Towards value-oriented product development roadmapping

Juuso Hyvönen

Master's thesis
UNIVERSITY OF HELSINKI
Department of Computer Science

Helsinki, November 18, 2015

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Computer Science

Tekijä — Författare — Author

Juuso Hyvönen

Työn nimi — Arbetets titel — Title

Towards value-oriented product development roadmapping

Oppiaine — Läroämne — Subject

Computer Science

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	November 18, 2015	58

Tiivistelmä — Referat — Abstract

Software companies have problems connecting business goals to actual software development tasks. This means that a lot of software development is done without linkage to business goals, which can lead to wasting time and money, and to bloated- and hard to maintain software and failed software projects. Roadmapping is a popular method to communicate future product development efforts but a problem with roadmaps is that they usually do not communicate the value the roadmapped future work is expected to create. This thesis presents an action oriented case study on three software planning techniques that fit into a lean software product development organization. The case study is about benefits of using Lean canvas, Impact mapping and Lego serious play as tools for value-oriented product development planning. Lean canvas and Impact mapping are promising techniques for helping a company connect business goals to actual software development tasks. They therefore aid in avoiding waste and creating value through putting effort only on the development tasks that create value. Lego serious play is an strategic planning method that utilizes Lego bricks and the knowledge of the whole development team to improve decision making and building shared knowledge. Results from the study show that Lean canvas proved to fulfil its promises to be a lightweight technique that improves shared understanding of the business model. Lego serious play has a similar benefit of improving shared understanding in the team, but it is more geared towards visualizing problems and finding solutions for them. Impact mapping was proved to be an effective way to find value creating tasks and to visualize the value assumptions behind each task. Based on these findings, an approach for value-oriented roadmapping is sketched.

ACM Computing Classification System (CCS):
Software and its engineering → Requirements analysis
Software and its engineering → Software design techniques

Avainsanat — Nyckelord — Keywords

design techniques, requirements analysis, roadmapping, lean software development, value

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

Acknowledgements

I would sincerely like to thank all the parties that supported me during my thesis process. Steeri Oy and Terho Norja for giving me and opportunity to start the research process and to use their valuable time for experiments. Solita Oy and Timo Lehtonen for giving me an opportunity to finish the research I started but was never able to finish. Jürgen Münch from University of Helsinki for working as my instructor and guiding me towards finishing my thesis. Tomi Männistö from University of Helsinki who also instructed me during the thesis process. Mom for keeping nagging me about my thesis not being finished. Special thanks for supporting me during the process (not in any order) also goes to the following people: Juho Sorvettula, Olli Rissanen, Mari Pursiainen, Paavo Toivanen and Iiro Uusitalo.

Thank you,
Juuso Hyvönen

1 Introduction

Software companies are like any other companies in that they have their visions, their strategies and their business goals. They set goals to effectively plan and to communicate their steps towards achieving their vision. Short term goals can be achieved in a few weeks while long-term goals can be set to five years, or even over a quarter century away. Companies set similar targets for their software products. They have their vision and their short term goals, but how do companies and projects link their everyday software development tasks to these goals? It is understood that linkage is vital for decision makers to be able to invest in the practices that create real value and help company achieve its goals.

From a software development point of view, modern software companies typically use agile software development methodologies such as Scrum, Kanban and XP. Software projects have their product backlogs and sprint backlogs, and sprints to develop backlog items from sprint backlogs. Development model for sprints is usually well documented. A great deal of literature to support each phase of development, with the exception of one very important part of the process, the black box called "product backlog". Finding valuable tasks is usually left outside of the development process and items are chosen by the managers and customer from a roadmap, or they are invented on demand.

Roadmapping is a popular method to try to link business level decisions to product development level actions and to visualize future and past [KS01, VLR02]. It can be an effective way to plan future development but often roadmaps are used rather to visualize what the next development steps will be than to show the real map of possibilities for the future. The value assumptions behind each possibility are usually not explicitly declared and the roadmap visualizations are more like parallel runways than road maps with crossing roads. They have one way and one way only with the goals becoming more abstract as they are further away in the future.

This is where this thesis started from. There exists a need for roadmaps that would not just show one dimensional future but would also have cross-roads and also maybe different types of roads. The roadmap should be oriented towards value-creation. After beginning this study, it was rapidly noticed that understanding of what is valuable seemed to be a hard problem for all the stakeholders of software projects, which led to a need for projects to be able to link their work to business goals and to improve the understanding of value expectations behind every development task. If developed features do not create value, all efforts to improve development process are worthless.

This thesis elucidates value-creation in software development and presents a case study on three techniques. The methods are Lean canvas [Mau12b], Impact Mapping [AB12] and Lego serious play [LEG10]. Based on the findings of the case study, a proposition for a lightweight value-oriented roadmapping solution is given.

Two papers have already been written in parallel with this thesis. The first one introduced Lean canvas and Impact mapping and analysed how they can be used to find software tasks that help in reaching a business goal [Hyv14b]. The second one studied Lego serious play and how it supports creating shared understanding [Hyv14a]. This thesis combines the knowledge of earlier papers and expands it further with a more detailed description of the case study and a proposition for value-oriented roadmapping.

The structure of this thesis is the following: Chapter two builds understanding on value creation in software development. It starts from the abstract concept of value and moves towards business value and then gives examples of how business value can be modelled and communicated. Then Lean software development, which works as a framework for this study, is introduced. After that, roadmapping and the three evaluated planning techniques are explained. Chapter three describes the case study, which was done in two Finnish IT companies. Chapter four presents the findings of the study and Chapter five discusses these findings. It showcases an example of how to introduce metrics to the methods tested in the case study and gives a proposition for value-oriented roadmapping approach based on the results. Chapter six summarises the thesis.

2 Value creation in software development

If anyone repeatedly uses a piece of software, it seems clear that it somehow creates value for its user. The value could be, for example, amusement, financial benefit or new skills. To create value, software creators have been been able to gather know-how of what are the value-creating connections between different entities in the domain of a certain problem. Then they have been able to deduce the features that the user of the software would consider useful.

Finding the value-creating features is a major challenge in software development. Even for simple problem domains, the value creation process can get surprisingly complex. Often the features chosen to be implemented are not the optimal ones. The problem gets even harder when the software creator is supposed to make profit.

There are many reasons for the features to not be optimal. Often the features could be working perfectly but no one would be using them because they actually do not solve any real problem. Sometimes the features solve a wrong problem. For example they could speed up one stage of production when the bottleneck is actually on a different stage of the production line. Sometimes features are simply thought to be useful and get implemented. Later on, no one uses them.

2.1 Value

Decision making in firms is based on a notion of perceived business value. In an optimal situation, the option with most business value is chosen. Based on this, to improve decision making, understanding of the business area should be improved. Before delving into the concept of business value, a good understanding of what is value is of great benefit.

2.1.1 Defining value

To be able to define value, a strategic management theory called *resource-based view* [Wer84] is used here. Resource-based view studies companies from a viewpoint of resources rather than from the viewpoint of products (Product market fit). Companies use their capabilities to use resources to generate new resources that their customers hopefully see as valuable. Consumers then buy products (resources) to create new kinds of valuable resources again. This means, for value, that there is always someone or something that creates a resource and someone who perceives that it is valuable. For anyone to consider resource A valuable enough for acquiring, the perceived value of resource A should be higher than the amount of money or some other resource B, the producer of the resource A is demanding in exchange for resource A.

