

UNIVERSITY OF TARTU VILJANDI CULTURE ACADEMY

Information Studies Department

Design and Development of Virtual Environments

Harri Tallinn

Henrik Aavik

Mihkel Virkus

Raul Liive

**USING SCRUM IN A SIDE PROJECT WITH
DISTRIBUTED TEAMS**

Master Thesis

Supervisor: Marge Sults, MSc

Ready for Defence: */signed digitally/*

Viljandi 2013

Table of Contents

RESÜMEE	4
INTRODUCTION	5
1. THEORETICAL BACKGROUND.....	8
1.1. Agile methodology theory overview	8
1.2. Other methods	10
1.3. Scrum	11
1.3.1. Scrum history.....	13
1.3.2. Roles in Scrum.....	13
1.3.3. Scrum artifacts.....	18
1.3.4. Scrum ceremonies.....	22
1.4. Virtual teams theory	24
1.5. Side project development theory.....	27
2. METHODS	29
2.1. Online Bridge card game development.....	29
2.1.1. Lean Startup and Business Model Generation	30
2.1.2. Background info on contract bridge	34
2.1.3. Validating idea.....	36
2.1.4. Design thinking.....	39
2.1.5. Development tools.....	41
2.2. Data collection.....	43
2.2.1. Observing our team	43
2.2.2. Mentoring other teams.....	44
2.2.3. Survey to early stage startups	45

3. RESULTS	47
3.1. Our product development.....	47
3.1.1. Building Minimum Viable Product	50
3.2. Findings from incubated teams.....	52
3.2.1. Team A	52
3.2.2. Team B	53
3.2.3. Team C	55
3.2.4. Summary of findings	58
3.1. Results of the quantitative survey	60
CONCLUSIONS.....	67
BIBLIOGRAPHY	70
APPENDIX'ES.....	84

RESÜMEE

Scrum on iteratiivne ja kasvava populaarsusega agiilse tarkvara arendamise raamistik. Käesoleva töö eesmärgiks on uurida, kas on võimalik kasutada Scrum raamistikku edukalt tootearenduse kõrvalprojektis, kui meeskonna liikmed töötavad erinevatest asukohtadest ning neil on väga erinevad võimalused ajaliselt panustada.

Uuringu keskmes on meeskonnaliikmete kommunikatsiooni, motivatsiooni ja produktiivsusega seonduvad küsimused ja võimalused, kuidas neid leevendada kasutades Scrum raamistikku. Selleks me rakendasime Scrum raamistikku enda tootearenduses jälgides selle mõjusid protsessile. Samuti aitasime Scrum'i sisse viia kolme sarnase meeskonna tootearenduse protsessi. Lisaks viisime läbi uuringu kvantitatiivsete andmete kogumiseks idufirmades rakendatavate koostöö metoodikate kaardistamiseks.

Eelmainitud uuringu tulemusena selgus, et suur enamus (77%) idufirmasid ei kasuta oma tootearenduses mingit konkreetset raamistikku. Käesoleva uurimistöö tulemustele tuginedes võime väita, et Scrum raamistiku rakendamine teeb kogu meeskonna jaoks tootearendusprotsessi läbipaistvamaks, suurendab meeskonna fookust või vähemalt aitab avastada meeskonnasiseseid motivatsiooniprobleeme.

INTRODUCTION

Scrum is the most popular agile software development methodology in use today (VersionOne, 2013), (West & Grant, 2010), (Begel & Nagappan, 2007). Many of the Scrum teams work in virtual distributed formations (VersionOne, 2013). The distributed work model introduces multiple burdens, like communication issues, matter of trust, time zones and cultural differences to the teams, which can severely affect performance (Deemer, s.a.). As a result, dislocated Scrum teams have a considerably lower success rate than collocated ones according to various surveys (Ambler S. W., 2008), (Ambler S. W., Agility at Scale Survey 2012, 2012), (Cohn, 2010).

This paper researches the topic of “Using Scrum in a side project with distributed teams” and consist of three main parts: theoretical background, methods and results.

In the theoretical background we cover the different software development methodologies, agile methodology in particular. We take a deeper look into Scrum; it's history, roles in Scrum, Scrum artifacts and ceremonies. We also briefly discuss virtual teams theory.

In the methods we cover the online contract bridge card game development, Lean Startup and Business Model Generation as part of business development, background info on contract bridge, validating the idea, the design and development tools used to build the minimum viable product and the methods of data collection for Scrum.

In the results we describe our implementation of Scrum, the building of the minimum viable product, and research made into other early stage startup teams to support our own findings.

The thesis started from a plan to develop a startup in the field of virtual products – a social version of the card game contract bridge. We met irregularly at school and divided some tasks between ourselves only to come back next time and see that things are not moving along. We needed some kind of framework that would help us to manage our team, motivate ourselves and help us to focus

on tasks. We composed a survey to find out whether other early stage startups in a similar position to ours are having similar problems.

As a startup finds itself in a situation where the team cannot meet on a daily basis physically, they will soon find that it's hard to implement Scrum (Alex & Christian, 2013). Still, it is possible to make it work virtually (Luz, Gazineu, & Teófilo, 2009).

Our hypothesis is that it is possible to effectively run a Scrum team for a side-project in a situation where all its members work remotely, and with varying amounts of time commitment from each participant. We will be focusing on early stage startups.

We will be focusing on our own Bridge card game development project, team and the implementation of Scrum. We will be observing our own Scrum and make conclusions from it. We will also use the Minimum Viable Product to determine whether or not the Scrum was successful.

In addition we are mentoring three other similar teams to find out whether their experience will validate our findings. The selected teams consist of members who have other daily obligations and who are working remotely. We made suggestions of best practices, but we let the mentored teams make their own decisions on how to exactly implement Scrum. This included decisions on communication platform, backlog management, how and when they handle Scrum ceremonies.

We map the team members, their obligations and family status. We look into their motivation to participate in the teamwork and how they keep the commitments in real life and how they later on assess their work themselves. The teams under focus were not using any formal development methods before our influence.

In the paper we define a team as group of 2-9 people working on a joint product development. We do not look at multi-team product development in this work, as it is not a usual setup for early stage startups.

While we study Scrum we use Lean Startup and Business Model Canvas methods alongside with design thinking to build our own product. Although the Lean Startup process does involve some core principles of Agile methodologies, the main difference between Lean Startup and Agile is that Lean eliminates anything that is not absolutely necessary, possibly including team meetings, tasks and documentation (Yau & Murphy, 2013). Lean Startup is a very good addition to our experimentation with our own approach to Scrum and virtual product development process - therefore we will discuss it under methods and results in more detail as well.

1. THEORETICAL BACKGROUND

1.1. Agile methodology theory overview

Agile methods including Scrum embrace an old truth – only constant in this world is change. We just do not know when and how frequently it is going to happen. If companies are not able to adapt quickly enough, they will go out of business, even if their profits are healthy today. The empirical nature of Scrum makes it suited to deal with innovation and all things new. It helps to cope with complex and unpredictable situations. (Pichler, 2010)

There have been three types of development frameworks: linear, iterative, and combination of both (Pavlenko, 2012). The term *linear* means “*progressing from one stage to another in a single series of steps*” (Oxford University Press, 2013).

One of the most common and well-known linear frameworks is Waterfall, where projects are executed in sequential phases. Such linear methods are also referred to as 'plan-driven' because of the fact that all development is based on the first stage of the process – planning. Every step in a waterfall process has to be finished before moving on to the next (see Figure 1). Because of this, the customer can see the actual product only when all stages are completed. The requirements set down in planning phase have to be exact, clear, and relatively static. (Sims & Johnson, Elements of Scrum, 2011).

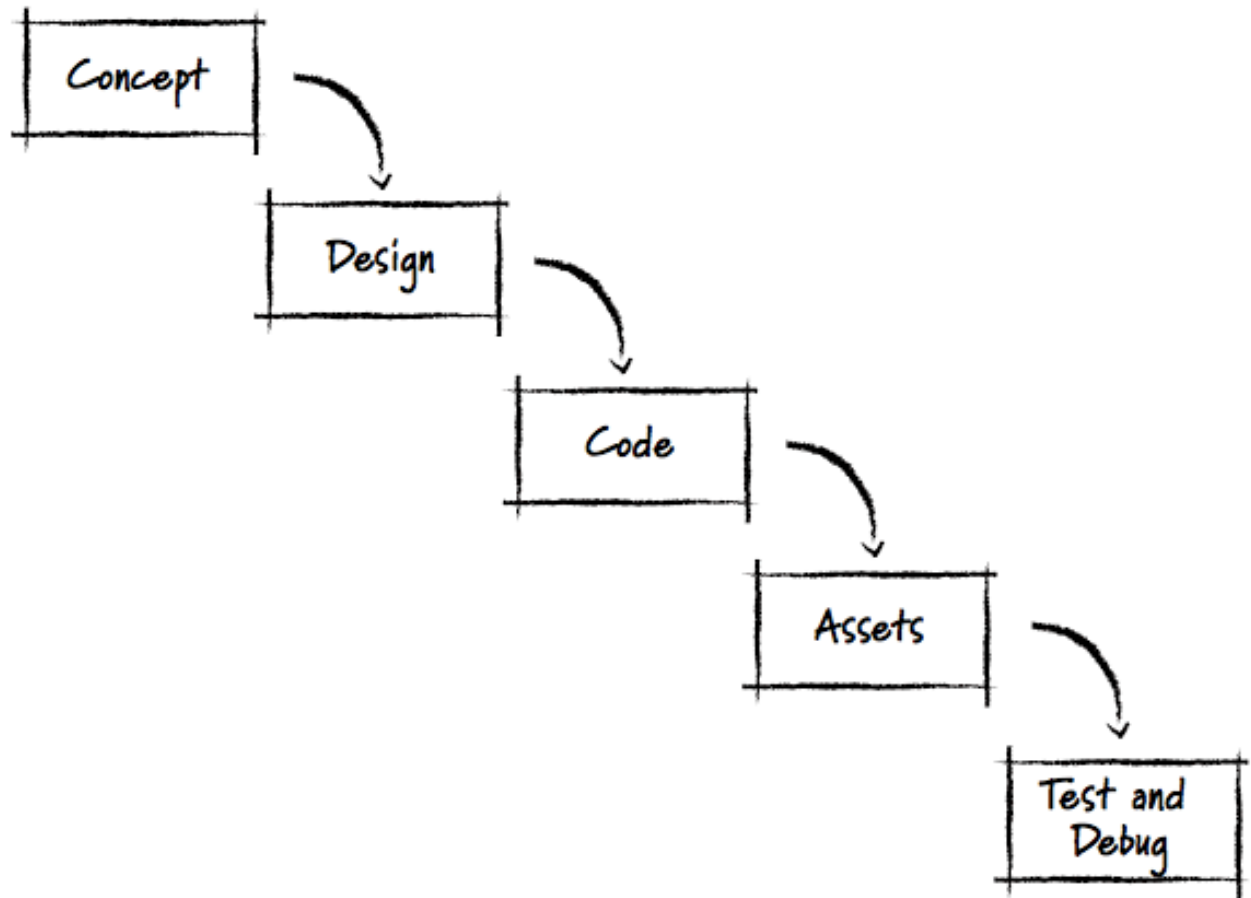


Figure 1. The linear workflow of the Waterfall methodology. (Keith, 2010)

Iterative development is the opposite of linear – it goes on in cycles. Unlike plan-driven linear methods, it does not need so much initial planning and focuses on constant changes. Scrum stimulates continuous revision and improvement of software, giving the customer the first glimpse of the product already at the end of the first iteration. The desired result is broken down into small pieces that are developed over a period and finally compiled together when ready. (Cockburn, 2008) Iterative frameworks can be also used in combination with linear methods, setting up such frameworks as Incremental, Spiral, Rapid Application Development, and Extreme Programming (Pavlenko, 2012).

Comparing to the linear process, iterative incremental development is more popular and widely applied in software companies. Its main advantage is flexibility, which is very important for developing a product so that the customers can constantly see the results and give input. That is

extremely important in the context of the rapidly developing software industry and software technologies. Customer expectations change quickly as they see the product getting better and better with every iteration and they may want to add features. (Williams, 2007)

1.2. Other methods

All though in this paper we focus on Scrum and compare it also with Waterfall, they are of course not only frameworks in the world. There are many more, we mention hereby just two most related to Waterfall and Scrum. There is a linear framework called the V-Model and an iterative style called Kanban.

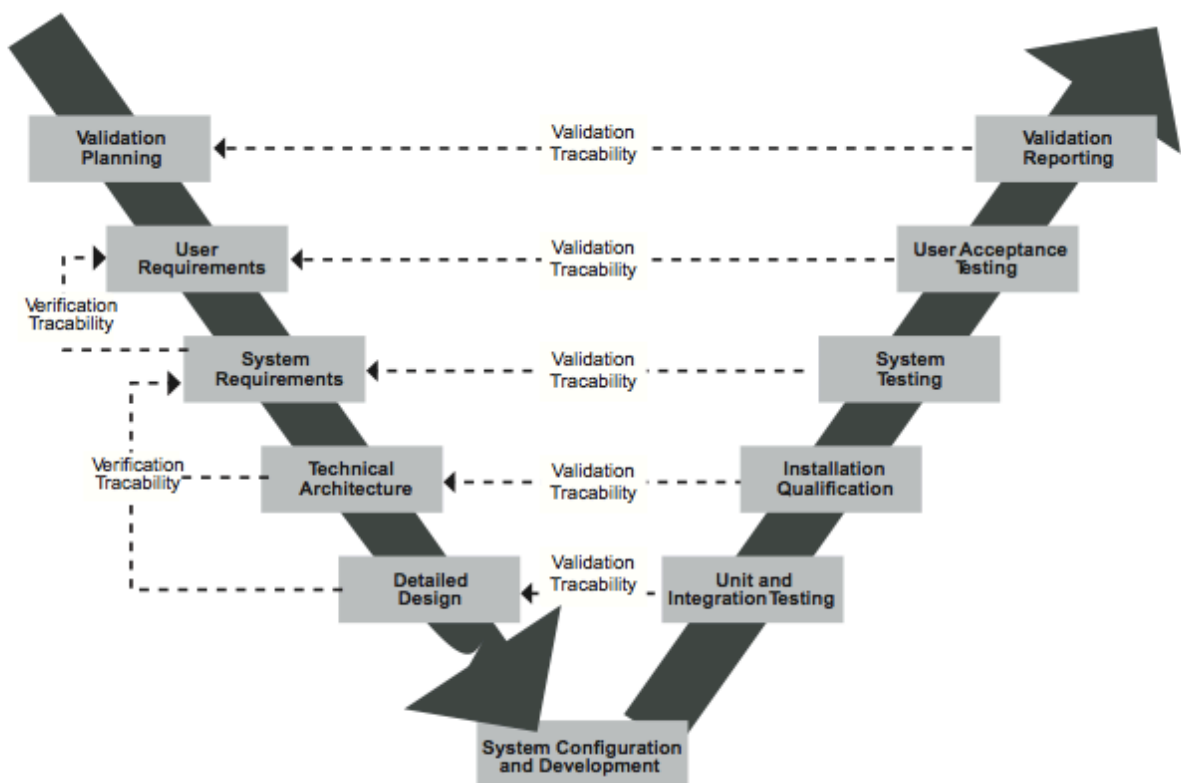


Figure 2. V-Model style diagram (Turner, 2007)

The V-Model is focused on early correctness, as systems engineering can sometimes value precision over accuracy and completeness over rapid user satisfaction. V-Model has evolved over time and changed somewhat, but its fundamentals are still the same – establish requirements, establish an architecture, decompose the system into subsystems, design the sub- systems, build the subsystems, test the subsystems, integrate the subsystems, and then test the system. (Turner, 2007)

Kanban is a new technique for managing a software development process in a highly efficient way. Kanban underpins Toyota's "just-in-time" production system. Although producing software is a creative activity and therefore quite different to mass-producing cars, the underlying mechanism for managing the production line can still be applied. (Kanban Blog, 2013)

Kanban does its magic by revealing bottlenecks in the production. For example: if the testers are only able to test 5 features per week whereas the developers and analysts have the capacity to produce 10 features per week, the throughput of the pipeline as a whole will only be 5 features per week because the testers are acting as a bottleneck. Kanban framework manages this problem by redeploing recourses and helping to maintain workflow even over iterations. (Kanban Blog, 2013)

1.3. Scrum

Scrum is a cooperation framework that is designed to help teams of people to develop complex products, doing it the agile way. A Scrum team usually consists of between three to seven people who work together in short bursts of activity called sprints. The teams plan their work ahead. Very important part of the process is reflection back to review the results and to search for areas of improvement. Scrum is based on a small set of core values, principles, and practices, collectively the Scrum framework. (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

It is quite common to mix up the terms 'Agile' and 'Scrum' in practice. The confusion is unsurprising, because Scrum is the most used Agile methodology (West & Grant, 2010).

Agile is a much wider concept than Scrum – it is a set of values and principles, whereas Scrum is one particular methodology based on those values and principles. Therefore, if someone says Scrum instead of Agile, or vice versa, the mistake is relatively small, for in most cases these terms are equal. (Pavlenko, 2012)

Agile42 (see Figure 3.) scheme demonstrates the usual Scrum workflow, where the Product Backlog items are chosen for the sprint backlog, then the sprint iterations happen where those items are developed and tested, and then the deliverable is presented to the customers and stakeholders.

