0	2.0	5.0	1.0
2011-2	9.0	17.4	12.0	18.7	48.0	1.0
2011-3	22.0	12.1	8.0	11.9	46.0	1.0
2011-4	12.0	17.8	16.5	16.5	43.0	1.0
2012-1	14.0	17.4	11.0	16.3	38.0	1.0
2012-2	1.0	1.0	1.0	-	1.0	1.0
2012-3	51.0	11.2	5.0	13.2	54.0	1.0
2012-4	16.0	10.6	6.0	11.3	45.0	1.0
2013-1	58.0	5.5	3.0	6.9	47.0	1.0
2013-2	60.0	6.2	7.0	3.6	15.0	1.0
2013-3	69.0	7.2	3.0	7.7	33.0	1.0
2013-4	13.0	21.6	4.0	24.5	53.0	1.0
Total	329.0	9.4	5.0	11.6	54.0	1.0

(b) Descriptive Statistic - Churn

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	1.0	3.0	3.0	-1.0	3.0	3.0
2011-1	3.0	7.7	3.0	9.0	18.0	2.0
2011-2	9.0	6.3	1.0	10.7	26.0	0
2011-3	22.0	4.0	0	9.0	26.0	0
2011-4	12.0	3.6	2.0	4.6	14.0	0
2012-1	14.0	0.4	0	1.3	5.0	0
2012-2	1.0	8.0	8.0	-1.0	8.0	8.0
2012-3	51.0	0	0	0	0	0
2012-4	16.0	0	0	0	0	0
2013-1	58.0	1.2	0	4.4	22.0	0
2013-2	60.0	0.0	0	0.1	1.0	0
2013-3	69.0	0	0	0	0	0
2013-4	13.0	0.6	0	2.2	8.0	0
Total	329.0	0.9	0	4.0	26.0	0

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

(a) Descriptive Statistic - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2011-1	1.0	2.0	2.0	-	2.0	2.0
2011-2	2.0	0	0	0	0	0
2011-3	1.0	0	0	-	0	0
2011-4	4.0	2.8	0.5	4.9	10.0	0
2012-1	7.0	0	0	0	0	0
2012-3	50.0	0	0	0	0	0
2012-4	12.0	0	0	0	0	0
2013-1	41.0	0	0	0	0	0
2013-2	54.0	0	0	0	0	0
2013-3	62.0	0	0	0	0	0
2013-4	12.0	0	0	0	0	0
Total	246.0	0.1	0	0.7	10.0	0

(b) Descriptive Statistic - Churn bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	1.0	3.0	3.0	-	3.0	3.0
2011-1	2.0	10.5	10.5	10.6	18.0	3.0
2011-2	7.0	8.1	2.0	11.6	26.0	0
2011-3	21.0	4.2	0	9.2	26.0	0
2011-4	8.0	4.0	3.0	4.8	14.0	0
2012-1	7.0	0.9	0	1.9	5.0	0
2012-2	1.0	8.0	8.0	-	8.0	8.0
2012-3	1.0	0	0	-	0	0
2012-4	4.0	0	0	0	0	0
2013-1	17.0	4.2	0	7.4	22.0	0
2013-2	6.0	0.2	0	0.4	1.0	0
2013-3	7.0	0	0	0	0	0
2013-4	1.0	8.0	8.0	-	8.0	8.0
Total	83.0	3.6	0	7.2	26.0	0

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

(a) Descriptive Statistic - Bugs

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1.0	1.0	1.0	-1.0	1.0	1.0
2010-4	7.0	1.1	1.0	0.4	2.0	1.0
2011-1	9.0	1.2	1.0	0.4	2.0	1.0
2011-2	16.0	1.6	1.0	1.5	7.0	1.0
2011-3	15.0	1.3	1.0	0.6	3.0	1.0
2011-4	13.0	1.2	1.0	0.6	3.0	1.0
2012-1	9.0	1.2	1.0	0.4	2.0	1.0
2012-2	2.0	1.0	1.0	0	1.0	1.0
2012-3	4.0	1.2	1.0	0.5	2.0	1.0
2012-4	2.0	1.0	1.0	0	1.0	1.0
2013-1	10.0	3.3	2.5	2.8	10.0	1.0
2013-2	4.0	1.0	1.0	0	1.0	1.0
2013-3	4.0	1.5	1.5	0.6	2.0	1.0
2013-4	1.0	1.0	1.0	-1.0	1.0	1.0
Total	100.0	1.5	1.0	1.3	10.0	1.0

(b) Descriptive Statistic - Bugs finished within quarter

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

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

A.9 Team 9 - Descriptive Statistics

(a) Descriptive Statistic -WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	92.0	0	0	0	0	0
2010-4	144.0	1.6	0	3.0	10.0	0
2011-1	180.0	5.9	2.5	6.5	19.0	0
2011-2	182.0	5.6	1.5	7.5	34.0	0
2011-3	184.0	6.4	3.0	7.0	24.0	0
2011-4	184.0	8.0	2.5	8.8	25.0	0
2012-1	182.0	8.1	4.0	8.8	30.0	0
2012-2	182.0	17.7	4.0	21.2	67.0	0
2012-3	184.0	16.3	7.5	17.3	51.0	0
2012-4	184.0	10.9	1.5	13.2	39.0	0
2013-1	180.0	10.7	3.5	12.1	38.0	0
2013-2	182.0	13.3	5.5	16.1	47.0	0
2013-3	184.0	8.0	3.0	9.1	35.0	0
2013-4	174.0	8.3	0	9.2	29.0	0
Total	2418	9.1	0	12.5	67.0	0

(b) Descriptive Statistic -Throughput

2010-4	52.0	4.5	4.5	3.4	10.0	0
2011-1	90.0	11.8	12.5	3.8	19.0	5.0
2011-2	91.0	11.2	8.0	6.9	34.0	3.0
2011-3	92.0	12.8	12.0	4.0	24.0	6.0
2011-4	92.0	16.0	17.0	5.1	25.0	5.0
2012-1	91.0	16.2	15.0	4.7	30.0	8.0
2012-2	91.0	35.4	33.0	16.4	67.0	8.0
2012-3	92.0	32.6	33.5	7.9	51.0	15.0
2012-4	92.0	21.8	23.5	10.4	39.0	3.0
2013-1	90.0	21.4	20.5	8.0	38.0	7.0
2013-2	91.0	26.6	21.0	12.7	47.0	11.0
2013-3	92.0	15.9	14.0	6.3	35.0	6.0
2013-4	84.0	17.1	17.0	4.5	29.0	7.0
Total	1140.0	19.2	16.0	11.6	67.0	0

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

(a) Descriptive Statistic -Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	12.0	1.7	1.0	1.0	4.0	1.0
2011-1	13.0	1.6	1.0	0.8	3.0	1.0
2011-2	5.0	1.4	1.0	0.5	2.0	1.0
2011-3	9.0	1.2	1.0	0.4	2.0	1.0
2011-4	17.0	1.7	1.0	1.4	6.0	1.0
2012-1	11.0	1.6	1.0	0.8	3.0	1.0
2012-2	23.0	2.9	3.0	1.6	6.0	1.0
2012-3	12.0	3.8	3.5	2.8	9.0	1.0
2012-4	20.0	2.7	2.0	1.7	6.0	1.0
2013-1	15.0	2.1	2.0	1.0	4.0	1.0
2013-2	24.0	3.6	3.0	2.8	12.0	1.0
2013-3	22.0	2.5	2.0	1.7	7.0	1.0
2013-4	31.0	2.4	2.0	1.5	7.0	1.0
Total	214.0	2.4	2.0	1.8	12.0	1.0

(b) Descriptive Statistic -Throughput bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	3.0	2.0	2.0	1.0	3.0	1.0
2011-1	17.0	1.9	1.0	1.1	4.0	1.0
2011-2	26.0	2.4	2.0	1.9	9.0	1.0
2011-3	18.0	1.7	1.5	1.0	5.0	1.0
2011-4	16.0	2.3	2.0	1.1	5.0	1.0
2012-1	30.0	2.7	2.5	1.5	5.0	1.0
2012-2	25.0	3.8	3.0	2.3	9.0	1.0
2012-3	41.0	3.1	3.0	2.0	9.0	1.0
2012-4	23.0	3.0	3.0	2.3	10.0	1.0
2013-1	36.0	3.2	3.0	1.9	9.0	1.0
2013-2	22.0	3.4	3.0	2.2	8.0	1.0
2013-3	28.0	2.6	2.0	1.9	9.0	1.0
2013-4	22.0	2.5	2.0	1.6	7.0	1.0
Total	307.0	2.8	2.0	1.9	10.0	1.0

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

(a) Descriptive Statistic - Lead time

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	16.0	9.9	8.0	9.2	31.0	2.0
2011-1	32.0	14.6	13.5	9.7	33.0	2.0
2011-2	41.0	11.5	10.0	6.2	29.0	2.0
2011-3	22.0	8.8	8.0	6.5	32.0	2.0
2011-4	34.0	8.3	5.5	7.1	30.0	2.0
2012-1	67.0	8.0	5.0	6.7	33.0	2.0
2012-2	99.0	7.5	6.0	7.0	32.0	2.0
2012-3	121.0	9.0	7.0	7.0	33.0	2.0
2012-4	65.0	11.7	9.0	8.7	28.0	2.0
2013-1	117.0	9.1	7.0	6.5	30.0	2.0
2013-2	117.0	9.9	8.0	6.7	28.0	2.0
2013-3	108.0	8.3	8.0	5.5	32.0	2.0
2013-4	84.0	8.9	7.0	6.3	29.0	2.0
Total	923.0	9.3	7.0	7.0	33.0	2.0

(b) Descriptive Statistic - Churn

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	16.0	77.9	54.5	102.2	396.0	4.0
2011-1	32.0	70.7	36.0	84.4	388.0	1.0
2011-2	41.0	53.9	21.0	82.4	460.0	1.0
2011-3	22.0	72.3	21.0	98.7	309.0	1.0
2011-4	34.0	129.8	71.0	135.7	450.0	2.0
2012-1	67.0	80.0	28.0	108.8	426.0	1.0
2012-2	99.0	60.9	25.0	87.5	465.0	1.0
2012-3	121.0	77.6	36.0	116.0	479.0	0
2012-4	65.0	55.7	26.0	79.7	367.0	0
2013-1	117.0	57.8	26.0	76.6	369.0	0
2013-2	117.0	117.2	60.0	134.2	454.0	0
2013-3	108.0	98.2	38.5	120.9	438.0	0
2013-4	84.0	99.6	43.5	126.5	460.0	0
Total	923.0	81.9	34.0	110.0	479.0	0

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

(a) Descriptive Statistic - Churn feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	11.0	88.5	50.0	120.5	396.0	5.0
2011-1	10.0	109.6	73.5	109.8	388.0	19.0
2011-2	3.0	127.7	162.0	67.4	171.0	50.0
2011-3	2.0	204.5	204.5	130.8	297.0	112.0
2011-4	14.0	229.9	232.5	135.6	450.0	61.0
2012-1	15.0	130.1	72.0	129.8	426.0	12.0
2012-2	29.0	119.5	74.0	126.0	465.0	9.0
2012-3	29.0	112.7	34.0	150.6	464.0	1.0
2012-4	24.0	100.3	45.5	110.4	367.0	0
2013-1	36.0	100.9	75.0	99.2	369.0	0
2013-2	51.0	160.6	108.0	155.0	454.0	0
2013-3	38.0	109.4	45.0	131.6	438.0	0
2013-4	33.0	192.0	186.0	142.3	460.0	0
Total	295.0	133.9	81.0	135.2	465.0	0

