

UNIVERSITY OF TARTU
Institute of Computer Science
Information Technology Curriculum

Magnus Meldre

**Using Impact Mapping in Agile Software
Product Management**

Bachelor's Thesis (6 ECTS)

Supervisor: Helle Hein, PhD

Tartu 2016

Using Impact Mapping in Agile Software Product Management

Abstract:

The author of this thesis has over 10 years of experience working with software products and is currently working as product owner (PO) and agile coach in a technology company. During the years industry trends have shifted towards Lean and Agile practices. The role of the PO defined by Scrum framework has emerged but the responsibilities for the role have not been unambiguously defined. To succeed with products and businesses from various backgrounds PO needs to vastly adapt and master different techniques. In this work, an inquiry on a strategic planning technique called Impact Mapping is performed and a unique combination of initial and follow-up materials is introduced to the reader. The technique is then analyzed through the problems that a PO faces in 2016, and conclusions are made on its compliance with the role and environment constraints.

Keywords:

Impact Mapping, Agile Product Management, Product Discovery, Product Delivery, Scrum, Product Owner

CERCS: T120 Systems engineering, computer technology

Mõju vastendamise kasutamine väledas tarkvara tootejuhtimises

Lühikokkuvõte:

Autor töötab tehnoloogiafirmas tooteomaniku ning väletreenerina ning omab tarkvaratoodetega üle 10 aasta kogemust. Aastate jooksul on tööstusharu trendid nihkunud väleda ja timmitud praktikate kasuks. Scrum raamistikus on defineeritud tooteomaniku mõiste, kuid selle rolli vastutusala ei ole üheselt defineeritud. Selleks, et olla erineva taustaga toodete ja äridega edukas, peab selle rolli täitja kiirelt kohanduma ja valdama erinevaid tehnikaid. Käesolevas töös uuritakse strateegilise planeerimise tehnikat, mida nimetatakse mõju vastendamiseks ning lugejale esitletakse ainulaadne kombinatsioon esialgsest ning hilisemast materjalist. Seejärel analüüsitakse mõju vastendamise tehnikat lähtuvalt tooteomaniku probleemidest aastal 2016 ning tehakse järeldusi mõju vastendamise rakendamise kohta sõltuvalt keskkonnapiirangutest.

Võtmesõnad:

Mõju vastendamine, väle tarkvara tootejuhtimine, toote avastamine, toote väljastus, Scrum, tooteomanik

CERCS: T120 Süsteemitehnoloogia, arvutitehnoloogia

Contents

Introduction	6
Goals	7
Outline	7
1. Different Aspects of Impact Mapping	8
1.1 About Impact Mapping	8
1.2 Overview and Usage of the Technique	9
1.3 The Goal	10
1.3.1 Defining Measurements	11
1.4 The Actors	12
1.5 The Impacts	13
1.6 The Deliverables	14
2. Origins of Impact Mapping	15
2.1 Goal Card Method	15
2.2 InUse Effect Mapping	16
3. Software Product Management in Agile Environment.....	18
3.1 Environment Constraints	19
3.1.1 Product Environment	19
3.1.2 Agile Frameworks	22
3.2 Responsibilities and Expectations	24
3.3 Challenges	26
4. Impact Mapping Benefits to the Product Owner	28
4.1 Learning Perspective	28
4.1.1 Validating Assumptions	29
4.1.2 A/B Testing and Cohort Analysis	30

4.2	Introduction to Product Discovery and Product Delivery	31
4.3	Product Discovery Perspective.....	32
4.3.1	Market and Customers	32
4.3.2	MVP and Lean UX.....	33
4.3.3	Disruptive Innovation.....	35
4.3.4	Stakeholders' Perspective	35
4.4	Product Delivery Perspective	36
4.4.1	Managing and Grooming Product Backlog.....	37
4.4.2	User Stories	38
	Conclusions	40
	Bibliography	41
	Appendices	44
I.	Abbreviations	44
II.	Licence	45

Introduction

Software product management is one of the core business functions in software companies that throughout the entire product life-cycle deals with planning, forecasting, development and marketing in order to maximize business revenue.

In today's world the market is not standing still until a pre-planned project in a full scope is delivered. In such an environment, Agile and Lean practices have emerged and gained popularity.

Teams are expected to learn from customer feedback faster, product discovery is an ongoing process where the focus is set on a customer and pre-defined requirements have a lesser role.

Communication and assumption sharing are more important than ever and some frameworks like Scrum expect product delivery people to work together with the development team in an ongoing basis. Since teams need to have a shared product vision, Scrum introduces the role of Product Owner (PO) - the person who is responsible for maximizing the value of the product and the work of the Development Team.

The expectations on a PO are high. However, the role itself in the context of the product from discovery to delivery has not been unambiguously defined.

The present thesis will look into the role of the PO and analyze the solutions to the problems that this role is facing through the technique called Impact Mapping. It is a strategic planning technique which visualizes the scope as solution ideas for the problems. The technique prevents organizations from getting lost while building products or delivering projects. Furthermore, it helps people from multiple domains to hold meaningful planning and prioritization sessions.

One of the challenges of the PO role is to know and master many different techniques in order to choose the best ones for a specific situation or environment, the constraints and goals in mind. Impact Mapping is just one of such techniques.

Goals

The present thesis carries the following goals:

1. To conduct an extensive inquiry on the materials published about Impact Mapping to fully understand the technique, its strengths and weaknesses. The subject has not been researched much and within this thesis, a unique combination of initial and follow-up material by Gojko Adzic, the author of the technique, and his peers is presented. When the Impact Mapping book was first published, the ideas were raw and the author wanted to get improvement ideas and feedback from practitioners.
2. To introduce the Impact Mapping technique to the reader; to discuss which environments it may be used in, and provide enhanced primary solutions to the product manager problems, solvable by using Impact Mapping.
3. To bring out environment constraints; to analyze problems and day-to-day challenges related to product management in an agile software development environment.
4. To give an overview of the problems of the domain where PO operates by describing industry trends; analyze how an already implemented Impact Map would be beneficial for learning to deal with those problems.

Outline

The work is organized as follows:

1. The first chapter introduces and analyzes the impact mapping technique, the suggestions on its usage and gives an overview of the impact map structure.
2. The second chapter presents the ideas from which Impact Mapping has emerged and highlights why it is mistaken for Effect Mapping.
3. The third chapter introduces the agile software product management core principles and the term Product Owner. It also analyses and gives an overview of expectations, challenges and constraints for the PO role based on the product environment and the agile framework preference, where Scrum is discussed in more detail.
4. The fourth chapter introduces and analyzes the benefits of the Impact Mapping technique from the point of view of a PO. The product discovery, product delivery and learning perspectives are observed.

Chapter 1

1. Different Aspects of Impact Mapping

1.1 About Impact Mapping

Impact Mapping is a strategic planning technique that complies with agile software delivery and lean software development environment. It visualizes the scope as solution ideas for the problems and reveals underlying assumptions by connecting a deliverable to the business goal. The intention is to bridge the communication gap between development teams and stakeholders, by providing a big picture view for all the participants involved in continuously building a software product, and to help align activities with overall business objectives [1, 2].

The Impact Map, in an essence a visualized product backlog for multiple product milestones, is presented as a structured mind-map which is created and updated collaboratively. Therefore, it supports adaptive planning while facilitating learning through iterative delivery and experiments. The comparison of an iterative and an incremental delivery is presented in Figure 1, which visualizes how the big picture is revealed only at the end. The mind-map is grown during a discussion by considering the following four aspects: business goal, actors who can help to achieve a goal, impacts to describe the behavior change and high-level deliverables which may or may not be software deliverables [3].



Figure 1. Incremental versus iterative delivery or planning. ¹

The purpose of the constantly shrinking, growing and re-prioritized mind-map is to help manage a product or project road-map in a rapidly changing environment in which most IT-projects operate. Impact map based road-map decisions are made for development

¹ <https://gojko.net/2007/12/04/waterfall-trap/>

iterations, whilst multiple alternatives are available to reach a desired business goals set as product milestone targets. Preferably the shortest path to the goal is desired at the given time, and eventually the whole map is not to be implemented, because the business goal are reached sooner [1, 4].

Impact Mapping, as introduced in 2012 by Gojko Adzic, is a variant the effect mapping technique combined with the high impact learning method, put into a modern agile or lean software development context [1]. The origins are described further in Chapter 2.