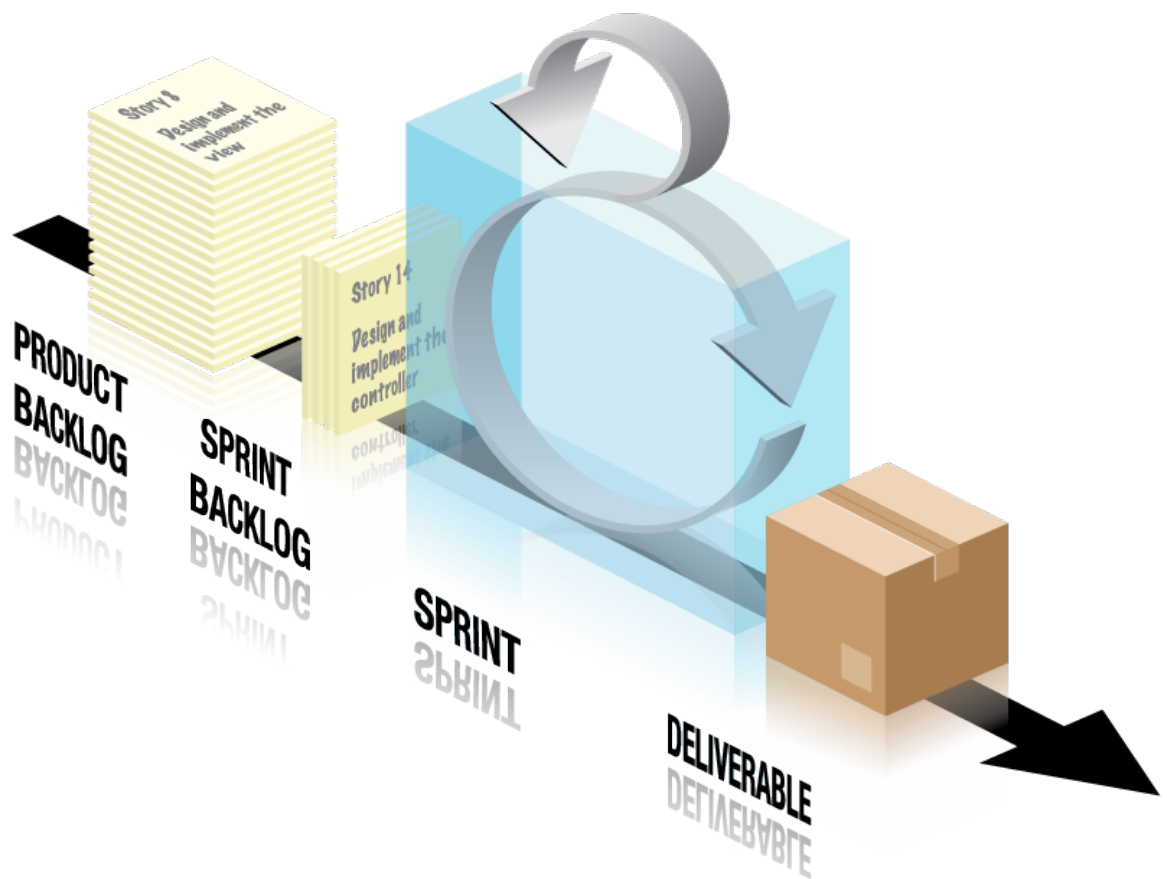


Figure 3: Scrum workflow (agile42.com, 2012)

1.3.1. Scrum history

Scrum was first defined as "*a flexible, holistic product development strategy where a development team works as a unit to reach a common goal*" as opposed to a "*traditional, sequential approach*" by Hirotaka Takeuchi and Ikujiro Nonaka in the "Harvard Business Review" (Takeuchi & Nonaka, 1986).

The word scrum comes from rugby football, where it refers a manner of game restarts after an infraction. Scrum as a framework was first used by Ken Schwaber in beginning of 1990s. He implemented it in company Advanced Development Methods. A similar way of working was developed at the same time by Jeff Sutherland, John Scumniotales and Jeff McKenna at Easel Corporation. The name Scrum to describe the framework was first used in Easel Corporation. First public presentation of the framework was done in conference Object-Oriented Programming, Systems, Languages & Applications in 1995, where it was raised in workshop of Business Object Design and Implementation by Sutherland and Schwaber. They continued to co-operate and during next few years they formed out of their initial ideas the Scrum as we know. (Sutherland, What is Scrum?, 2013)

1.3.2. Roles in Scrum

There are three roles in a Scrum team – Product Owner, Scrum Master and Team Member.

Product Owner's jobs is to maximize the return on investment (further ROI) by choosing what the team should work on first, based on how much value it brings at any given moment. The Product Owner controls the priority of the items in the product backlog. No one but the Product Owner is authorized to ask the team to do work and to prioritize the product backlog items.

(Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012).

The Product Owner is responsible person for the product success. He is leading the product development He stands between the Scrum team and the stakeholders and needs to constantly look both ways. He needs to understand the needs of the organizational stakeholders, the customers, and the users well enough to talk on their behalf. He decides what to build in which order and communicates that to the rest of the team. (Rubin, 2012)

The team need to fully understand the requirements of the worth. It helps to raise the value of the work as you avoid building the wrong items. It's the Product Owner's responsibility to record the requirements, often in form of user stories (see Figure 4). By that he builds the Product Backlog. (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

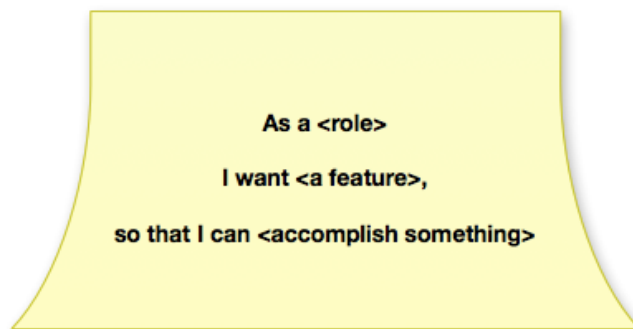


Figure 4. User story (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

The Product Owner also ensures that the criteria for accepting features are specified, often referred as definition of done. In this respect, the Product Owner is part business analyst and part tester. He has to define the acceptance criteria for each product backlog item – what are the functional and nonfunctional requirements to be met for marking item "done". This can be done in the form of automatic tests, by consulting experts or development team members. Those criteria should be created before an item is even considered at a sprint-planning meeting. The Product Owner will have to verify the acceptance criteria by the end of the sprint execution but before the sprint review. (Rubin, 2012)

It is also the Product Owner's responsibility to ensure that good economics decisions are continuously made at the release, sprint, and product backlog levels. (Rubin, 2012)

At the release level, the Product Owner sometimes has to make trade-offs in date, scope, budget and quality as additional info is arises during product development iterations – what is in focus at the beginning might not be relevant in the presence of new information that emerged during or after the first release. Also, at the end of each sprint, the Product Owner has to make a decision whether to fund the next sprint or not. He might conclude that shipping the product early is economically more sensible. The flexibility to deliver early is enabled by the inherent logic in Scrum that the higher value items in the product backlog are developed first. (Rubin, 2012)

In addition to release-level economics, the Product Owner is also responsible for sprint-level economics, making sure that good ROI is constantly delivered. He knows the cost of the next sprint and should ask himself at the planning phase whether the possible value delivered is higher than the sprints cost. (Rubin, 2012)

Backlog-level economics are addressed by prioritizing the product backlog. When economic conditions change, the priorities in the product backlog may change as well. If the value/price ratio of a feature has dramatically changed, the Product Owner should reprioritize the Product Backlog according to this new knowledge. (Rubin, 2012)

During product planning, the Product Owner works with the stakeholders and carries the vision for the product. He works with the stakeholders and the team during planning to define the content of the next release. In sprint planning, the Product Owner works with the development team to define the sprint goal. He also provides input to the team that helps them to select a set of product backlog items that they can realistically deliver by the end of the sprint. (Rubin, 2012)

The Product Owner has to be one person in Scrum team, not a group of decision makers or committee, team has to know who makes decisions. Though, this person can be Product Owner for several Scrum teams or can also be a team member in the same Scrum team. Product Owner can also have an assistant in the team, who takes over from him some responsibilities. (Rubin, 2012)

The Scrum Master is a servant leader (Narayanaswamy, 2012). While the team's deliverable is the product, the Scrum Master's deliverable is the team – a high-performing and self-organizing team. In doing that, he is not boss – this is a peer position on the team, set apart by knowledge and responsibilities, not rank. (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012). The Product Owner and the Scrum Master roles complement each other: The Product Owner is primarily responsible for the “what” – creating the right product. The Scrum Master is primarily responsible for the “how” – using Scrum the right way. (Pichler, 2010)

The Scrum Master coaches both the Product Owner and the development team. He removes barriers between them and helps the Product Owner to drive development. He observes how the team is using Scrum and does what he can to help it get better in performance. The Scrum Master helps the new Product Owner to understand and execute his Product Owner's responsibilities and provides him with ongoing assistance for activities such as grooming. (Rubin, 2012)

The Scrum Master is sometimes described as a servant leader of the Scrum team. He solves upcoming problems and ensures that highest-priority needs of the team are being met. He is the process authority, empowered to ensure that the team connects, understands and respects the Scrum values, principles and practices. The Scrum Master helps the team to define its own process for making sure that the work gets done instead of being responsible for the work itself. The Scrum Master protects the team from outside interferences and removes all obstacles quickly, so the team can keep its focus on delivering business value. (Rubin, 2012)

The Scrum Master has to have vast knowledge of Scrum. He also needs to understand most of the technical issues that the team needs to address and technologies that will be used by the team. Thirdly, he needs to understand business issues as working knowledge of business is very helpful for understanding how to deliver business value. (Rubin, 2012)

The Scrum Master coaches the team members by asking smart questions. A great Scrum Master should try to avoid answering any questions directly, but instead answer with his own question, thereby helping the team realize that they have the insight to find their own answers, a form on

Socratic questioning (Paul & Elder, 2007). Because this, the Scrum Master needs to be patient, giving the team time to arrive at appropriate answers on their own. He has to have excellent collaboration skills to work with the Product Owner, the development team, and all the other parties, even those who might not be directly involved with Scrum. (Rubin, 2012)

As the Product Owner and Scrum Master roles are designed to balance each other. One individual should never be both Scrum Master and Product Owner at the same time. (Pichler, 2010)

Full authority over how the work gets done belongs to the team members. Scrum changes the teams mindset from "doing my job" to "doing the job". Also the change is in focus from work ("what we are doing") to results ("what is getting done"). (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

A Scrum team is a cross functional team which has in the team all needed skills to create the shippable product (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012). Scrum team includes several competences, which may include architect, programmer, tester, database administrator, user interface designer, and so on. The development team in Scrum is simply defined as a multi- and cross-functional collection of these jobs. (Rubin, 2012)

In Scrum team the team as whole has to do all of the work to produce one or more deliverables of a working product's functionality each sprint. This includes design, development, integration, and testing of the functionality. During sprint, the team has to perform creative work on designing, building, and testing the product. They do all the work to develop product from scratch till public release. (Rubin, 2012)

It usually takes some time for a group of people to become a true team – a strong unit with members who trust and support each other and who work together effectively. Scrum teams should be filled with the same people for as long as possible, as they have to grow together. Changing the team's setup makes the team-building process start all over again, and probably productivity and self-organization will suffer. Scrum teams should also “grow” together with product during longer

period of time and this relationship should be maintained – product should be developed by dedicated teams. This not only facilitates learning, but it simplifies the allocation of people and resources. (Pichler, 2010)

1.3.3. Scrum artifacts

There are some "Scrum things" that have to be present in the process and in Scrum framework they are called artifacts. These are things every Scrum team needs to create for themselves so they are most suitable for them. But they should always be there in one way or another to make Scrum really work. (Sims & Johnson, Elements of Scrum, 2011)

Those physical things are (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012):

- The Product Backlog;
- Scrum board;
- The Sprint Backlog;
- Burn down Charts;
- Task Board;
- Definition of Done.

The Product Backlog is a cumulative list of desired deliverables for the product. This includes features, bug fixes, documentation changes, and anything else that might be meaningful and valuable to produce. These are called product backlog items, which have often been written in form of user stories. (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

Scrum Board (see Figure 5) should display all user stories in prioritized order and development state be visible to team.

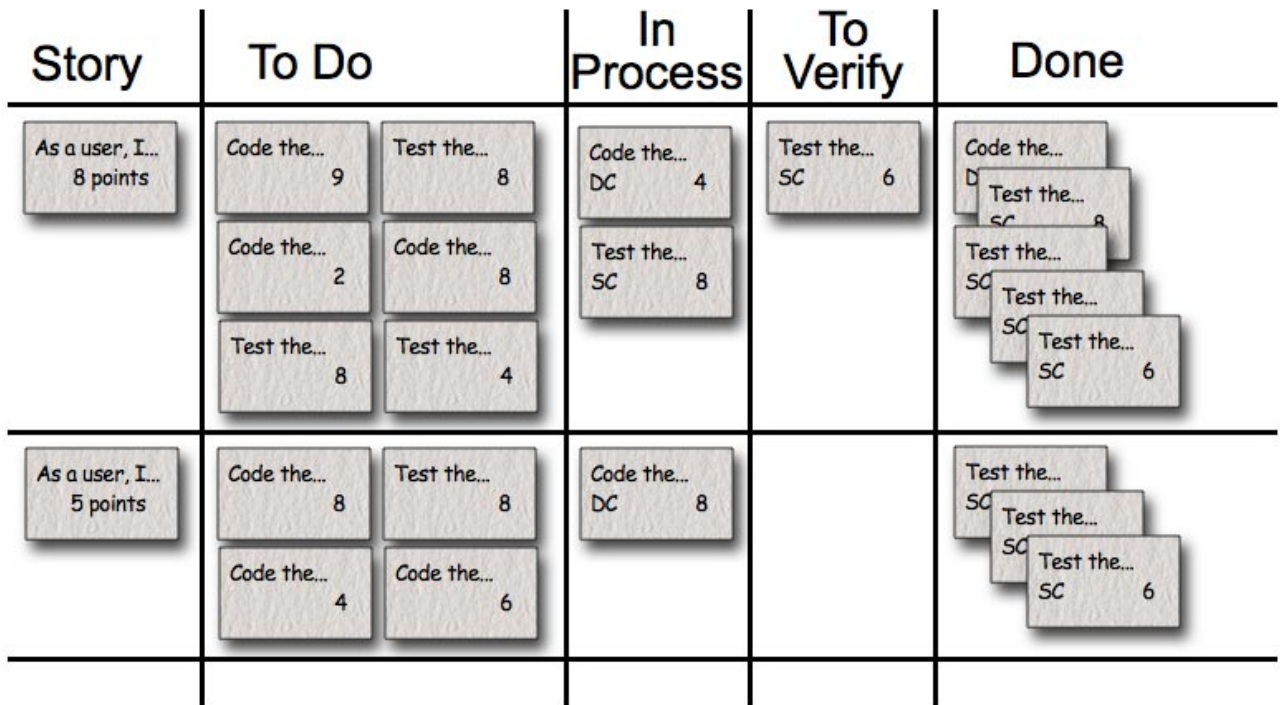


Figure 5: Scrum board scheme. (Mountain Goat Software, 2013)

The list of user stories should be arranged by the Product Owner so, that the most important story, the one that the team should develop next, is at the top of the list. The stories get less important/urgent moving further down the list. The first items at the top of the list have to be fully understood by the team, whereas the ones lower down can be larger and vaguer, as it will be some time before the team starts working on them. (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

The Product Backlog is like a living thing as it is a constantly evolving artifact. Items are added, deleted, and reviewed by the Product Owner as business conditions change based on market situation, customer feedback or value created. Changes can also occur when the Scrum teams understanding of the product grows through feedback of the software produced during each sprint. (Rubin, 2012)

Each user story in the product backlog should include the following information (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012):

- Which users will benefit from the story (Who is it for?)
- A short description of the needed functionality (What needs to be built?)
- The main reason why this story is valuable (Why we should do it?)
- An estimate of time as to how much work the story requires to finish
- Acceptance criteria that tell the team when this story has been fully implemented.

Before final prioritizing, ordering from team, or otherwise arranging the product backlog, the size of each item in the product backlog should be known as size equates to cost. A Product Owners needs to know an item's cost to correctly assess its priority. (Rubin, 2012)

Estimates for larger tasks are less accurate than the ones for small tasks. The limit of task size is arbitrary, but generally 16 hours is considered to be a smart work amount limit before the task needs to be broken down into smaller parts. (Keith, 2010)

A product backlog usually represents many weeks or even months of work, which is much more than can be completed in a single sprint. To determine the set of most important stories and tasks in the product backlog, items to build in the next sprint, the team preforms a meeting called sprint planning before each sprint. In sprint planning, the team agrees on the sprint goal, which defines what the upcoming sprint should achieve. Using this goal, the development team reviews the product backlog to determine the high-priority items to do. The items have to be chosen, keeping in mind, that the team has to realistically accomplish them in the course of the upcoming sprint while working at a normal pace. (Rubin, 2012)

Unlike product backlog, the sprint backlog has a finite lifespan – the length of the upcoming sprint. It includes only those stories that the team has committed to delivering during the sprint and tasks that also need to be done to get to the agreed goal. The stories are deliverables, and we can think of them as incremental units of value. Tasks are things that must be done, in order to deliver the stories, and so tasks can be thought of as little units of work. Therefore a story is the result a team delivers and a task is the particle of work that the person does. Each story will usually require several tasks. (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

A burn down chart is a table that shows us the relationship of scope and time (Figure 6). Time is on the horizontal axis and scope on the vertical axis. It shows how much of the scope the team has got done over a period of time. Each time something new is completed, the line on the chart has to move up. If the work scope is changed, the chart also changes. On the chart, the line goes up when work is added and down when it is removed from the plan. (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

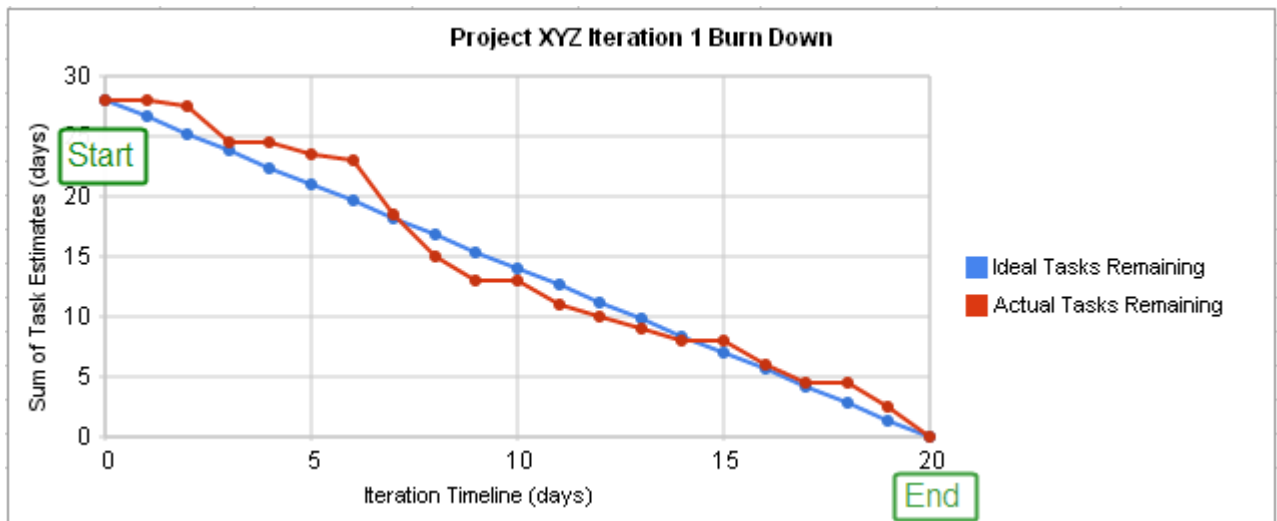


Figure 6: Burndown Chart example. (Wikipedia, 2013)

In a Scrum team, tasks should be visible to everyone, if possible right from across the room. A simple task board consists of three columns: to do, doing, and done. This visibility helps the team inspect their current situation and adapt as needed. (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

Definition of done is the acceptance criteria for a user story – the team has to agree on what it means to get a thing done. As there is sometimes confusion, a Scrum team creates their own definition of the word "done". The teams' own definition may be based on things like code written, code reviewed, unit tests passing, regression tests passing, documentation written, Product Owner sign-off, and so on. No user story is done before all its tasks are done. That definition will usually be

printed out as checklist and posted next to task board. (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

1.3.4. Scrum ceremonies

Ceremonies set the way for teamwork in Scrum. They help to foster the product development and to keep the focus by the team (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012).