Bowman and Ambrosini [BA00] see value as a compound of two main components, *perceived use value* and *exchange value*. Perceived use value is a subjective value, which is the value that the customer sees or which the customer deduces the resource to have. For example, a hammer could be seen to have a use value of being useful for hammering nails or opening piggy banks. A concept near the perceived use value is the value the customer is prepared to pay for the the hammer which is the *total monetary value* perceived by the customer. The total monetary value is paid fully only in the case that the resource is only produced by one supplier, meaning that there is a monopoly situation. In all other situations, competition reduces the sum of money or other resources that has to be paid to acquire the product and the amount paid will be less than the total monetary value.

Total monetary value does not equal the perceived use value of the product. The total monetary value is usually higher than that. The difference is called *consumer surplus*. This is the same value consumers typically call *a value for money*. If two products have the same price but the other has more consumer surplus (maybe the hammer is very decorative), the one with more consumer surplus is chosen.

The other component of value, the exchange value is the value which is the value realized for the producer of the goods when the customer decides to pay for the perceived value of the goods. This can be for example money or services. *Profit* is made when the amount of value from realized exchange value is greater than the sum spent on acquiring and transforming the sold use value.

2.1.2 Creating value

Acquired resources need to be worked on to activate and further develop their use values. For example, iron ore needs a lot of work to become steel. These *input resources* can also be immaterial such as information and brands [BA00]. Immaterial resources require work just as physical resources to realize their value, and their value can even decrease if they are not worked on. Resources are often combined and worked on to create new valuable resources, such as, when creating steel, carbon and small amounts of other elements are added to improve iron's properties and to form a new and more valuable resource. Brands can merge to create something bigger such as Exxon-Mobil or two brands can cooperate to create more consumer surplus such as Apple Watch with Hermès wristband. Information can be combined to create new information that someone perceives as valuable.

The amount of new value created from existing resources can be determined when the new product is sold. Sometimes use value grows radically. For example, steel is a much better metal for most tools than plain iron. Sometimes it is the consumer surplus that grows. Apple Watch with Hermès wristband has the the same use value as Apple Watch with basic wristband but many might be willing to pay more for an Apple Watch with Hermès wristband. Of course, some might also consider the increased status achieved by owning a more expensive watch a use value, as it can be beneficial in some social situations.

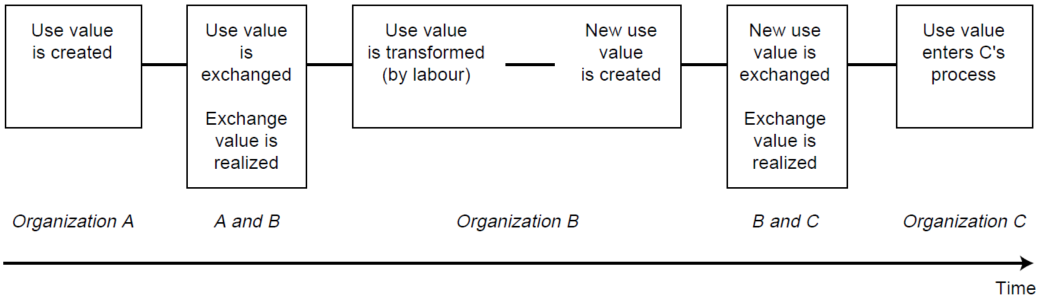

Figure 1: Value creation process [BA00].

Figure 1 demonstrates the process of value creation. From left to right: something or someone A creates a resource that has use value. Then B perceives that the resource is valuable and decides to get it. If the resource was owned by someone, that someone A gets a compensation in a form of realized exchange value. This can be, for example, money or services. B then transforms the newly acquired use value into something that potentially adds new use value. Then C perceives that the resource is valuable and decides to get it. B gets a compensation in a form of realized exchange value. This can be, for example, money or services. C starts the process of transforming the newly acquired resources into something that creates new use value and so on...

2.1.3 Improving profit

Resource-based view gives a great and simple way to demonstrate profit differences between firms. The view states that "if all inputted resources are homogeneous, and freely traded, competing firms will produce identical products, incurring identical costs of production. All firms in this market would produce identical perceived use values and identical amounts of exchange value, and profit, would be realized. This equates with neo-classical perfect competition." [BA00]. For software development companies, resources are basically identical and standardized, and only thing that can create difference between firms is people. Bowman and Ambrosini suggest three categories of labour to explain profit differences between firms.

Generic labour is work that does not create any noticeable difference between firms. It is work that is needed to produce the goods but it cannot be the source of great profit differences between firms. This kind of labour preserves the purchase value of bought resources and tries to minimize additional costs.

Differential labour is the labour that creates a lot of value. It is the source of major profit differences between companies. It is something unique the company has. This can be, for example, very talented designers and salespeople or analysts. It can also be the way the company is able to build teams and use resources in a particularly clever way.

Unproductive labour is labour that destroys value. It has a negative effect on the profit compared to other firms. Resources could be wasted or managers might get better salaries than what they are worth. Unnecessary features could be added to software products. As the company does more this kind of work, it gets less profitable against its competitors.

Based on this, Bowman and Ambrosini suggest that it is the use value created from the acquired resources what matters. Minimizing the unproductive labour and maximizing the differential labour the company improves its competitive profits.

2.2 Business Value

Early economics assumed that people are rational and therefore systematically and carefully consider the available options before making decisions, for example, purchasing services. But to be able to make the best possible decision, one would have to understand all the aspects of the decision. That sort of understanding is usually not possible. Term *business value* is used to describe value assumptions behind *business decisions*, which could be defined as decisions that have effect on a firm on the long run. Still, there seems to be no consensus on what exactly is business value [RDS09, MBH$^+$10].

The concept of business value is often found in agile literature. Racheva et al., in their systematical literature review, found out that there exists a lot of discussion about creating business value but majority of the papers, practitioners' and scientific, on agile software development did not define the concept [RDS09]. They were also trying to find out in which way agile projects create business value. For that question as well, they were unable to find any answers. The reason for that finding is, they think, that the concept of business value is more slippery and volatile than most authors seem to assume.

The definitions Racheva et al. were able to find about business value are the following:

- "... business value, as measured in business revenue, stock price, market share, or other business metrics. Value is in the eyes of the customer... "
- "Business value is something that delivers profit to the organization paying for the software in the form of an increase in revenue, an avoidance of costs, or an improvement in service"
- "Business value is a communication vehicle: we use business value to communicate value, priorities, motivation"
- "Business value is what management is willing to pay for; value can only be defined by the ultimate customer. And it's only meaningful when expressed in terms of a specific product (a good or a service, and often both at once), which meets the customer's needs at a specific price at a specific time"
- "Might not be possible to define the business value of IT independently of other activities. What is business value:

  Business value = F(x) + F(y) + F(z) + ...

  That is, a complex function where we must balance multiple things ...while they are changing!"

All the found definitions were from practitioners' articles. They could not find any scientific publications and hypothesized that business value in the context of agile software development is not studied in academia because of an assumption that it is a self-evident concept, and as noted earlier, it seems that business value is actually not that self-evident.

To make identifying business value easier, Racheva et al. deducted five characteristics of business value based on their literate review.

1. Business value in practice tends to be qualitative
2. Business value tends to be subjective
3. The sources of business value drive requirements prioritization
4. Business value of the IT solution requires a degree of trust
5. The business value of IT solution tends to be dependant on non-IT business processes

These characteristic still leave the definition of business value very vague. Racheva et al. summarize that "The key distinguishing feature of the agile practice is reprioritization, based on an assessment of business value that appears to be uncertain and changing over time. The idea that re-prioritization is driven by calculating a cost function can be discarded as overly simplistic; it seems evident that some non-trivial decision making is involved."