(b) Descriptive Statistic - Churn bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	5.0	54.6	59.0	43.5	120.0	4.0
2011-1	22.0	53.0	30.0	65.6	204.0	1.0
2011-2	38.0	48.1	18.0	81.3	460.0	1.0
2011-3	20.0	59.1	15.5	88.5	309.0	1.0
2011-4	20.0	59.7	35.5	82.8	299.0	2.0
2012-1	52.0	65.5	15.0	98.7	367.0	1.0
2012-2	70.0	36.7	17.0	49.0	256.0	1.0
2012-3	92.0	66.5	36.0	101.3	479.0	0
2012-4	41.0	29.5	8.0	35.6	110.0	0
2013-1	81.0	38.7	19.0	54.7	265.0	0
2013-2	66.0	83.7	36.0	105.2	353.0	0
2013-3	70.0	92.1	38.5	115.1	432.0	0
2013-4	51.0	39.8	19.0	65.7	428.0	0
Total	628.0	57.5	23.5	85.7	479.0	0

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

(a) Descriptive Statistic - Bugs

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	2.0	1.0	1.0	0	1.0	1.0
2010-4	13.0	1.9	2.0	1.1	4.0	1.0
2011-1	18.0	1.8	1.0	0.9	3.0	1.0
2011-2	20.0	2.2	1.0	2.6	12.0	1.0
2011-3	14.0	1.6	1.5	0.6	3.0	1.0
2011-4	23.0	1.8	1.0	1.2	5.0	1.0
2012-1	33.0	2.1	2.0	1.5	7.0	1.0
2012-2	43.0	2.2	2.0	1.5	7.0	1.0
2012-3	40.0	2.7	2.0	1.9	9.0	1.0
2012-4	33.0	2.2	2.0	1.7	8.0	1.0
2013-1	34.0	2.2	1.0	1.7	8.0	1.0
2013-2	39.0	2.1	2.0	1.6	6.0	1.0
2013-3	38.0	2.1	1.5	1.5	7.0	1.0
2013-4	42.0	1.5	1.0	0.8	4.0	1.0
Total	403.0	2.1	1.0	1.5	12.0	1.0

(b) Descriptive Statistic - Bugs finished within quarter

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

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

A.10 Team 10 - Descriptive Statistics

(a) Descriptive Statistic -WIP

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	24.0	2.7	3.0	0.9	5.0	1.0
2010-4	92.0	13.1	13.5	6.0	26.0	2.0
2011-1	90.0	8.1	8.0	6.2	22.0	0
2011-2	91.0	6.0	4.0	4.8	17.0	0
2011-3	92.0	0.9	1.0	0.8	3.0	0
2011-4	92.0	16.7	17.5	13.7	40.0	1.0
2012-1	91.0	24.6	24.0	3.8	36.0	17.0
2012-2	91.0	34.5	35.0	8.4	51.0	18.0
2012-3	92.0	12.7	10.0	8.7	44.0	4.0
2012-4	92.0	25.8	19.5	13.6	59.0	10.0
2013-1	90.0	16.3	6.0	14.5	49.0	5.0
2013-2	91.0	8.9	8.0	4.5	21.0	5.0
2013-3	92.0	12.5	12.0	5.8	29.0	3.0
2013-4	57.0	15.4	15.0	4.8	26.0	7.0
Total	1177.0	14.8	12.0	12.2	59.0	0

(b) Descriptive Statistic -Throughput

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	5.0	1.6	1.0	0.9	3.0	1.0
2010-4	44.0	2.2	2.0	1.6	7.0	1.0
2011-1	29.0	2.5	2.0	1.8	7.0	1.0
2011-2	21.0	1.9	1.0	1.4	6.0	1.0
2011-3	8.0	1.0	1.0	0	1.0	1.0
2011-4	34.0	2.6	2.0	1.7	7.0	1.0
2012-1	32.0	1.8	1.0	1.2	6.0	1.0
2012-2	52.0	2.7	2.0	1.6	7.0	1.0
2012-3	38.0	1.7	1.0	1.1	6.0	1.0
2012-4	47.0	2.7	2.0	3.0	16.0	1.0
2013-1	25.0	2.9	2.0	1.8	8.0	1.0
2013-2	10.0	1.9	1.5	1.1	4.0	1.0
2013-3	36.0	1.8	1.5	1.2	5.0	1.0
2013-4	23.0	1.8	1.0	1.2	5.0	1.0
Total	404.0	2.2	2.0	1.7	16.0	1.0

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

(a) Descriptive Statistic -Throughput feature

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	5.0	2.0	1.0	2.2	6.0	1.0
2011-1	2.0	2.0	2.0	0	2.0	2.0
2011-2	1.0	6.0	6.0	-	6.0	6.0
2011-3	3.0	1.0	1.0	0	1.0	1.0
2011-4	7.0	3.0	2.0	2.2	7.0	1.0
2012-1	7.0	1.1	1.0	0.4	2.0	1.0
2012-2	15.0	2.8	2.0	1.8	7.0	1.0
2012-3	6.0	1.0	1.0	0	1.0	1.0
2012-4	11.0	3.0	1.0	4.4	16.0	1.0
2013-1	2.0	2.5	2.5	0.7	3.0	2.0
2013-2	1.0	1.0	1.0	-	1.0	1.0
2013-3	4.0	1.2	1.0	0.5	2.0	1.0
2013-4	5.0	2.4	2.0	1.7	5.0	1.0
Total	69.0	2.3	1.0	2.3	16.0	1.0

(b) Descriptive Statistic -Throughput bug

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	5.0	1.6	1.0	0.9	3.0	1.0
2010-4	39.0	2.2	2.0	1.5	7.0	1.0
2011-1	27.0	2.6	2.0	1.9	7.0	1.0
2011-2	20.0	1.7	1.0	1.1	5.0	1.0
2011-3	5.0	1.0	1.0	0	1.0	1.0
2011-4	27.0	2.5	2.0	1.5	6.0	1.0
2012-1	25.0	1.9	1.0	1.3	6.0	1.0
2012-2	37.0	2.7	3.0	1.5	6.0	1.0
2012-3	32.0	1.8	1.0	1.2	6.0	1.0
2012-4	36.0	2.6	2.0	2.6	13.0	1.0
2013-1	23.0	2.9	2.0	1.9	8.0	1.0
2013-2	9.0	2.0	2.0	1.1	4.0	1.0
2013-3	32.0	1.9	2.0	1.2	5.0	1.0
2013-4	18.0	1.7	1.0	1.0	4.0	1.0
Total	335.0	2.2	2.0	1.6	13.0	1.0

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

(a) Descriptive Statistic - Lead time							(b) Descriptive Statistic - Churn						
Quarter	N	Mean	Median	Std.Dev	Max	Min	Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	1.0	18.0	18.0	-	18.0	18.0	2010-3	1.0	5.0	5.0	-	5.0	5.0
2010-4	50.0	10.8	7.5	9.7	42.0	2.0	2010-4	50.0	79.8	15.5	129.3	469.0	1.0
2011-1	39.0	7.7	6.0	6.0	28.0	2.0	2011-1	39.0	53.6	16.0	75.1	266.0	0
2011-2	24.0	12.1	7.0	12.7	44.0	2.0	2011-2	24.0	25.8	1.0	57.0	266.0	0
2011-3	7.0	9.0	6.0	7.2	21.0	3.0	2011-3	7.0	70.0	1.0	131.8	358.0	0
2011-4	62.0	12.6	11.0	9.8	43.0	2.0	2011-4	62.0	31.1	0	69.5	309.0	0
2012-1	32.0	17.4	15.5	11.1	42.0	2.0	2012-1	32.0	29.7	0	84.2	441.0	0
2012-2	111.0	16.8	15.0	10.8	42.0	2.0	2012-2	111.0	34.3	0	84.8	438.0	0
2012-3	43.0	14.2	7.0	12.0	44.0	2.0	2012-3	43.0	25.6	0	60.1	267.0	0
2012-4	97.0	14.1	12.0	10.9	42.0	2.0	2012-4	97.0	60.9	7.0	102.8	401.0	0
2013-1	52.0	9.8	8.0	8.1	37.0	2.0	2013-1	52.0	93.3	23.5	127.9	432.0	0
2013-2	20.0	18.2	15.5	11.8	44.0	2.0	2013-2	20.0	31.9	5.5	45.8	123.0	0
2013-3	44.0	13.3	10.0	11.9	44.0	2.0	2013-3	44.0	67.5	1.0	122.6	469.0	0
2013-4	27.0	12.4	10.0	8.9	33.0	2.0	2013-4	27.0	50.4	7.0	93.4	321.0	0
Total	609.0	13.4	10.0	10.6	44.0	2.0	Total	609.0	50.4	4.0	96.9	469.0	0

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

(a) Descriptive Statistic - Churn feature							(b) Descriptive Statistic - Churn bug						
Quarter	N	Mean	Median	Std.Dev	Max	Min	Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-4	1.0	219.0	219.0	-	219.0	219.0	2010-3	1.0	5.0	5.0	-	5.0	5.0
2011-1	5.0	74.4	14.0	113.7	266.0	0	2010-4	49.0	76.9	15.0	129.0	469.0	1.0
2011-2	3.0	29.3	25.0	27.8	59.0	4.0	2011-1	34.0	50.6	17.0	69.7	241.0	0
2011-3	2.0	47.5	47.5	67.2	95.0	0	2011-2	21.0	25.2	1.0	60.5	266.0	0
2011-4	12.0	25.8	0	89.2	309.0	0	2011-3	5.0	79.0	1.0	156.7	358.0	0
2012-1	1.0	441.0	441.0	-	441.0	441.0	2011-4	50.0	32.4	1.0	64.9	263.0	0
2012-2	34.0	67.8	0	136.0	438.0	0	2012-1	31.0	16.4	0	38.8	195.0	0
2012-3	4.0	79.0	24.5	127.4	267.0	0	2012-2	77.0	19.5	0	40.6	189.0	0
2012-4	18.0	36.3	0	76.7	290.0	0	2012-3	39.0	20.1	0	48.8	253.0	0
2013-1	3.0	45.7	13.0	68.1	124.0	0	2012-4	79.0	66.5	8.0	107.5	401.0	0
2013-2	2.0	48.0	48.0	67.9	96.0	0	2013-1	49.0	96.3	27.0	130.5	432.0	0
2013-3	4.0	199.8	248.5	138.9	302.0	0	2013-2	18.0	30.1	5.5	45.1	123.0	0
2013-4	3.0	78.3	0	135.7	235.0	0	2013-3	40.0	54.3	0	114.5	469.0	0
Total	92.0	66.0	0	119.2	441.0	0	2013-4	24.0	46.9	7.5	90.2	321.0	0
							Total	517.0	47.7	5.0	92.3	469.0	0

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

(a) Descriptive Statistic - Bugs

Quarter	N	Mean	Median	Std.Dev	Max	Min
2010-3	11.0	2.3	1.0	3.0	11.0	1.0
2010-4	32.0	2.6	2.0	1.7	8.0	1.0
2011-1	29.0	2.1	2.0	1.3	6.0	1.0
2011-2	24.0	1.3	1.0	0.7	4.0	1.0
2011-3	15.0	1.5	1.0	0.6	3.0	1.0
2011-4	37.0	2.5	2.0	2.1	9.0	1.0
2012-1	26.0	1.6	1.0	0.9	4.0	1.0
2012-2	34.0	2.0	2.0	1.5	8.0	1.0
2012-3	29.0	1.6	1.0	0.9	4.0	1.0
2012-4	35.0	2.0	1.0	1.5	7.0	1.0
2013-1	29.0	2.3	1.0	2.7	13.0	1.0
2013-2	16.0	1.5	1.0	0.6	3.0	1.0
2013-3	22.0	2.3	2.0	1.8	7.0	1.0
2013-4	19.0	1.4	1.0	0.6	3.0	1.0
Total	370.0	1.9	1.0	1.6	13.0	1.0

(b) Descriptive Statistic - Bugs finished within quarter

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

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

Part I

Notes

6.10.1 WIP correlation

As shown in table 6.51 bugs, throughput, churn, churn bugs and throughput bugs have a significant correlation with WIP. In table 6.52 is descriptive statistic for WIP shown. As shown by the N column all quarters besides 2010-3 give reasonable data since in each quarter has 90 days give or take.