Scrum starts with first ceremony that is backlog grooming. Product backlog needs constant attention and tuning – grooming. New items are discovered and added, existing ones changed or removed, items in higher positions will be prepared for upcoming sprints and estimations on item sizes are changed as team learns more about product and the working tempo. Product Owner will have the last say on what happens with Backlog, but grooming is collaborative process – Scrum allocates up to 10% of teams time to grooming process. (Pichler, 2010)

During backlog grooming the team estimates the Product Backlog items size using points from the Fibonacci scale (Downey & Sutherland, 2012). The estimates are done by comparing the new work item with few known reference points. This will help to determine the team's velocity and it gives input of investment to the Product Owner. (Pichler, 2010)

After the team has passed a few sprints they will approximately know how many points they are able to complete in any given sprint. This will be the team's velocity. It's important information for Product Owners on ordering the product backlog. (Rubin, 2012)

Sprints are rhythm of Scrum. As Scrum works in iterations, the team constantly comes back to the same things and activities. For scrumming successfully, it is important for the team to get in the right rhythm. Scrum iterations i.e. Sprints should be equal in length. Most often they are one or two weeks long. (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

The duration of a sprint is agreed on between the stakeholders and the Scrum team. If there is a disagreement, the Product Owner makes the decision. The duration shouldn't be changed too frequently because changes to the length of sprints are disruptive. It takes some time for the team to adjust to the rhythm and pace of a particular sprint length. It also takes time to refine their ability to accurately estimate items sizes in the sprint backlog after changing the sprint duration. So the length of the sprint should not be changed unless there is a dire need for it, and even then it can be done only between sprints. (Keith, 2010)

Sprints are time boxed – they always have a fixed start and end date. A new sprint planning immediately follows the completion of the previous sprint. As a rule no changes in personnel or scope is done during the sprint. However, sometimes the business needs can make following this rule unwise. (Rubin, 2012)

The tasks completed in sprint should always create some tangible value to the customer or the user. (Rubin, 2012)

The team members participate in the sprint planning. In the beginning of every cycle they help the Product Owner to establish the goals for the upcoming sprint, so that those goals are achievable and include measurable value for stakeholders. At the end of each sprint all team members participate in two other activities to make their teamwork better by inspecting-and-adapting: sprint review and sprint retrospective. (Rubin, 2012)

Daily standups are meetings are daily 15 minutes sync up meetings where the every team member gives answer to three questions, “What you did yesterday?”, “What you will do today?”, “What are your impediments?”. All team members are expected to participate in them. (Rubin, 2012)

They have to be brief and their goal is to inspect and adapt the work team is doing on completing stories they need to deliver. They inspect during meeting, adaption can happen later – problems have to surface and team members will be decided who will address those issues. Solutions will be found after meeting. Sometimes daily standups are also called daily scrums, as they are the ceremonies that

gave name to Scrum. (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

Sprint review meeting is ceremony after every sprint being a public end of it, any and all stakeholders are invited to that meeting. Team demonstrates there its accomplishments – stories that have met agreed Definition of Done. Stakeholders have the opportunity to see how product has evolved during the sprint. (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

Review meetings are informal, not show, preparing time for reviews should be kept minimum and slides avoided. (Pichler, 2010)

Next ceremony is sprints retrospective – teams own private meeting to improve itself. They need to inspect and adapt also their own processes. This is done after public review meeting at the very end of sprint and team looks on what have they learned during sprint and how they can use this new knowledge. No long lists of cons and pros are compiled – teams focus on just one or two strategic changes that could be applied on their process on next sprint. (Sims & Johnson, SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition), 2012)

1.4. Virtual teams theory

Virtual team is commonly defined as a group of people who work together from multiple locations, where they mostly communicate with each other using electronic means (Lipnack & Stamps, 2000) (Powell, Piccoli, & Ives, 2004).

Working in a virtual team is rather different to working in a physical team from one location. There are many challenges to overcome, which are more likely to hit virtual teams in comparison with teams in one location. With the advances made in the collaboration and communication tools over network, it has got much easier to run a successful virtual team. Especially together with the

globalized economy, which has led to a boom of virtual team usage. A virtual team needs different management style than a regular one. (Lepsinger & DeRos, 2010)

Prior research by Paasivaara et al. (2010) has resulted in an easy diagram which shows which communication medium is effective in a given situation (Figure 7). We can see that the richer a communication tool is, the more complex questions can be solved over it. (Paasivaara, Ornäs, Hynninen, Lassenius, Niinimäki, & Piri, 2010).

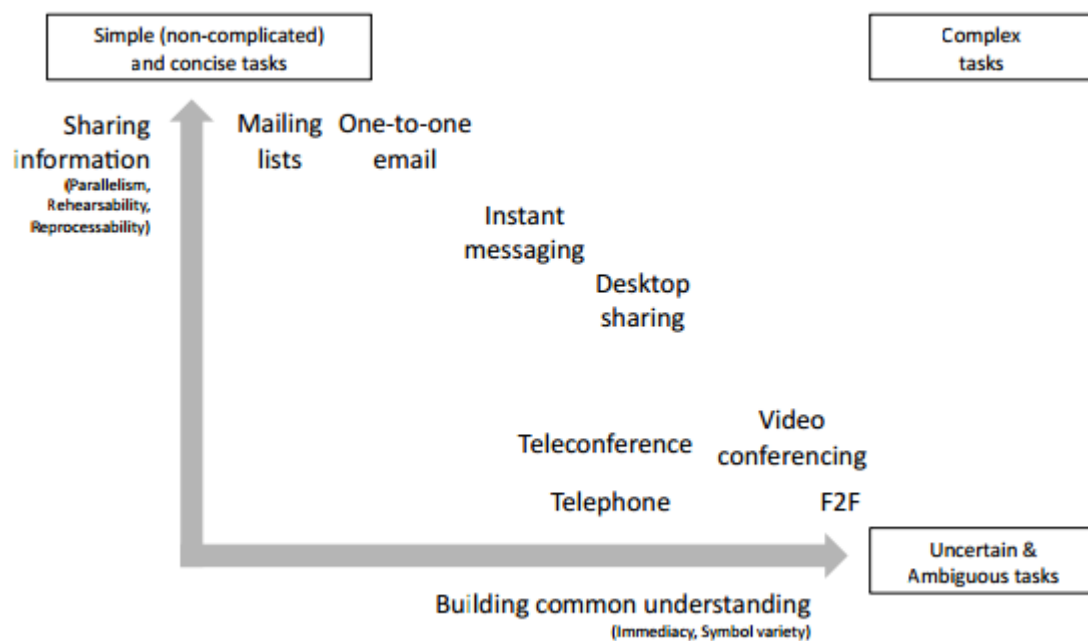


Figure 7: Communications complexity matrix (Paasivaara, Ornäs, Hynninen, Lassenius, Niinimäki, & Piri, 2010).

The main challenges of virtual teams, according to the surveys, conducted by Lepsinger and DeRos, are the lack of face-to-face communication; the lack of resources; time zone differences; the team members may be on more than one team and can devote limited time to project. Also team members don't share relevant info with each other and lack in training. (Lepsinger & DeRos, 2010) Language and cultural background can have additional effects on the team results. If the team communicates and works daily in a language, which is not the native language for most of the team members, then it adds additional communications challenges (Govindrajana & Gupta, 2001).

There are set of guidelines to follow to have a successful virtual team. First of them is to have a stable team membership. Getting to know the other team members and to build personal ties with them is much harder in virtual form. Smaller teams have a better success ratio than very big ones – consisting of more than 13 people. Cross-disciplinary teams have a higher failure rate than teams working in a common area of work. Effective meeting culture, leadership and well-used technology also make a crucial difference in the results. (Lepsinger & DeRos, 2010)

In face-to-face meetings, a number of familiar clues help us determine whom we can trust and whom we should not. Unable to rely on repeated, frequent face-to-face communication, distributed teams need to take additional measures and special care to build trust among team members, as trust is needed in coherent teams. (Cohn, 2010)

Team should use traveling ambassadors, starting meetings with casual conversations, in-person meetings of the full team now and then, and other similar activities. What also helps is early pressure for the team to produce working software by the end of each sprint, even the earliest ones. (Cohn, 2010)

Running virtual teams for software development, product management and outsourcing over different cultures, languages and time zones have showed that the most important issue in virtual work assignments is personal relationships between the people. To build and to keep the relationships alive, it is key to have face-to-face meetings with all of the team members at the start of an engagement, at least once a year and whenever there are bigger changes or very important discussions to be held. Meeting with people helps to avoid many of the common issues as one of their causes is misunderstanding in communication. The richer the communication tools used in the team are, the more effective the teamwork and collaboration results. (Liive, 2013).

The main benefit of having virtual teams is the ability to bring together the best available talent to a project no matter where their located geographically (Maynard, Mathieu, Rapp, & Gilson, 2012). Virtual teams also contribute to the contemporary way of doing business, where many of the companies work with clients across multiple locations. (Lepsinger & DeRos, 2010).

Virtual team performance can surpass the regular team ones. High performance can be achieved when the team implements effective communication, avoids conflicts, actively partakes in building a global culture and has an effective process in place (Siebdrat, Hoegl, & Holger, 2009).

1.5. Side project development theory

Side project development can be looked up in theory through the terms of secondary employment, supplementary employment and moonlighting job depending on geography and the exact usage of the terms by the authors of publications. Though none of these terms exactly describes working from home and developing your own project. In this paper we look into the practical advices, as we were unable to find comprehensive researches on the matter.

Developing a company while still maintaining a day job is a preferred way of development as it is possible to mitigate a lot of the risks. One should work out the business plan and the early stages of development until one has a much clearer idea of the potential success of the business. (Levit, 2009) (York, 2013). It is key not to let your day job be affected by the work being done for the future company (Levit, 2009).

When stating to build a business, it's important to investigate into possible clashes of interest with your current employer by fields of operation and potential intellectual property ownership matters based on your employment contract. It's good to be open with the current employer about the entrepreneurship plans. One might get them to become a customer or even invest in the startup. It's suggested to avoid using company equipment and time for any kind of development work on the project. It's good to build up a capital base, as much as possible, for the moment when it's time to leave the job. (York, 2013) (Wight, 2012).

It's hard to find the time and energy for developing one's own business being employed full time. The easiest way to make time is by devoting to a recurring timeslot each day. (Campbell, 2008). It's important to make time for one's family and leisure activities in order to keep oneself sharp (Wight,

2012). On the other hand family needs to understand that there is much less time for them than before. Developing a new business raises a set of costs and the family needs to understand that there's a need to save on the costs while the new business has not yet fully started. All in all everyone must keep an eye on their health – it's the main asset. (Ruben Anlacan, s.a.).

2. METHODS

2.1. Online Bridge card game development

The first step with a startup is to write down the initial vision and to share it with at least one person (Maurya, Running Lean, 2010).

Our team member, Henrik Aavik, introduced the idea and we started to develop the concept of a more social version of online Contract Bridge, with extra emphasis on playing with friends rather than other anonymous players back in November 2012. The essence of the product and the business model is given in the following 30-second pitch:

“There are about 100 million bridge card game players on our planet, but no sane and easy way to play it if people are not located in same place. We are creating a platform that lets people play bridge over Facebook. We use freemium business model where plain play is free, but extra services like keeping personal statistics and participating on tournaments cost money. We are team of 5, including highest level of design, marketing and engineering know-how.”

The hypotheses we raised in the beginning of our project were following:

1. The market size is well focused and big enough;
2. There is only one main competitor at the moment on the market: Bridge Base Online (BBO);
3. Today no one is offering a well-designed and user-friendly environment to play bridge with your friends online.

In order to understand whether to pour hard cash into the startup, one has to first validate the idea and then build a minimum viable product (later mentioned as MVP) in order to validate the whole business model and test customer reactions. For that we used Lean Startup and Business Model

Generation methodologies, well known in startup field (the latter one often also referred as Business Model Canvas – BMC). In developing the MVP, design-thinking principles are applied alongside with implementing Scrum.

2.1.1. Lean Startup and Business Model Generation

Lean thinking itself originated from Toyota Production System and it is a systematic methodology for identifying and eliminating waste in manufacturing and administrative processes through continuous improvement by flowing the product and service at the pull rate of the customer (Widman, Hua, & Ross, Applying Lean Principles in software development process – a case study, 2010). Eric Ries was the one who used the term **Lean Startup** for the very first time in September 2008 in his popular blog Startuplessonslearned.com: *“My belief is that these lean startups will achieve dramatically lower development costs, faster time to market, and higher quality products in the years to come”* (Ries E. , The Lean Startup, 2008).

Opposite to what is sometimes assumed, the Lean Startup method is not about cost. It is about speed – bringing product or service to customers as quickly as possible at the same time eliminating as much waste as possible (Ries E. , Lean Startup for Final Post, s.a.). The main argument for choosing the Lean startup methodology in early idea development phase is that it helps us to reduce the time between major iterations and by doing that we can increase our odds of success. The latter one is supported also by Guy Kawasaki: *“Don’t worry about shipping an innovative product with elements of crappiness. The first version of an innovation is seldom perfect /.../. If a company waits until everything is perfect, it will never ship, and the market will pass it by.”* (Kawasaki, Reality Check, 2011).

According to Ries a startup is a human institution designed to deliver a new product or service under conditions of extreme uncertainty (Ries E. , The Lean Startup, 2011). He even goes as far as putting the equals sign between startup and an experiment. Steve Blank has similar, but still different definition for a startup: a startup is a temporary organization in search of a scalable, repeatable, profitable business model. So in order to figure out where the startups stands with its product and

whether it moves in the right direction or not, one has to get feedback from its customers. (Blank & Dorf, The Startup Owner's Manual, 2012).

Our goal in using the Lean Startup methods was to figure out if we have the right product to build - a product customers want and would be willing to pay for - as quickly as possible. To that end the Build-Measure-Learn feedback loop as the core of the Lean Startup model can be used (Figure 8).

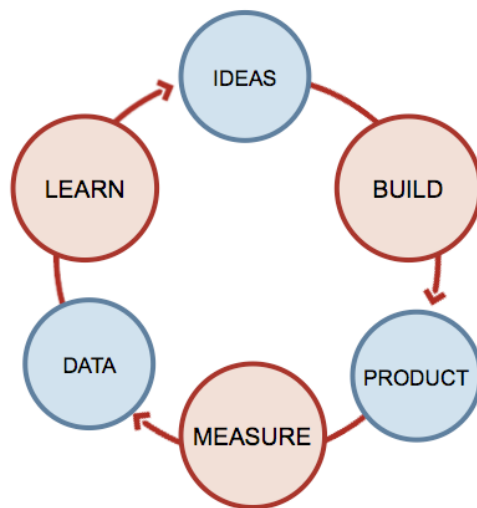


Figure 8: Build-measure-learn loop (Ries E. , The Lean Startup, 2011)

Usually the first idea is tested with simple sketching on a paper to visualize the product. After getting feedback one has to learn, what is good and what is bad about initial product, to make necessary adjustments and start all over again. Ries calls it validated learning - a process where all the effort that is not absolutely necessary for learning what customers want is eliminated and it's demonstrated by positive improvements in the startup's core metrics. (Ries E. , The Lean Startup, 2011) Alternatively collecting data from real customers and making necessary adjustments is called iterating to better results (Widman, Hua, & Ross, Applying Lean Principles in software development process – a case study, 2010).

While going through the loop with continuous iterations and customer feedback, a lean startup is testing its hypotheses e.g. validating its business model. It is necessary to make a difference between business model and business plan. Business model brings out the essence of a business – how

company creates, delivers and captures value while business plan is rather a static document collecting the facts. Steve Blank has said on the topic: *“Unless you have tested the assumptions in your business model first outside the building, your business plan is just creative writing.”* (Blank, No Plan Survives First Contact With Customers – Business Plans versus Business Models, 2010)

Successful business models have a common set of attributes (Barringer & Ireland, Entrepreneurship – successfully launching new ventures, 2008):

- Core strategy (How firm competes?)
- Strategic resources (How a firm acquires and uses its resources?)
- Partnership network (How firm structures and nurtures its partnerships?)
- Customer interface (How a firm interfaces with its customers?)