2.2.1 Stakeholders and value

A major problem in finding the value is that different interest groups inside a company might have completely different value expectations for the same system [BAB+06]. For example, software developers merely want to make a product that works and is of high technical quality. Product and project managers often have a higher level vision and a sales department might expect value to be something that can be clearly communicated to potential customers. A customer's expectations of the system can also highly differ from the ones of a software provider. All these different groups are *stakeholders*.

Stakeholders are persons, groups and organizations that have to be taken into account when business decisions are made [Bry04]. Other common definitions according to Bryson include:

- ”Any person group or organization that can place a claim on the organization's attention, resources, or output, or is affected by that output” [Moo95]
- ”People or small groups with the power to respond to, negotiate with, and change the strategic future of the organization” [EA98]
- ”Those individuals or groups who depend on the organization to fulfill their own goals and on whom, in turn, the organization depends” [JS02]

The definitions above share a similar meaning that stakeholders are the groups that count when decisions about any business activities are made. Software business being a business like any other, stakeholders are as important to it as they are to any business. Stakeholders are something that can create or reduce the value of the software system. This makes understanding stakeholders a necessary tool for value creation.

Even stakeholder groups do not always share uniform value expectations. Important notion by Racheva et al. [RDS09] is that when agile organizations refer to "customer", they mean a multi-stakeholder setting in a client organization. It means that the customer is not just one stakeholder but that the customer can consists of multiple different value opinions.

Bryson hypothesizes in [Bry04] that "strategic management processes that employ a reasonable number of competently done stakeholder analyses are more likely to be successful – that is, meet mandates, fulfill missions and create public value – than those that do not". He also states that what constitutes to "reasonable" amount of stakeholder analysis is not clear yet.

Stakeholder analysis can be an expensive and time consuming process because of the high amount of people and time it can require. Performing excessive stakeholder analysis can be seen as a waste of time and money [Bry04]. Based on the notion that excessive stakeholder analysis is waste, the biggest benefit can be achieved when a right, not too much, nor too little, amount of planning is executed. Sometimes it might be enough to make project team aware that there are multiple stakeholders and the other times a deep and systematical stakeholder analysis could be beneficial. The systematic approach is especially important for high value public service projects as they often directly affect people's lives and get a lot of public coverage.

2.2.2 Business value models

To be able to steer the work of a company or a product to a right direction, a thorough understanding of different dimensions of business value should be built. For this, business value models have been suggested. This chapter will introduce two of these models. The first model is by Mandić et al. [MBH+10]. It incorporates risk/benefit model for the existing GQM+Strategies [BLR+10] which is helpful when analyzing the GQM+Strategies grid. The Mandić et al. model consists of two components. A benefit/investment analysis which is represented in Figure 2 and GQM+Strategies business goals' risk assessment which is represented in Figure 3.

Value Goal	Level	Level-*i*		Lower levels		Estimated		ARE
(Object)		Cost	Benefit	Cost	Benefit	Investment	Benefit	
V1(B1)	1	300FTE	1200FTE	228.5FTE	54FTE	Small	Large	*VH*
V2(B2)	2	28.5FTE	18FTE	200FTE	36FTE	Small	Medium	*H*
V3(B3)	3	200FTE	36FTE	n/a	n/a	Small	Small	*M*

Figure 2: Benefit/investment analysis [MBH+10].

The benefit/investment analysis consists of rows of value goals that are analysed on multiple aspects. In Figure 2, value goal 1(V1) is linked to business goal 1 (B1). It is highest level goal (1), meaning that other goals are derived from it. The cost and benefit of the value goal is described as

Full-Time Empoyees (FTE). The Level-i Cost and Benefit quantify the cost and benefit of implementing that specific goal. The Lower levels Cost and Benefit quantify the cost and benefit of the lower level goals related to the specific goal. The Investment describes how big an investment the goal is in relation to the total investment budget allocated for the entire business goal. Benefit describes how big of a benefit is estimated from implementing the value goal successfully. ARE stands for acceptable risk exposure. It is used to describe what level of risk is acceptable for the specific goal. V1 is of very high risk (VH) but the lower level goals are of lower risk exposure (High and Medium).

The second part of the Mandić et al. model, the business goal risk assessment lists the business goals similarly to the value goals that are listed in the benefit/investment analysis. The business goal 1 (B1) is listed on the first row. Key assumption for the B1 is listed next. Next the Likelihood (certainty) of the assumption is described as High. Assumption Type is listed as + which means that the assumption is of positive nature. Risk exposure (RE) for the goals is listed as medium (M).

Business Goal	Key Assumptions	Likelihood (certainty)	Goal Impact	Assumpt. Type [+/-]	RE
B1	[A1]: Existing core competences can be adequately extended with IP testing knowledge/skills in relatively short time period.	High	Significant	+	*M*
B2	[A2]: Product manager and marketing department have sufficient understanding of customers' needs in new business domain.	Medium	Significant	+	*H*
B3	[A3]: Stable product requirements will be ready and specified on time.	Low	Moderate	+	*H*

Figure 3: Business goal risk assessment [MBH+10].

Mandić et al. also define more precisely how ARE and RE are formulated but question is outside of interest of this thesis.

The second model is by Heidenberg et al. [HWMP12]. They propose a model that can be used to help in understanding the different dimensions of business value. They propose a model of six attributes that have a scale from one to four. They claim that the attributes were chosen in compliance with agile values. The attributes support "Communication and interaction between customer representatives and developers." and "Support both the customer's and the development organization's goals.". The attributes are said support "long-term growth and development instead of solely aiming for short-term economic revenue." and that they help "developers understand business goals." and "business stakeholders understand development goals.". The model is supposed to help in product backlog prioritization, sprint planning, and business information radiator and it should be useful for business stakeholders and technical staff.

The first attribute is *monetary value* which is the estimated monetary value of the business case of the feature. Feature would be categorized in four categories, for example, in the following way.

I 0-50Ke
II 50-100Ke
III 100-200Ke
IV 200Ke+

The second attribute is *market enabler*. A feature is considered market enabler if the feature opens possibilities for new markets. For example, support for special characters could make the product more fit for countries with a more demanding character set. Heidenberg et al. suggest the following categories.

I No enabler, can be added at any time without missing a marketing window.

II Feature can potentially work as a market enabler, alternatively, delays might affect marketing or sales aspects.

III Feature is to a significant degree a market enabler, alternatively, delays will to a significant extent harm or counteract marketing and sales efforts.

IV Feature is critical to complete before any other features are considered.

The third attribute is *technical enabler*. When the feature makes it possible to later create new functionality, it is a technical enabler. Technical enabling can be put for example into following categories.

I No enabler, separate from all other features.
II Feature contains some foundation for further functionality.
III Feature contains significant foundation for further functionality.
IV Feature is critical to complete before any other features are undertaken.

The fourth attribute is *competence growth*. Developing new features always has an effect in the competence of the developers, and competence is an important factor in a company's success. The following categories are suggested for competence growth.

I Maintaining knowledge in some fields.

II More complex feature, maintain knowledge in multiple fields, using technologies or techniques not used on a daily basis.

III Feature where some new tools and/or technologies needs to be used. Some delays due to learning are expected, teams competence grows into new fields.

IV Complex feature where a significant amount of new tools and/or new technologies are needed for progress. Delays are expected and the team's competence grows significantly.

The fifth attribute is *employee satisfaction*. Satisfied employees are more productive and low satisfaction often leads to employees changing their work place which means that employee satisfaction is a big factor in software companies' success.

I Work consists of routine tasks. Team members either do not understand why the feature is important or how it connects to the product. Neutral or even possibly some slight negative satisfaction.