1.2 Overview and Usage of the Technique

An impact map is a visualization of the scope in connection with measurable business objective(s). The visualization helps to clearly show the impacts that technical or organizational deliverables are supposed to produce from a business goal perspective [3]. The key purpose of impact mapping sessions, led by a PO or a consultant acting as a PO, is to facilitate discussion between technical and business people and align everyone with underlying assumptions to think towards a common shared goal from different perspectives each participant represents. ²

By methodology, creating an initial impact map consists of two sessions [1]:

1. Preparation: discovering goals and determining measurements;
2. Creating a map: drawing a skeleton, finding alternatives, identifying key priorities and defining experiments to test assumptions.

In the follow-up sessions, participants go through what has been learned thus far and make road-map decisions for the upcoming iterations until the product milestone is proven a success.

Technically, an impact map is a mind-map that comprises at the very least the following groups of hierarchical nodes as presented in Figure 2, which answer the following questions [1, 3]:

1. Goal: “Why are we doing this?”
2. Actors: “Who can produce the desired effect? Who can obstruct it?”
3. Impacts: “How should our actors’ behavior change? How can they help us to achieve the goal? How can they obstruct or prevent us from succeeding?”

² <https://groups.google.com/forum/#!topic/impact-mapping/tWJ8vDfsXeA>

4. Deliverables: “What can we do, as an organization or a delivery team, to support the required impacts?”

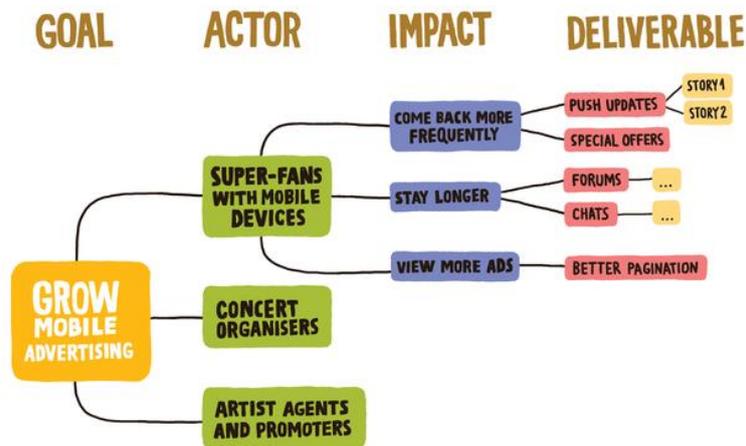


Figure 2. An example of an impact map.³

Hereby the nodes of an impact map are introduced and analyzed in more detail.

1.3 The Goal

The center node of an impact map coincides with the ultimate business goal to be achieved, answering the most important question: Why do we do this? [3]

Since all the following branches of an impact map are assumptions towards achieving the desired effect, the target here is to provide a context and general understanding of the shared goal. Otherwise, the assumptions would be worthless [4].

The goal is defined either as a problem to be solved or as a desired business benefit that a product or project is expected to produce. The goal should not hint to a solution [1].

A good goal is S.M.A.R.T.⁴, that is: Specific, Measurable, Assignable, Realistic and Time-related. This will prove useful later on when scope, cost or time schedule changes. For commercial projects, the goal should preferably be linked to money [1].

³ <https://www.impactmapping.org/drawing.html>

⁴ https://en.wikipedia.org/wiki/SMART_criteria

1.3.1 Defining Measurements

Once the goal(s) have been set, product milestone measurements need to be defined. A product milestone is a medium-term target which is pursued through iterative planning and delivery [1].

Impact mapping method advocates using a simplified version of the metrics defined by Tom Gilb in his book [5]. A measurement in impact mapping consists of the following five items [1, 5]:

1. Scale (of measure) – an attribute that defines the units for measurement.
2. Meter – a method or location for measuring and testing the Scale attribute.
3. Benchmark – the current level as defined by Adzic, most similar to Gilbs' attribute Record which corresponds to the best-known numeric value result in the past but not exactly the same.
4. Constraint – the minimum acceptable value, the break-even point for an investment as defined by Adzic, could be either Fail or Survival attribute in Gilbs' book.
5. Target – the desired value as defined by Adzic, most similar to Gilbs' attribute Wish which corresponds to a future dream level and is too big to even plan or promise.

The method proposes consultants to experiment in naming the measurement points, whatever is understandable to the audience, given the audience's background.

Hereby we construct an example of the usage of the above introduced metrics' attributes by thinking of an imaginary startup company's next Product Milestone. Table 1 presents a simultaneous combination of the following desired effects: three times more active users while scaling up user acquisition budget profitably, 100% increase in Gross Burn allowed.

The goals and metrics for this example have been defined based on the ideas from Andreessen Horowitz' blog post *16 Startup Metrics*. Monthly gross burn rate corresponds to the measure on how much capital a startup company is willing to spend each month. Paid CAC is calculated as show in Table 1 and shows investors whether a company is capable of optimizing its marketing activities, scaling up at the same time [6].

Table 1. An example of the measurements for the upcoming product milestone.

	More Users in 6 Months	Monthly Gross Burn Rate	Paid Customer Acquisition Cost (CAC)
Scale	The number of monthly active users	Hosting cost + marketing cost + salaries	Total acquisition cost / new customers acquired through paid marketing
Meter	Users database	Financial accounts	Financial accounts
Benchmark	350 000	€50,000	€5
Constraint	800 000	€100,000	€10
Target	1 050 000	€50,000	€8

1.4 The Actors

The following step is to define the main actors by answering the question “Who can produce the desired effect?” [3]. The target here is to find the people either in power of helping to achieve the goal, or who can obstruct in reaching it. Actors may be stakeholders, users of a system, specific user segments, specific user personas, people from other departments etc. Impact mapping method suggests observing three types of actors that can influence the outcome:

1. Primary actors who are the users of the product, preferably grouped as user segments or personas.
2. Secondary actors who are related to the service or the product which is provided.
3. Offstage actors who represent internal or external decision makers. These are the people or organizations that are in position of power who may work towards or against the business goal.

All actors expect value to be delivered to them. Each actor has separate needs, goals and preferences to be studied and fulfilled in order for a product to be successful. Impact Mapping advises serving the needs of the most important actors first, in order to get most of the value as early as possible.

1.5 The Impacts

The second branch level of an impact map corresponds to the impacts, which may either be positive or negative. In general, the impacts coincide with the job to get done to produce a desired effect. The actors are set to the perspective of the business goal by asking how their behavior is expected to change in terms of the goal. Since the behavior may change for the worse it is likely that some risks are to be revealed.

As presented in Figure 3, an impact map consists of assumptions. The first three levels of nodes are related to business or market assumptions. Changing those assumptions without eliciting some change in the actions is generally out of direct control, but in the sphere of influence [7]. The main target of impact mapping is to make the assumptions visible for all the participants, whereas most commonly the assumptions are only defined in vision documents. In a worse case, assumptions are not even present in writing.

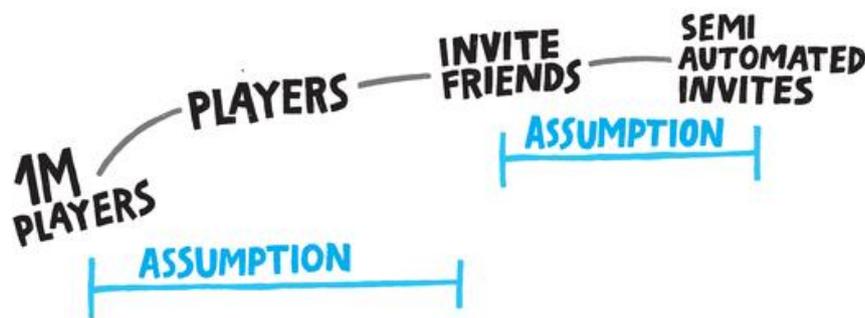


Figure 3. Impact map is made up of assumptions.⁵

In Figure 3, the first assumption is that in order to reach one million players target, players would invite their friends. The other assumption that may be used as an experiment is that semi-automated invites will show us that players will invite their friends. If that happens, the process could be fully automated, but there is no reason to invest too much into that assumption until proven to be true [8].