Today most of the startups business models come from Alexander Osterwalder’s creation of **Business Model Canvas** and business model generation concept, which is a simple and explicit way to create a business model for any product or service (Greenwald, 2012). The business model itself is described through nine blocks that show the logic of how the company intends to make money. Those blocks cover the four main areas of business: customers, offer, infrastructure (also called resources) and financial visibility. (A. Osterwalder Y. P., Business Model Generation, 2010)

The bottom of the BMC (Appendix 1) is related to financials – costs and revenues expected to run through the business. Canvas lists main cost and revenue sources, which underlies the business model. Often including a reference to some main competitor(s) is useful in the planning process.

On the left side of the canvas key partners, resources and activities are elaborated. The partnership building block includes the network of suppliers and partners that are needed in order to make the business model work. Key activities complements it with the most important activities the company has to do to run the business – for example to design, develop and deliver the product in superior quality as it is the case with our project. Key resources building block is related to assets needed in order to run the business model – physical, intellectual, human or financial. (A. Osterwalder Y. P., Business Model Generation, 2010)

In the center of the canvas is the value proposition – what value do we generate to our clients or what kind of problem do we solve for them. It can be related to services, which create value for a specific customer segment. It is the core of a startup idea as such and should show whether there is something in it or not. (A. Osterwalder Y. P., Business Model Generation, 2010) The best startup ideas tend to have three main common lines: they are something that founders themselves want, that they can build and what only few others are realizing as worth doing (Graham, 2012).

There are three sections in the BMC related to customers: customer segments, relationships and channels. The key questions to consider in relation to those are (A. Osterwalder Y. P., Business Model Generation, 2010):

- Who are the most important customers to whom we create value?
- How do we establish the relationships with the customers?
- How are we reaching each customer segment and what are the preferred channels?

So it all comes together on the business model canvas that helps to think through all the necessary aspects of the idea and test it with the Lean Startup approach. The latter one puts into the center of product development the most important asset for a company – customer (Ries E. , The Lean Startup, 2011).

The customer development process starts with customer discovery, which first captures the founder's vision and turns it into a series of business model hypotheses, which in turn develops to a plan to test customer reactions to those hypotheses (Figure 9). Then customer validation tests whether the resulting business model is repeatable and scalable – if not, the process starts again from customer discovery. Customer creation is the start of the execution, including the scaling of a business. Company building transitions the organization from a startup to a company focused on executing a validated model. (Blank & Dorf, The Startup Owner's Manual, 2012)

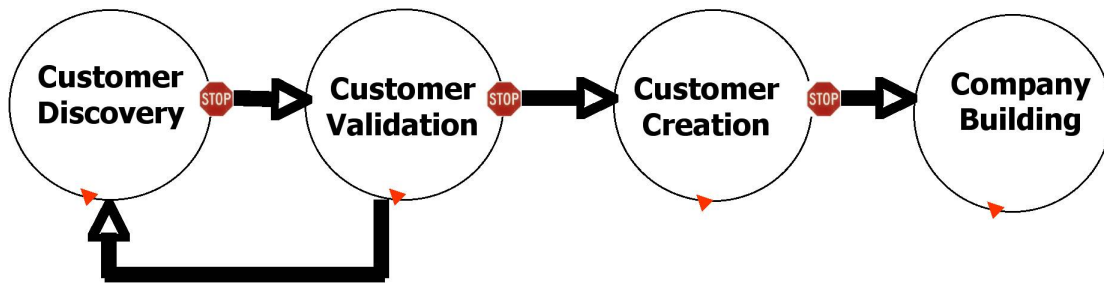


Figure 9. Customer development process (Blank & Dorf, *The Startup Owner's Manual*, 2012)

BMC helps one to capture the outcomes of such customer development process by creating another canvas template if something has changed (in addition to customer blocks also other blocks may change during that process). After the first two cycles are fully executed (customer discovery and customer validation), the BMC is validated and one can start to build a business out of it.

2.1.2. Background info on contract bridge

Contract bridge (or simply bridge) is a card game played with a deck of 52 cards and four people sitting across a table forming pairs. All the cards are dealt out (13 cards per player) and the aim of the game is to win as many tricks as possible. (American Contract Bridge League, s.a.) The start of bridge traces its origins to the British game called whist, first played in the 16th century. Known as contract bridge today it began in the 1920s when Harold Vanderbilt came up with an early scoring system. (American Contract Bridge League, s.a.)

The World Bridge Federation (further: WBF) is the main organization promoting and governing the card game bridge all over the world. It also federates National Contract Bridge Associations. There are eight zonal organizations under WBF which coordinate the activities on different countries in total of 693 845 official members from 123 countries. The biggest organizations are European Bridge League (393 164 members from 47 countries) and American Contract Bridge League (160 912 members from three countries). (World Bridge Federation, 2009)

The numbers given above are related to official members in bridge clubs who usually are more advanced professional bridge players. There is no concrete information source to figure out how many casual bridge players there are. The American Contract Bridge League has carried out an official survey with the methodology used in political and consumer polls and weighted survey data to reflect U.S. Census population profile (American Contract Bridge League, 2005).

According to this survey there are 45.8 million people that are familiar with bridge and 25.1 million people who know how to play bridge, from which 4.1 million (16.3%) play bridge online. Among online players 35.8% play bridge at least once a week. (American Contract Bridge League, 2005)

In our product we consider the 25.1 million players in U.S. as the widest target market we could have for our product. Since there is no such research available with later details in the U.S. nor in other countries, then in order to find out the number bridge players in the world we made an assumption that the general proportions between professional players in club statistics and casual players hold also in other countries. We believe this assumption holds rather well since bridge is considered to be played even more in European countries as it is in U.S. The outcomes are given in Table 1.

Table 1. Estimated amount of bridge players

Region	Official players	Multiplayer ¹	Estimated players (millions)
North-America (ACBL)	160 912	176	28.3
U.S.	142 592 ²	176	25.1
Europa (EBL)	393 164	176	69.2
World (WBL)	693 845	176	122.1

¹ Multiplier is found with the following calculation: people who can play by survey 25.1 million divided by the number of official members in ACBL U.S. clubs 142,592 = 176 players per one official player.

² The number was found when from total number of ACBL members Canadian and Mexican numbers were deducted. The information was obtained from same WBF page as numbers for official players.

So according to our analysis and taking into account the number of people who can play bridge and could be in the target group for our product amounts to 69.2 million players in Europe and 122.1 million players worldwide.

2.1.3. Validating idea

In order to test the hypotheses set under point 2.1, we first conducted an online research using Google to map out possible competitors and to analyze them. Summarizing the outcome, we can say that the hypotheses we made were true albeit with some admissions:

1. The market size is around 122 million players, from which ca 25.1 million are located in U.S.
2. In addition to Bridge Base Online (further: BBO) there are several different bridge playing platforms available, but they are either focused on gambling and they have similar usability and design issues as BBO. There are also some well-designed applications available, but they are platform or device dependent.
3. One can play bridge with friends also in BBO, but it means that they have to be registered users and one has to know their “code name” – the users in BBO are not using their real names and it makes social play more complicated. There are also other online play environments available, but they lack also in usability.

We investigated which are the most used and recommended online environments for playing bridge. To answer that, we analyzed the Google search results and different websites and forums. We also compared the info acquired to the user base information we found. The results are given in the following table:

Table 2. Main online Bridge play environments according to some websites and our own analysis (What is the best place to play bridge online?, 2010) (Play Bridge Online – The Best Places, s.a.).

Online environment	User base
Bridgebase.com (BBO)	50 000 players daily (Online Clubs: Bridge Base Online, s.a.), 100 000+ community (Introduction, s.a.)
OKBridge.com	13 000 members (Why OKBridge?, s.a.)
Swangames.com	4 338 players have earned points (Tournaments, s.a.)
Gambler.ru	Peak online users 2 586 (Бридж на Гамблере, s.a.) ³

There were also mentions of places like MSN games (which actually is a skin over BBO), Yahoo games and places we have found with Google like Bridge Club Live, Bridge Doctor, Bridge Island, but after our analysis the one's mentioned in Table 2 make up a list of the most used and recognized ones.

BBO is the biggest and most influential online bridge play environment in the world and its founders are backed with such known players in the software industry as Bill Gates and Warren Buffet, the first of whom has entered into the shareholders team with some other investors (People behind Bridge Base, 2011).

We analyzed the pros and cons of BBO environment on the basis of info we found in forums (Bodell, 2012), feedback from our adviser Oliver Loper (who is active top ranked user of BBO) and our own analysis we brought out five most positive and relevant elements of the site:

³ In order to find out the number of users (no statistics available) we visited website in a period of a week and recorded the users on the site on different times of a day: between 8-12, 12-18 and 18-24.00. Number in the table is average.

1. BBO has a big player's community – with different levels, so anyone should be able to find people to play with.
2. One can mark players as friends and see when they are online and ready for playing.
3. BBO arranges tournaments, which are popular especially among higher level players (costs money).
4. One can play against robots (costs money) and compare the outcome with others who have played against same hand.
5. Vugraph – there are online streams on the main bridge competitions around the world, which are very popular among the professional bridge players.

So why we should create another place to play bridge? Five things that could be improved on, compared to BBO by the same sources given above are as follows:

1. User interface and usability is bad – nor simple or intuitive.
2. The design of the site and its overall look is back where it was created – in the 90ies.
3. Flash interface usage that is not supported by some tablets and phones.
4. Hard to know how good the opponent is and it leads to different game experiences.
5. One does not know the real person behind the code-style names what BBO users use.

The players face most of the abovementioned problems in other online bridge environments as well. Most common are the first and second point from the list and none of the environments are really using advantages of social play (like play over Facebook). The latter one is considered by us as one of the most important elements of bridge play as the bridge is a social play in its essence and the game has been usually learned from friends or family – as high as 87.9% of U.S. bridge players have learned it from friends and family (American Contract Bridge League, 2005). Therefore social element is key for bringing in new players into the bridge community.

With that in mind it became apparent that Facebook's software platform was the right fit. The current leader of social media sites is Facebook still with over 1.2 billion users (Statistic Brain, 2012). Also, an interview with our consultant, Oliver Loper, had revealed that the countries where bridge is most popular and widely played (United States, Turkey and Poland) are also well

represented in Facebook. Number one in the table of users by country is The United States of America. Facebook users in the United States amount to 158,855,340 (Socialbakers, 2013). Turkey with 32,716,660 monthly active users takes 6th place (Socialbakers, 2013). Poland with 10,676,820 monthly active users ranks 23rd (Socialbakers, 2013).

What concerns direct competitors, then there is one online bridge play application in Facebook called Our Bridge. By our analysis there are several factors affecting the usage of the app (including the design and UX of the application), but the most inconvenient one is that one has to reject three different advertisements before one can access the game. Its overall monthly and daily active user's number has decreased and at the date of 22.04.2013 it was respectively 483 and 11 (Metricsmonk, 2013). Our assumption is that the biggest reason for this is that bridge play is for this company one among the others (Our.com, s.a.). Therefore they are not focused on taking the bridge play social.

Alongside validating the idea with potential customers and advisers we also decided to apply to an accelerator in order to validate our idea as investable business. The answer was positive and we received an offer to join the accelerator on 12.02.2013.

After analyzing all of the information above, we decided to continue with the next phase – developing minimum viable product, which we discuss in third chapter.

2.1.4. Design thinking

First of all, because the word design has many, according to the Longman Dictionary of Contemporary English as much as 7 definitions for it, it must be stated that this paper will look at design in the light of design thinking (Bullon, 2003). As Tim Brown puts it: "*Design thinking is a discipline that uses the designer's sensibility and methods to match people's needs with what is technologically feasible and what a viable business strategy can convert into customer value and market opportunity.*" (Brown T. , 2008)

This definition is suitable because it covers methods - meaning it isn't an intuitive artistic pursuit,

but there are tools to carry out the work. It takes into account the user experience, which gives us input and feedback about the development in the absence of other metrics. Technological feasibility is important because it forces to take into account the technological boundaries so that the designs can actually be implemented. It also takes into account the economic viability, which means there are business goals that can be measured against. The following section will expand on some of the methods that are relevant in the context of Scrum.

Benchmarking is essentially the process of determining the highest standards for products, services etc. in the industry, and using them to make the necessary improvements to reach or surpass these standards. (Lewis, 2009) The process is relevant because it is a tool that takes into account the ever-changing world and competition. The results of benchmarking are reflective of the time they are carried out in. It can be used continuously to reassess the situation as new competition arises.

The results of the benchmarking give input for sketching in the form of standards. Jonathan Fish and Stephen Scrivener have shed light on the subject of **Sketching**, saying: *"Sketching, or the production of untidy images to assist in the development of visual ideas, is one of the oldest and most familiar activities of artists and designers"* (Fish & Scrivener, 1990). The untidiness of the process implies speed, with which the designer can implement different visual and structural ideas. Speed is a crucial factor in agile development and more specifically Scrum.

Furthermore, Fish and Scrivener bring out that *"the necessity to sketch arises from the need to foresee the results of the synthesis or manipulation of objects without actually executing such operations."* (Fish & Scrivener, 1990) This allows a Scrum team to already deliver something tangible for the user without actually having to build anything in the early stages of development.

A step up in complexity is using **wireframes**, defined by Dan M. Brown as *"A simplified view of what content will appear on each screen of the final product, usually devoid of color, typographical styles, and images. Also known as schematics, blueprints."* (Brown D. M., 2011) Wireframes already display decisions and structure whereas sketching was about developing ideas. Yet again the lack of detail makes it a fast and therefore suitable tool for the agile processes.

Mockups are generally one page static representations of the design. High-fidelity mockups are more visually detailed version of the latter. It allows to polish and to refine visual details usually in later stages of design. (Brinck, Gergle, & Wood, 2002) Since high fidelity mockups are about working with details, it fits into the overall incremental development of the project as a logical step after wireframes.

Prototyping is creating a working model, essentially a demo, of a product (PC Magazine, s.a.). In the context of web applications it allows to test not only static pages but also the interaction with the user and the flow between pages. Prototyping is used mainly for user testing, gathering feedback about the developing product. They can be made more quickly since they don't require time-consuming coding-work (UserTesting.com, 2012). It is an efficient way to facilitate learning about user needs.

2.1.5. Development tools

In this paragraph we take a look at the technologies we use in our online product development.

Java is a programming language. In the Java Language Specification it is defined as "*a general-purpose, concurrent, class based, object-oriented language.*" It is designed to be simple enough that many programmers can achieve fluency in the language. (Gosling, Joy, Steele, Bracha, & Buckley, 2013) Today Java is one of the most used programming languages in the world (Welton, 2011). Java was mainly chosen because of the availability of the programmer in the team. However being one of the most used programming languages it would suggest it is relatively easy to find additional programming work force.

A **web application framework** is a foundation designed to help developers build web applications. A framework provides core functionality that is common to web applications. These functionalities may include things like session management, data persistence etc. Using a framework can therefore save the developer valuable time. (DocForge, 2013)

The play framework, which is a high-productivity Java (and Scala) web application framework, uses lightweight, stateless, web-friendly architecture. It has minimal resource consumption, which makes it good for highly scalable applications. (Play 2.1 documentation , s.a.) Because it was “*Written by web developers for web developers*”, the framework is more in line with the Architecture of the World Wide Web than it is with Java Enterprise conventions (Hilton, 2010). It includes good usability features like showing errors with the corresponding code in the browser and automatically reloading Java classes, making it relatively easy to use (Schmidt, 2010).

Hyper Text Markup Language or HTML is a language for describing web pages (Musciano & Kennedy, 2000). HTML5 is the newest version of that language and it is currently still in development (W3C, 2011). However the latest versions of Safari, Chrome, Fireworks, Opera and Internet Explorer all support HTML5, if not all its elements. (Deveria, 2013) (Koch, HTML5 compatibility, 2013) The language is backward compatible, so using it is no problem. HTML5 enables using SVG (Scalable Vector Graphics) vector graphic elements inside the document. This allows us to use fewer images in the designs and therefor reduce file size and make faster pages. (W3C, 2011)

The most popular language for defining website style and layout is Cascading Style Sheets, which is more commonly know by its abbreviation – CSS. (Meyer, 2000)

CSS3 is a term used for all further developments that come after CSS2.1. The World Wide Web Consortium (W3C), international community that develops open standards to ensure the long-term growth of the Web, has stopped using the version system since it makes development slow and maintaining it difficult. Instead the development has been divided into independent modules that can be added on to the language. (Atkins Jr, 2013)

All though, similarly to HTML5, CSS3 is still in development, the decision to use the latest version came easy because already modern browsers tend to support many of the new properties (Deveria, 2013) (Koch, CSS - Contents and compatibility, 2013). CSS3 provides new stylistic opportunities like opacity, shadows and rounded corners, which can be used to enhance the user experience (De Silva, 2010). In addition CSS3 can be used for doing animation, which can further enhance usability (Waterhouse, 2011).

JavaScript is a scripting language. It is mostly used in web browsers. More specifically we are talking about one version of JavaScript – client side JavaScript. As Flanagan puts it: "*When a JavaScript interpreter is embedded in a web browser, the result is clientside JavaScript*". That is also the most common version of JavaScript. It allows user interaction on an otherwise static page. It can be easily used in HTML pages and is supported by all modern web browsers. (Flanagan, JavaScript - The Definitive Guide, 2006) For example it allows to ask users to input data, to move elements (Flanagan, JavaScript Pocket Reference, 2012).

2.2. Data collection

In this section we are going to give an overview of data collection activities that we use to conduct research. We cover gathering information about our own implementation of Scrum in section 2.2.1, gathering qualitative comparative data in 2.2.2 and gathering quantitative data to find out whether other early stage startups in a similar position to ours are having the same problems in 2.2.3.