II Work consists of a combination of routine and varying tasks. Team members have some understanding of why the feature is important and how it connects to the product. Positive or neutral satisfaction.

III Work consists of tasks not normally carried out, exploring new possibilities and viewpoints. Team members understand why the feature is important and how it connects to the product. Positive satisfaction.

IV Work consists of self-fulfilling tasks, which are novel and creative. Strong positive satisfaction by all team members.

The sixth attribute is *customer satisfaction*. Customer satisfaction is important because it makes the customer continue collaboration. Positive customer satisfaction could also bring new customers as customers communicate their satisfaction with the company's services. Negative customer satisfaction conversely has a negative effect on future collaboration and new customer acquisition.

I A low risk feature or enabler that does not deliver high value but is demanded by the customer and is easy to verify as correct. Neutral satisfaction.

II A feature that delivers business value to the customer and helps other features by reducing risks, enabling or solves a problem in the customer's environment. Neutral satisfaction.

III A feature with higher risk that delivers substantial business value to the customer or is otherwise important to the project. Successful implementation always gives the customer positive satisfaction.

IV A critical feature that delivers a lot of value to the customer or is otherwise very important to the project. Successful implementation is crucial to the project and is much appreciated. Always gives strong positive satisfaction.

To visualize the model, Heidenberg et al. suggest a polar area diagram presented in Figure 4.

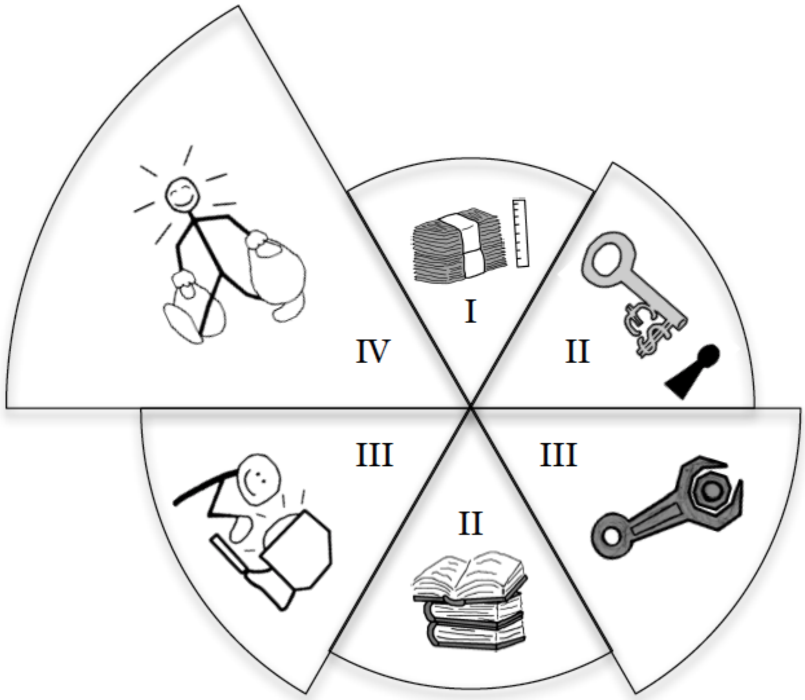

Figure 4: An example of the agile value model [HWMP12].

When comparing the two value models, it can be seen that the Mandić et al. model is geared towards high level decision makers and is clearly centered around financial feasibility of the business goals. It is all about finding the balance between risk and benefit. The agile value model describes different attributes (or dimensions) of value, which is useful for understanding what value the business goal is expected to create. The model lacks the risk dimension of Mandić et al. model. This kind of model could be useful during product development process.

2.3 Lean Software Development

Lean software development [PP03] is a software development method adapted by Mary and Tom Poppendieck from Toyota Automobile Production System [Ohn88]. It does not compete with agile development methods but is rather a complementary philosophy to support producing high quality software as fast as possible. Lean software development gave birth to the Lean startup which is a method for establishing IT startups, made popular by the book of the same name [Rie11]. It provides a set of core principles such as minimum viable product and actionable metrics to assure that the company is making the right decisions at the right time. Recently, also more mature software organizations have started to adapt lean ideas.

2.3.1 Waste

Lean software development is centered around a concept of waste and how to eliminate it. Eliminating waste equals to avoiding all the actions that do not directly produce value. Lean software development also has a principle of empowering the team which means that the software developers are seen as true professionals of the field and their experience should be utilized to the maximum. This could be seen as a bottom-up approach as it flips the traditional way of seeing management as higher level work.

Lean manufacturing methodology declares seven types of waste and the Poppendiecks were able to find comparable wastes from software development. Too big *inventory* is comparable to *partially done work*. Having partially done features means that time has been spent on something that never got validated. The product of the time spent is just laying somewhere unused. There is no knowledge of what problems a feature still might have and whether it solves the business problem it was invented to solve.

Extra processing can be compared to *extra processes*. Software development often has paperwork that adds little or no value. Some of it is necessary but paperwork should still be kept as compact and high level as possible.

Overproduction is comparable to *extra features*. Having more features does not usually make a product better. What matters is to have the right features. Every feature requires work to get it ready and after it is ready, it has to be maintained. There is no point in maintaining features that no one uses.

Transportation is comparable to *task switching*. Task switching typically interrupts the work flow and causes the work to progress slower than it would if one could finish one task at a time.

Waiting means the same in both manufacturing and software development. Waiting is always unproductive time. Waiting means that parts of the processes are not working as well as they should be.

Motion is understood as the same for both manufacturing and software development. The less people have to move to find information they need, faster they can work. Moving is also comparable to task switching in that it interrupts concentration and can lead to decreased productivity. Poppendiecks also make a note that not only people move, also artifacts move. Document handed to next person can usually not include all the information the next person will need even when the creator of the document knew what information the next person would need.

Defects are the same for manufacturing and software development. The longer a defect goes unnoticed, more it causes damage (waste). Noticing defects early is much cheaper than when the product is already in use.

A type waste not described in original lean manufacturing processes but an important one that the Poppendiecks bring up is unnecessary *management*. Management activities do not directly create value but can not be considered waste either. Management should be optimized to keep the right amount of work in pipeline, minimizing unfinished work and adding new work just-in-time.

Value stream is a concept introduced by lean manufacturing. It is a map that maps and visualizes the lifetime of a product from raw materials to a ready product and its consumption. The idea is to map how much time is spent producing and using the product and how much time is spent waiting. For lean software development, the similar map can be build by mapping for example how a software feature is born. This map is then used to optimize the process. Mapping should start from the initial idea and can end when the feature is deployed to production.

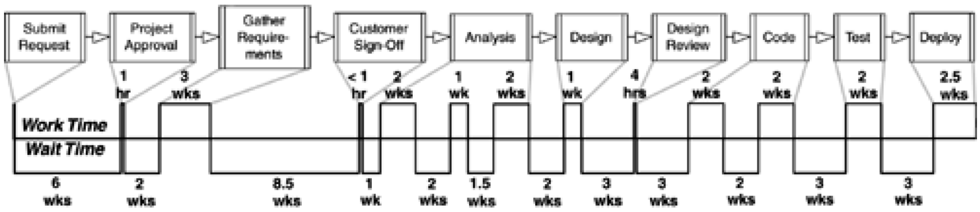

Figure 5: Value stream in lean software development [PP03].

Figure 5 is an example of a value stream. From left to right it shows a initial feature request and ends when the feature is deployed. Also the time spent working and time spent waiting are clearly visualized. It can easily be identified that the customer sign-of takes a very long time (8.5 weeks) and on overall there are long gaps between each work phase. It takes three weeks for a design review to start after the initial design is finished, even though the actual review only takes four hours. It also takes almost six weeks to deploy the feature to production after it has been tested.