When the impact map is ready to be prioritized, it will be first done against the impacts and then choosing the best deliverables.

⁵ <https://www.impactmapping.org/delivering.html>

1.6 The Deliverables

The furthest set of nodes corresponds to the deliverables that are mapped to business objectives. The deliverables may or may not be software features, because some could be organizational activities as well [1, 3].

In general, deliverables define the project or product milestone scope as a list of possible actions with a context that is in the sphere of control. Deliverables are based on assumptions and support the organization or delivery team in achieving the desired impacts. Since all the deliverables are options, it is not important to implement all of those. Instead, it is only necessary to find the most valuable path through the map to the desired goal.

Impact mapping method suggests dividing the deliverables into two groups:

1. Mandatory high-level deliverables which correspond to the actions.
2. Optional low-level deliverables that are the sub-tasks for the actions.

When deliverables are hard to slice, impact mapping suggests using User Story Mapping [1]. It is a technique invented by Jeff Patton to organize hierarchical backlogs by setting a user to the perspective of how they use software or to a perspective of a customer journey. A story map is a grid where user activities, which are most likely to happen, are mapped to the user stories or releases that are in the sphere of control. The technique also allows finding and discussing steps that might be skipped or made semi-automatic to get to the initial release sooner [1, 7, 9].

Impact Mapping advises organizations to set up a learning budget and find the simplest ways of experimenting to test assumptions [1]. Splitting learning from earning is described more detail in Gojko Adzic's and David Evans' book [10] as a time-boxed set of learning user stories, in order to prevent a long upfront analysis and keep teams from pretending to be good at estimating learning, while in reality, such pretending delivers no apparent value. In addition, teams should themselves decide the output for learning [1, 10].

Chapter 2

2. Origins of Impact Mapping

Impact Mapping is based on a Swedish interaction design process called InUse Effect Mapping and Management [1, 11] – about setting goals and strategies [12].

2.1 Goal Card Method

The principal idea of linking business goals to usage goals during the product design process was first presented by Balic et al [13]. In order to bridge a gap between business objectives and the actual design of the product a three-level Goal Card method was introduced. To formulate a hypothesis to be evaluated during the design process, one must cover [13]:

1. the business goal (the “Why?” question);
2. the means to achieve the business goal as usage goals (the “How?” question);
3. the steps within the product to ensure the business goal (the “What?” question).

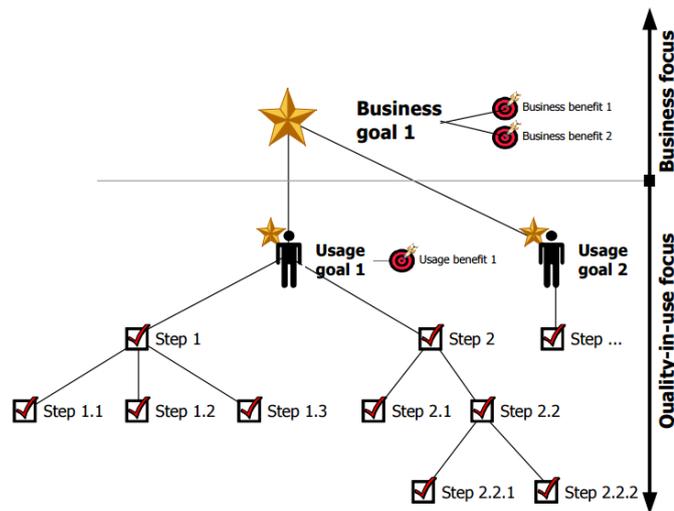


Figure 4. A Goal Card [13].

The idea of the method was to visualize hypothesis as a mind map towards measurable business goals — which are described as changes in the business — and measurable usage goals — which in turn describe the benefit for the users or user groups. Further nodes within a map correspond to steps towards achieving business goals. The bottom-up traceability within the associative network provides an overview that is easy to understand

for all team members. The Goal Card resembles highly an Impact Map as presented in Figure 4.

The intention of the Goal Card method was to use it in traditional industrial solutions' projects' environments in order not to exceed time estimates and budgets. The latter, by hypothesis, in turn happen because of misunderstanding of user requirements, causing a scope change.

The other hypothesis was related to visualizing the scope during the discussions with stakeholders in order to prioritize features to be implemented after the scope change. In today's agile software development environment, scope change is no longer seen as a negative effect and a constant feature prioritization, as described in Chapter 3 and analyzed in Chapter 4, is one of the key responsibilities of a PO.

2.2 InUse Effect Mapping

Over some years, the concept of the Goal Card method was detailed into the Impact Management method, seen as an addition to any project management method because many IT projects focus very weakly on benefit and effect once the development project has commenced [11]. The Impact Management method was first published in 2004 in the book *Effektstyrning av IT* (in Swedish), later translated into English and published as *Effect Managing IT* [14].

The structure of the effect management mind map highly resembles the one used in impact mapping as seen in Figure 5, but the way the maps are used is different. In addition, Impact Mapping works in an incremental way whereas Impact Management is designed for iterative way of working [1].

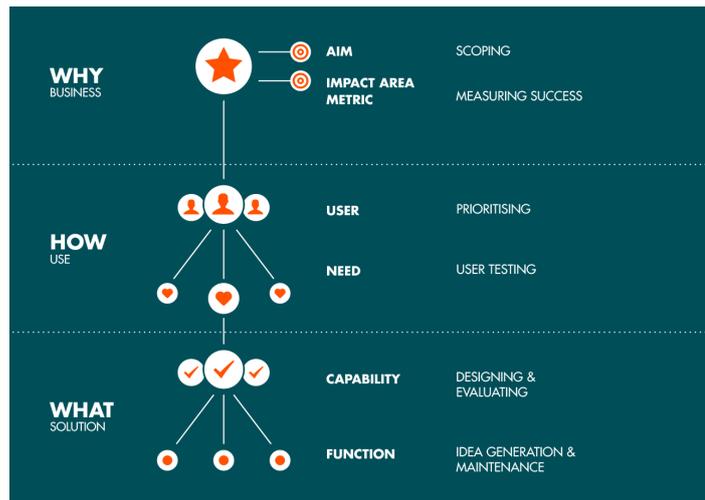


Figure 5. Effect mapping can easily be confused with Impact Mapping.⁶

The effect mapping method is primarily meant to be used for product design and user experience (UX) design, where the most important user needs must be satisfied early and details are left for later [14]. The agile development and delivery aspect was described by Gojko Adzic in the 2012 [1] and re-named to Impact Mapping in order to sort out the confusion in a fact that the practitioners of effect mapping claimed the method was wrongly used [1, 2, 13].

The product discovery, product design and (Lean) UX design decisions, including the interaction design aspect of PO work, are described and analyzed in Chapter 4.

⁶ <http://inuseexperience.com/img/needs.png>

Chapter 3

3. Software Product Management in Agile Environment

Software product management is one of the core business functions in software companies that throughout the entire product life-cycle deals with planning, forecasting, development and marketing in order to maximize business revenue.

To achieve success, a product has to be delivered with unique benefits compared to competitors on the market and with superior value to the customer. From the business perspective, in order to maximize the return of investment (ROI) from a business opportunity, reduced time to market is usually required.

In today's world, where the market is not standing still until we deliver a pre-planned project in a full scope, the only constants remaining are change and unpredictability. A research⁷ published in 2008 by BCS concluded that only one in eight IT projects can be considered successful. Commercial organizations across the European Union lost 142 billion EUR on failed IT projects in 2004 alone, mostly because of poor alignment with business objectives or business strategies becoming obsolete during delivery.

In a rapidly changing environment, Agile and Lean practices have emerged and increased in popularity. Development teams are expected to ship features more often in order to learn from the customer feedback faster. Product discovery is an ongoing process where requirements are not expected to be fully pre-defined any more. Due to that, communication and sharing assumptions are more important than ever, and some frameworks like Scrum expect product people to work together with the development team on an ongoing basis [15].

With those trends, the focus has shifted from process and department boundaries to teams with mixed skill-sets, capable of doing every task that the team commits to. Since teams need to have a shared product vision, Scrum introduces the one person role of PO who is responsible for maximizing the value of the product and the work of the Development Team [16].

⁷ <http://www.bcs.org/content/ConWebDoc/19584>

Hereby the discipline of product management is seen in a more narrow perspective and the focus is set on software product management in an agile and lean environment, that is constant improvements to a product in a non-project way.