2.2.1. Observing our team

Participant observation is a way of collecting data in field research, used in anthropology and sociology, but other research as well. In this method, the investigator studies the life of a group by taking part in its activities. (Guest, Namey, & Mitchell, 2013)

Of the five different categories introduced by Spradley, we are using Active Participation. In the case of one of our research we can say that Complete Participation was used. Researcher becomes a member of the group by fully embracing skills and customs for the sake of complete comprehension. This method permits the researcher to become more involved in the population. There is a risk of "going native" as the researcher strives for an in-depth understanding of the population studied. (Spradley, 1980)

Taking into account the risk of going native and considering the fact that in our case we are observing our own work, we use triangulation to minimize the biases of single researchers. By triangulation we mean using multiple researchers to gather information about the same event. (Lincoln & Guba, 1985)

2.2.2. Mentoring other teams

To test the hypothesis, posed in the introduction of this research paper, we decided to mentor three other similar teams in using Scrum to find out whether their experiences can validate our own findings.

Mentored groups will be referred to as Team A, Team B and Team C. We call the teams with arbitrary names to protect their privacy. All mentored teams are developing software products and are in the very beginning of the product development cycle. They fit well into our scope of early startups. All the teams followed are working mostly remotely where they have rarely physical meetings. Most of the team members have a day job in addition to their new development venture. Team members from Team A and Team B are all physically located in Estonia. Team C has a team member from Austria in addition to Estonians.

We helped the teams to switch to Scrum from their previous development model. Within the mentored teams we took on the role of Scrum Master. We mentored Teams A and B for two one week sprints. With Team C we had a longer mentoring session due to their results. In this paper we have covered three sprints results Team C. We gave them a course in Scrum and its ceremonies. We made suggestions of best practices, but we let the mentored teams make their own decisions on how to exactly implement Scrum. This included decisions on the communication platform, backlog management, how and when they handle Scrum ceremonies.

We collected background data on the teams and their members, including their family status, obligations outside the project and motivation. We mapped the findings before and after the

mentoring and asked for their own opinion how the Scrum worked and whether it improved their product development practices. The results were compared to the survey results what we executed.

2.2.3. Survey to early stage startups

We have developed a survey to collect data about the team dynamics, obligations and team member's personal lives to investigate if and how it affects the teamwork and product development. The main reason for the survey is to find out whether other early stage startups in a similar position to ours are having the same problems.

Kirkman and Benson (1999) define a good survey questionnaire, which consists of 26 questions on team empowerment measurements. They have also looked into how team performance is affected by demographics. (Kirkman & Rosen, 1999) Their research has been expanded on by Kirkman, Rosen, Tesluk and Gibson with special focus on measuring virtual team performance (Kirkman B. L., Rosen, Tesluk, & Gibson, 2004). We can learn from the past researches that the team empowerment level will have a big effect on the efficiency and results of the team (Kirkman & Rosen, 1999). We also looked at sample team management surveys put together by online sites like QuestionPro (Question Pro LLC, 2013) and agile software development measurement metrics (ScrumSense CC, 2009).

One of the additional aims of the survey is to study the aspects of working from home. We assume that most of the projects we follow are mainly being worked on from home. Working from home is a good way to have time for a side-job like the project under investigation in this paper, but there are many challenges to this. People get tired from their day jobs, many have families and children who need attention and people need to take care of their households. Working from home requires also a great deal of self-discipline, focus and motivation (Truex, 2009). There are different times, which suit different people when it comes to executing the work. Some prefer mornings, some evenings. Then challenge is how to balance the time so it suits everyone. (LaRowe, 2009)

Based on abovementioned previous research and our thoughts on the monitored teams, we have put together our survey questionnaire (Appendix 2). We have divided the survey into three subsections, which pertain to demographics and personal data, person's involvement in the project and leadership of the team respectively.

We sent the survey out with the ability to identify how each receiving team answered. First we sent it out to a few specially selected teams whom we knew before as a validity check. We reviewed their answers and compared it to the knowledge we had about them to ensure that we have valid answers. By doing this we were able to get assurance that our survey was worded in an easily understandable way. To determine the needed sample size we looked at the sampling rate theory prepared by Jacob Cohen, which was easily visualized by David (David, 1987, p. 215). Based on these actions we can say that our survey is representative.

3. RESULTS

3.1. Our product development

Our core team consists of four members who are all male and in the age group 25-49. Three of the team members have kids and all are working from home with the project as a side-job. Time spent for product development varies – two are spending up to five hours, one five to ten hours and one ten to 20 hours per week. There is an additional team member, a programmer, who joined the team after validating the idea. Whereas we acknowledge him as a team member he does not belong to the four core team members in the context of this research.

All of the four core team members participated in observing the process of using Scrum. We observed ourselves developing our product over the course of 4.5 months. From the 9th November 2012 to the 26th February 2013. This time includes the long and slow process that preceded the actual implementation of the Scrum framework and the 2 weeks of Scrum sprints. The observation was cut short because we lost our developer on the 26th of March due to unforeseeable circumstances.

On the 9th of November 2012, we agreed on developing the product in the agile way. We used design methods that support incremental, iterative process however we did not use any concrete framework at that point. At the start of the project we used only one of the Scrum artifacts – the task board. At that point we used two tools: Trello, for housing the task board and Skype chat for communication. The first task board consisted of items like: login, bidding, dealing cards. The board itself was divided into three columns: To do, Doing and Done.

Until the beginning of February we met irregularly at school and divided some tasks between ourselves only to come back next time and see that things are not moving along. We needed some kind of framework that would help us to manage our team, motivate ourselves and help us to focus better on tasks. We also realized that the "To do" items are far too big to perform as tasks and were varying in sizes. There were gaps in communication ranging from a few days to a whole month at one point. It became clear from our own experience that the lack of overview is demotivating to the team. Those were the reasons that made us aware that a change was needed.

From the 11th February, we started adopting Scrum. We introduced sprints and daily standups in the form of Skype calls. The daily standups were set to take place in the evenings, post work, 10.30 PM.

It was immediately clear that daily standups increased visibility of the work being done. The daily standups also increased communication between team members. Before adopting Scrum work was mainly done on the weekends. After adopting Scrum it appeared that more and more work was done on weekdays after business hours, increasing the speed of the project development.

There was some uncertainty regarding how to proceed after the first sprint. The question was whether to hold a retrospective meeting or just continue with the second sprint. It was agreed that the retrospective meeting should be held and new deliverable for the upcoming sprint should be agreed upon.

We held the retrospective meeting on the same day that the second sprint began. In the meeting we discussed what went well and what could be improved. The main issues were the lack of time for the development work and the time of the daily standup, which we all felt was too late.

As a result of the first sprint we managed to deliver the first HTML, CSS coded layout of the bidding screen, getting to the table, using the Fischer-Yates shuffle. Despite that, the first sprint failed because we took last minute commitments than we could not deliver. Also the daily standups ran over the 15-minute limit, because we were adjusting to the format and got off topic.

One of the findings from first sprint was that our backlog wasn't detailed enough. So we created a

new Scrum board with an added backlog column on the 15th of February. The backlog items had to be rewritten in more detail by Product Owner.

The 18th of February – start of the second sprint. We moved the time of daily standups a half hour earlier – to 10:00 PM.

During the sprint several daily standups were missed. Again we concluded that evenings might not be the ideal time for daily standups. It often happened, that during the day, some unexpected new obligations or tasks outside the bridge project turned up. That moved the standup to a later time, thereby ruining the rhythm, or causing team member to fail to show up all together.

As a result of the second sprint we managed to deliver only the designs for the gameplay screens (Open cards in south, open cards in south and east, open cards in south and west, open cards south and north). None of the development commitments were completed by the end of the sprint. Due to different conflicting commitments outside the project some of the team members were not able to take commitments on daily standups.

There was a significant drop in the team member's appearance in daily standups. To address this communication issues we decided to hold a physical meeting – Retrospective and grooming of the backlog, planning meeting for the next sprint. The retrospective issues were largely the same. We pinpointed the biggest issue – the daily standups were still too late and we all agreed that evening standups were not working at all and need to be changed. We further improved the backlog for the next two sprints. We added a new column to our Scrum board in order to separate backlog and sprints completed from current sprint.

The start of the third sprint was initially delayed because we wanted to be thorough in the planning we had started at the end of the previous sprint, to improve our quality of work. It is at this point that we found out that our only developer had to step out of the project. At the time of writing this research paper, we have found a new developer and are in the process of introducing him to the team and project. We are prepared to continue with the Scrum.

We used Scrum for our product development for a total of two weeks. We agree that there are daily virtual meetings and the virtual tools in use were effective. We found that they increase visibility and productivity. We concluded that physical meetings help to keep the virtual communication going. After implementing Scrum, the division of tasks within the team became clearer. We agree that using Scrum helped us to divide work into small and manageable tasks.

3.1.1. Building Minimum Viable Product

This section will describe the building process of the minimum viable product. In the context of this paper we define it as:

A working application on Facebook, where It is possible for four players to join the table, start a game of contract bridge, with a set contract of one no trump (1NT), having a deck of cards shuffled, dealt out and be able to play till 200 points, after which the winners will be announced.

We describe the MVP in the terms of design and development tools.

We started with benchmarking the competition, collecting data about the most popular current bridge sites. We tested different leading competitors' sites, collected screenshots, analyzed them and talked to people why they use these sites and what they think of it. As suspected we found evidence that there were problems with information design and usability.

Based on our findings we made assumptions. We assume that the fewer steps it takes to get to the game, the higher the user retention rate will be. We also assume if we increase contrasts between different functions, we can guide the users towards their main goals, making the experience more enjoyable. All though these factors seemed plausible, we didn't know whether they would have the desired effects.

So based on our assumptions we formed initial hand drawn sketches. We simplified the design in contrast to the competing applications, cleaned up the interface and reduced the amount of clicks it takes to get to the game. For example: in case of our main competitor, BBO, the amount of clicks to start a game with your friends was 13 clicks. We reduced that amount by half. The sketches allowed us to quickly gather feedback to visual ideas. Based on the feedback to sketches, we then adjusted our design and went into more detail. For that we used digital wireframes.

Thus far we had been dealing with feedback too ideas and concepts. With the digital wireframe we could add navigation functionality. This was our first prototype. We could actually test how the user interacted with the design uninterrupted. This allowed us to collect feedback about the functionality and usability. When we had ironed out most of the big flaws we moved our focus to the visual appearance.

To that end we created high fidelity mockups, which enabled to collect feedback on things like typographic styles, colors, graphics etc. This process was more time consuming than the less detailed wireframes and sketches beforehand but it let us test the full experience. The mockups we created were static, because there is no point in wasting potential development time to judge aesthetics. Now that we had also sufficient feedback on visual development, it was time to start building the actual product.

We cut the design into pieces and started inserting them as images into HTML and CSS code as images to get the general structure. Then we started gradually working through the images in the code, rewriting them into HTML objects and defining their visual styles in CSS. The first coded screens did not look a lot like the mockups because it was important at that time to deliver front-end functionality. It was at this time that the backend development process got started by laying down the framework. The development started with the game mechanics like shuffling and dealing cards. As the designs became HTML and CSS files, they could now be integrated into the software. Shortly after that we lost our developer and the project grinded to a halt.

As of writing this research paper, the minimum viable product as described has not been developed.

We have found a new developer and started working him into our team. We plan to continue with Scrum and the product development until we meet our goal, but in the context of this paper, we failed to deliver.

3.2. Findings from incubated teams

In this section we look at the results of the three early stage startup teams we mentored in the implementation of Scrum.

3.2.1. Team A

Team A is developing an online solution for small kids to securely share pictures of their doings. The team consists of five members of whom three are male and two female. The team had one member who is younger than 25, rest were in the age group of 25 to 49. Most of the team has a family and kids. One of the team members conducts his work for the project from his main workplace while the rest work on it from home. All team members are employed elsewhere and are working on this project from their free time. The team members dedicated up to 10 hours a week on their project development, including two who worked maximum five hours a week on it. One of the team members was not sure about the co-operation on solving the tasks, while the rest agreed that they work effectively together.

The team is not fully happy with their choice of the virtual tools they use and they have issues with role division where none of the members are agreeing that they have a clear division of roles. They agree that their work tasks are small and manageable. The team members have a different view on their meetings structure, where they think that they have virtual meetings either daily, few times a week or irregularly.

Mentor helped the Product Owner to prepare the backlog before Scrum kickoff. The team decided to use Trello for product backlog and Skype for daily communications.

The first sprint failed Scrum wise. Despite that the team members found that implementing Scrum raised visibility of their project. One of the tasks was testing the designs on a target group and analyzing the existing interview results. Using Scrum helped the team to realize, that there was no need for conducting further interviews.

Also they liked the standup format over Skype as meetings covered several time zones and one participant used usually public transportation during morning standup time. "It is really, good having daily standups, as I understand what team does now", said one participant. "Standups make me commit more to project", said another.

For second Sprint we helped teams Product Owner to prepare better and helped them create decent sprint backlog into Trello environment. It was good, that we had first Sprint started "on fly" and team understood now at once why those parts of Scrum are needed.

Second Sprint failed also mostly, but was more successful than first one and team knows now what they do. They plan to continue with Scrum methodology for creating their product.

3.2.2. Team B

Team B's business idea is to build an environment for sharing and exchanging 3D drawings, they planned to make money by offering 3D printing service.

The team consists of three male founders who are all in the age group of 25 to 49. They all do their work on the project from home; one of them has a family and children. All of the members of Team B are employed outside the project.

Two team members have stated that they work on the project 5 to 10 hours a week and one is able to dedicate to it up to 5 hours a week. They all believe that their tasks are divided into manageable

chunks. The team uses Trello for managing the work items and Skype as communications platform. They do not rate their virtual tools highly, only one team member thinks their effective.

Survey results show that there's different understanding on many questions inside of the team. One of the members does not rate high the co-operation in working together and there's disagreement on the role division of the team members. To question, "There is a clear division of roles in your team" the responses were "disagree", "agree" and "Neither disagree or agree". There is also a difference in the responses to question: "How often you have planned virtual meetings?" One responder says that they have virtual meetings once a day others think they have them 2-3 times a week.

The team built a backlog together in Trello and agreed to have one-week sprints and to have a daily standup every weekday evening at 7 PM over Skype call.

At the start of sprint 1 the team decided which tasks they will do and preliminary also set personal commitment to do these tasks during the sprint.

Team B had problems with people's commitments. On daily standups one person was present on all meetings, one was present only a few times a week and one participated only once a week.

Not much happened in first sprint and the sprint failed, as none of the tasks were done. They still got the feeling that work is started and going to happen. People knew what exactly they needed to do, who will do what part and they gave constant input how things are going on their side.

Retrospective findings for sprint one were:

- Team members noted that they day job affected their ability to dedicate to the project work.
- Team members noted that their social life stopped them from participating in the project work.
- One team member raised an idea to stop doing daily standups, but it was voted down by other team members.

As an improvement over sprint one team member promised to improve his/her time management and participation on the daily standups.

On the second sprint we saw similar issues to the first one – some team members did not participate on the daily standup, as they should have. On this sprint they completed a few tasks, but they were unable to finish all of the planned sprint backlog items. Therefore the second sprint also failed.

The main finding from the second sprint was that if someone does not show up, then other team members should try to reach that person, not just start without him. It was also noted that often times the obligations outside the project took priority.

In summary we can see that the Team B had big issues in time management between their day job, moonlighting project and social life. Their motivation to work on the project was questionable and it probably had an effect on the results. One of their team members felt hard to participate on the daily standups, as he did not complete his promised work often. All in all the team found that implementing Scrum did help them to make the issues visible for the whole team and to raise motivation. At the end they admitted to liking the framework and were optimistic about applying it in the future.

3.2.3. Team C

Team C is a nonprofit project that aims to develop a solution to track the production chain of various goods from raw materials till disposal. They want to show from where each ingredient of each product came from, how it was processed and manufactured.

The Team C software development team consists of six people. They have in the team a software developer, database developer, designer who is also Product Owner and two software testers. Their team has four male and two female members. They're all from an age group of 25 to 49. Their database developer is located in Austria and rest of the team is in different places in Estonia. Half of the team has kids. Some of the team members are not employed full-time. They are either at home

or working as freelancers. They are able to dedicate more of their time to the project. One person in the team conducts the work for Team C matters in his day job office. All of them are dedicating more than 10 hours a week for the project and many of them more than 20 hours.

Everyone on the team agrees that they have a good co-operation ongoing to develop the project. There is a person who does not agree that they have a good online collaboration tools in use. There is an issue with roles clarity in the team one person does not think that they have it solved well and two are unable to make up their mind about it. The team has in addition to regular virtual meetings also physical meetings for few members who are working on the product backlog.

The team was already before using Pivotal Tracker to track their backlog and work, now they amended the task items to fit them into Scrum structure. They agreed to one-week sprints, with sprint retrospective, demo and planning being held on Sunday evening at 7 PM. They have their daily standups over Skype group chat with a rule that the update must be published every morning by 10 AM Eastern European Time (subsequently EET). They have once a week a sprint planning session where they plan the next weeks sprint backlog.

Team had prebuilt their backlog for sprint 1 planning meeting, we had to work on adjusting the items a bit only to fit them for Scrum and to split few of them to smaller tasks.

The team kept their commitments well during the sprint. The daily updates were posted to the chat by everyone and on time. They had few syncing issue at the end of the sprint where the software testers did not get the information about development finish, which delayed the start of testing process.

The first sprint failed. The team had managed to complete all sprint backlog items, which were planned for the week, but the team had decided during the week to take on an additional task, which they were not able to deliver.

Sprint one retrospective notes were:

- People were happy with the process and especially with the fact that they saw a live result of their work.
- Visibility increased
- Scrum was treated as a new thing
- There were information syncing issues around the development progress.

The team agreed to speed up their communications to avoid misinformed team members. It was agreed whenever a part of work is completed it would immediately be communicated to other team members in the common group chat.

The second sprint failed. The failure was caused by a fact that one tasks development was finished few minutes before the start of a sprint demo, which meant that it was not published to agreed server and that it was not fully tested. The team found that the first weeks agreed improvement step worked well for them.

Sprint two retrospective notes were:

- The team should report also testing results in case no issues are found.
- There were issues with bug management.
- Last development item got delivered after the deadline.
- The weekly meeting on Sunday evening seemed not to fit everyone.