2.3.2 Lean principles

Lean software development declares seven principles which one should always bear in mind when making decisions. Lean principles according to Poppendiecks are:

Eliminate Waste All actions that do not create value are considered waste. This could be seen as the main lean principle as the rest of the principles could be derived from eliminating waste. Eliminating waste means, for example, avoiding extra software features, avoiding partially done work and avoiding waiting.

One major problem in finding the waste is that different interest groups inside the company might have completely different value expectations for the same product [BAB+06].

Amplify Learning Amplifying learning means that the team tries to constantly improve their understanding of their area of business and their product. This can mean building feedback loops as short as possible. which includes fast testing cycles and short iterations, as well as fast customer feedback.

Decide as Late as Possible If decisions are made too early, there might not be enough data to enable wise decision making and waste could be created. If decisions are made too late, waiting happens and waste is generated again. For software developers, this can be described as Just-In-Time decisions comparable to Just-In-Time compilation.

Deliver as Fast as Possible When decisions are made as late as possible, software delivery needs to be fast. Rapid delivery simply makes it possible to make important decisions later than with less rapid delivery. Rapid delivery is achieved by utilizing the time spent at work as effectively as possible.

Empower the Team Traditionally managers tell employees what to do and how to do their work. Lean software development sees this relation in a completely different way. Workers are seen as experts in their own field and managers job is more about making it possible for the team to use their expertise the best way possible. This has a direct relation to the notion of management as waste by Poppendiecks. Less management is needed if more power (and responsibility) is given to the team.

Build Integrity in Customer's perceived integrity is about the whole experience with the product. The customer should feel that all features are relevant and that they work as they are expected. Conceptual integrity is all about understanding the business domain and creating components that work well together. The key for integrity is understanding the whole system.

See the Whole "A system is not just the sum of its parts - it is the product of their interactions." [PP03] Optimizing one part does not mean that it will improve the system. The whole system should be understood, and only in that way the interaction between all the components of the system can be optimized to create the best system possible.

2.4 Techniques to support value creation

There exists a countless amount of different ways to support value creation in software development. When depending on lean methodology, finding lightweight tools that use the professional knowledge of whole development team to link tasks to business goals could be seen to potentially improve the productivity of a company and to help it it reach its business goals. Roadmapping and three other methods to improve value-creation are introduced next.

2.4.1 Roadmapping

Roadmapping is a popular way to plan product development. The name roadmap is a metaphorical reference to more widely used concrete road maps: "Generically, 'road map' is a layout of paths or routes that exists (or could exist) in some particular geographical space." [KS01]. In software development, roadmaps are documents that links company's scientific and technological resources over a period of time [KS01, VLR02]. Typically the roadmaps summarise and communicate results of decisions that have already been made.

There exists roadmaps for wide variety of purposes for wide area of different fields. Some of them include science/research roadmaps, industry roadmaps (e.g. Roadmap for Semiconductors), technology roadmaps, product roadmaps, product–technology roadmaps and project/issue roadmaps. The types of roadmaps can be divided into four categories Science&Technology roadmaps, industry technology roadmaps, corporate or product-technology roadmaps, product/portfolio management roadmaps [KS01]. The scope of this study lies between product roadmaps and project/issue roadmaps and in corporate or product-technology roadmaps and product/portfolio management roadmaps. Furthermore, according to Kostoff and Schaller [KS01], technology roadmapping can be separated into two different approaches: expert-based and computer-based.

Computer based roadmapping is mostly restricted to retrospecive analysis of scientific research and revelopment (R&D) projects as it needs a large and relevant textual database to build a citation network based on linguistic and thematical analysis [KS01]. Aim of these roadmaps is typically to find key events that happened during a development of a certain technology or product. They are also used to communicate that existence of the department that created the map is essential.

In the expert-based approach, a team or teams of experts gather to create a roadmap or roadmaps to analyze past and to plan for the future. Systematically done roadmapping process can be time consuming and expensive and is usually executed in large corporations and organizations. A less systematic way can be beneficial for smaller entities.

Kostoff and Schaller [KS01] claim that "roadmaps should have a sufficiently flexible structure to incorporate these dynamic changes. Thus, the linkage relationships should be functional, not static, and changes inserted at any node in the roadmap network should automatically impact the other network nodes through the linked functional relationships." This quote shows how the value of the roadmap is seen to be in planning future actions in, for example, testing, HR and marketing but a static roadmap does no support prioritization and experimenting with features, as the decisions have already been made.

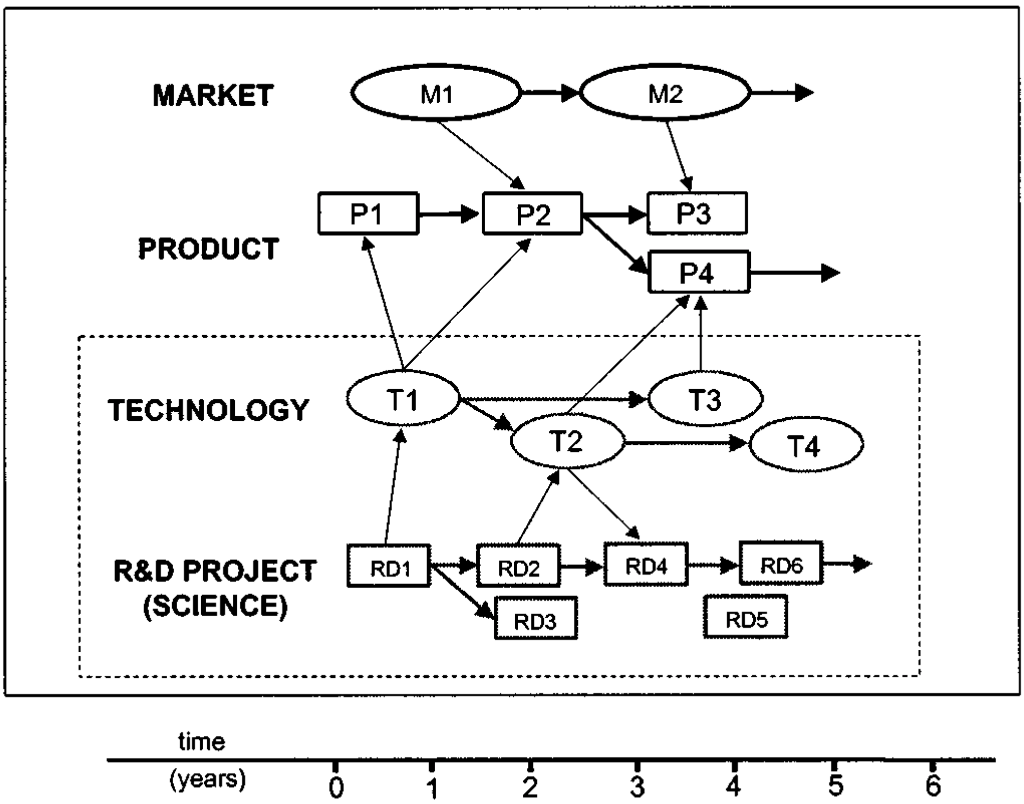

Figure 6: A generic roadmap [KS01].

Figure 6 depicts a typical product-technology roadmap. It has separate lanes for Market, Product, Technology and R&D. The R&D nodes are linked to Technology nodes which then are linked to Products which then have Market nodes linked to them. Time in the map is progressing from left to right. The map can be read so that product P2 is aimed for market M1 and that the product P2 is dependant on technology T1 which is created in R&D project RD1.