3.1 Environment Constraints

Since the nature of the PO's day-to-day work tasks and expectations on outcomes change vastly while working with products from various product life cycle states, say early stage or mature, we hereby first observe the constraints set by the environment. The investment horizon and agile framework constraints are discussed in more detail.

3.1.1 Product Environment

A product and its features or components may be seen as an investment, which is usually true from the perspective of the investor or a company building its own product.

The author of this thesis works as a PO and an agile coach in a technology company that works with various products at the same time. On the portfolio level, products may fall into different categories by investment risk and investment horizon as presented in Figure 6.

For a smart technology company such as Google, looking to innovate effectively while continuously growing, all the horizons are important. However, each horizon requires a different set of skills from the employees, who then, in turn, use their wisdom to choose the best frameworks and methodologies that work the best with the environment constraints.

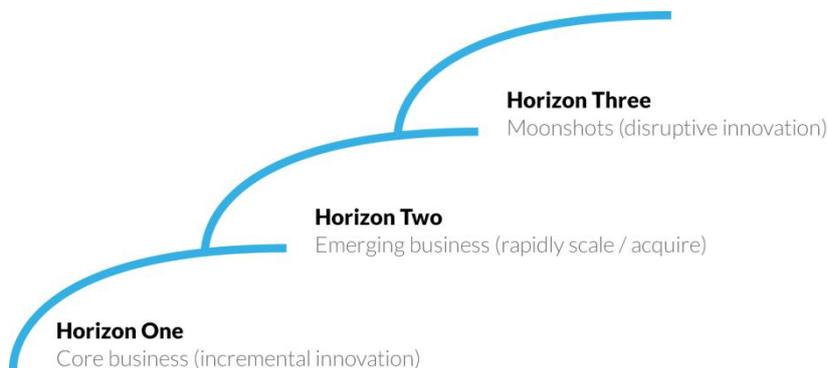


Figure 6. Innovation portfolio horizons.⁸

⁸ <http://digintent.com/crafting-the-ideal-innovation-portfolio/>

The idea of the innovation portfolio for a company is to keep and expand its current business, while maximizing the likelihood of future success [17]. The horizons and industry’s best risk management practices for budget allocation are described in more detail in Table 2.

Table 2. Descriptions and budget allocation shares of an innovation portfolio.

Horizon	Allocation of Budget	Comment
Core business	50% - 70%	A part of a business that has scaled up and provides the main income for the company today. Investment needs to be protected through constant incremental innovation; company’s own employees can manage to do that.
Emerging business	20% - 25%	A part of a business that has not scaled up yet but the concept already has been proven to be a success with the customers. For an established business, this usually means a partnership with another innovative company or a startup which itself is missing the competences it needs.
Moonshots	10% - 25%	Disruptive innovation attempts that depend on the business’s risk aversion. In a venture capital way, usually many small bets at a same time to the future are made while most of them fail, which is acceptable to the business. The competences and required mindset are usually different from today's ways of working.

The PO may be expected to work with products from all following investment horizons:

1. either the ones that require continuous incremental improvements;
2. the ones that have not scaled up yet;
3. the ones that are convinced as disruptive innovation and require extensive Research and Development (R&D).

Working with products that have already been accepted by its market is very much different from disruptive innovation products where the market has a problem to be solved

but solution as a product does not even exist yet. The product discovery aspect of the PO work is described and definition of disruptive innovation is given in Chapter 4.

The latter requires a startup mindset, tools and methodologies since at any time new info may cause a pivot decision to be taken by the PO, which is analyzed in Chapter 4. Usually there are fewer of the company's people involved, the focus is on the Minimum Viable Product (MVP) and scaling up is not an issue, since the product might not even exist yet. Therefore methods related to product discovery are used, for example Design Sprint invented by GV (formerly Google Ventures). It is a five-day process consisting of various intense time boxed activities for answering critical business questions through design, prototyping, and testing ideas with customers [18].

Working with an existing product provides a larger playground for shipping new features, but on the other hand legacy features need to live up to market expectations as well. Most of the techniques, including Impact Mapping, and frameworks, such as Scrum, address the products, teams and decision makers from this investment horizon.

Much to a surprise of the author of this thesis, the aspect of how the expectations differ while working with the above described products has not been widely recognized in the industry today. By the understanding of the author, this is one of the root causes for waste and failed processes.

This above mentioned hypothesis was researched in University of Helsinki by Juuso Hyvönen in his work [19] where the Lean Canvas, the Impact Mapping and the Lego Serious Play techniques were studied and compared based on a case study which was conducted in two Finnish medium-sized software companies.

The key conclusion after a case study was that the Impact Mapping method is not optimal for defining a MVP scope [19]. The author of this thesis believes that to be true, since impact mapping by definition as defined today is meant to be used only for medium-term plans and works best with existing products where incremental improvements are pursued. The problem has also been occasionally discussed in the impact mapping practitioners' Google Groups' [20] forum. The author of this thesis has struggled with the similar problems when practicing the technique on consumer product discovery himself.

The author of this thesis proposes that one possible research idea for the future would be to further investigate Impact mapping compliance with defining the MVP scope. During the case study by Hyvönen many mistakes in goal setting were made as noted later in the

work also [19]. The author of this thesis suspects that the goals were not S.M.A.R.T. enough. The specific hypothesis cannot be verified since no info exists in Hyvönen's work about the actual presentation of the business objective, nor is it important in the context of this thesis. Impact mapping technique based meetings consist of two sessions. A conclusion may be drawn that the input from the first session is crucial for the success of the second meeting with a larger group. Some experience with the technique is required to best facilitate meetings as also concluded in Hyvönen's work.

Therefore, based on the case study mentioned above and based on the experience with impact mapping in general, it can be concluded that impact mapping today does not fully comply with product discovery. In the future, the technique should be further researched and improved. Until then, some other (emerging) techniques, such as GV Design Sprint, should be used instead.

3.1.2 Agile Frameworks

The second set of constraints for the PO work originates from the agile (development) methodologies or frameworks, where the most popular ones in today's industry are Scrum, Kanban, and Scrumban. In recent years some multi-team agile frameworks such as SAFe (Scaled Agile Framework) and LeSS (Large Scale Scrum), both of which have a background in Nokia, have gained popularity in our region, especially in Finland, but those will not be covered in this thesis.

Hereby, based on the experience of the author of this thesis, the most important single-team framework related constraints are described and analyzed in more detail.

Scope limit defines the work to be done within an iteration. In Scrum, the workload is limited by a sprint, which most commonly lasts for 2-3 weeks. The work has to be finished within the iteration and the scope cannot change. In Kanban and Scrumban, the work is limited only by the items that are currently in progress.

Iteration length in Scrum is fixed and planning is required, whereas in Kanban the iterations are not set and work is done continuously without planning sessions. Scrumban is the combination of the two approaches: work is pre-planned, but done continuously until the next planning occurs.

In all the mentioned frameworks the work is assigned to the teams in a centralized way and the team members choose the tasks they start to work on. The frameworks do not

regulate how the work is done until it reaches the backlog and therefore each PO may choose one's techniques and tools oneself. Based on the experience of the author of this thesis, this liberty may be somewhat problematic, particularly for the less skillful PO-s, because the in-debt knowledge of different techniques is usually adequate and the product environment aspect is quite often not taken into account. To overcome this obstacle, successful companies have introduced a dedicated role of an agile coach or an agile leader, but most commonly technology companies still fail to recognize the problem in the first place.

The performance metrics measure the velocity or the work to be done, for example through the burn-down chart, as presented in Figure 7, or through cumulative diagram. However, the metrics suggested by the most popular development frameworks do not focus on the fact that the work should benefit the customer or the user. Instead, the author of this thesis believes that this is mildly assumed to be foreseen by the customer-focused or customer-faced PO.