Agreed improvements for the next sprint were to move the weekly meeting to Monday 6.30 PM EET.

The third sprint succeeded. The team was able to deliver all the planned tasks. The team velocity was around the same level as on the second sprint. The team found that newly agreed meeting time works very well for them. On sprint three the Product Owner was away for the duration of the sprint.

Sprint three retrospective notes were:

- First succeeded sprint.
- Issues with daily updates after public holiday.
- One team member was few days away and did not give notice to the team.

Agreed improvement for next sprint was that everyone who is going to be away will need to give notice to the team on the sprint planning to take it into account when deciding the team commitments.

When planning the third sprint, team was looking on the development task estimates. After they had taken into sprint a certain amount of point's Product Owner raised the awareness of the team about the usual velocity line. Team C was able to predict velocity by the end of the third sprint.

Team C team is highly motivated to solve their cause. They work well as a team. They were willing to note in case something did not go as planned or in case they failed to do what they promised last time during the daily standup. Implementing Scrum helped them to gain additional visibility to their development process and to focus on the most important development items. The team continued using Scrum after we finished mentoring.

3.2.4. Summary of findings

Our mentored teams acted rather differently throughout the mentoring period. Motivation levels to develop their project varied among the three teams. In teams A and B the motivation levels differentiated vastly between the team members. Team C had a considerably higher motivation compared to teams A and B. They all believed in their causes and they put their best efforts into delivering the results.

Team C never had trouble with show-up on daily standups or other meetings, while it was a common issue with other teams. Team C members contributed much more of their time to the project development work. Survey questions, which measured the team co-operation and single

understanding, showed that the Team C answers were overlapping much more by the team members than on other teams. They had a higher trust among the team members than other teams. We found on Team B that one of the team members was negatively minded about daily standups the reason was that he was often unable to keep the daily commitments. He was feeling bad about reporting on no progress.

All the sprints from the mentored teams failed, but this is not a bad thing according to Scrum theory, because on average half of the sprints are failing according to surveys. It's important that the teams learn from failure reasons and adopt their processes to avoid doing the same mistakes again. The teams learned from their failure reasons and improved their working process. Second sprint had a different reason for sprint failure than the first sprint for all the teams.

We did not notice any differences to the teams results based on which choices they made for daily standups, collaboration and communication platforms.

We were unable to verify a predictable velocity in our mentoring sessions. Only Team C was using the velocity estimation using the Fibonacci scale and they were unable to do a velocity estimated with input from all the team members. The rest of the teams were unable to make any meaningful velocity estimations. The accuracy of the velocity measurement was not possible to verify, as the mentoring period for the teams was too short.

All mentored teams gave us feedback that Scrum helped them to improve their project development process. Main positive notes were about raised visibility and focus. They all promised to continue using Scrum after the end of mentoring period. We offered to them a further guidance on next steps in Scrum in case they feel it to be necessary. We will continue to follow the teams from longer distance to see their future progress.

We found with mentoring that for all the mentored teams Scrum raised visibility of the progress, straightened the focus on the most important tasks and brought out issues. These findings are the three of the main results of Scrum implementation according to theory.

3.1. Results of the quantitative survey

In this section we look at the results of the survey we composed to find out whether other early stage startups in a similar position to ours are having similar problems.

We sent the survey out to 101 team members from 29 teams from which 56 people from 24 teams responded (rates accordingly 55.4% and 82.8%). The teams were selected as the ones with their product development rather in early stage - therefore we picked out 21 teams from Ajujaht competition and eight teams from the DDVE master course. The products developed ranged from virtual services to physical products on different fields.

All the respondents fell into two age groups: under 25 (10) and between 25-49 (46). Most of the respondents were male (40) and half of the team members had kids (28). The one's having kids fell into the second age group with some exceptions; distribution of kids between men and women was equal. Our assumption that most of the team members worked from home hold – 42 (72%), 13 did it at their current workplace and three on other locations. Two people responded that they work on their current workplace as well as home.

As our thesis deals with the issue of product development as a side project, we asked whether team members were employed outside current project as well – 75% responded yes to this question.

So as we can see, all the elements described in part 2.2.3 as possible factors affecting the person's ability to contribute to overall teamwork and product development exist among our respondents (having full time job, kids and working from home). So in following pages we give an overview of the responses given to the questions and in more detailed we analyze the most important outcomes combined with the questions analyzed above.

Our experience suggests that there are two main factors that we consider important to investigate as factors affecting the ability to contribute to teamwork: having children and regular day job.

As we know starting up a successful business is very much related to time spent on developing a product and investors look how much team members are dedicated to company before they invest. The general outcome of our survey is given on Chart 1.

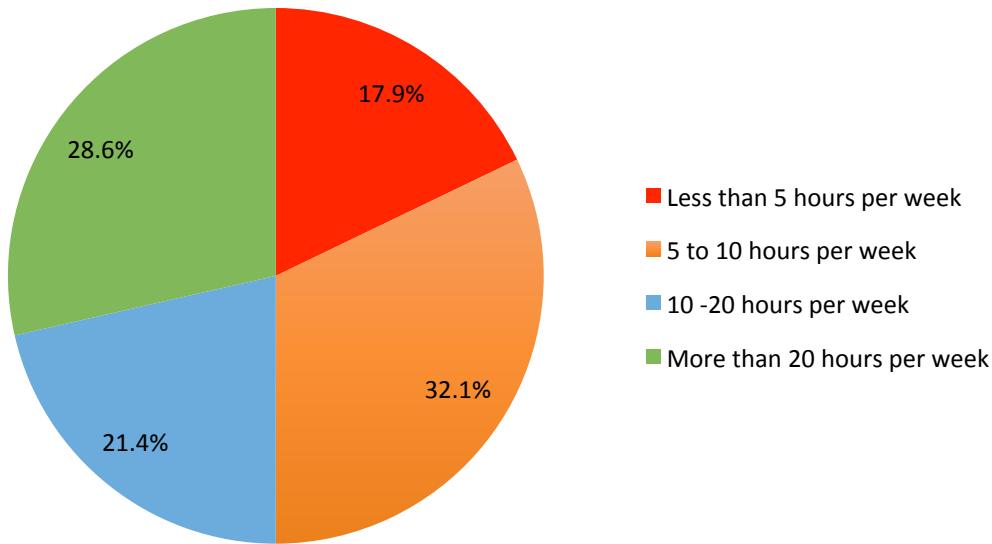


Chart 1. Time spent on developing a product (all respondents).

Most of the respondents (60.7%) spent either 5-10 hours or more than 20 hours per week on their product development. We analyzed the relationship between having the kids and ability to contribute to teamwork to find out if such connection is present.

As we can see from Chart 2, respondents who have kids, follow the pattern of total outcome i.e. the time spent is divided into the same groups as before (either 5-10 hours or more than 20 hours per week). People without kids have more flat distribution in the time spent on projects.

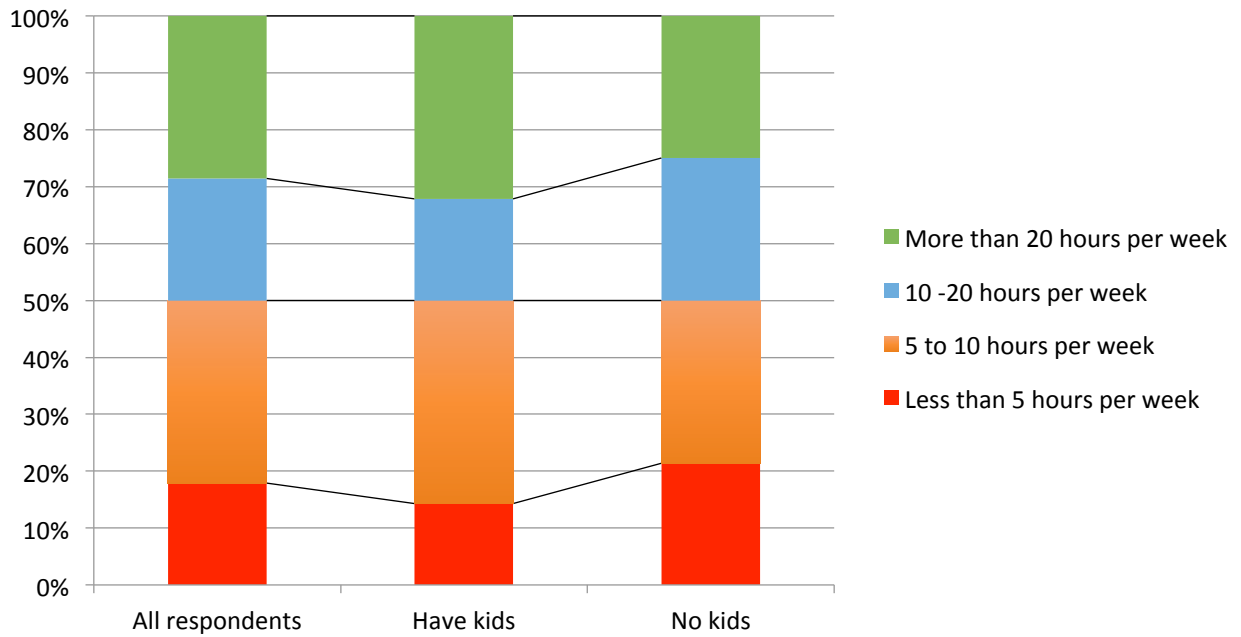


Chart 2. Having kids compared to time dedicated to teamwork.

We analyzed the responses to time spent question given by respondents having kids versus the ones who had no kids and it came out that having kids is not affecting the ability to contribute to teamwork as it was assumed in the beginning. That finding was also confirmed when we looked at the more detailed level of data from responses from Ajujaht team members – which of them made it to final TOP10 (semifinalists) and which one’s dropped out. After further analysis we found out that the reasons were not related to whether team members had kids or not, but there was much more simple reason: the teams who made it to semifinalists on average contributed more of their time to their product development (Chart 3).

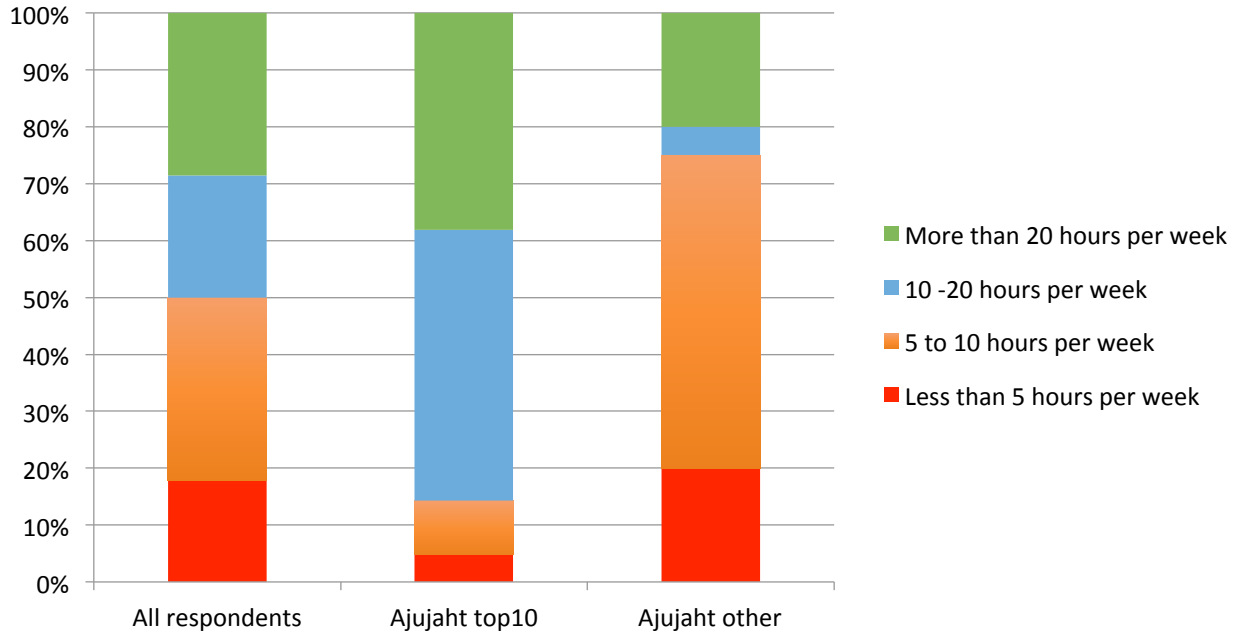


Chart 3. Ajujaht teams time contribution compared to average.

Next we looked how much employment outside the current teamwork influences the amount of time spent on product development (Chart 4).

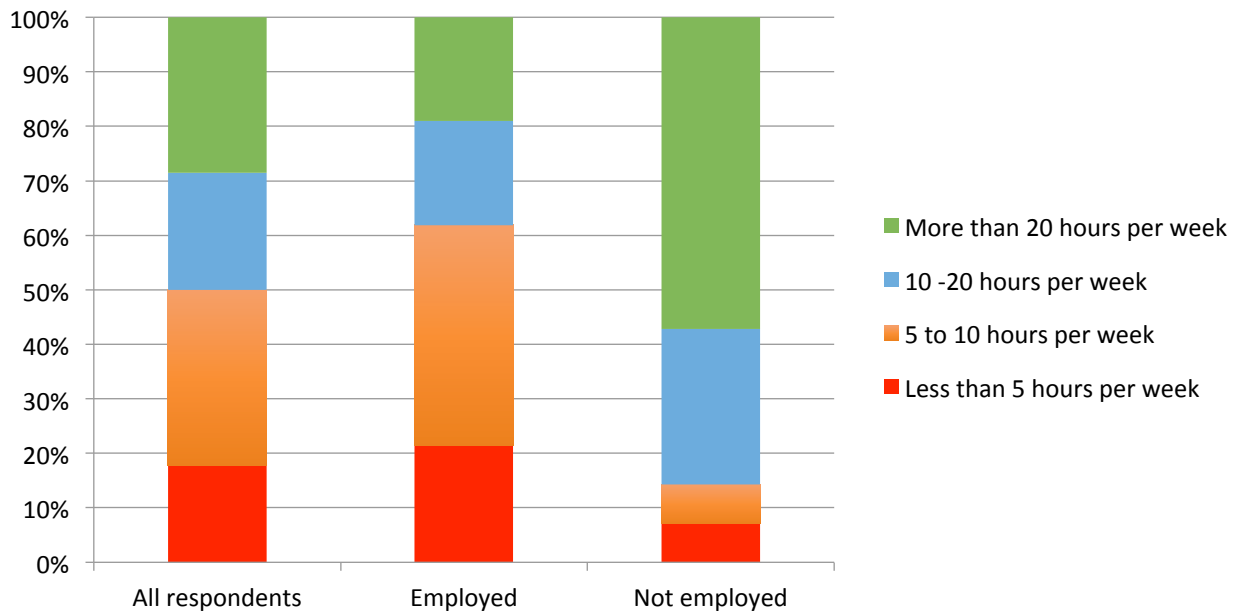


Chart 4. Employment and contribution to teamwork.

As one can see the team members who are not currently employed proportionally contribute more - taking percentage from total employed team members 19% of them contribute more than 20 hours per week to current project versus 57% of those who are currently not employed. This assumption is true that people who contribute to teamwork as side-job cannot do it as much as the ones who do it more or less full time.

When we looked at the Ajujaht teams who made it to finals, then percentage wise more employed team-members made it to semifinals than those who were not employed (accordingly 53% vs. 44%). The latter one can be explained by the number of people working for teams as side-project (32 vs. 9) and therefore the influence to final combined input to teamwork had higher impact (or alternatively- just the core product idea developed was better).

Now it's time to take a look how the teams work and how the team members assess their teamwork as such. On Chart 5 we have overall answers to team management questions given.

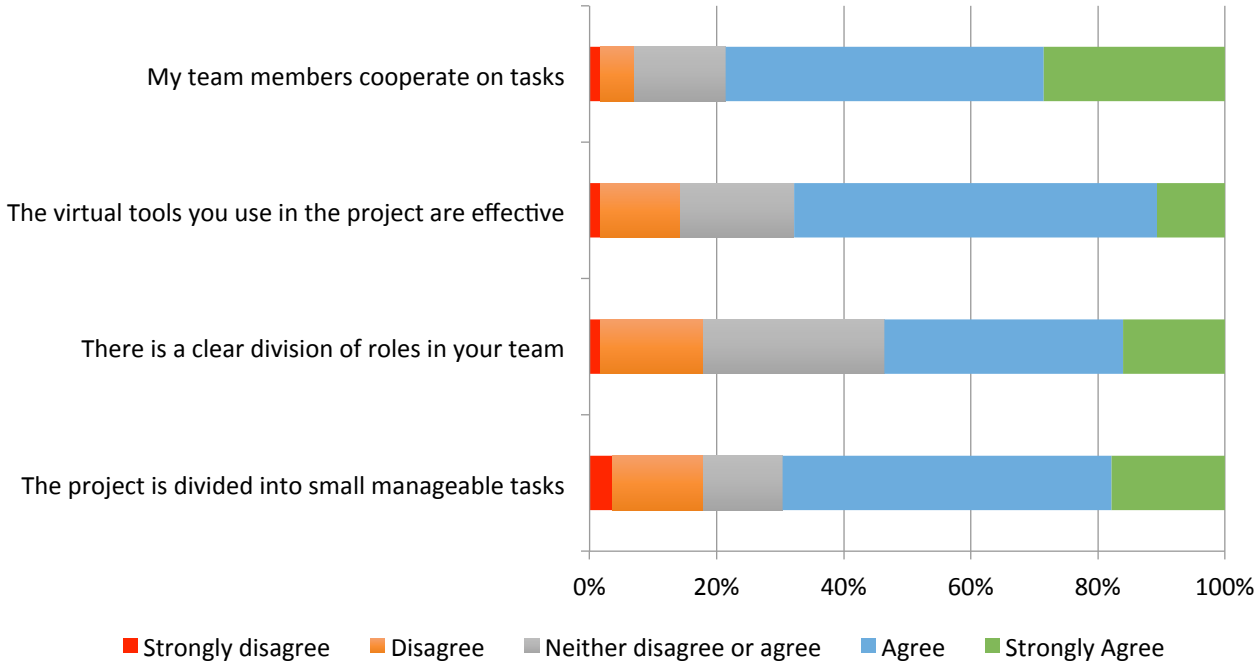


Chart 5. Cooperation and involvement in teamwork

As we can see from the chart in general team members agree that they cooperate on tasks. The more detailed we get with the questions the more opinions differ, especially when we get to the questions related to division of tasks and roles inside the team. The latter one was expected – this most often comes down to communication - if there is not enough managed communication between team members then such misunderstandings are easy to come (it comes out also from Chart 6- frequency of meetings, we'll come to that later).