Correlation	Bugs	Throughput	Churn	Churn Bugs	Throughput bugs
WIP	.753*	.637**	.738*	.708*	.800**

Table 6.51: Team one - Correlation - WIP

* Correlation is significant at the 0.05 level (2-tailed).

** Correlation is significant at the 0.01 level (2-tailed).

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
360	2010-3	39	5.82	5	2.55	15	1
	2010-4	92	1.7	2	0.70	4	1
	2011-1	90	4.37	2	6.87	31	1
	2011-2	91	14.25	5	14.497	52	3
	2011-3	92	2.4	3	0.97	4	1
	2011-4	92	14.26	4	22.718	97	1
	2012-1	91	13.35	5	16.62	67	4
	2012-2	91	15.35	3	27.145	94	2
	2012-3	92	16.97	13.5	8.719	52	5
	2012-4	90	9.07	2	17.073	74	1
Total		860	9.99	4	16.04	97	1

Table 6.52: Descriptive statistics for WIP - Team one

6.10.1.1 WIP and throughput correlation

Both throughput and throughput bugs have a significant correlation relationship with WIP. This is inconsistent with Kanban philosophy and what the research says about WIP limit. All the research and the principles of Kanban suggest lowering WIP in order to get high throughput. In team one's case the higher the WIP the higher the throughput.

In table 6.52 the mean column shows indication that team one could have experimented with WIP limit the first five quarters. But in the last quarter, the WIP limit decreased. In the mean column in table 6.53 if one take a look at the quarter 2012-4, the throughput are second highest, the same quarter as WIP decreased. This link could have several reasons. One could be that they experimented with WIP and found out after the quarter 2012-3 that they needed to lower the WIP in order to increase the throughput, which it did. Or maybe team one didn't care about setting a WIP limit as stated in section 2.4.2 by Shinkle.

The throughput bugs variable and throughput variable both have a significant correlation with WIP, which make sense since throughput bugs are a subset of throughput. But throughput feature on the other hand has a -0.441 correlation with WIP. As shown in table 6.56. The correlation number -0.441 is not a significant correlation, but it's on the opposite side level of throughput and throughput feature although it's also a subset of throughput. This also goes in the literatures favor, because when WIP is low, more throughput feature is produced. Maybe this could have something to do with the relationship bugs and feature? Features are often more challenging and more fun to work with than bugs. One could also argue that maybe it's was significant fewer tasks labeled as bug than feature or vice versa, but as shown in table 6.56 that's not the case.

Team	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
360	2010-3	5	2.6	2	1.34	4	1
	2011-1	7	5.71	6	4.03	13	1
	2011-2	26	3.38	3	1.96	8	1
	2011-3	1	1	1	.	1	1
	2011-4	15	6.13	4	4.984	18	2
	2012-1	16	6.56	6.5	3.59	14	1
	2012-2	13	10.15	10	5.742	23	1
	2012-3	22	3.41	2	3.33	12	1
	2012-4	14	8.14	7	6.5	21	1
	Total	119	5.55	4	4.694	23	1

Table 6.53: Descriptive statistic for throughput - team one

Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
2010-3	14.00	3.07	2.00	3.47	14.00	1.00
2010-4	71.00	2.73	2.00	2.05	11.00	1.00
2011-1	67.00	2.55	2.00	1.96	13.00	1.00
2011-2	38.00	1.97	2.00	1.03	4.00	1.00
2011-3	43.00	1.65	1.00	0.92	4.00	1.00
2011-4	53.00	1.83	1.00	1.31	6.00	1.00
2012-1	40.00	1.90	1.00	1.46	7.00	1.00
2012-2	27.00	2.07	2.00	1.36	6.00	1.00
2012-3	40.00	2.70	2.00	2.43	13.00	1.00
2012-4	3.00	1.00	1.00	0	1.00	1.00
Total	396.00	2.26	2.00	1.82	14.00	1.00

Table 6.54: Descriptive statistic for throughput feature - team one

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
360	2010-3	9	1.11	1	0.33	2	1
	2010-4	3	1	1	0	1	1
	2011-2	26	5	3	4.45	17	1
	2011-3	1	1	1	.	1	1
	2011-4	24	6.79	4	6.35	20	1
	2012-1	18	4.83	4	3.20	11	1
	2012-2	19	7.84	9	5.843	17	1
	2012-3	6	2.67	1.5	2.25	6	1
	2012-4	11	3.18	3	1.77	7	1
	Total	117	5.08	3	4.88	20	1

Table 6.55: Descriptive statistic for throughput bugs - team one

Number of features	Number of bugs
660	594

Table 6.56: Number of throughput divided into bugs and feature - Team one

6.10.1.2 WIP and bugs correlation

Only looking at WIP and throughput correlation without taking other factors into account may give bias.

Bugs and churn have also have a significant correlation with WIP. A positive bug correlation could help favor what said about WIP and throughput in literature. Since both WIP and throughput increases and decreases at the same time, the same goes for bug. This could mean that when WIP is high, and throughput is high team one also has a lot of bugs in their code. This means that they don't do their job well, and this could be because they have a lot of task switching due to high WIP limit. As stated in table 6.57 one can see that almost each bug that was reported was finished in the same quarter. The outlier could be the bugs reported just before the quarter end. Worth noticing in quarter 2011-1 there was no bugs recorded. This conclude the statement that the WIP limit is set too high. 0.467

Quarter	Percent of bugs finished in the same quarter as reported
2010-3	100%
2010-4	100%
2011-1	0%
2011-2	97.74%
2011-3	20%
2011-4	97.5%
2012-1	96.51%
2012-2	98.63%
2012-3	82.35%
2012-4	100%

Table 6.57: The percent number of bugs finished in the same quarter as reported - Team one

6.10.1.3 WIP and churn correlation

Churn and WIP have a positive correlation. This means that when team one has more tasks in progress the tasks requires more coding. This is odd and could be a coincidence. Churn and churn bugs have also here as with throughput a close correlation with WIP. But churn feature has a correlation of 0.467

As shown in table 6.58 and the column 'maximum' the biggest feature task requires almost 60 times more code editing than the biggest bug task as showed in table 6.59. In the same table the column std. deviation shows way more proliferation for churn feature. This indicates that the numbers in churn feature are way more spread, which again indicates that it's random of how big the tasks are. In the churn bugs the std. deviation is much lower than in the feature table. This could explain why churn feature not has a positive correlation as churn and churn bugs. Both median and mean favors that churn feature requires more work than churn bugs

Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
2010-3	5	434	106	808.67	1879	30
2011-1	7	0	0	0	0	0
2011-2	26	1199.54	139	4695.424	24139	0
2011-3	1	0	0	.	0	0
2011-4	15	516.4	148	1063.92	3946	0
2012-1	16	5775.25	410	20692.09	83317	0
2012-2	14	7569.14	522.5	24655.70	93108	0
2012-3	23	6.74	0	25.44	118	0
2012-4	16	408	47	1056.23	4275	0
Total	123	2001.29	65	11380.06	93108	0

Table 6.58: Descriptive Statistic for churn feature - Team one

Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
2010-3	7	57.14	22	73.91	193	4
2010-4	2	30	30	41.012	59	1
2011-2	21	386.33	177	431.94	1604	2
2011-3	1	2	2	.	2	2
2011-4	19	306.89	148	437.13	1557	12
2012-1	16	312.81	160	307.73	1035	4
2012-2	13	296.92	169	292.29	1019	18
2012-3	7	37.71	9	56.494	126	0
2012-4	8	118.75	84	103.09	292	3
Total	94	260.48	135	346.25	1604	0

Table 6.59: Descriptive Statistic for churn bugs - Team one

6.10.2 Throughput

In table 6.60 is correlation for throughput for team one stated. Throughput has a positive correlation to bugs, WIP, churn, churn bugs, churn feature, throughput bugs and churn average. In table 6.61 is descriptive statistic for throughput shown. The range from 2011-2 to 2012-4 is the most valuable quarters based on the number of tasks finished if one excludes the quarter 2011-3.

Correlation	Bugs	WIP	Churn	Churn Bugs	Churn feature	Throughput bugs	Churn Average
Throughput	.945**	.816**	.962**	.776**	.736*	.940**	.638*

Table 6.60: Team one - Correlation - Throughput

* Correlation is significant at the 0.05 level (2-tailed).

** Correlation is significant at the 0.01 level (2-tailed).

Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
2010-3	5	2.6	2	1.34	4	1
2011-1	7	5.71	6	4.03	13	1
2011-2	26	3.38	3	1.96	8	1
2011-3	1	1	1	.	1	1
2011-4	15	6.13	4	4.984	18	2
2012-1	16	6.56	6.5	3.59	14	1
2012-2	13	10.15	10	5.742	23	1
2012-3	22	3.41	2	3.33	12	1
2012-4	14	8.14	7	6.5	21	1
Total	119	5.55	4	4.694	23	1

Table 6.61: Descriptive statistic for throughput - team one

6.10.2.1 Throughput and bugs correlation

The correlation between throughput and bugs is 0.945, which is significantly high. The quarters from 2011-2 to 2012-4 minus 2011-3 is also the most valid in bugs. The correlation between bugs and throughput means that when team one has high throughput of tasks they also produce more bugs. This correlation can be used to favor the literature of WIP as stated in section 6.10.1. Since team one produces more bugs when they finished feature or bugs could be an indicator that they may have a too high WIP limit so they cannot focus enough on each task.

Throughput and bugs also have the same relationship as WIP and bugs, because throughput and throughput bugs has a significant correlation to both, but throughput

feature not. Throughput feature have correlation of -0.340 compared to -0.441 in WIP. What this means is that when team one has high throughput of tasks, they mostly produces tasks labeled as bug. This could be because the feature tasks are more comprehensive than the tasks labeled as bug so the lead time is longer for feature tasks than bug tasks. Or it could due the reason that when there is an upcoming release, all the bugs are pushed through the Kanban system.

In table 6.62 the hypotheses that feature tasks is more comprehensive is verified. Four out of the six valid quarters churn feature is significant higher. In table 6.63 one can see that throughput bug's lead time is higher than throughput feature in four out of six quarters. The lead time cannot be counted, since bugs usually have longer lead time than feature, because feature usually has higher priority than bugs.