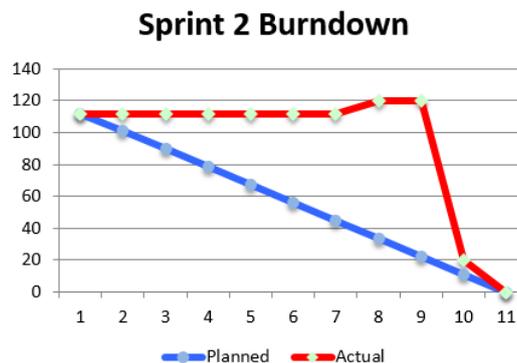


Figure 7. Scrum sprint burn-down chart.⁹

In Figure 7, a Scrum Sprint's burn-down chart is presented and hereby we elaborate on what may be seen from the diagram. On x-axis, the numbers correspond to the working days during the Sprint. In this case it is a classical two-week Sprint. On y-axis, the numbers correspond to the story points, a complexity value defined to each work task by all team members during a discussion. The total number of story points makes up team's velocity. For this team the velocity is a bit over hundred story points. Once the Sprint Planning is complete and the Sprint is started, an average plan may be created as visualized with the blue line. The red line measures the actual effort of closing tasks

⁹ <https://refinem.com/wp-content/uploads/2015/04/Sprint-Burndown-Large-Peak.png>

during the Sprint. The specific Scrum Team has taken new tasks to the Sprint after the Sprint is started, which is not allowed. Most probably the team will discuss what happened during the next Sprint Retrospective and collectively agree on improvements for the upcoming Sprint. However, the team managed to deliver all the items by the end of the Sprint, therefore the Sprint did not fail.

The agile framework to be chosen could be decided by the development team or it may be enforced by corporate procedures. Either way the constraints have to be kept in mind. The same goes with for benefits. From the experience of the author of the thesis, Kanban is usually the best framework to discover bottlenecks in the ongoing process and should be used when teams are new, not yet familiar with each other and common habits have not emerged yet. Later on Scrum is the most beneficial because it builds up the teamwork the best.

Hereby throughout this thesis, the methodology focus is set to Scrum and product management aspect is handled by the term of PO.

3.2 Responsibilities and Expectations

The expectations to the PO by Scrum methodology are [16]:

1. Preparing and continuously grooming the Product Backlog and making it visible and understandable. Different tools may be used for that, starting from the traditional post-its, finishing with some software like JIRA Agile.
2. Prioritizing the Product Backlog items to best achieve medium and long term business goals and missions.
3. Optimizing that the short-term work in hand brings the most value. This also includes planning upcoming Sprints with the Development Team.
4. Making sure that each item in the Product Backlog is understandable. This most often consists of writing user stories and communicating them to the development team.

From the experience of the author of this thesis, the above mentioned list does not fully cover all the responsibilities of a PO, because scrum focuses solely on the product owner, scrum master and development team interaction, events and artefacts. All of the above mentioned items are related to the product backlog, so the sole responsibility of a PO is to generate and maintain the product backlog. From scrum's point of view, the only

important aspect is that the PO is one person, not a committee, and PO is held accountable, but nevertheless may delegate tasks. The other axes of PO communication and successful product are presented in Figure 8.

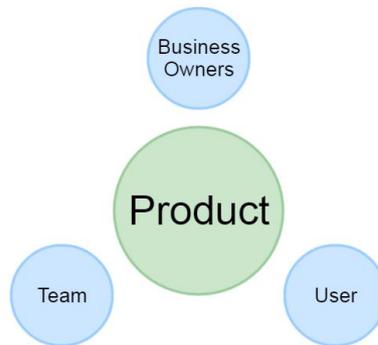


Figure 8. Three axes of the product (ownership).

Typically a product has its place on the market, usually with competitors that want to win the market share. A successful product serves the needs of its users and generates value to the business via serving customers and keeping them happy. In order to design and to provide good UX, the market's needs should be fulfilled at any time, in an ideal world having unique (unfair) advantage(s), and the users' needs should be continually monitored to react to changes. In order to build and keep a product a success, a product manager or a PO needs to understand, monitor and communicate all of those aspects.

Usually the users' perspective is covered with market researches, behavior monitoring, meeting users directly or collecting feedback. In a lean environment, continuous improvements are tackled with experimenting on new features or different versions of shippable features as discussed in more detail in Chapter 4.

In Figure 8, the node Team is not defined only as the self-organizing cross-functional development team as defined by the Scrum framework, but it could refer to multiple teams or highly-specialized independent out of the team resources such as UX designers etc. In addition, during the product discovery phase, when working with high reward and high risk initiatives, the node Team could refer to the specially set up R&D team.

Each product has stakeholders or business owners who are directly or indirectly investing into development. The stakeholders are the ones interested in seeing the targets being reached and budgets not being overrun. One of the key responsibilities of a PO is to make information and ongoing activities transparent to all stakeholders. Usually empirical

business objectives are defined as some Key Performance Indicators (KPI) and stakeholders have the authority to influence the decisions that a PO is taking.

To illustrate the fact that the responsibilities of the PO have not been defined in detail neither by the industry in general, nor by the lightweight Scrum framework, other categorization possibility is presented. According to Roman Pichler [21] the Agile PO responsibilities may also be defined by the depth of the ownership as he recently stated in May 2016. He argues that a PO may be concluded “small” when only working with tactical decisions and “big” when the responsibilities also cover the strategic decisions about the product.

The author of this thesis in general agrees with this division of labor, because it is actually happening quite often, especially for the core business products. On the other hand, the author of this thesis does not fully agree with the presented definition in detail, because the “small” PO then would only be a proxy-PO who could not actually be held accountable for the product or the component, including the tactical decisions, as required by the Scrum framework. The goal of this thesis is not to unambiguously define the role, therefore within this thesis only the “big” PO responsibilities are defined as the role of PO.

Hereby we have covered the high-level list of expectations to the PO. The most important challenges shall be linked to those expectations and be reviewed further in the next section.

3.3 Challenges

Based on longtime experience in the product development industry, the author of this thesis suggests that the key challenge to product management is internal communication with the stakeholders and the teams. The PO should present a shared product vision before the work commences and afterwards, during the experimenting or development, all the participants should be kept in loop. The aim is to bridge the communication gap between decision makers and the people actually doing the work. This could be first done in the planning phase or later on continuously. By solving a communication problem some other future problems like budgeting issues should at least in principle have lesser impact.

When internal communication and product vision problems are solved at planning phase, some meetings should be held together with technical and business people. As such, everyone could have a say and could agree that the ideas generated collectively are theirs

as well and not coming top-down. In order to facilitate a meaningful and productive problem solving ideation meeting, various techniques like Impact Mapping, Requirements Elicitation, and Design Sprint etc. could be used. In general, most of them follow the same similar principle of facilitating discussion in some structured time-boxed way. Within this thesis one of those techniques is observed.

One of the most critical challenges when setting up a Lean organization is the shift in mindset, but also the learning from feedback or (external) communication. The feedback may come from customers, users, internal sources, measured activities etc. In some cases, feedback is not received at all and some new ways to collect info are to be defined firsthand, which in turn might cause an overhead in the amount of data to be processed to an input for the decision. Either way, after meaningful feedback, continuous improvements to the product and to the way of working are to be expected.

The challenges and expectations described in the current and previous section are each analyzed further and put into context of using impact mapping to solve those in Chapter 4.

Chapter 4

4. Impact Mapping Benefits to the Product Owner

The impact mapping technique proposes the method to be used for the following roles and only under following pre-conditions [1, 3]:

1. Strategic planning, if the strategic goals can be set and decision-makers and technical people can participate in the meetings together.
2. Defining quality, if an agreement can be made that actor's behavior change can be considered as proof of quality.
3. Road-map management, if the scope can be planned in an adaptive way and the teams can deliver in an iterative way.

By using impact mapping or creating an impact map, there are various benefits to the PO. Hereby within this thesis the following perspectives are discussed and analyzed: learning, product discovery and product delivery.

4.1 Learning Perspective

The core principle in lean thinking is to maximize customer value (benefit to the customer) and minimize waste (activities that do not create value). The lean thinking may be applied to a team, a product or even the company in general. In order to understand if value was created by implementing or experimenting on our hypothesis a Build-Measure-Learn feedback loop was introduced as presented in Figure 9. The general optimization idea is to minimize the total time through the loop in order to learn from customer's feedback faster [22, 23].