Figure 7 depicts a product management roadmap. It has similar structure to the roadmap in Figure 6 but the lanes have different focuses. There are lanes for Services, Release management, product components, platforms and resource requirements.

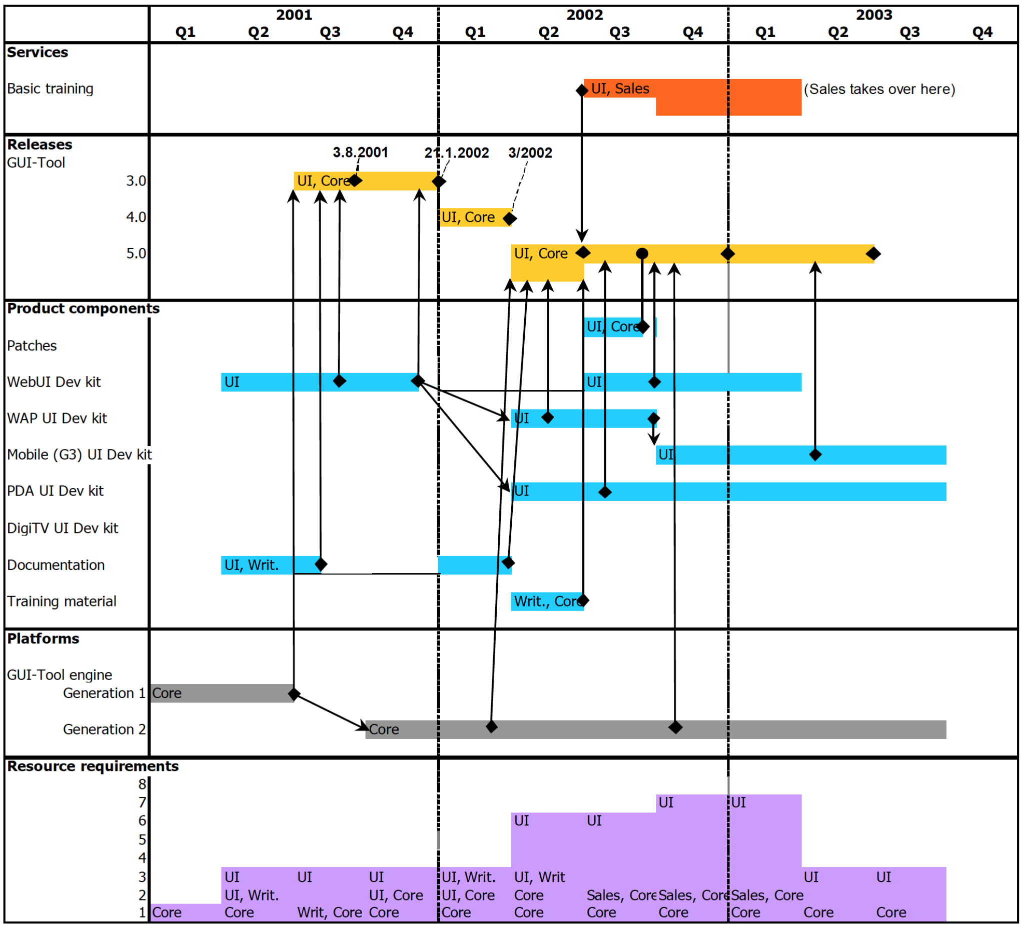

Figure 7: Product development roadmap proposal [VLR02].

As it was already noted earlier, roadmaps typically communicate the current plan to certain goal. Both of the presented roadmaps only communicate what decisions have already been made. They do not communicate what were the options from which these plans were made and they do not communicate what kind of impact the roadmap makers assume the plans to have on their customers, the product business goals and company level business goals. The downside of this approach is that the future is set. There could be other, better ways to reach the business goals the roadmap was made to help reaching.

2.4.2 Lean canvas

Canvas tools provide a simple design template that helps to guide design and analysis of business. The idea is that a simple template is fast and easy to fill and iterate over. It is also good for explaining a business idea quickly to others to build shared understanding. The first canvas to make the method popular was Business model canvas (Figure 8) which was originally presented in a book Business Model Generation [OP10]. The book gives the following definition of *business model*. "A business model describes the rationale of how an organization creates, delivers and captures value". According to the book, business model is best described as nine different building blocks: The blocks are Customer Segments, Value proposition, Channels, Customer relationships, Revenue streams, Key resources, Key activities, Key partnerships and Cost structure. Business model canvas has a section for each building block to give an overall look on a business model. Business model canvas is already used in large organizations such as IBM, Ericsson and Government Services of Canada [OP10].

Nowadays, there exists a plethora of canvas methods for business planning. All of them claim to provide some new sort of value compared to previous ones. Other canvas tools include *The Happy startup canvas* [McC15] which is intended to help businesses communicate their core values and purpose. The idea is that happy people working for what they really believe in and have passion for would make the company more successful. More product centric canvas called *Product canvas* [Pic15] is intended to support user centered design and to ease selection of the right features. It can be used to replace a traditional product backlog. *Validation board* [LSM15] is a canvas centered around validating experiments. It is trying to make entrepreneurs become faster learners.

To build a unified view of the business, a canvas tool used by lean startups called Lean canvas was selected to be evaluated. It is used to clarify and communicate a business model. Lean Canvas [Mau12a] is an adaptation of Business model canvas. The purpose of it is to crystallize and communicate a business model in a simplified way so that it can be clearly understood. This

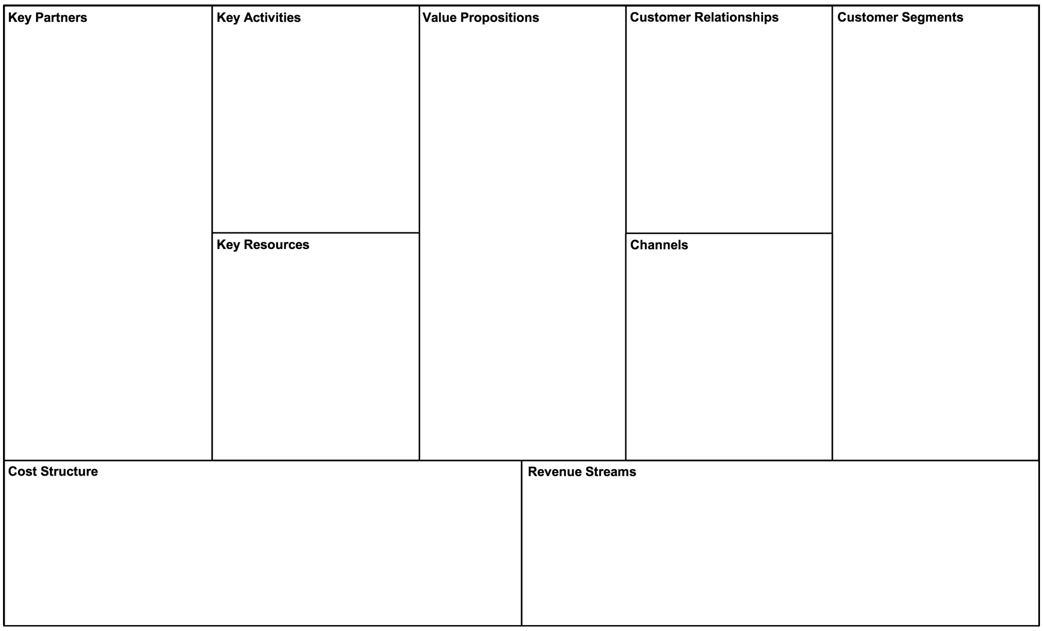

Figure 8: Business model canvas.

is achieved by using a simple template that has sections for each important area of a business model, exactly like in Business model canvas. According to the creator of Lean canvas, the canvas was designed for entrepreneurs, not consultants, customers, advisors, or investors. Lean canvas should be seen as a living artifact that should be updated from time to time.