Quarter	Churn feature	Churn bug
2010-3	0	29.3
2010-4	0	20
2011-1	0	0
2011-2	1199.54	59.14
2011-3	0	2
2011-4	516.4	76.66
2012-1	5775.25	1009.9
2012-2	7569.14	707.46
2012-3	6.74	15.68
2012-4	408	168.28

Table 6.62: team one - Average churn for feature and bugs - Bugs

Quarter	Throughput feature	Throughput bug
2010-3	23	51.33
2011-1	38.85	6
2011-2	19.73	25.46
2011-3	1	5
2011-4	30.6	36.38
2012-1	20.37	45.56
2012-2	35.30	55.89
2012-3	30.72	17.67
2012-4	30.57	8.82
Total	27.30	35.09

Table 6.63: Lead time for throughput feature and bugs per quarter

6.10.2.2 Throughput and churn correlation

Churn, churn bugs, churn feature and the average of churn all have a positive correlation with throughput which mean that when throughput is high, the tasks requires more coding. From table 6.64 in the number column one can see that in each quarter there is a lot of tasks. Although from the median column one can see that in 6 of the valid quarters, three of them have a median of 0, which gives and indicator that in these three quarter there is probably minority of tasks which has high churn. So the correlation between throughput and churn may not be valid.

Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
2010-3	11.00	17.00	11.00	18.94	60.00	0
2011-1	73.00	0	0	0	0	0
2011-2	136.00	14.40	3.00	21.51	83.00	0
2011-3	48.00	1.67	0	7.71	50.00	0
2011-4	156.00	14.81	6.50	19.91	86.00	0
2012-1	157.00	9.47	0	19.55	86.00	0
2012-2	238.00	10.11	0	18.50	83.00	0
2012-3	53.00	3.02	0	9.62	53.00	0
2012-4	26.00	21.42	14.50	20.55	67.00	1.00
Total	902.00	10.14	0	18.55	86.00	0

Table 6.64: Descriptive statistic for churn - team one

6.10.3 Bugs

Correlation	Throughput	WIP	Churn	Churn Bugs	Churn feature	TP bugs
Bugs	.945**	.753*	.992**	.813**	.730*	.984**

Table 6.65: Team two - Correlation - Throughput

* Correlation is significant at the 0.05 level (2-tailed).

** Correlation is significant at the 0.01 level (2-tailed).

It's hard to find any correlation between lead time, WIP and throughput. In quarter three in 2011, lead time, WIP and throughput dropped to their lowest point. This was due to the summer vacation. In quarter two of 2011 and two of 2012 bugs dropped but both Throughput and WIP increased.

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
360	2010-3	7	1.29	1	0.48	2	1
	2010-4	4	1	1	0	1	1
	2011-2	32	4.16	3.5	3.63	14	1
	2011-3	5	1	1	0	1	1
	2011-4	26	6.15	4	5.80	22	1
	2012-1	21	4.09	4	2.7	11	1
	2012-2	17	8.59	9	5.59	19	1
	2012-3	7	2.43	2	1.512	5	1
	2012-4	11	3	3	2	6	1
	Total	131	4.53	3	4.44 t	22	1

Table 6.66: 360 - Descriptive statistic - Bugs

6.11 Team 2

6.11.1 WIP

For team two there is no significant correlation between WIP or any of the variables that have been measured. In table 6.68 one can see that every quarter beside 2010 is a valid quarter based on the number of dates in quarter. An excerpt of the correlation data is shown in table 6.67. As in section 6.10.1 for team one, throughput is fairly high, not significant high as for team one, but fairly high. For team two the correlation between bugs and WIP is quite low, so the conclusion stated in section 6.10.1 may not suit team two.

In table 6.69 the correlation for throughput for team two is shown. Here one can see that throughput and bugs have a significant correlation. That means when WIP increases, throughput increases and when throughput increases the number of bugs reported increases. Which could mean that the WIP limit is to high.

	Throughput	Bugs	Churn	Lead time
WIP	0.341	0.278	-0.446	0.096

Table 6.67: Team two - Correlation - WIP

* Correlation is significant at the 0.05 level (2-tailed).

** Correlation is significant at the 0.01 level (2-tailed).

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
Frontend	2010-3	24	9.28	10	6.43	23	1
	2010-4	92	13.92	13	3.931	25	5
	2011-1	90	15.3	15.5	3.94	23	7
	2011-2	91	23.51	24	4.19	37	13
	2011-3	92	20.13	20	5.43	32	9
	2011-4	92	21.33	21	6.92	34	7
	2012-1	91	22.91	22	6.63	40	11
	2012-2	91	21.93	21	3.36	32	17
	2012-3	92	25.93	27	4.92	36	19
	2012-4	67	19.18	15	9.66	41	9
Total		822	20.18	20	6.93	41	1

Table 6.68: Descriptive statistic - WIP - Team two

Correlation	Bugs	Throughput feature	Bugs finished in the same quarter
Throughput	.885**	.669*	.640*

Table 6.69: Team one - Correlation - Throughput

* Correlation is significant at the 0.05 level (2-tailed).

** Correlation is significant at the 0.01 level (2-tailed).

6.11.2 Lead time

Correlation	Churn Bugs	Churn union	Precent bugs fininshed	Churn Average
Lead time	-.771**	.704*	-.697*	.766**

Table 6.70: Team one - Correlation - Lead time_union

Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
2010-3	14.00	3.07	2.00	3.47	14.00	1.00
2010-4	71.00	2.73	2.00	2.05	11.00	1.00
2011-1	67.00	2.55	2.00	1.96	13.00	1.00
2011-2	38.00	1.97	2.00	1.03	4.00	1.00
2011-3	43.00	1.65	1.00	0.92	4.00	1.00
2011-4	53.00	1.83	1.00	1.31	6.00	1.00
2012-1	40.00	1.90	1.00	1.46	7.00	1.00
2012-2	27.00	2.07	2.00	1.36	6.00	1.00
2012-3	40.00	2.70	2.00	2.43	13.00	1.00
2012-4	3.00	1.00	1.00	0	1.00	1.00
Total	396.00	2.26	2.00	1.82	14.00	1.00

Table 6.71: Descriptive statistic - Throughput - Team two

6.11.3 Analyzed data per quarter

In figure 6.1 the average WIP, Throughput, Lead time and Bugs per quarter for team 360 are shown. As stated in section 2.4.2, to get best possible throughput the WIP limit should be low. According to the figure 6.1, when WIP is high, throughput is also high and when WIP is low, throughput also low. This team data is contrary to what is suggested from Kanban and research.

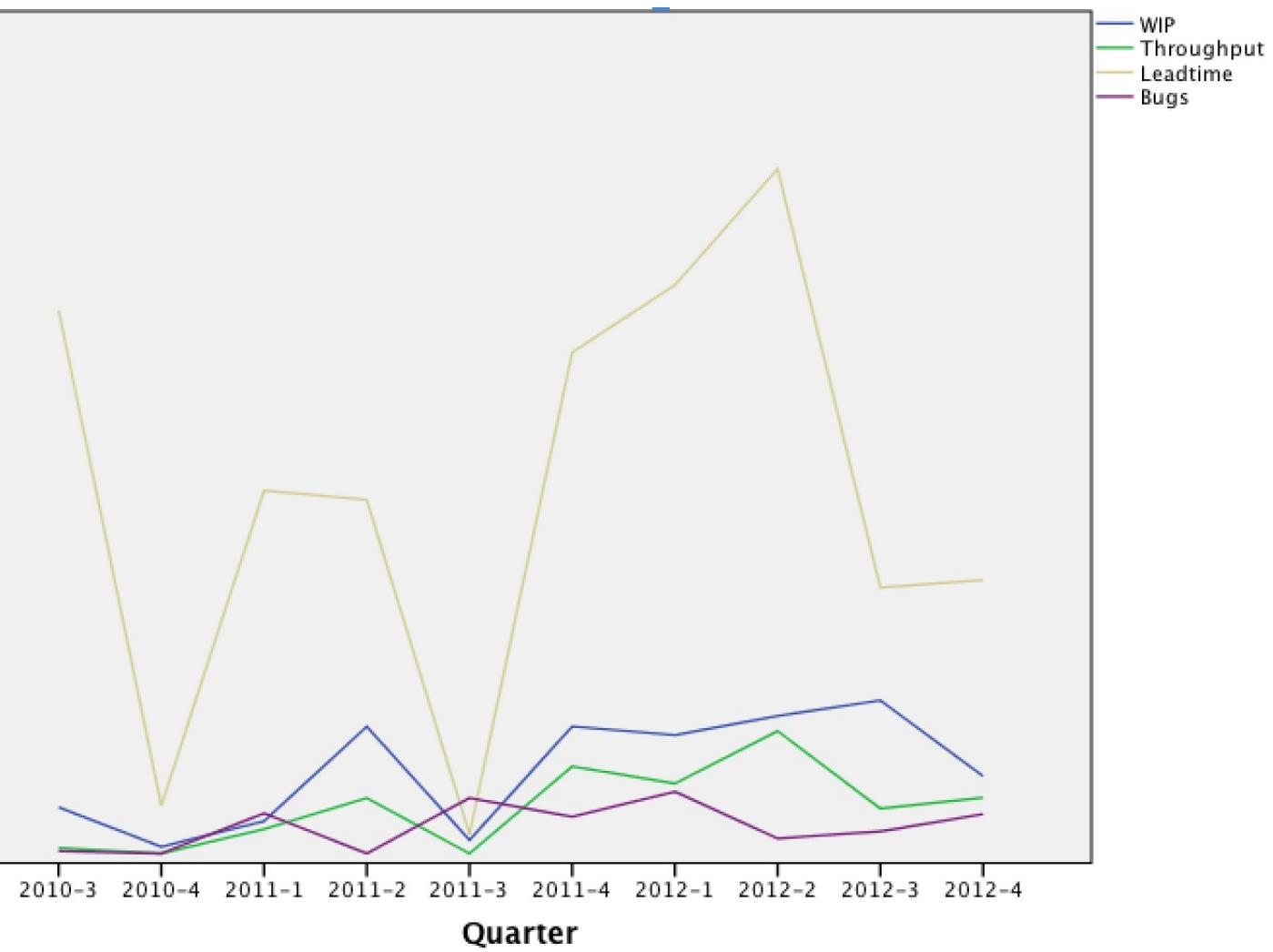


Figure 6.1: WIP, Throughput, Lead time and Bugs for 360 per quarter

In figure 6.2 the average WIP, Throughput, Lead time and Bugs per quarter for team Frontend are shown