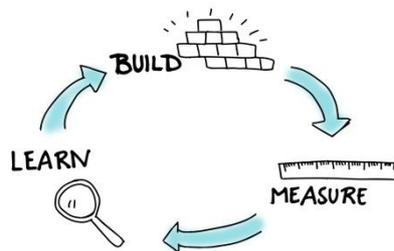


Figure 9. The traditional build-measure-learn feedback loop. ¹⁰

¹⁰ <http://www.meeteor.com/wp-content/uploads/2015/11/Build-Measure-Learn.jpg>

The feedback may be divided into qualitative and quantitative, what customers like or don't like and how customers use the product or what they find valuable, respectively. For a lean organization the information to learn is more valuable than the money earned, because learning helps to improve the business before it is ready to scale up [23].

The opposite of lean thinking is the traditional management thinking which optimizes resources and costs to please investors or managers with short-term results, therefore treating learning and failing as cost.

Most of the corporations owning software products today do not learn fast enough from customer feedback because organizations' culture is not a learning culture. The feedback loop is measured in years or months, not days as it should be. For traditional corporations, the feedback loop is there but the loop is not closed to go to the next cycle [24].

4.1.1 Validating Assumptions

Validating learning means intentionally testing assumptions and is backed by real empirical data collected from customers. Impact mapping compliments lean startup ideas by visualizing assumptions [1, 23].

The impact mapping technique facilitates collectively creating and testing assumptions, the technique taken from design thinking method where the problem is stated, focus is on doing not planning, and many options are created instead working deeply on the single best idea. The process for design thinking is to visualize all the steps along the way in order to activate the creative, organizational, communicative the right part of the brain instead of number-oriented left side [1, 25].

Assumptions can eventually be proven wrong. In order to "fail fast", as Lean Startup method suggests, assumptions need to be validated as early as possible. By putting assumptions in the context of goals, as impact mapping technique suggests, it is easy to decide which experiments are more valuable to be validated first [1, 23].

In Figure 3, the out-of-control assumption that players would invite their friends was presented. For validation, the semi-automated invites feature is first shipped in order to minimize the time through the feedback loop and measure progress against milestone metrics. When the assumption is proven to be wrong, the second most valuable in-control solution assumption is tested. When multiple feature assumptions fail, another market

assumption is taken into experimentation, because the ultimate goal is to reach the business goal.

During the delivery phase, teams may run into tasks that require technical investigation. Impact mapping suggests separating learning into a separate task, setting objectives and timeboxing the learning to get feedback sooner. It also suggests organizations to set up a learning budget so that stakeholders would see the benefit in learning and approve it [1, 10].

Therefore, it can be concluded that impact mapping is designed for amplifying learning through continuously validation assumptions, and moving organizations closer to using the lean principles.

4.1.2 A/B Testing and Cohort Analysis

A/B testing or the split-test experiment is a practice used widely in many industries where different versions of a product are offered to customers at the same time. The practice has its roots in the advertising industry where the success of two sets of posted ads was compared based on customer conversion. The A/B testing was named after a practice of assigning letter names to each variation representing separate paths to a common goal [23].

In the cohort analysis, data from the same data pool is divided into several groups. For example, people who graduate school in different years¹¹. It can be considered to be an A/B test controlled for dependencies, whereas regular A/B testing sees users as one big group.

Impact mapping defines the node actor who may be stakeholder, users of a system, specific user segment, specific user persona, person from other departments etc. A/B testing and cohort analysis in the context of assumptions may be run on the actor as an experiment. The impact mapping method has not set any limitations on the deliverable node. Therefore the author of this thesis concludes that impact mapping fully complies with A/B testing and cohort analysis principles, even if the actor is not a user of the product.

¹¹ <http://cohortanalysis.com/>

4.2 Introduction to Product Discovery and Product Delivery

Product discovery starts when a markets’ problem or an opportunity has been defined and the decision has been made for it to be worth to investigate.

The input to product discovery may come either from the stakeholders, customers or by analyzing data. The goal of the product discovery is to answer the question “What to build?” from the customer’s need point of view. The product discovery is usually led by the PO who is also responsible of introducing the product vision whenever needed. The discovery itself is a collaborative process.

Product delivery starts when the idea has been validated by the PO and consists in answering the question “How to build it?” so the solution would be reliable, adaptable and scalable, followed by the actual implementation. The output of product delivery is hereby in this thesis defined as the shipped features that provide value to the users.

Product discovery and delivery process is presented in Figure 10. All of the steps need to be addressed in the best possible way in order to launch a successful product. The process presented in Figure 10 may also be viewed in the context of a component or a feature.

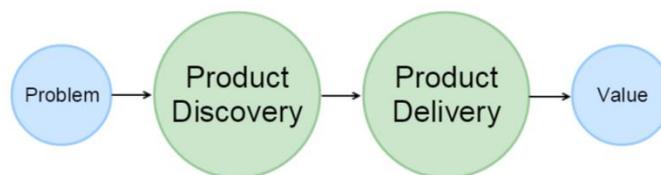


Figure 10. Product discovery and delivery process.

Product delivery is a discipline that the software development industry has been improving ever since the Agile Software Development Manifesto was first released in 2001. Over the years the industry has evolved a lot: development teams are used to agile frameworks, iterative delivery is used over long projects, continuous integration reduces quality issues, and continuous deployment allows shipping value faster [26].

Product discovery is a discipline that, today in 2016, as perceived by the author of this thesis, has not really been solved yet in a sense of the key principles being agreed on how to do it. Good methodologies exist but PO-s and teams have not adopted those yet. The author of this thesis predicts that in the upcoming years big mindset shifts in that respect are to be expected in the (software) product development industry. Building a product in a

way that it is successful to the business and loved by its users is a challenge that requires understanding both the market and the customer.

4.3 Product Discovery Perspective

Traditionally, product discovery in the software development industry has been consisting in gathering requirements and ideas from business representatives that order the development. The author of this thesis sees that as a business needs' oriented discovery, which should nevertheless be taken into account but the PO needs to think of fulfilling a user's needs as the method to achieving business objectives, not vice versa. The author of this thesis hereby defines the product discovery as a user or customer need oriented discovery.

The mindset shift is particularly important when working with consumer products and has milder impacts when products used in-house are developed. The purpose of the product discovery is to ship products that users love and would recommend to a friend. In a corporate world, users may be forced to use the product even if before mentioned criterias are not fulfilled. The expectation to the successful PO is to syndicate user need oriented discovery and business need oriented discovery together, so that a product would solve both at a time, which in turn leads to distilling business strategy into product strategy.

4.3.1 Market and Customers

The PO must at any time be ready to introduce the product vision and defend the product strategy. The PO is also responsible of acting as a representative of the customer in the eyes of the development team. In order to do that, one needs to know and understand a great deal about the market, competitors and customers or users.

From the experience of working as a PO, the author of this thesis claims that if asked directly from the customers what they want, they usually want it all until they realize they must learn to manage it all. So the most important part of the PO's work is to say "no" to new features. An impact map helps to visualize assumptions by linking them to the business objective and therefore it is easier to prevent scope creep or pet features that do not link to the value being pursued, and eliminate waste originating from miscalculations in planning. Also, by visualizing many options at a time, it is easy to target most the valuable options and prioritize those with a smaller value lower. The methodology also facilitates finding high-value low-hanging-fruits. It is likely that some organizational

activities might even bring more value to customers than implementing yet another feature or improvement [1].

4.3.2 MVP and Lean UX

Hereby we define the term MVP for this thesis. It is a tactic of shipping a limited feature set product to earlyvangelists in order to get feedback fast and to reduce engineering waste. The author of the idea is a Silicon Valley entrepreneur and academician Steve Blank and the idea was popularized in a bit different phrasing by Eric Ries book [23]. Earlyvangelists¹² are defined by Blank as the ones having a problem and money to pay for a solution, who have looked for solutions and since none exists, have joined separate pieces together as the current solution. Ries uses the term early adopters instead [23].

Over the years, the concept has gained much controversy from its adopters and based on the experience on coaching clients and agile teams the author of this thesis concludes that it is mainly due to the fact that the MVP idea has been misunderstood while reading Ries' book [23]. The most common misunderstanding presumes that it is a tool to save money by building a product stepwise or shipping a product with poor quality. Some adopters quickly find out that their MVP is not working. Because of that, they claim it is overrated and overused. Based on the definition given earlier, the author of this thesis claims that MVP is a process, not a product.