Lean canvas helps put business people and technical personnel on the same line when discussing business and technological decisions. It therefore realizes the lean principle of empowering the team. It also helps in seeing the whole as it compresses the essentials of the domain. A well designed template for a business model forces its users to think about all the corners of a business model. This helps avoiding gaps that might not be found when not using a template. The result of planning can be distributed to all the interest groups inside a company. It can also be used to elucidate a business model, for example, to potential investors.

Main difference [Mau12b] between Lean canvas, which is presented in Figure 9 and Business model canvas (Figure 8) is that Lean canvas is more problem/solution centric while Business model canvas is focusing more on the different activities required to run a company. The differences between Business model canvas and Lean canvas are the following [Mau12b]:

First, instead of "Key partners", Lean canvas lists the top three problems of a customer. This was changed because Lean canvas is focusing on startup

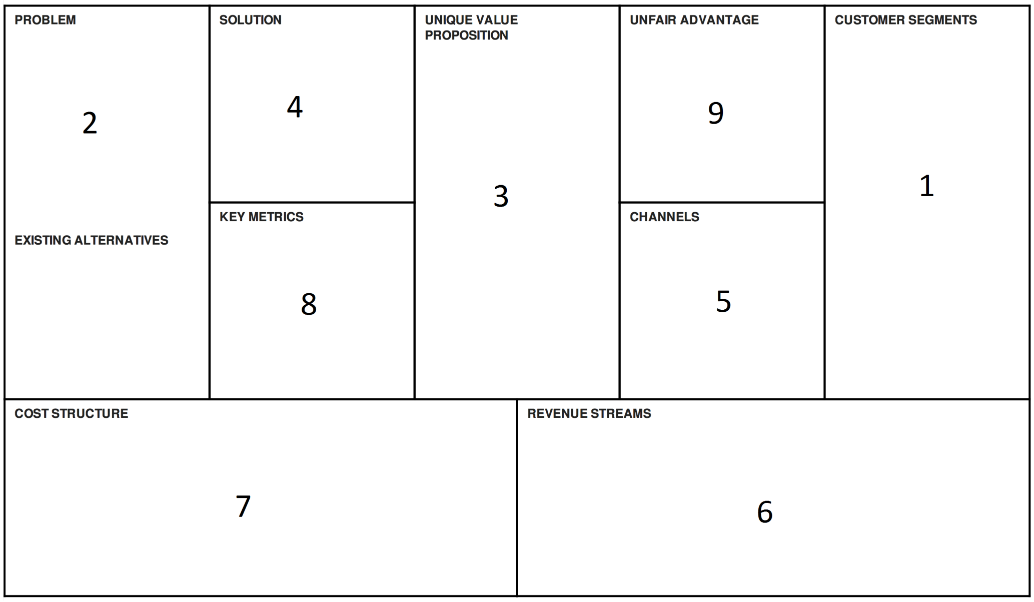

Figure 9: Lean canvas.

companies which are trying to find a problem/solution fit for their product. Partners are a critical part of a company's success from the very beginning but having a product that truly answers to someones problems is seen as more important.

Second, instead of "Key activities", Lean canvas lists "Solutions" for the former problems. The idea behind this change is that key activities should be derived from the solutions. In a software company, listing key activities would mean self-evident things such as "software development" and "customer development". Listing these kinds of activities does not create value to a software company as they are intrinsic to all software companies.

Third, instead of "Key resources", Lean canvas lists "Key metrics". Because of open-source software and cloud services today, fewer resources are needed. Even one skilled person could build and maintain a reasonably big product. Having metrics to lead development of products is seen as the most important thing in product development.

Fourth, instead of "Customer relationships", Lean canvas lists "Unfair advantage". When building new products, direct customer relationships are seen as the base of the whole start up. Interviews and customer observation should be done constantly. Unfair advantage, on the other hand, is something not everyone has. Unfair advantage is something that defends a company when others try to copy their product. Many startups have no real unfair advantage when they start up but they will develop it later if they achieve success.

2.4.3 Impact mapping

Impact mapping [Adz11, AB12] is a strategic planning technique which helps in finding links between business goals and actual product development work. Impact mapping is basically a mind mapping technique with strictly defined structure to visualize assumptions behind each deliverable. This helps in eliminating unnecessary features that have no clear link to value to back them up. This means eliminating waste. Impact mapping is advertised as a tool to help building software that makes an impact [AB12]. Impact mapping is a variation of effect mapping described in Effect-managing IT [OB07].

The process of impact mapping follows a simple format. It starts from defining a goal that should be reached. This goal can be a low level goal like “Improve product stability” or a higher level goal like “Reach one million users”. The goal should be a problem that needs to be solved, not a solution for a problem. For example, "Build a marketing platform" is not a valid goal. For commercial organizations there are goals that usually have an obvious link to money. The goal is the *Why* part of the Impact map presented in Figure 10.

Next step is to list stakeholders that could help in reaching the declared goal. These can be any kinds of actors such as users, marketing, project managers and even customers or software developers. Mapping the stakeholders helps prioritizing the needs of the most important actors first. Stakeholders are the *Who* level of Impact map.

The third step is finding out *How* the selected stakeholders could help fulfill the goal and how their behaviour should be altered. These are not software features. The focus should be on the business activities of the stakeholders such as inviting friends to use a software or handling documents faster.

Fourth and last part is finding out *What* we as team or as an organization can do to help the stakeholders fulfill the activities that could help us in reaching our goal. These are tasks. The tasks that are found using Impact map can be, for example, epics that should be implemented, features that should be implemented or tasks for one feature that should be implemented. Likewise, a task can also be a marketing campaign to run or, for example, a task to tidy up a storage room.

At this stage, all the connections between mapped items are assumptions if there exists no data to support the assumptions. Mapping assumptions this way creates a tree-like graph where each edge could be seen as an assumption. This creates assumption chains for each feature. An important notion is that