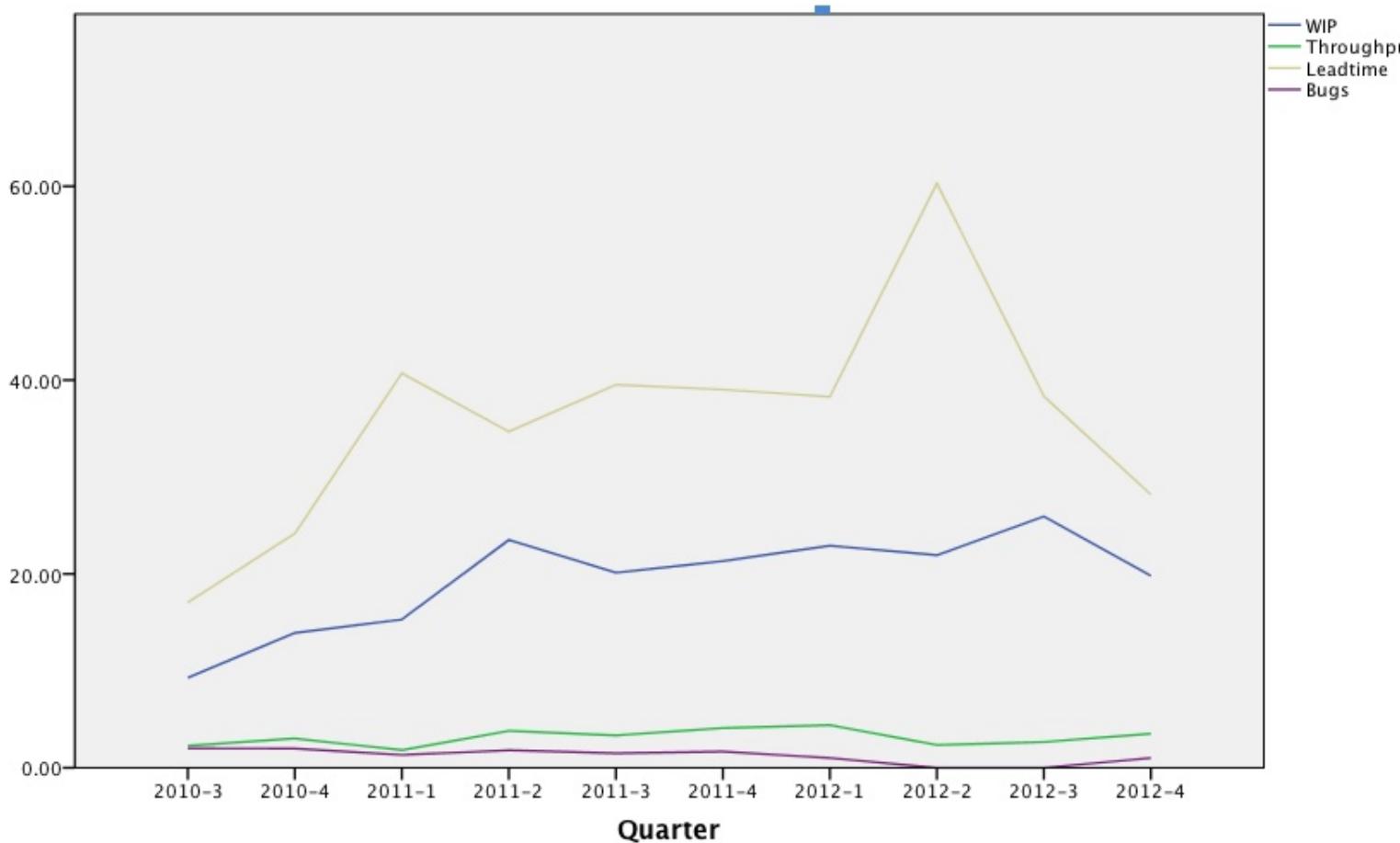


Figure 6.2: WIP, Throughput, Lead time and Bugs for Frontend per quarter

In figure 6.3 the average WIP, Throughput, Lead time and Bugs per quarter for team Krypton are shown

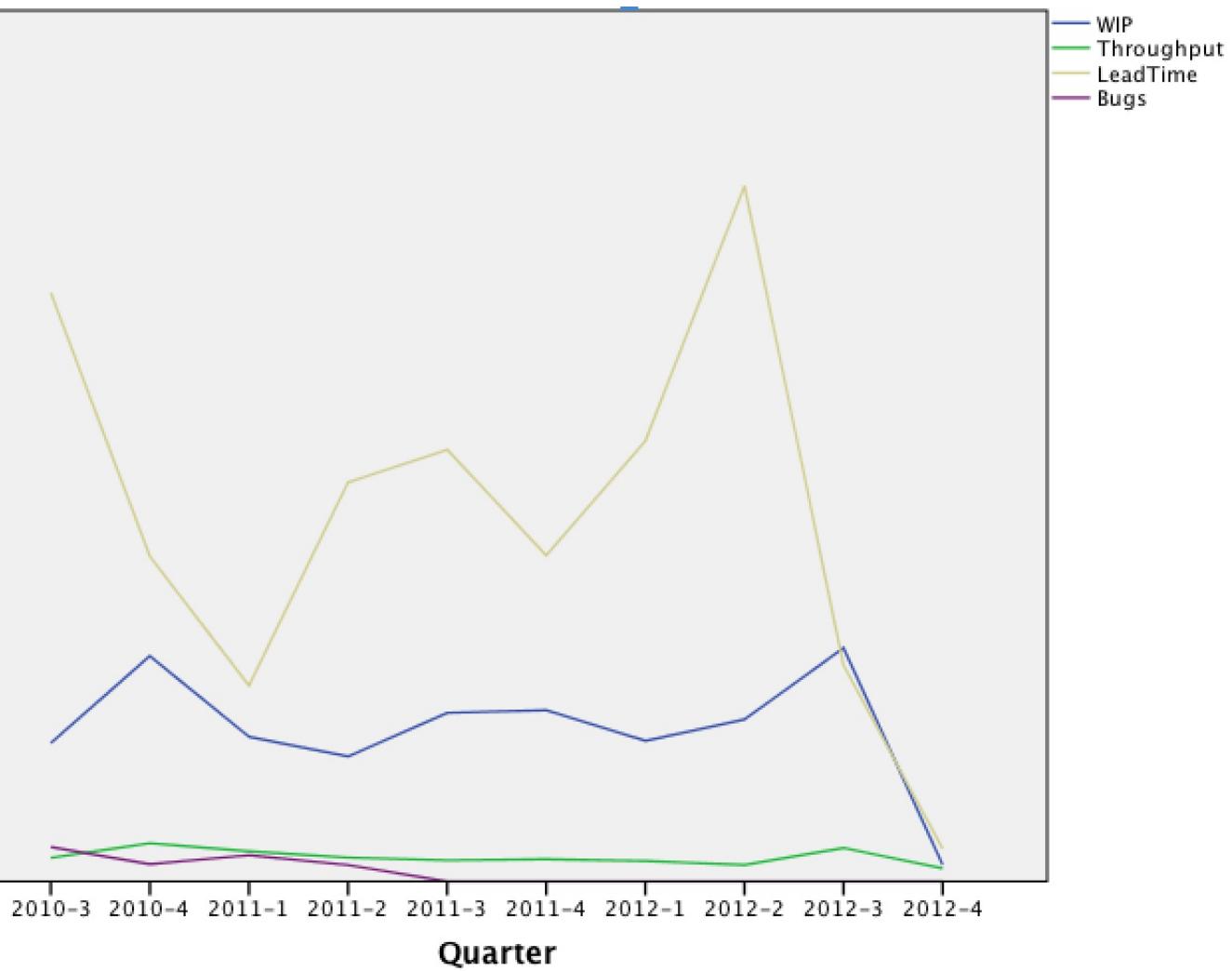
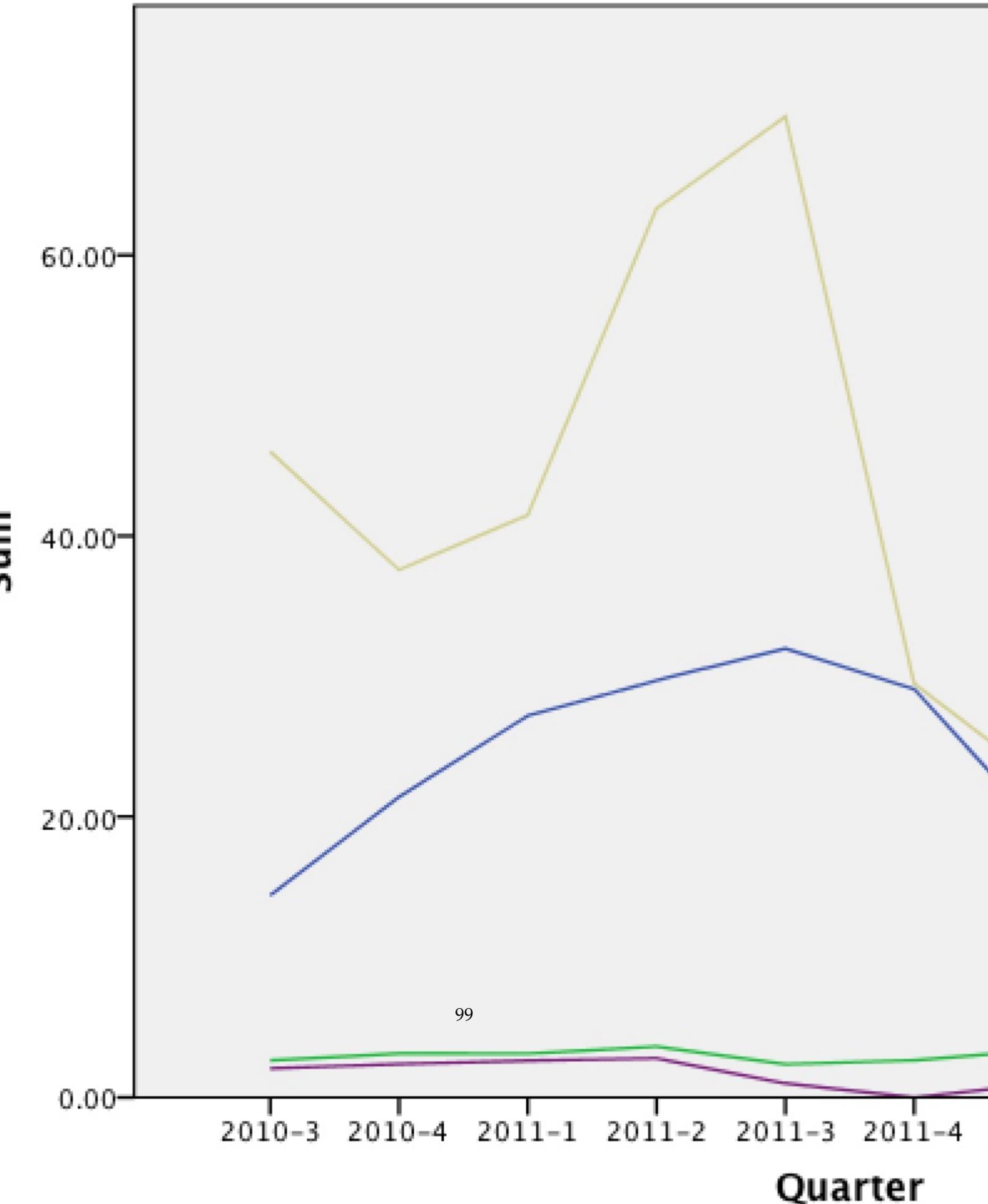


Figure 6.3: WIP, Throughput, Lead time and Bugs for Krypton per quarter

In figure 6.4 the average WIP, Throughput, Lead time and Bugs per quarter for team Neon are shown