After analyzing the above questions by team members who were employed or currently not employed we did not find very significant differences. Still we noticed one small but important pattern in answers: the ones who were not employed outside the project were more biased to positive answers (i.e. agree or strongly agree) than the one's employed outside the project. Interestingly we found the same comparing the question to having children or not.

So as we told before, often problems arise when people are not managing their communication effectively. Chart 6 depicts the answers given to questions how often the teams have virtual or physical meetings.

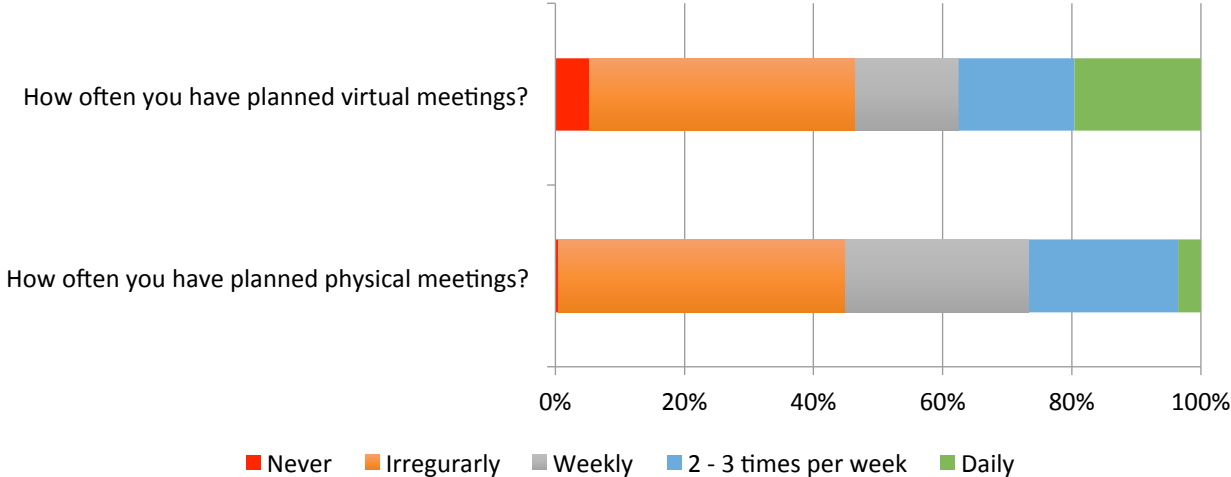


Chart 6. Frequency of physical and virtual meetings.

As one can see large proportion of the teams have very irregular meetings pattern. The virtual meetings are held more often than physical one's. When we analyzed the pattern in the case of

Ajujaht top ten teams *versus* the one's that did not make it to semi-finals, we saw that the one's who made it, had more regular meetings, incl. those who had more daily virtual meetings (8 vs 1).

From the analysis above we can already see that the teams have similar problems as we did in the beginning of card game bridge development – missing some guiding framework to work with in order to share effectively information and tasks among team members. In order to find out if the teams use any kind of framework we asked it as well: 62.5% of the respondents told that they do not use any kind of framework to manage their work. The ones who told they use something agile or Scrum and lean startup was mentioned, also some mentioned Trello, which is actually just one online collaboration tool, not framework as such.

Among the respondents who use some kind of framework were also two teams we were mentoring already during the poll (and we know they did not use Scrum before mentoring sessions): subtracting those and the one's answering Trello tool, only 23% using some kind of framework to manage their work is left.

In conclusion the questionnaire reveals that early stage startups are not using any kind of framework to manage their work – as much as 77% of the respondents admitted to having no framework in place.

Having kids or working from home does not affect input to teamwork according to our survey. Employment outside the project affects the ability to contribute to teamwork, as suspected.

CONCLUSIONS

The paper researched a Scrum implementation in a side project with distributed teams.

As a method to verify hypothesis that it is possible to effectively run a Scrum team for a side-project in a situation where all its members work remotely, and with varying amounts of time commitment from each participant, we observed our own project development process using Scrum as a framework. We also defined our Minimum Viable Product to determine whether or not the Scrum was successful.

To verify findings from our own project development we mentored three similar early stage startups. We also prepared and executed a survey to find out whether other early stage startups are experiencing similar problems to ours.

Unfortunately we have not been able to complete our own minimum viable product by the time of finishing the writing of this paper. In that sense we failed to run Scrum effectively. We got delayed on the software development work as we needed to replace the programmer who left in the middle of the development.

Despite the failure to deliver the MVP the team unanimously decided that using Scrum helped in development process:

- The daily standups increased productivity because people felt obligated to take commitments.
- The virtual tools used helped the communication.
- We found that the Scrum board increased visibility.

- We found that physical meetings helped to keep the virtual communication going.
- After implementing Scrum, the division of tasks within the team became clearer.
- We agreed that using the product backlog helped us to divide work into small and manageable tasks.

All of the mentored teams gave us the feedback that implementing Scrum had helped them to improve their project development process. The main points that they brought out were raised visibility of progress, increased focus and discovering issues that were otherwise unnoticed. All the teams stated that they would use Scrum in the future as they found it beneficial.

Our survey confirmed our assumption that early stage startups are not using any kind of framework in order to manage their work. An interesting pattern emerged from the survey - the ones who were not employed outside the project were more biased to positive answers than the one's employed outside the project. Interestingly the same applies for people who have children.

When the first observation can be explained by the fact that the people not employed outside the current project have on average more time to analyze better different tasks and roles necessary to fulfill, then the latter could be explained by the fact that team members with the kids are probably more valuing their time and therefore focusing more to the tasks while they are working on them. There might be many other reasons for that as well - that could be an interesting topic to rise as a separate research question in the future

The survey also revealed that remote working from home does not have a significant effect on results and the more often communication tools were used, the less issues came up from remote working.

Previous researches have suggested that Scrum can be too process heavy for early startup companies with only a few founders on board. We believe that this concern can be addressed by using a lighter version of Scrum in the beginning and extend the toolset as the team grows and gets used to it.

Minimum required elements of running a Scrum are daily standups, product demonstration, planning, retrospective meetings and some form of a backlog. Later the teams should consider implementing also backlog grooming to be able to take advantage predictable velocity and definition of done.

We found in our research as a side effect that the highest impact on team performance is the teams motivation to execute the project. The motivation deficit became apparent in delayed or skipped work items, in not taking commitments and in not participating in Scrum ceremonies. Even if that is the case, using Scrum can reveal the shortcomings in motivation within the team.

Taking into account the fact that most early startups did not use any kind of framework in order to manage their work, based on our experience and the feedback from the teams with whom we practiced Scrum, we are able to say that implementing Scrum can be very beneficial for raising visibility of progress, increasing focus or at least discovering whether there is a problem with motivation within a team.

BIBLIOGRAPHY

- Бридж на Гамблере.* (s.a.). Retrieved 04 24, 2013, from <http://www.gambler.ru/%D0%91%D1%80%D0%B8%D0%B4%D0%B6>
- A. Osterwalder, Y. P. (2009). *Business Model Generation*. <http://www.slideshare.net/techdude/business-model-generation>: Self published.
- A. Osterwalder, Y. P. (2010). *Business Model Generation*. New Jersey: John Wiley & Sons, Inc.
- agile42.com. (2012, 11 1). *Scrum Artefacts*. Retrieved 4 1, 2013, from Agile42 - The Agile Coaching Company: <http://www.agile42.com/en/agile-coaching-company/agile-info-center/scrum-artefacts/>
- Alex, Y., & Christian, M. (2013). *Is a Rigorous Agile Methodology the Best Development Strategy for Small Scale Tech Startups?* Retrieved April 27, 2013, from University of Pennsylvania: http://repository.upenn.edu/cgi/viewcontent.cgi?article=2025&context=cis_reports&seiredir=1&referer=http%3A%2F%2Fscholar.google.com%2Fscholar%3Fq%3Dscrum%2Bin%2Bstart%2Benvironment%26btnG%3D%26hl%3Den%26as_sdt%3D0%252C5#search=%22scrum%20start%20envi
- Ambler, S. W. (2008). *Agile Adoption survey February 2008*. Retrieved March 27, 2013, from Ambysoft: <http://www.ambysoft.com/downloads/surveys/AgileAdoption2008.ppt>
- Ambler, S. W. (2012). *Agility at Scale Survey 2012*. Retrieved March 27, 2013, from Ambysoft: <http://www.ambysoft.com/downloads/surveys/StateOfITUnion201209.pptx>
- Ambler, S. W. (2002). *Lessons in Agility from Internet-Based Development*. Retrieved April 27, 2013, from Agile Alliance: <http://cf.agilealliance.org/articles/system/article/file/1025/file.pdf>
- American Contract Bridge League. (2005). *Bridge Trumps Food and Fire*. Retrieved 04 24, 2013, from 2005 ACBL Nationwide Bridge Survey: <http://web2.acbl.org/documentLibrary/marketing/NationwideSurvey05.pdf>

- American Contract Bridge League. (s.a.). *How to Play Bridge?* Retrieved 04 30, 2013, from American Contract Bridge League: <http://www.acbl.org/learn/howToPlay.html>
- American Contract Bridge League. (s.a.). *Our Game, Our History*. Retrieved 04 30, 2013, from American Contract Bridge League: <http://www.acbl.org/about/ourGame.html>
- Atkins Jr, T. (2013, February 19). *A Word About CSS4 - Tab Completion*. Retrieved May 3, 2013, from Tab Completion: <http://www.xanthir.com/b4Ko0>
- Barringer, R. R., & Ireland, R. D. (2008). *Entrepreneurship – successfully launching new ventures* (second edition ed.). New York: Pearson/Prentice Hall.
- Barringer, R. R., & Ireland, R. D. (2008). *Entrepreneurship – successfully launching new ventures* (second edition ed.).
- Begel, A., & Nagappan, N. (2007). *Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study*. Retrieved March 26, 2013, from <http://research.microsoft.com/en-us/um/redmond/groups/hip/papers/agiledevatms.pdf>
- Berczuk, S. (2007). *Back to Basics: The Role of Agile Principles in Success with an Distributed Scrum Team*. Retrieved April 28, 2013, from Agile Alliance: http://agile2007.agilealliance.org/downloads/proceedings/081_Back%20to%20Basics_465.pdf
- Blank, S. (2010, 4 8). Retrieved 04 10, 2013, from No Plan Survives First Contact With Customers – Business Plans versus Business Models
- Blank, S. (2010, April 8). *No Plan Survives First Contact With Customers – Business Plans versus Business Models*. Retrieved March 10, 2013, from Steve Blank: <http://steveblank.com/2010/04/08/no-plan-survives-first-contact-with-customers-%E2%80%93-business-plans-versus-business-models/>
- Blank, S. (2010, April 8). *No Plan Survives First Contact With Customers – Business Plans versus Business Models*. Retrieved March 10, 2013, from <http://steveblank.com/2010/04/08/no-plan-survives-first-contact-with-customers-%E2%80%93-business-plans-versus-business-models/>
- Blank, S. (2011, December 13). *The Startup Team*. Retrieved April 16, 2013, from Steve Blank | Entrepreneurship and Conversation: <http://steveblank.com/2011/12/13/the-startup-team/>

- Blank, S., & Dorf, B. (2012). *The startup owner's manual*. K&S Ranch Publishing Division.
- Blank, S., & Dorf, B. (2012). *The Startup Owner's Manual*. Pescadero: K&S Ranch Publishing Division.
- Bodell, M. (2012, 05 08). *What is good and what is bad about playing online Bridge at BridgeBase Online?* Retrieved 11 30, 2012, from Quora: <http://www.quora.com/Bridge-card-game/What-is-good-and-what-is-bad-about-playing-online-Bridge-at-BridgeBase-Online>
- Bridge Base Online*. (s.a.). Retrieved 04 20, 2013, from <http://www.bridgebase.com/intro/introduction.php>
- Bridge Trumps Food and Fire*. (2005). Retrieved 04 24, 2013, from 2005 ACBL Nationwide Bridge Survey: <http://web2.acbl.org/documentLibrary/marketing/NationwideSurvey05.pdf>
- Brinck, T., Gergle, D., & Wood, S. D. (2002). *Usability for the Web: Designing Web Sites that Work*. Burlington: Morgan Kaufmann Publishers.
- Brodzinski, P. (2011, 4 3). *When to Use Waterfall, When to Use Scrum?* Retrieved 4 30, 2013, from StackExchange: <http://pm.stackexchange.com/questions/389/when-to-use-waterfall-when-to-use-scrum>
- Brown, D. M. (2011). *Communicating Design: Developing Web Site Documentation for Design and Planning*. Berkeley: New Riders Publishing.
- Brown, T. (2008, June 1). Design Thinking. *Harvard Business Review* , pp. 84-92.
- Bullon, S. (2003). *Longman Dictionary of Contemporary English*. Harlow: Pearson Education Limited.
- Campbell, A. (2008). *Tips for Incubating your Small Business Idea While Still Working Full-Time*. Retrieved April 14, 2013, from Enteprenaur Incubator Academy: <http://www.nbmedia.co.za/Downloads/Entrepreneurship/Tips%20for%20Incubating%20your%20Small%20Business%20Idea%20While%20Still%20Working%20Full-Time.pdf>
- Cockburn, A. (2008, 5 1). *Using Both Incremental and Iterative Development*. Retrieved 4 30, 2013, from Crosstalk Online: <http://www.crosstalkonline.org/storage/issue-archives/2008/200805/200805-Cockburn.pdf>

- Cohn, M. (2010). *Succeeding with Agile: Software Development using Scrum*. Boston: Addison-Wesley.
- Daniela, D., Casper, L., Maria, P., Arber, B., & Adrian, S. (2012). *Teaching a Globally Distributed Project Course Using Scrum Practices*. Retrieved April 27, 2013, from https://www2.lirmm.fr/lirmm/interne/BIBLI/CDROM/INFO/2012/ICSE_2012/icse2012/icsews12ctgdsd/p30-damian.pdf
- David, K. (1987). *Statistics for the social and behavioral sciences*. Boston: Little, Brown.
- De Silva, M. (2010, June 8). *What's New In CSS3? - Notes Of Genius*. Retrieved May 1, 2013, from Notes Of Genius: <http://notesofgenius.com/new-css3/>
- Deemer, P. (s.a.). *The Distributed Scrum Primer*. Retrieved March 26, 2013, from GoodAgile: <http://www.goodagile.com/distributedscrumprimer/DistributedScrumPrimer.pdf>
- Deveria, A. (2013, April 3). *Can I use... Support tables for HTML5, CSS3, etc.* Retrieved May 11, 2013, from Can I use... Support tables for HTML5, CSS3, etc: <http://caniuse.com>
- DocForge. (2013, April 9). *Web application framework - DocForge*. Retrieved May 2, 2013, from DocForge: http://docforge.com/wiki/Web_application_framework
- Downey, S., & Sutherland, J. (2012). *Scrum Metrics for Hyperproductive Teams: How They Fly like Fighter Aircraft*. Retrieved from Scrum Log Jeff Sutherland: <http://jeffsutherland.com/ScrumMetricsHICSS2013BWSsubmissionFinal.pdf>
- E.Ries. (12.03.2013). *The Lean startup methodology principles (blog)*.
- Fish, J., & Scrivener, S. (1990). Amplifying the Mind's Eye: Sketching and Visual Cognition. *Leonardo* , 117-126.
- Flanagan, D. (2006). *JavaScript - The Definitive Guide*. Sebastopol, California: O'Reilly Media, Inc.
- Flanagan, D. (2012). *JavaScript Pocket Reference*. Sebastopol, California: O'Reilly Media, Inc.
- Franke, N., Marc, G., Harhoff, D., & Henkel, J. (2006). Venture Capitalists' Evaluations of Start-up Teams: Trade-offs, Knock-out Criteria, and the Impact of VC Experience. Vienna.