MVP should mainly be used to sell a vision and as a first step to learn market's needs. In order to learn one doesn't even need software, a prototype is as good. The MVP cannot be tested on engineers and designers; it needs to be tested on real customers to go through the entire build-measure-learn feedback loop as a solution provider in order to test the hypothesis that the current solution is already good enough, and that the customers are already willing to pay money for it [23].

Based on conclusions made in Chapter 3, impact mapping does not fully comply with defining the MVP scope, as such a thing could not be defined for the product milestone in advance in the first place. In Figure 3 it was presented and in Chapter 3 it was discussed that the first three nodes of an impact map define the market need assumptions that are out of the influence sphere. If this restriction is kept in mind while having workshops, impact mapping could be used with MVP-s but for the author of this thesis it is hard to see

¹² <https://steveblank.com/2010/03/04/perfection-by-subtraction-the-minimum-feature-set/>

justifications in doing so because other better techniques and methods meant for this purpose are available. One of the suggested methods could be the Effect Mapping method introduced in Chapter 2, which is similar to impact mapping but is designed for interaction design and UX design purposes. Another method already suggested in Chapter 2 is the GV Design Sprint where the user testing of prototypes is already defined by the technique itself.

As described in this section already, assumptions may be tested on (early adopter) users without developing software. Lean UX is a lean prototyping practice for designers having the ultimate goal of putting the power of validating the design's direction to the customers as early as possible. The designer's output emphasis is not on deliverables like internally used documentation but the focus is set on the experience being designed and validated through the build-measure-learn feedback loop. The initial focus of the process is on the design and then improving the user's experience [27].

Based on the experience of the author of this thesis, Lean UX is sometimes mistaken or mixed with the term Agile UX. The latter refers to the delivery phase or to the fact that internal work of the designers has been organized using some agile framework. Lean UX within this thesis refers to the discovery phase actions. The purpose of the PO is to help teams build the right thing, whereas in agile environment according to the Scrum framework it is the Scrum Master who helps teams to build the thing right according to processes set by agile frameworks.

Impact mapping advocates that once the goal is achieved, further actions must be stopped. A follow-up session for a strategy rethink is required, even if the whole budget was not spent [1]. The author of this thesis concludes that this complies with the lean startup methodology where regular pivot or persevere meetings are held in participation with product development and business leadership teams, to decide if the current path is followed (persevere) or a fundamental direction change is required (pivot). This is not odd, because startups by definition are searching their business opportunities on the market and testing their MVP, therefore a startup's runway is the number of pivots it can make. Pivot is a strategical planning decision; therefore it complies with the impact mapping method [23].

4.3.3 Disruptive Innovation

The concept of “Innovator’s dilemma” was first introduced by Joseph L. Bower and Clayton M. Christensen in 1995¹³. It corresponds to the market situation where an established business is making incremental improvements to their existing product driven by the needs of their existing customers and not making enough new risky innovations as reviewed in Chapter 3. Then a new entrant to the market introduces an innovation that only jeopardizes a small part of the established company’s business, the one not very significant to the current market leader in terms of revenue. Since the established company does not start to compete, it will lose its market in some time in general because by that decisive time, the current market leader is already far behind in technology and cannot serve the new needs of the customers any longer [28].

Hereby the categories of the disruptive innovation are defined as introduced by Peter Thiel in his book [29]:

1. a new product;
2. a new technology to produce a product;
3. a new way to distribute a product;
4. a new way to provide services.

Lean practice may be seen as methodology, not a goal itself. Making small changes to things that already exist might lead to a local maximum but not to a global maximum. Therefore an established company, even if it uses lean practices, is not superior to the new ideas that fall into disruptive innovation category [29].

4.3.4 Stakeholders’ Perspective

In Chapter 1 it was introduced that one of the main objectives of impact mapping is to facilitate discussion between technical and business people, to bridge a communication gap, and align everyone with underlying assumptions to think towards a common shared goal. When workshop participants are given a shared canvas or a template, it makes the meetings effective. In addition, impact mapping routes people to think of the problems instead of delivering a shopping list of solutions to the PO.

¹³ <https://hbr.org/1995/01/disruptive-technologies-catching-the-wave>

Impact mapping is a powerful technique to evoke discussion and it is also “very good for leading the discussion of a workshop” as discovered during the case study on impact mapping by Juuso Hyvönen, discussed in Chapter 3. Additionally it was discovered that “the well defined structure of the map was very effective at keeping the discussion on topic” [19]. Based on the background with facilitating impact mapping meetings, the author of the thesis agrees with Hyvönen. In addition, when pre-prepared workshops generate great results and receive good feedback it also helps to keep relationships better.

The most important strength of using impact mapping comes from the mapped secondary actors who are related to the service or the product which is provided. The technique channels workshop participants think not only about the users of the software, but also about organizational activities. Therefore it reveals how marketing department or some other actor may generate an even bigger impact on the business goal than some software feature. With the traditional PowerPoint planning meetings it is unlikely to be discovered.

By mapping offstage actors, the ones that are in position of power to obstruct like government regulators for example, some risks may be revealed and those can be managed.

As stated in Chapter 3, one of the responsibilities of a PO is to make information and ongoing activities transparent to all stakeholders. This includes the generation of roadmaps, and planning sessions, which — if held in the participation of both business and technology people — could lead to less explanation work required from the PO afterwards.

Based on the observations by the author of the thesis, usually workshops with more than six people are not very effective; therefore everyone involved cannot be invited. Having a visualized workshop result to present to the people afterwards makes the assumption communication easier for the PO, since with every handover on sharing info, some information gets lost.

4.4 Product Delivery Perspective

Hereby the usage of impact mapping in the product delivery perspective is observed. The product backlog management, grooming and describing backlog items is analyzed in more detail.

4.4.1 Managing and Grooming Product Backlog

Managing and grooming the product backlog is the key responsibility of a PO. The product backlog, as seen by the Scrum framework, is a list of items waiting to be developed that has been organized by business priority, and where the top of the list has been evaluated together with the development team. The product backlog is dynamic and in its essence is never ready because new requirements come in; some options expire and are therefore removed from the list.

In order to communicate requirements and underlying business assumptions to the development team, one has to know the assumptions in the first place. Impact mapping is a technique that helps to gather and visualize assumptions about the market and the assumptions about deliverable-market relations, which is info that is traditionally hard to obtain. On the other hand, this info is useful when designing a good product that fulfils market expectations and the needs of the users that get value out of a product.

The Impact Mapping community website has listed three mapping/backlog tools: EffectCup, SpecLog and MindMup [3]. The author of this thesis has tried them out but in practice most commonly the teams that develop software as their main job have their preferred and accustomed tools in use already. For example JIRA Agile is neither compatible with suggested tools, nor does JIRA have a plugin to visualize product backlog as a mind map as a workaround. In practice, it is inconvenient to enter the same data twice and synchronize in two locations. For that reason impact mapping based backlog management shall be dropped. Therefore, the author of present thesis avows that impact mapping could be used to manage a product backlog as advocated by the author of the methodology but, in practice, the tools to do that have not evolved enough.

The other possible suggested usage of an impact map is to make roadmap decisions. A roadmap in traditional development models is seen as a linear project with a start and a finish. Agile and Lean practices tend to see a roadmap as a tool for navigation in order to move from point A to point B where the route may change during the trip. Impact map works in a similar way as it provides many options and it is up to its user to select and implement the best ones. Therefore, the author of this thesis concludes that using impact mapping to make roadmap decisions in correspondence with agile and lean principles is a good idea.

4.4.2 User Stories

A good User Story describes a feature from the point of view of the customer or some specific user segment, helping to keep engineers focused on the customer's perspective throughout the UX design and development process.

A traditional user story template was first introduced by Rachel Davies and Tim McKinnon in 2001 and after that it has not changed much [30].

Agile frameworks do not say exactly that the backlog items have to be user stories; instead just the term item is used. Therefore the items could be in whichever format. However, in general, user stories are considered to be *de facto* standard in the software development industry nowadays.

A typical user story is constructed in a following way:

As a <user or stakeholder type>

I want <some software feature>

So that <some business value>

The format is easy for everyone to understand but in order to write a good user story, a lot of experience is needed. The point of the story is in the telling, not the template [10].

From the practice and gathered know-how of the author of this thesis, most common pitfalls are the following two:

1. Generic actor „as a user“ or „as a customer“ is used, whereas adjectives such as “busy executive”, “desperate housewife” would help to bring the story to life.
2. Story does not describe the change in users’ behavior.