Til Dag

6.12 360

Correlations		Bugs	Throughput	WIP	Churn	TP_Feature	Churn_Bugs	Lead_time_Union	Churn_Union	Churn_feature	TP_bugs	Average_Days_Backlog	Bugs_churn_average	Average_ft_churn	Percent_bugs_finished	Churn_Average	
Bugs		Pearson Correlation	1	.945**	.753*	.992**	-.328	.813**	-.240	0.433	.730*	.984**	0.377	0.581	0.119	0.543	0.622
Sig. (2-tailed)			0	0.012	0.012	0.034	0.004	0.505	0.212	0.026	0	0.282	0.078	0.744	0.104	0.055	
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	
Throughput		Pearson Correlation	.945**	1	.816**	.962**	-.340	.776**	-.177	0.399	.736*	.940**	0.274	0.626	0.030	0.420	.638*
Sig. (2-tailed)			0	0	0.004	0	0.036	0.008	0.624	0.253	0.024	0	0.443	0.053	0.924	0.227	0.047
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	
WIP		Pearson Correlation	.753*	.816**	1	.738*	-.441	.708*	-.336	0.365	.0467	.800**	0.297	0.430	0.243	0.495	0.478
Sig. (2-tailed)			0.012	0.004	0	0.015	0.202	0.022	0.342	0.299	0.205	0.005	0.255	0.215	0.498	0.146	0.363
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	
Churn		Pearson Correlation	.992**	.962**	.738*	1	-.317	.804**	-.203	0.372	.741*	.974**	0.349	0.581	0.087	0.475	0.617
Sig. (2-tailed)			0	0	0.015	0.015	0.372	0.005	0.574	0.289	0.022	0	0.323	0.078	0.811	0.165	0.057
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	
TP_Feature		Pearson Correlation	-.328	-.340	-.441	-.317	1	-.419	0.376	-.460	-.208	-.360	0.097	-.171	-.295	-.795**	-.223
Sig. (2-tailed)			0.354	0.336	0.202	0.372	0	0.228	0.284	0.181	0.307	0.789	0.637	0.408	0.006	0.536	0.006
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	
Churn_Bugs		Pearson Correlation	.813**	.776**	.708*	.804*	-.419	1	-.126	0.516	.597	.884**	.700*	0.560	0.523	0.551	.665*
Sig. (2-tailed)			0.004	0.008	0.022	0.005	0.228	0	0.729	0.127	0.090	0.001	0.024	0.092	0.121	0.099	0.036
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	
Lead_time_Union		Pearson Correlation	-.240	-.177	-.336	-.203	0.376	-.126	1	-.060	0.138	-.280	-.012	-.003	0.158	-.598	0.059
Sig. (2-tailed)			0.505	0.624	0.342	0.574	0.284	0.729	0	0.868	0.724	0.433	0.974	0.994	0.663	0.068	0.871
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	
Churn_Union		Pearson Correlation	0.433	0.399	0.365	0.372	-.460	0.516	-.060	1	0.202	0.466	0.231	0.146	0.410	.634*	0.198
Sig. (2-tailed)			0.212	0.253	0.299	0.289	0.181	0.127	0.868	0.603	0.175	0.522	0.687	0.239	0.049	0.584	0.049
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	
Churn_feature		Pearson Correlation	.730*	.736*	0.467	.741*	-.208	.597	0.138	0.202	1	.681*	0.073	.924**	0.008	0.362	.947**
Sig. (2-tailed)			0.026	0.024	0.205	0.022	0.591	0.090	0.724	0.603	0	0.044	0.852	0	0.984	0.338	0
N			9	9	9	9	9	9	9	9	9	9	9	9	9	9	
TP_bugs		Pearson Correlation	.984**	.940**	.800**	.974**	-.360	.384**	-.280	0.466	.681*	1	0.494	0.574	0.192	0.568	0.620
Sig. (2-tailed)			0	0	0.005	0	0.307	0.001	0.433	0.175	0.044	0	0.146	0.083	0.596	0.087	0.056
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	
Average_Days_Backlog		Pearson Correlation	0.377	0.274	0.397	0.349	0.097	.700*	-.012	0.231	0.073	0.494	1	0.101	0.595	0.130	0.217
Sig. (2-tailed)			0.282	0.443	0.255	0.323	0.789	0.024	0.974	0.522	0.852	0.146	0.782	0.070	0.721	0.548	0.548
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	
Bugs_churn_average		Pearson Correlation	0.581	0.626	0.430	0.581	-.171	0.560	-.003	0.146	.924**	0.574	0.101	1	-.127	0.322	.973**
Sig. (2-tailed)			0.078	0.053	0.215	0.078	0.637	0.092	0.994	0.687	0	0.083	0.782	0	0.726	0.365	0
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	
Average_ft_churn		Pearson Correlation	0.119	0.030	0.243	0.087	-.295	0.523	0.158	0.410	0.008	0.192	0.595	-.127	1	0.291	0.088
Sig. (2-tailed)			0.744	0.934	0.498	0.811	0.408	0.121	0.663	0.239	0.984	0.596	0.070	0.726	0.415	0.808	0.088
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	
Precent_bugs_finished		Pearson Correlation	0.543	0.420	0.495	0.475	-.795**	0.551	-.598	.634*	0.362	0.568	0.130	0.322	0.291	1	0.369
Sig. (2-tailed)			0.104	0.227	0.14	0.165	0.006	0.099	0.068	0.049	0.338	0.087	0.721	0.365	0.415	0.294	0.294
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	
Churn_Average		Pearson Correlation	0.622	.638*	0.478	0.617	-.223	.665*	0.059	0.198	.947**	0.620	0.217	.973**	0.088	0.369	1
Sig. (2-tailed)			0.055	0.047	0.163	0.057	0.536	0.036	0.871	0.584	0	0.056	0.548	0	0.808	0.294	0.294
N			10	10	10	10	10	10	10	9	10	10	10	10	10	10	

** Correlation is significant at the 0.01 level (2-tailed).

* Correlation is significant at the 0.05 level (2-tailed).

Table 6.72: 360 - Correlation

Team	Quarter		Churn	Lead time
360	2010-3	N	11	11
		Mean	17	3.55
		Median	11	3
		Std. Deviation	18.93	2.50
		Maximum	60	8
		Minimum	0	1
	2010-4	N	3	3
		Mean	0.67	1.67
		Median	1	2
		Std. Deviation	0.57	0.57
		Maximum	1	2
		Minimum	0	1
		N	136	136
	2011-2	Mean	14.4	2.7
		Median	3	2
		Std. Deviation	21.51	1.85
		Maximum	83	8
		Minimum	0	1
		N	48	48
	2011-3	Mean	1.67	3.06
		Median	0	2.5
		Std. Deviation	7.71	2.13
		Maximum	50	8
		Minimum	0	1
		N	156	156
	2011-4	Mean	14.81	2.25
		Median	6.5	2
		Std. Deviation	19.90	1.53
		Maximum	86	8
		Minimum	0	1
		N	157	157
	2012-1	Mean	9.47	2.68
		Median	0	2
		Std. Deviation	19.54	1.758
		Maximum	86	8
		Minimum	0	1
		N	238	238
	2012-2	Mean	10.11	2.78
		Median	0	2
		Std. Deviation	18.50	1.597
		Maximum	83	8
		Minimum	0	1
		N	53	53
	2012-3	Mean	3.02	1.68
		Median	0	1
		Std. Deviation	9.61	1.18
		Maximum	53	7
		Minimum	0	1
		N	26	26
	2012-4	Mean	21.42	2.08
		Median	14.5	2
		Std. Deviation	20.54	1.16
		Maximum	67	5
		Minimum	1	1
		N	902	902
	Total	Mean	10.14	2.7
		Median	0	2
		Std. Deviation	18.55	1.798
		Maximum	86	8
		Minimum	0	1

Table 6.73: Descriptive statistic - Lead-time and churn - 360

Frontend

Correlations			WIP	Throughput	Bugs	Churn	TP_feature	Churn_Bugs	Churn_union	Lead_time_union	Churn_feature	Tp_bugs	Average_Days_Backlog_Bugs	Bugs_Churn_average	Churn_feature_Average	Precent_bugs_fininished	Churn_Average
WIP	Pearson Correlation	1	0.491	0.278	-0.446	0.380	-0.446	0.316	0.096	0.362	0.508	-0.178	0.257	0.337	0.367	0.223	
	Sig. (2-tailed)		0.150	0.436	0.196	0.279	0.196	0.373	0.791	0.304	0.134	0.622	0.473	0.341	0.297	0.537	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Throughput	Pearson Correlation	0.491	1	.773**	-0.036	.741*	0.217	-0.494	-0.539	-0.433	.758*	0.055	0.259	-0.609	.745*	-0.575	
	Sig. (2-tailed)		0.150	0.009	0.921	0.014	0.548	0.147	0.108	0.212	0.011	0.879	0.470	0.062	0.013	0.082	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Bugs	Pearson Correlation	0.278	.773**	1	0.259	0.534	0.398	-0.555	-0.581	-0.478	0.271	0.005	0.296	-0.571	0.498	-0.624	
	Sig. (2-tailed)		0.436	0.009	0.471	0.111	0.254	0.096	0.078	0.162	0.449	0.988	0.407	0.085	0.143	0.054	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Churn	Pearson Correlation	-0.446	-0.036	0.259	1	-0.219	.791**	-0.522	-0.449	-0.517	-0.110	-0.174	0.172	-0.485	-0.014	-0.471	
	Sig. (2-tailed)		0.196	0.921	0.471	0.543	0.006	0.122	0.192	0.126	0.762	0.631	0.634	0.155	0.970	0.170	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
TP_feature	Pearson Correlation	0.380	.741*	0.534	-0.219	1	0.214	-0.244	-0.312	-0.026	0.427	-0.256	-0.007	-0.294	.640*	-0.319	
	Sig. (2-tailed)		0.279	0.014	0.111	0.543	0.553	0.497	0.381	0.943	0.219	0.475	0.984	0.409	0.046	0.369	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Churn_Bugs	Pearson Correlation	-0.446	0.217	0.398	.791**	0.214	1	-.707*	-.771**	-0.460	0.040	-0.378	-0.035	-0.590	0.388	-.689*	
	Sig. (2-tailed)		0.196	0.548	0.254	0.006	0.553	0.022	0.009	0.181	0.912	0.282	0.924	0.073	0.267	0.027	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Churn_union	Pearson Correlation	0.316	-0.494	-0.555	-0.522	-0.244	-.707*	1	0.704*	.844**	-0.368	-0.139	0.152	.841**	-0.312	.981**	
	Sig. (2-tailed)		0.373	0.147	0.096	0.122	0.497	0.022	0.023	0.002	0.296	0.702	0.674	0.002	0.379	0	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Lead_time_union	Pearson Correlation	0.096	-0.539	-0.581	-0.449	-0.312	-.771**	0.704*	1	0.492	-0.301	0.327	-0.227	0.631	-.697*	.766**	
	Sig. (2-tailed)		0.791	0.108	0.078	0.192	0.381	0.009	0.023	0.148	0.398	0.357	0.528	0.050	0.025	0.010	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Churn_feature	Pearson Correlation	0.362	-0.433	-0.478	-0.517	-0.026	-0.460	.844**	0.492	1	-0.314	-0.358	-0.201	.921**	-0.172	.842**	
	Sig. (2-tailed)		0.304	0.212	0.162	0.126	0.943	0.181	0.002	0.148	0.377	0.310	0.579	0	0.635	0.002	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Tp_bugs	Pearson Correlation	0.508	.758*	0.271	-0.110	0.427	0.040	-0.368	-0.301	-0.314	1	0.096	0.034	-0.393	0.622	-0.406	
	Sig. (2-tailed)		0.134	0.011	0.449	0.762	0.219	0.912	0.296	0.398	0.377	0.792	0.926	0.261	0.055	0.245	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Average_Days_Backlog_Bugs	Pearson Correlation	-0.178	0.055	0.005	-0.174	-0.256	-0.378	-0.139	0.327	-0.358	0.096	1	-0.267	-0.272	-0.489	-0.035	
	Sig. (2-tailed)		0.622	0.879	0.988	0.631	0.475	0.282	0.702	0.357	0.310	0.792	0.456	0.446	0.151	0.925	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Bugs_Churn_average	Pearson Correlation	0.257	0.259	0.296	0.172	-0.007	-0.035	0.152	-0.227	-0.201	0.034	-0.267	1	-0.197	0.371	0.016	
	Sig. (2-tailed)		0.473	0.470	0.407	0.634	0.984	0.924	0.674	0.528	0.579	0.926	0.456	0.585	0.291	0.966	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Churn_feature_Average	Pearson Correlation	0.337	-0.609	-0.571	-0.485	-0.294	-0.590	.841**	0.631	.921**	-0.393	-0.272	-0.197	1	-0.389	.846**	
	Sig. (2-tailed)		0.341	0.062	0.085	0.155	0.409	0.073	0.002	0.050	0	0.261	0.446	0.585	0.267	0.002	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Precent_bugs_finished	Pearson Correlation	0.367	.745*	0.498	-0.014	.640*	0.388	-0.312	-.697*	-0.172	0.622	-0.489	0.371	-0.389	1	-0.445	
	Sig. (2-tailed)		0.297	0.013	0.143	0.970	0.046	0.267	0.379	0.025	0.635	0.055	0.151	0.291	0.267	0.197	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Churn_Average	Pearson Correlation	0.223	-0.575	-0.624	-0.471	-0.319	-.689*	.981**	.766**	.842**	-0.406	-0.035	0.016	.846**	-0.445	1	
	Sig. (2-tailed)		0.537	0.082	0.054	0.170	0.369	0.027	0	0.010	0.002	0.245	0.925	0.966	0.002	0.197	
N		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	

** Correlation is significant at the 0.01 level (2-tailed).

* Correlation is significant at the 0.05 level (2-tailed).