- Gibson, C. B., & Gohen, S. G. (2003). *Virtual Teams That Work: Creating Conditions for Virtual Team Effectiveness*. San Francisco: Jossey Bass.
- Gosling, J., Joy, B., Steele, G., Bracha, G., & Buckley, A. (2013). *The Java® Language Specification*. Boston: Addison-Wesley Professional.
- Govindrajan, V., & Gupta, A. K. (2001). Building an Effective Global Business Team. *MIT Sloan Management Review*, 4.
- Graham, P. (2012, 11). *How to get startup ideas*. Retrieved 4 10, 2013, from How to get startup ideas: <http://paulgraham.com/startupideas.html>
- Great Bridge Links*. (s.a.). Retrieved 04 20, 2013, from <http://www.greatbridgelinks.com/gblPLAY/BridgeBaseOnline.html>
- Greenwald, T. (2012, January 31). *Business Model Canvas: A Simple Tool For Designing Innovative Business Models*. (Forbes Magazine) Retrieved March 11, 2013, from Forbes.com: <http://www.forbes.com/sites/tedgreenwald/2012/01/31/business-model-canvas-a-simple-tool-for-designing-innovative-business-models/>
- Guest, G. S., Namey, E. E., & Mitchell, M. L. (2013). *Collecting Qualitative Data - A Field Manual for Applied Research*. Thousand Oaks, California: SAGE Publications, Incorporated.
- Hilton, P. (2010, March 15). *Lunatech Blog: Play framework usability*. Retrieved May 10, 2013, from Lunatech Blog: <http://blog.lunatech.com/2010/03/15/play-framework-usability>
- Hossain, E., Babar, M. A., & Paik, H.-y. (2009). Using Scrum in Global Software Development: A Systematic Literature Review. *2009 Fourth IEEE International Conference on Global Software Engineering*. IEEE Xplore.
- Introduction*. (s.a.). Retrieved 04 20, 2013, from Bridge Base Online: <http://www.bridgebase.com/intro/introduction.php>
- Jessup, A. (2012). *Scrum: Can you build startups with scrum methodology?* (Quora) Retrieved April 28, 2013, from Quora.com: <http://www.quora.com/Scrum/Can-you-build-startups-with-scrum-methodology>

- Jonas, S., Robert, W., & Maic, M. (2010). *ABC-Sprints: Adapting Scrum to Academic Game Development Courses*. Retrieved April 27, 2013, from University of Duisburg-Essen: http://www.uni-due.de/~hk0377/papers/Schild_ABC-Sprints_Adapting_Scrum_to_Academic_Game_Development_Courses_FDG2010.pdf
- Kanban Blog. (2013). *What is Kanban?* Retrieved 5 6, 2013, from Kanban Blog: <http://www.kanbanblog.com/>
- Kawasaki, G. (2011). *Reality Check*. New York: Penguin.
- Keith, C. (2010). *Agile Game Development With Scrum*. Boston, USA: Addison-Wesley.
- Khaled, A. A., Eddie, B., Dean, S., & Scott, S. (2007). How do we manage student projects? *Journal of Computing Sciences in Collegues* , 22 (4).
- Kirkman, B. L., Rosen, B., Tesluk, P. E., & Gibson, C. B. (2004). The Impact of Team Empowerment on Virtual Team Performance: The moderating Role of Face-To-Face interaction. 47 (2).
- Kirkman, L. B., & Rosen, B. (1999). Beyond Self-Management: Antecedents and Consequences of Team Empowerment. *The Academy of Management Journal* , 42 (1), 58-74.
- Koch, P.-P. (2013, May 11). *CSS - Contents and compatibility*. Retrieved May 11, 2013, from QuirksMode - for all your browser quirks: <http://www.quirksmode.org/css/contents.html>
- Koch, P.-P. (2013, May 11). *HTML5 compatibility*. Retrieved May 11, 2013, from QuirksMode - for all your browser quirks: <http://www.quirksmode.org/html5/index.html>
- LaRowe, M. (2009). *Working Mom's 411: How To Manage Kids, Career and Home*. Ventura: Regal.
- Lepsinger, R., & DeRos, D. (2010). *Virtual Team Success: A Practical Guide for Working and Leading from a Distance*. San Francisco: Wiley.
- Levit, A. (2009). *Starting a New Business While Still Employed*. (Dow Jones & Company Inc) Retrieved April 14, 2013, from The Wall Street Journal: <http://online.wsj.com/article/SB125098393224651701.html>

- Lewis, K. N. (2009, January 19). *Scribd*. Retrieved April 28, 2013, from Evaluating Web Site Quality: A Benchmarking Approach: <http://www.scribd.com/doc/10934868/Evaluating-Web-Site-Quality-A-Benchmarking-Approach>
- Liive, R. (2013). *Kogemuslugu: Tarkvara arendus ja testimine scrum metoodika alusel*. Tallinn: Äripäev AS.
- Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic Inquiry*. Thousand Oaks, California: SAGE Publications.
- Lipnack, J., & Stamps, J. (2000). *Virtual teams: People Working Accross Boundaries with Technology*. New York: Wiley.
- Luz, M., Gazineu, D., & Teófilo, M. (2009). Challenges on Adopting Scrum for Distributed Teams in Home Office Environments. *World Academy of Science, Engineering and Technology* (35).
- Magretta, J. (2002). Harvard Business Review - Why Business Models Matter. <http://info.psu.edu.sa/psu/fnm/asalleh/MargarettaWhyBusModelMatter.pdf> (R0205F).
- Malka, H. (2011). *Startups: What are best ways in which startups can manage employees (Software Developers) working from home?* (Quora.com) Retrieved April 28, 2013, from Quora.com: <http://www.quora.com/Startups/What-are-best-ways-in-which-startups-can-manage-employees-Software-Developers-working-from-home>
- Maurya, A. (2010). *Running Lean* (Vol. First Edition). San Fransisco: Ash Maurya.
- Maynard, T. D., Mathieu, J. E., Rapp, T. L., & Gilson, L. L. (2012). Something(s) old and something(s) new: Modeling drivers of global virtual team effectiveness. *Journal of Organizational Behavior* (33).
- Metricsmonk. (2013). *Metricsmonk: Our Bridge*. Retrieved 04 22, 2013, from Metricsmonk: <http://metricsmonk.com/apps/facebook/205121939508558-our-bridge?from=appstats>
- Metricsmonk*. (s.a.). Retrieved 04 22, 2013, from Our Bridge: <http://metricsmonk.com/apps/facebook/205121939508558-our-bridge?from=appstats>
- Meyer, E. A. (2000). *Cascading Style Sheets: The Definitive Guide*. Sebastopol, California: O'Reilly Media, Inc.

Mountain Goat Software. (2013, 4 23). *Scrum Task Board Training*. Retrieved 4 29, 2013, from Mountain Goat Software: <http://www.mountaingoatsoftware.com/scrum/task-boards>

Musciano, C., & Kennedy, B. (2000). *HTML & XHTML: The Definitive Guide*. Sebastopol, California: O'Reilly Media, Inc.

Narayanaswamy, P. (2012, December 5). *Is Your Scrum Master a Servant Leader?* Retrieved from Centare: <http://www.centare.com/is-your-scrum-master-a-servant-leader/>

OKBridge. (s.a.). Retrieved 04 20, 2013, from <http://www.okbridge.com/new-signup/deployed/whyokbridge.php>

Online Clubs: Bridge Base Online. (s.a.). Retrieved 04 20, 2013, from Great Bridge Links: <http://www.greatbridgelinks.com/gblPLAY/BridgeBaseOnline.html>

Our.com. (s.a.). Retrieved 04 22, 2013, from <http://www.our.com/games/>

Oxford University Press. (2013, 04 01). *Apple Dictionary: New Oxford American Dictionary*. Los Angeles, California, USA.

Paasivaara, M., Ornäs, N. H., Hynninen, P., Lassenius, C., Niinimäki, T., & Piri, A. (2010). *Practical Guide To Managing Distributed Software Development Projects*. Espoo: Helsinki University of Technology.

Paul, R., & Elder, L. (2007). *The Art of Socratic Questioning*. Tomales: The Foundation for Critical Thinking.

Pavlenko, T. (2012). *Master Thesis: Applying Agile Methodologies to Design and Programming*. (D. Lamas, Ed.) Tallinn: Tallinn University Institute of Informatics.

PC Magazine. (s.a.). *prototyping Definition from PC Magazine Encyclopedia*. Retrieved May 01, 2013, from PC Magazine: <http://www.pcmag.com/encyclopedia/term/49886/prototyping>

People behind Bridge Base. (n.d.). Retrieved 04 17, 2013, from http://webutil.bridgebase.com/v2/news_fetch.php?id=560

People behind Bridge Base. (2011). Retrieved 04 17, 2013, from http://webutil.bridgebase.com/v2/news_fetch.php?id=560

Pichler, R. (2010). *Agile Product Management with Scrum: Creating Products that Customers Love*. Massachusetts: Addison-Wesley.

Picker, R. (2010, September). *The Razors-and-Blades Myth(s)*. Retrieved March 10, 2013, from THE UNIVERSITY OF CHICAGO: <http://www.law.uchicago.edu/files/file/532-rcp-razors.pdf>

Play 2.1 documentation . (s.a.). Retrieved May 10, 2013, from Play framework: <http://www.playframework.com/documentation/2.1.1/Home>

Play Bridge Online – The Best Places. (s.a.). *Learnin Bridge*. Retrieved 04 17, 2013, from Play Bridge Online – The Best Places: <http://learningbridgespot.com/category/play-bridge-online>

Powell, A., Piccoli, G., & Ives, B. (2004). Virtual teams: a review of current literature and directions for future research. *The DATA BASE for Advances in Information Systems* , 35 (1).

Prettyman, G. (1999). *Advertising, Utopia, and Commercial Idealism: The Case of King Gillette*. Cambridge: Cambridge University Press.

Question Pro LLC. (2013). Retrieved March 31, 2013, from Question Pro: <http://www.questionpro.com>

Quora, What is good (n.d.). Retrieved 11 30, 2012, from What is good and what is bad about playing online Bridge at BridgeBase Online?: <http://www.quora.com/Bridge-card-game/What-is-good-and-what-is-bad-about-playing-online-Bridge-at-BridgeBase-Online>

Quora, What is the... (2010). Retrieved 11 30, 2012, from What is the best place to play bridge online?: <http://www.quora.com/Bridge-card-game/What-is-the-best-place-to-play-bridge-online>

Ries, E. (s.a.). *Lean Startup for Final Post*. Retrieved 03 09, 2013, from Lean Enterprise Institute: http://www.lean.org/downloads/lean_startup_for_final_post.pdf

Ries, E. (2008). *Startup lessons learned*. Retrieved 03 09, 2013, <http://www.startuplessonslearned.com/2008/09/lean-startup.html>

Ries, E. (09.03.2013). *The Lean Startup*. Retrieved 03 22, 2013, from Startup Lessons Learned: <http://www.startuplessonslearned.com>

Ries, E. (2008, 09 08). *The Lean Startup*. Retrieved 03 22, 2013, from Startup Lessons Learned: <http://www.startuplessonslearned.com/2008/09/lean-startup.html>

Ries, E. (2011). *The Lean Startup*. New York: Crown Business.

Ruben Anlacan, J. (s.a.). *Starting a Business While Still Employed*. (BusinessCoach Inc) Retrieved April 14, 2013, from <http://www.businesscoachphil.com/starting-a-business-while-still-employed>

Rubin, K. S. (2012). *Essential Scrum. A Practical Guide to the Most Popular Agile Process*. Boston: Addison-Wesley.

(s.a.). Retrieved 04 22, 2013, from Our.com: <http://www.our.com/games/>

Schmidt, S. (2010, March 22). *@codemonkeyism Playing with Play Framework for Java »*; *@codemonkeyism*. Retrieved April 30, 2013, from *@codemonkeyism the blog for developers*: <http://codemonkeyism.com/playing-play-framework-java/>

Schmitt, J. (2011). *Startups: What are best ways in which startups can manage employees (Software Developers) working from home?* (Quora.com) Retrieved April 28, 2013, from Quora.com: <http://www.quora.com/Startups/What-are-best-ways-in-which-startups-can-manage-employees-Software-Developers-working-from-home>

ScrumSense CC. (2009, May 16). *Towards a Useful Set of Agile Metrics*. Retrieved March 31, 2013, from ScrumSense CC: <http://www.scrumsense.com/wp-content/uploads/2009/07/Towards-Agile-Metrics1.pdf>

Siebrat, F., Hoegl, M., & Holger, E. (2009). *How to Manage Virtual Teams*. Retrieved April 7, 2013, from Massachusetts Institute of Technology: <http://sloanreview.mit.edu/article/how-to-manage-virtual-teams/>

Sims, C., & Johnson, H. L. (2011). *Elements of Scrum*. Foster City: Dymaxicon.

Sims, C., & Johnson, H. L. (2012). *SCRUM: a Breathtakingly Brief and Agile Introduction (Kindle edition)*. Seattle: Dymaxicon.

Socialbakers. (2013, March 12). *Social Media Marketing, Statistics & Monitoring Tools | Socialbakers*. Retrieved March 12, 2013, from United States Facebook Statistics by Countries | Socialbakers: <http://www.socialbakers.com/facebook-statistics/united-states>

Socialbakers. (2013, March 12). *Social Media Marketing, Statistics & Monitoring Tools | Socialbakers*. Retrieved March 12, 2013, from Turkey Facebook Statistics by Countries | Socialbakers: <http://www.socialbakers.com/facebook-statistics/turkey>

Socialbakers. (2013, March 12). *Social Media Marketing, Statistics & Monitoring Tools | Socialbakers*. Retrieved March 12, 2013, from Poland Facebook Statistics by Countries | Socialbakers: <http://www.socialbakers.com/facebook-statistics/poland>

Spradley, J. P. (1980). *Participant observation*. Danville, California: Holt Rinehart & Winston.

Statistic Brain. (2012, December 11). *Statistic Brain*. Retrieved March 12, 2013, from Social Networking Statistics | Statistic Brain: <http://www.statisticbrain.com/social-networking-statistics/>

Sutherland, J. (2013, 02 19). *What is Scrum?* Retrieved 04 29, 2013, from Agile Software Project: <http://www.agilesoftwareproject.com/tag/jeff-sutherland/>

Sutherland, J., Viktorov, A., Blount, J., & Puntikov, N. (2007). *Distributed Scrum: Agile Project Management with Outsourced Development Teams*. Retrieved April 28, 2013, from Jeff Sutherland: <http://jeffsutherland.com/SutherlandDistributedScrumHICCS2007.pdf>

Swan Games. (s.a.). Retrieved 04 20, 2013, from <http://www.swangames.com/tourney/leaders.php?stopct=100>

Takeuchi, H., & Nonaka, I. (1986). *New Product Development Game*. Harvard: Harvard Business Review.

Tournaments. (s.a.). Retrieved 04 20, 2013, from Swan Games: <http://www.swangames.com/tourney/leaders.php?stopct=100>

Truex, L. (2009). *The Work-at-Home Success Bible: A Complete Guide for Women: Start Your Own Business; Balance Work and Home Life; Develop Telecommuting Strategies*. Avon: Adams Media.

Turner, R. (2007, 4 1). *Toward Agile Systems Engineering Processes*. Retrieved 5 6, 2013, from CrossTalk: <http://cross5talk2.squarespace.com/storage/issue-archives/2007/200704/200704-Turner.pdf>

UserTesting.com. (2012, October 23). *Prototyping Tools and Testing* | *UserTesting.com*. Retrieved May 01, 2013, from UserTesting.com: <http://www.usertesting.com/blog/2012/10/23/prototyping-tools-and-testing/>

VersionOne. (2013, 03 26). *7th Annual State of Agile Development Survey*. Retrieved from VersionOne LLC: <http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf>

W3C. (2011, April 5). *HTML5 differences from HTML4*. Retrieved May 1, 2013, from World Wide Web Consortium (W3C): <http://www.w3.org/TR/2011/WD-html5-diff-20110405>

Waterhouse, T. (2011, September 14). *The Guide To CSS Animation: Principles and Examples* | *Smashing Coding*. Retrieved May 1, 2013, from Smashing Magazine — For Professional Web Designers and Developers: <http://coding.smashingmagazine.com/2011/09/14/the-guide-to-css-animation-principles-and-examples/>

Welton, D. N. (2011, April 13). *Programming Language Popularity*. Retrieved May 1, 2013, from Programming Language Popularity: <http://www.langpop.com>

West, D., & Grant, T. (2010, January 20). *Agile Development: Mainstream Adoption Has Changed Agility*. Retrieved from http://pms2012.programmedevelopment.com/public/uploads/files/forrester_agile_development_mainstream_adoption_has_changed_agility.pdf

What is the best place to play bridge online? (2010). Retrieved 11 30, 2012, from Quora: <http://www.quora.com/Bridge-card-game/What-is-the-best-place-to-play-bridge-online>

Whitten, J. L., & Bentley, L. D. (2007). *System Analysis and Design Methods*. Boston: McGraw-Hill Irwin.

Why OKBridge? (s.a.). Retrieved 04 20, 2013, from OKBridge: <http://www.okbridge.com/new-signup/deployed/whyokbridge.php>

Widman, J., Hua, S. Y., & Ross, S. C. (2010). *Applying Lean Principles in software development process – a case study*. Retrieved 03 22, 2013, from International Association for Computer Information Systems: http://iacis.org/iis/2010/635-639_LV2010_1497.pdf

- Widman, J., Hua, S. Y., & Ross, S. C. (2010). *APPLYING LEAN PRINCIPLES IN SOFTWARE DEVELOPMENT PROCESS – A CASE STUDY* (J Widman, SY Hua, SC Ross - Issues in Information Systems, XI, 2010 - iacis.org ed.).
- Wight, E. (2012). *Live Q&A: How to start a business while you're still employed*. (The Guardian) Retrieved April 14, 2013, from The Guardian: <http://www.guardian.co.uk/small-business-network/2012/oct/05/starting-up-business-employment>
- Wikipedia. (2013, 03 25). *Agile software development*. Retrieved 04 01, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Agile_software_development
- Wikipedia. (2013, 3 1). *Burn down chart*. Retrieved 4 20, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Burn_down_chart
- Wikipedia. (2013). *Socratic questioning*. Retrieved 05 05, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Socratic_questioning
- Williams, L. (2007). *A Survey of Agile Development Methodologies*. Retrieved 4 30, 2013, from Realsearchgroup: <http://agile.csc.ncsu.edu/SEMaterials/AgileMethods.pdf>
- Wirtz, B. W. (2011). *Business Model Management: Design – Instruments – Success Factors*. Retrieved March 10, 2013, from http://www.businessmodelcommunity.com/fs/Root/8v33g-110314_Extract_BMM.pdf
- World Bridge Federation. (2009). *Geographical zones*. Retrieved 04 24, 2013, from World Bridge Federation: <http://www.worldbridge.org/geographic-zones.aspx>
- Yau, A., & Murphy, C. (2013). *Is a Rigorous Agile Methodology the Best Development Strategy for Small Scale Tech Startups?*
- York, M. (2013). *How to Start Up While Still Employed*. (InterActiveCorp) Retrieved April 14, 2013, from About.com: <http://entrepreneurs.about.com/od/becominganentrepreneur/a/HowtoStartUpWhileStillEmployed.htm>

Zwilling, M. (2013, January 19). *6 Key Characteristics of Innovative Startups Teams*. (Martin Zwilling) Retrieved April 17, 2013, from Startup Professionals Musings:
<http://blog.startupprofessionals.com/2013/01/6-key-characteristics-of-innovative.html>

APPENDIXES