An impact map and its nodes may be used to generate user stories that help to communicate assumptions and describe business value to its reader as seen in Figure 11.

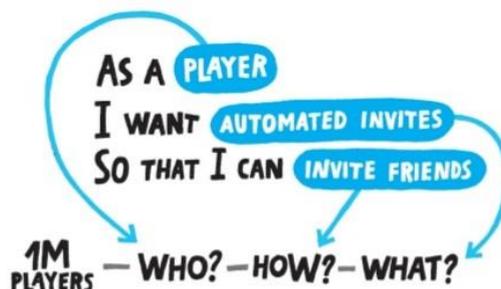


Figure 11. An example of generating a user story based on impact map [1].

The author of present thesis agrees with the author of the technique that the user stories created based on an impact map provide a great context about assumptions and how the behavior of an actor is expected to be changed. Therefore those user stories would be useful for all the participants that can use the user stories to agree on an iteration scope, to generate UX wireframes and design components, and to develop and test working software.

Another new usage proposed by the author of this thesis is to use the impact mapping technique and already created impact maps for learning purposes for the novice PO-s or business analysts, for the purpose of mastering the art of creating meaningful design thinking driven user stories, even when in the future impact mapping facilitated discussions are not used for defining scope.

Conclusions

Within this thesis an inquiry on a technique called Impact Mapping was conducted. An overview was given as a mixture of initial material from the author of the technique and follow-up material from the author, his peers, practitioners and researchers. The creation of the combined overview was induced by the fact that the initially published material was somewhat raw and the author wanted to get feedback on how the idea progresses. Over the years new ideas have emerged and discoveries have been made, also some constraints of the technique have revealed themselves.

Later within this thesis the impact mapping technique was put into agile software product management context and, based on the experience of the author of present thesis, an overview of day-to-day challenges and expectations was given. The product discovery, product delivery and learning perspectives were analyzed in the context of the trends and problems of today's agile software product development industry, and impact mapping as a tool to help to tackle those was proposed.

Within this thesis it was discovered that impact mapping is a great technique for a product manager to use, but the context of the usage and constraints have to be understood.

In addition, it was discovered and proposed that impact mapping could be used as a learning tool for novice PO-s or as a gateway to other techniques. To become a great product manager, one has to know different tools and choose the best ones given the constraints revealed in this thesis; impact mapping is just one possible option to choose from.

One of the important conclusions made within this thesis was related to product discovery. It was concluded that that impact mapping does not fully comply with the products that do not exist yet and it is best to use it for learning through experiments pursuing medium-term targets in an incremental improvement way of working.

The author of present thesis hopes that the author of the technique will take new perspectives into account and will publish an updated version of the book soon to cater the new needs within the software development industry today. It would be an honor if some perspectives analyzed in this thesis would get covered.

Bibliography

- [1] G. Adzic, Impact Mapping: Making a Big Impact with Software Products and Projects. Provoking Thoughts, 2012.
- [2] G. Adzic, I. Domingues, J. Berndtsson, "Getting the most out of impact mapping", November 2014 [Online] <http://www.infoq.com/articles/most-impact-mapping>.
- [3] "Impact Mapping community web site", <https://www.impactmapping.org/> [Accessed April 2016].
- [4] D. Krimmer, "How to make an impact with your software product", November 2014, <http://www.dkrimmer.de/2014/11/21/how-to-make-an-impact-with-your-software-product/> [Accessed April 2016].
- [5] T. Gilb, Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage. Butterworth-Heinemann, 2005.
- [6] J. Jordan, A. Hariharan, F. Chen, P. Kasireddy, "16 Startup Metrics". Andreessen Horowitz, August 2015, <http://a16z.com/2015/08/21/16-metrics/> [Accessed April 2016].
- [7] C. Hassa, "Impact Maps and Story Maps". Swiss Requirements Day Zürich, June 2014, <http://www.slideshare.net/chassa/2014-0618srdimpact-mapsstorymaps> [Accessed April 2016].
- [8] N. Talens, "Impact Mapping and Continuous Validation". Gareth, November 2015, <http://craftsmenlabs.github.io/gareth//2015/11/16/Impact-mapping-and-continuous-validation.html> [Accessed April 2016].
- [9] J. Patton, User Story Mapping: Discover the Whole Story, Build the Right Product. O'Reilly Media, 2014.
- [10] G. Adzic, D. Evans, Fifty Quick Ideas to Improve Your User Stories. Neuri Consulting LLP, 2014.
- [11] I. Ottersten, M. Balic, Effect Managing IT. Copenhagen Business School Press, 2007.
- [12] C. Smith, R. Troughton, T. Ponton, "Episode 83: Making Impacts with Gojko Adzic". The Agile Revolution, November 2014, <https://theagilerevolution.com/2014/11/30/episode-83-making-impacts-with-gojko->

[adzic/](#) [Listened April 2016].

- [13] M. Balic, J. Berndtsson, I. Ottersten, M. Aldman, From Business to Buttons. 7th International Design Conference Dubrovnik, 2002, https://www.designsociety.org/publication/29623/from_business_to_buttons [Accessed April 2016].
- [14] "inUse web site", <http://www.inuseexperience.com/> [Accessed April 2016].
- [15] R. Pichler, "What is Agile Product Management?". Pichler Consulting, March 2010, <http://www.romanpichler.com/blog/what-is-agile-product-management/> [Accessed April 2016].
- [16] "The Official Scrum Guide", <http://scrumguides.org/scrum-guide.html#team-po> [Accessed April 2016].
- [17] S. Johnson, "Crafting the Ideal Innovation Portfolio". Digital Intent LLC, <http://digintent.com/crafting-the-ideal-innovation-portfolio/> [Accessed April 2016].
- [18] J. Knapp, J. Zeratsky, B. Kowitz, Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days. Simon & Schuster, 2016.
- [19] J. Hyvönen, Towards value-oriented product development roadmapping. University of Helsinki, November 2015, <https://helda.helsinki.fi/bitstream/handle/10138/159190/thesis.pdf> [Accessed May 2016].
- [20] "Impact Mapping forum on Google Groups", <https://groups.google.com/forum/#!forum/impact-mapping> [Accessed May 2016].
- [21] R. Pichler, "The Agile Product Owner Responsibilities". Pichler Consulting, March 2016, <http://www.romanpichler.com/blog/the-product-owner-responsibilities/> [Accessed May 2016].
- [22] M. Poppendieck, T. Poppendieck, Lean Software Development: An Agile Toolkit. Addison-Wesley, 2003.
- [23] E. Ries, The Lean Startup. Crown Business, 2011.
- [24] T. Chi, "Rapid Prototyping & Product Management". Mind the Product San Francisco, July 2015, <https://vimeo.com/131447795> [Accessed May 2016].
- [25] J. Liedtka, T. Ogilvie, Designing for Growth. Columbia University Press, 2011.
- [26] T. Torres, "The Rise of Modern Product Discovery". Product Talk, March 2016,

<http://www.producttalk.org/2016/03/rise-modern-product-discovery/> [Accessed May 2016].

[27] J. Gothelf, J. Seiden, Lean UX. O'Reilly Media, Inc., 2013.

[28] C. M. Christensen, The Innovator's Dilemma. Harvard Business School Press, 1997.

[29] P. Thiel, B. Masters, Zero to One. Crown Business, 2014.

[30] A. Marcano, "Old Favourite: Feature Injection User Stories on a Business Value Theme". Antony Marcano, March 2011,

http://antonymarcano.com/blog/2011/03/feature_injection_user_stories/ [Accessed May 2016].

Appendices

I. Abbreviations

1. CAC - Customer Acquisition Cost
2. GV - Google Ventures
3. KPI – Key Performance Indicator
4. MVP - Minimum Viable Product
5. PO - Product Owner
6. R&D - Research and Development
7. ROI - Return of Investment
8. UX - User Experience

II. Licence

Non-exclusive license to reproduce thesis and make thesis public

I, Magnus Meldre,

1. herewith grant the University of Tartu a free permit (non-exclusive license) to:
 - 1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
 - 1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

Using Impact Mapping in Agile Software Product Management,
supervised by Helle Hein.
2. I am aware of the fact that the author retains these rights.
3. I certify that granting the non-exclusive license does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **12.05.2016**