Table 6.74: Frontend Correlation

Team	Quarter		Lead time	Churn
Frontend	2010-3	N	58	58
		Mean	7.43	127.93
		Median	6	53.5
		Std. Deviation	5.567	187.011
		Maximum	22	1072
		Minimum	1	0
	2010-4	N	130	130
		Mean	10.32	158.72
		Median	4	14
		Std. Deviation	15.30	444.805
		Maximum	78	4308
		Minimum	1	0
	2011-1	N	95	95
		Mean	28.01	566.73
		Median	11	47
		Std. Deviation	52.183	3696.125
		Maximum	434	35917
		Minimum	1	0
	2011-2	N	132	132
		Mean	12.17	496.8
		Median	5	19
		Std. Deviation	23.548	3240.172
		Maximum	184	35956
		Minimum	1	0
	2011-3	N	146	146
		Mean	10.66	174.07
		Median	6	14
		Std. Deviation	12.577	655.578
		Maximum	79	5650
		Minimum	1	0
	2011-4	N	167	167
		Mean	9.85	110.98
		Median	7	3
		Std. Deviation	10.9	480.32
		Maximum	62	5672
		Minimum	1	0
	2012-1	N	188	188
		Mean	13.9	118.64
		Median	8	5.5
		Std. Deviation	21.92	568.03
		Maximum	150	7565
		Minimum	1	0
	2012-2	N	77	77
		Mean	22.53	849.42
		Median	17	15
		Std. Deviation	25.52	3575.50
		Maximum	106	27170
		Minimum	1	0
	2012-3	N	93	93
		Mean	13.76	458.92
		Median	8	25
		Std. Deviation	18.023	1272.32
		Maximum	107	8600
		Minimum	1	0
	2012-4	N	82	82
		Mean	8.8007	428.67
		Median	4	54
		Std. Deviation	14.978	1178.471
		Maximum	83	9368
		Minimum	1	0
	Total	N	1168	1168
		Mean	13.35	305.62
		Median	7	15
		Std. Deviation	23.15	1883.16
		Maximum	434	35956
		Minimum	1	0

104

Table 6.75: Descriptive statistic - Lead-time and churn - Frontend

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
Frontend	2010-3	14	2.79	2	2.91	12	1
	2010-4	39	2	1	1.53	8	1
	2011-1	20	1.85	1	1.226	5	1
	2011-2	29	2.86	2	3.21	16	1
	2011-3	37	2.41	2	1.93	10	1
	2011-4	35	2.89	1	5.02	30	1
	2012-1	39	3.56	2	3.90	23	1
	2012-2	31	1.81	1	1.4	7	1
	2012-3	24	2.54	2	1.79	7	1
	2012-4	14	2.86	2	1.79	7	1
Total		282	2.56	2	2.895	30	1

Table 6.76: Frontend - Descriptive statistic - Bugs

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
Frontend	2010-3	8	1.25	1	0.46	2	1
	2010-4	35	1.63	1	0.877	4	1
	2011-1	32	1.53	1	0.67	3	1
	2011-2	31	1.74	2	0.89	4	1
	2011-3	31	1.97	2	1.25	6	1
	2011-4	28	1.93	1	1.94	11	1
	2012-1	22	2.18	1	2.01	7	1
	2012-2	23	1.48	1	1.123	6	1
	2012-3	21	1.48	1	0.873	4	1
	2012-4	19	2.47	2	1.679	8	1
Total		250	1.78	1	1.294	11	1

Table 6.77: Frontend - Descriptive statistic - TP feature

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
Frontend	2010-3	15	2	2	1.363	6	1
	2010-4	41	1.98	2	1.387	8	1
	2011-1	18	1.33	1	0.48	2	1
	2011-2	35	2.50	2	2.161	10	1
	2011-3	38	2.18	2	1.60	8	1
	2011-4	40	2.6	2	2.52	14	1
	2012-1	41	3.44	3	2.61	10	1
	2012-2	29	1.9	1	1.472	7	1
	2012-3	30	2.13	1.5	1.69	8	1
	2012-4	25	2.12	1	2.10	10	1
Total		312	2.31	2	1.982	14	1

Table 6.78: Frontend - Descriptive statistic - TP bugs

6.13 Krypton

Correlations		WIP	Throughput	Bugs	Churn	TP_feature	Churn_Bugs	Churn_union	Lead_time_union	Churn_feature	Tp_bugs	Average_Days_Backlog_Bugs	Bugs_Churn_average	Churn_feature_Average	Precent_bugs_finished	Churn	
WIP	Pearson Correlation	1	0.491	0.278	-0.446	0.380	-0.446	0.316	0.096	0.362	0.508	-0.178	0.257	0.337	0.267	0.22	
	Sig. (2-tailed)	0.150	0.436	0.196	0.279	0.196	0.373	0.791	0.304	0.134	0.622	0.473	0.341	0.297	0.533		
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
	Throughput	0.491	1	.773**	-0.036	.741*	0.217	-0.494	-0.539	-0.433	.78*	0.055	0.259	-0.609	.745*	-0.52	
	Sig. (2-tailed)	0.150	0.009	0.921	0.014	0.548	0.147	0.108	0.212	0.011	0.879	0.470	0.062	0.013	0.082		
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
	Bugs	Pearson Correlation	0.278	.773**	1	0.259	0.534	0.398	-0.555	-0.581	-0.478	0.271	0.005	0.296	-0.571	0.498	-0.62
	Sig. (2-tailed)	0.436	0.009	0.471	0.111	0.254	0.096	0.078	0.162	0.049	0.988	0.407	0.085	0.143	0.05	0.143	
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
	Churn	Pearson Correlation	-0.446	-0.036	0.259	1	-0.219	.791**	-0.522	-0.449	-0.517	-0.110	-0.174	0.172	-0.485	-0.014	-0.47
	Sig. (2-tailed)	0.196	0.921	0.471	0.543	0.006	0.122	0.192	0.126	0.762	0.631	0.634	0.155	0.970	0.17		
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
TP_feature	Pearson Correlation	0.380	.741*	0.534	-0.219	1	0.214	-0.244	-0.312	-0.026	0.427	-0.256	-0.007	-0.294	.640*	-0.31	
	Sig. (2-tailed)	0.279	0.014	0.111	0.543	0.553	0.497	0.381	0.943	0.219	0.475	0.984	0.409	0.046	0.369		
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
	Churn_Bugs	Pearson Correlation	-0.446	0.217	0.398	.791**	0.214	1	-.707*	-.771**	-0.460	0.040	-0.378	-0.035	-0.590	0.388	-.689
	Sig. (2-tailed)	0.196	0.548	0.254	0.006	0.553	0.022	0.009	0.181	0.912	0.282	0.924	0.073	0.267	0.02		
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
	Churn_union	Pearson Correlation	0.316	-0.494	-0.555	-0.522	-0.244	-.707*	1	.704*	.844**	-0.368	-0.139	0.152	.841**	-0.312	0.981
	Sig. (2-tailed)	0.373	0.147	0.096	0.122	0.497	0.022	0.023	0.002	0.296	0.702	0.674	0.002	0.379	0		
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
	Lead_time_union	Pearson Correlation	0.096	-0.539	-0.581	-0.449	-0.312	-.771**	.704*	1	0.492	-0.301	0.327	-0.227	0.631	-.697*	0.766
	Sig. (2-tailed)	0.791	0.108	0.078	0.192	0.381	0.009	0.023	0.148	0.398	0.357	0.528	0.050	0.025	0.016	0.016	
Churn_feature	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
	Pearson Correlation	0.362	-0.433	-.478*	-0.517	-0.026	-0.460	.844**	0.492	1	-0.314	-0.358	-0.201	.921**	-0.172	.842	
	Sig. (2-tailed)	0.304	0.212	0.162	0.126	0.943	0.181	0.002	0.148	1	0.377	0.310	0.579	0	0.635	0.00	
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
	Tp_bugs	Pearson Correlation	0.508	.758*	.271	-0.110	0.427	0.040	-0.368	-0.314	1	0.096	0.034	-0.393	0.622	-0.40	
	Sig. (2-tailed)	0.134	0.011	0.449	0.762	0.219	0.912	0.296	0.398	0.377	0.792	0.926	0.261	0.055	0.24		
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
	Average_Days_Backlog_Bugs	Pearson Correlation	-0.178	0.055	0.005	-0.174	-0.256	-0.378	-0.139	0.327	-0.358	0.096	1	-0.267	-0.272	-0.489	0.003
	Sig. (2-tailed)	0.622	0.879	0.988	0.631	0.475	0.282	0.702	0.357	0.310	0.792	0.456	0.446	0.151	0.929		
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
Bugs_Churn_average	Pearson Correlation	0.257	0.259	0.296	0.172	-0.007	-0.035	0.152	-0.227	-0.201	0.034	-0.267	1	-0.197	0.371		
	Sig. (2-tailed)	0.473	0.470	0.407	0.634	0.984	0.924	0.674	0.528	0.579	0.926	0.456	0.585	0.291	0.96		
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
	Churn_feature_Average	Pearson Correlation	0.337	-0.609	-0.571	-0.485	-0.294	-0.590	.841**	0.631	.921**	-0.393	-0.272	-0.197	1	-0.389	0.846
	Sig. (2-tailed)	0.341	0.062	0.088	0.155	0.409	0.073	0.002	0.050	0	0.261	0.446	0.585	0.267	0.00		
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
	Precent_bugs_finished	Pearson Correlation	0.367	.745*	0.498	-0.014	.640*	0.388	-0.312	-.697*	-0.172	0.622	-0.489	0.371	-0.389	1	-0.44
	Sig. (2-tailed)	0.297	0.013	0.143	0.970	0.046	0.267	0.379	0.025	0.635	0.055	0.151	0.291	0.267	0.19		
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	
	Churn_Average	Pearson Correlation	0.223	-0.575	-0.624	-0.471	-0.319	-.689*	.981**	.766**	.842**	-0.406	-0.035	0.016	.846**	-0.445	1
	Sig. (2-tailed)	0.537	0.082	0.054	0.170	0.369	0.027	0	0.010	0.002	0.245	0.925	0.966	0.002	0.197		
	N	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	

*Correlation is significant at the 0.05 level (2-tailed).

Table 6.79: Krypton - correlation

Team	Quarter		Lead time	Churn
Krypton	2010-3	N	46	46
		Mean	7.89	159.38
		Median	3.5	26.5
		Std. Deviation	11.651	206.584
		Maximum	71	647
		Minimum	1	0
	2010-4	N	139	139
		Mean	7.38	97.7
		Median	3	28
		Std. Deviation	12.88	145.84
		Maximum	85	700
		Minimum	1	0
	2011-1	N	153	153
		Mean	5.83	71.38
		Median	2	17
		Std. Deviation	7.383	130.40
		Maximum	32	726
		Minimum	1	0
	2011-2	N	51	51
		Mean	14.37	110.45
		Median	9	31
		Std. Deviation	16.09	165.053
		Maximum	90	719
		Minimum	1	0
	2011-3	N	49	49
		Mean	13.71	119.43
		Median	8	27
		Std. Deviation	15.97	159.39
		Maximum	88	604
		Minimum	1	1
	2011-4	N	66	66
		Mean	10.5	112.36
		Median	6	36
		Std. Deviation	11.975	164.935
		Maximum	57	675
		Minimum	1	1
	2012-1	N	45	45
		Mean	23.73	99.09
		Median	8	23
		Std. Deviation	40.47	149.02
		Maximum	180	636
		Minimum	1	1
	2012-2	N	45	45
		Mean	23	109.07
		Median	7	47
		Std. Deviation	32.08	131.58
		Maximum	135	536
		Minimum	1	1
	2012-3	N	74	74
		Mean	6.92	82.39
		Median	3	37
		Std. Deviation	9.26	122.637
		Maximum	35	713
		Minimum	1	1
	2012-4	N	2	2
		Mean	1.5	44
		Median	1.5	44
		Std. Deviation	0.70	1.41
		Maximum	2	45
		Minimum	1	43
	Total	N	683	683
		Mean	10.55	98.6
		Median	4	29
		Std. Deviation	18.1	149.01
		Maximum	180	726
		Minimum	1	0

Table 6.80: Descriptive statistic - Lead-time and churn - Krypton

Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
2010-3	14.00	3.07	2.00	3.47	14.00	1.00
2010-4	71.00	2.73	2.00	2.05	11.00	1.00
2011-1	67.00	2.55	2.00	1.96	13.00	1.00
2011-2	38.00	1.97	2.00	1.03	4.00	1.00
2011-3	43.00	1.65	1.00	0.92	4.00	1.00
2011-4	53.00	1.83	1.00	1.31	6.00	1.00
2012-1	40.00	1.90	1.00	1.46	7.00	1.00
2012-2	27.00	2.07	2.00	1.36	6.00	1.00
2012-3	40.00	2.70	2.00	2.43	13.00	1.00
2012-4	3.00	1.00	1.00	0	1.00	1.00
Total	396.00	2.26	2.00	1.82	14.00	1.00

Table 6.81: Frontend - Descriptive statistic - Throughput

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
Krypton	2010-3	25	14.12	14	5.093	27	9
	2010-4	92	23	23.5	7.819	48	12
	2011-1	90	14.76	13.5	5	26	6
	2011-2	91	12.75	12	4.64	22	3
	2011-3	92	17.2	16	6.476	32	8
	2011-4	92	17.47	17	6.32	30	4
	2012-1	91	14.35	15	3.911	24	7
	2012-2	91	16.53	14	6.03	34	10
	2012-3	92	23.83	20	10.387	43	7
	2012-4	67	1.69	0	3.016	11	0
Total		823	16.11	15	8.43	48	0

Table 6.82: Krypton - Descriptive statistic WIP

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
Krypton	2010-3	17	1.76	2	0.83	3	1
	2010-4	28	1.71	1.5	0.85	4	1
	2011-1	39	3.1	2	2.51	12	1
	2011-2	26	2.04	2	1.28	5	1
	2011-3	25	1.64	1	1.14	5	1
	2011-4	26	2	2	1.35	7	1
	2012-1	26	1.85	1	1.617	9	1
	2012-2	19	2.52	2	2.48	11	1
	2012-3	31	2.16	2	1.44	6	1
	Total	241	2.12	2	1.69	12	1
	2010-3	20	2.65	2	3.54	17	1

Table 6.83: Krypton - Descriptive statistic - Bugs

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
Krypton	2010-3	10	2.20	1.5	1.619	5	1
	2010-4	48	2.79	2	2.278	11	1
	2011-1	30	1.7	1	0.95	4	1
	2011-2	17	1.65	1	1.115	4	1
	2011-3	20	1.45	1	0.82	4	1
	2011-4	31	1.48	1	0.89	4	1
	2012-1	14	1.71	1	1.139	5	1
	2012-2	11	1.73	1	0.90	3	1
	2012-3	17	1.53	1	0.8	4	1
	2012-4	3	1	1	0	1	1
Total		201	1.9	1	1.48	11	1

Table 6.84: Krypton - Descriptive statistic - TP feature

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
Krypton	2010-3	6	3.5	2.5	3.2090	9	1
	2010-4	34	1.76	2	0.89	4	1
	2011-1	45	2.67	2	2.20	13	1
	2011-2	28	1.68	2	0.77	4	1
	2011-3	26	1.62	1	0.752	3	1
	2011-4	27	1.89	1	1.45	6	1
	2012-1	28	1.86	1	1.407	7	1
	2012-2	18	2.06	2	1.552	6	1
	2012-3	28	2.93	3	2.19	9	1
	Total	240	2.13	2	1.66	13	1

Table 6.85: Krypton - Descriptive statistic - TP bugs

6.14 Neon

* Correlation is significant at the 0.05 level (2-tailed).

** Correlation is significant at the 0.01 level (2-tailed).

Table 6.86: Neon - Correlation

Team	Quarter		Churn	Lead time
Neon	2010-3	N	62	62
		Mean	25.27	7.69
		Median	9	6
		Std. Deviation	39.448	6.88
		Maximum	193	24
		Minimum	0	1
	2010-4	N	125	125
		Mean	33.13	6.66
		Median	15	5
		Std. Deviation	47.46	6.01
		Maximum	214	27
		Minimum	0	1
	2011-1	N	132	132
		Mean	23.81	6.53
		Median	9	5.5
		Std. Deviation	41.22	5.43
		Maximum	301	27
		Minimum	0	1
	2011-2	N	134	134
		Mean	36.1	10.71
		Median	7	6
		Std. Deviation	59.067	35.19
		Maximum	271	408
		Minimum	0	1
	2011-3	N	99	99
		Mean	40.86	8.34
		Median	17	6
		Std. Deviation	50.60	6.61
		Maximum	239	27
		Minimum	1	1
	2011-4	N	98	98
		Mean	63.31	8.5
		Median	38	7
		Std. Deviation	74.35	6.84
		Maximum	294	27
		Minimum	1	1
	2012-1	N	110	110
		Mean	56.75	5.16
		Median	21	4
		Std. Deviation	72.52	4.46
		Maximum	296	22
		Minimum	1	1
	2012-2	N	81	81
		Mean	44.95	5.05
		Median	22	4
		Std. Deviation	54.23	4.43
		Maximum	300	23
		Minimum	1	1
	2012-3	N	174	174
		Mean	66.12	5.78
		Median	33.5	4
		Std. Deviation	81.02	4.35
		Maximum	299	21
		Minimum	1	1
	2012-4	N	120	120
		Mean	75.93000000000007	5.3
		Median	32	4.5
		Std. Deviation	86.49	4.
		Maximum	303	17
		Minimum	1	1
	Total	N	1155	1155
		Mean	47.76	7
		Median	18	5
		Std. Deviation	66.37	13.112
		Maximum	303	408
		Minimum	0	1

Table 6.87: Descriptive statistic - Lead-time and churn - Neon

Team	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
Neon	2010-3	10	2	1.5	1.333	5	1
	2010-4	34	1.68	1	1.173	6	1
	2011-1	31	1.71	1	1.03	5	1
	2011-2	6	1.17	1	0.4	2	1
	2011-3	9	1	1	0	1	1
	2011-4	1	1	1	.	1	1
	Total	91	1.62	1	1.06	6	1

Table 6.88: Neon - Descriptive statistic - Throughput

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
Neon	2010-3	25	14.4	15	6.19	23	6
	2010-4	92	21.41	20	7.17	41	9
	2011-1	90	27.2	27.5	4.90	38	17
	2011-2	91	29.71	27	14.42	62	12
	2011-3	92	31.98	30	8.90	55	18
	2011-4	92	29.09	29	10.09	45	12
	2012-1	91	19.03	18	4.63	30	7
	2012-2	91	24.34	25	10.282	50	5
	2012-3	92	23.24	21.5	7.89	44	10
	2012-4	92	19.48	22	10.99	45	1
	2013-1	11	2.18	2	0.60	3	1
	Total	859	24.45	24	10.54	62	1

Table 6.89: Neon - Descriptive statistic - WIP

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
Neon	2010-4	40	2.45	2	1.694	9	1
	2011-1	47	2.43	2	1.80	8	1
	2011-2	42	3.57	3	2.52	13	1
	2011-3	45	2.47	2	2.33	13	1
	2011-4	48	2.48	2	1.59	7	1
	2012-1	36	3.25	3	3.03	16	1
	2012-2	36	2.19	2	1.48	8	1
	2012-3	44	3.43	2	2.55	10	1
	2012-4	33	3.12	2	2.63	10	1
	2013-1	1	1	1	.	1	1
	Total	404	2.75	2	2.306	17	1

Table 6.90: Neon - Descriptive statistic - Bugs

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
Neon	2010-3	10	2	1.5	1.333	5	1
	2010-4	34	1.68	1	1.173	6	1
	2011-1	31	1.71	1	1.038	5	1
	2011-2	11	1.27	1	0.64	3	1
	2011-3	34	1.53	1	0.99	5	1
	2011-4	16	1.5	1	0.73	3	1
	2012-1	23	1.3	1	0.70	4	1
	2012-2	32	1.63	1	0.83	4	1
	2012-3	32	2.02	2	0.93	4	1
	2012-4	40	1.93	2	1.11	6	1
Total		263	1.69	1	0.997	6	1

Table 6.91: Neon - Descriptive statistic - TP feature

TeamName	Quarter	N	Mean	Median	Std. Deviation	Maximum	Minimum
Neon	2010-3	10	2	1.5	1.333	5	1
	2010-4	34	1.68	1	1.173	6	1
	2011-1	31	1.71	1	1.038	5	1
	2011-2	11	1.27	1	0.64	3	1
	2011-3	34	1.53	1	0.99	5	1
	2011-4	16	1.5	1	0.73	3	1
	2012-1	23	1.3	1	0.70	4	1
	2012-2	32	1.63	1	0.83	4	1
	2012-3	32	2.02	2	0.93	4	1
	2012-4	40	1.93	2	1.11	6	1
Total		263	1.69	1	0.997	6	1

Table 6.92: Neon - Descriptive statistic - TP bugs

Chapter 7

Conclusion

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.
- 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/smrv.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.
- Langford, Eric, Neil Schwertman and Margaret Owens (2001). 'Is the Property of Being Positively Correlated Transitive?' English. In: *The American Statistician* 55.4, pages. ISSN: 00031305.
- Leonardo Campos Rafael Buzon, Eric Fer (Mar. 2013). *Kanban Pioneer: Interview with David J. Anderson*. URL: <http://www.infoq.com/articles/David-J.-Anderson-Kanban/> (visited on 30/09/2013).
- Manning, James (2013). 'Lean Software Development'. In:
- Mark L. Spearman, David L. Woodruff and Wallace J.Hopp (1990). 'CONWIP: a pull alternative to kanban'. In: 28.5, pp. 879–894.
- 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.
- Yin, Robert K. (2008). *Case Study Research: Design and Methods (Applied Social Research Methods)*. SAGE Publications, Inc. ISBN: 1412960991.