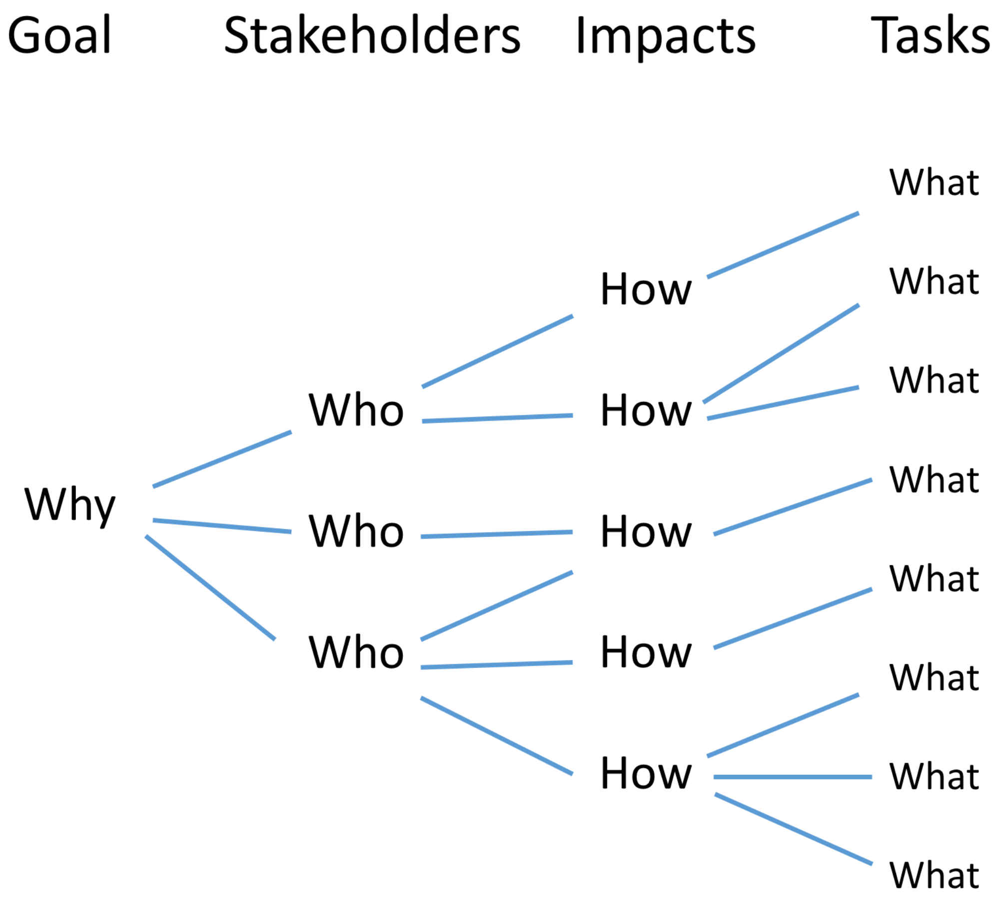

Figure 10: Impact map structure.

the top level goal also includes a hidden assumption linked to a higher level goal. For example, the goal “Reach one million users” includes an assumption that reaching one million users would improve the product’s profitability or some other important business factor. The chain clearly visualizes all the assumptions for each task. In lean terminology, this could be seen as form of amplifying learning and seeing the whole. Also, if a feature request can not be fitted into the graph, it probably should be left out of a development scope. Based on this, Adzic [AB12] suggests that if there is an existing backlog, the existing items should be reverse engineered into Impact maps.

Assumption chains also make it possible to attach metrics to each assumption if needed. This way impact map could be used to create actionable metrics. The tasks that are assumed to be the riskiest or the most important are implemented first. If after implementing a task, the metrics don’t reach a boundary value set for it, an assumption can be seen as false and the other nodes linked the action can be removed from backlog. This can prevent a lot of work spent on building features that are not needed. An example of how to implement metrics is presented in Chapter 5.

2.4.4 Lego serious play

Lego serious play (LSP) is a consultation product which was developed at the Lego Group in the late 1990s to help middle and top managers of the company in strategic planning. The method was first used only internally at Lego but it was later developed into a full commercial consultant product. A comprehensive description of how the method was born can be found in [RVS04]. In 2010, a part of LSP was made open-source. This thesis uses the open-source part of the method as a starting point for study.

Lego serious play consist of three different kinds of resources [LEG10]:

1. The Lego serious play basic principles and philosophy, upon which everything else is built
2. The Lego serious play materials - sets of specially selected Lego bricks and pieces
3. Lego serious play ”applications”- detailed roadmaps of different workshops which make use of the principles and philosophy and the materials.

The first two of these are the parts that were released to public in 2010.

LSP bases on an assumption that ”the answers are already in the room” [LEG10]. It means that external experts are not necessarily needed and that a team participating a workshop is probably already capable of solving the problems in hand. The participants are asked to ”think with their hands”, meaning that they should build the ideas they possess. This helps in clarifying concepts and demonstrating them to other participants and this way sharing understanding more effectively than they could by just telling a story or drawing diagrams.

Lego bricks make it easy for the participants of a workshop to build three dimensional structures to metaphorically resemble ideas in their minds. People, at least in western countries, are also typically already familiar with the bricks, and even if they were not, building Lego bricks is fast to learn because of the simple nature of the bricks. Figure 11 shows an example of a Lego brick metaphor for "A typical monday".

The benefit of building things, according to Lego and Schulz and Geithner [SG11], is that making something with hands makes the brain work a different

Figure 11: A typical Monday.

way which can open new perspectives. An emphasis is also on an idea that leaders do not have all the answers and they need to hear the whole team and allow each member to contribute and speak out. There is a presumption that people naturally want to contribute and to be a part of something bigger and also take ownership. Often team members are not heard and managers make decisions on their own, thus making a team work suboptimally. LSP, on the other hand, gives everyone an opportunity to express their views of a problem and this way build a shared understanding of a problem. People might have very diversified views concerning the matter, and, especially, how to solve the issue.

There are a many different problem types Lego serious play can be utilized to solve. Originally it was used to improve strategic work at the Lego Group [RVS04]. Other documented use cases include using it to elucidate project plan and different work roles in a long-term research project [SG11] and facilitating and pursuing on organizational interventions [SO08]. LSP has also been used in product development by capturing user stories and envisioning possibilities for service improvements [Swa11].

LSP workshop always has an external facilitator. The benefit of this is that the facilitator is usually someone who does not permanently work in the company. The facilitator takes care of all the arrangements required for the workshop and they also designs the workshop with a help of the customer company representative, for example, a product owner or project manager.

Lego serious play is practiced in a form of workshops. These workshops can be anything between three hours to multiple days in duration. The workshop consists of three phases that are repeated for the whole duration of the workshop. Phase 1 is called "The Challenge". During that phase the facilitator gives a challenge for the participants. Phase 2 is "Building". During that phase the participants build a Lego model that, they think, represents an answer to the challenge. The answer can be a concrete answer or it can be a metaphor that represents the answer. In Phase 3, "Sharing", everyone shares the meaning of their model. Explaining the answer can include telling a story by moving Lego parts around the structure (playing seriously) to visualize their ideas. After part 3, the facilitator poses the next challenge and the whole process repeats. The workshops start with simple challenges like ”Build a tower” to make everyone familiar with the bricks and the method and proceeds towards more challenging tasks. In the end, cooperative challenges can be presented.

One potential use case for Lego serious play would be stakeholder analysis and building visions for products. LSP powerfully visualizes stakeholder-issue interrelationships. Lego sells a separate LSP Connection Kit to aid in networking different parts of the Lego structures. LSP is not specifically designed to systematically map stakeholders but the potential of stakeholder analysis lies in utilizing the knowledge of all the participants and creating revealing metaphors of a business area. The focus of a workshop can also be on cooperative and broader subjects such as building an unified vision of a product. Even when stakeholder are not the main focus point, many aspects of them get analysed.

Lego serious play utilizes the potential of all the participants of the workshop for building a shared understanding. Schulz and Geithner studied LSP as a tool for building shared understanding for a researcher team [SG11]. They found out that the combination of a brick model and story telling added value compared to traditional forms of group meetings. This was partly due that that all the parts of the brick model are named and explained and that the participants can ask questions about the model to avoid misunderstandings. Schulz and Geithner also found out that building a physical model helped in reifying and reflecting the builder’s own understanding. They also claim that even though it could be said that the Lego model is too static, the model is only an anchor for a metaphor and it can be used to connect a story to the model to provide meaning.

Schulz and Geithner also claim that the first part of the workshop, the individual challenge solving, is about expression of personal understandings, the individual awareness of them and making their understanding explicit to the others. The cooperative part of the workshop, building a shared model, can be seen as building a shared understanding. Building a shared model forces the participants to modify their views to make them fit together. The final solution will then represent all the different views on the same issue and that way it provides a shared understanding which can be utilized for innovation. They argue that to be able to collaborate effectively, individuals should have a collective shared understanding of the domain area they are working on.

There are many ways how Lego serious play could fit into a lean software development organization. High level of shared understanding provided by Lego serious play could help in *eliminating waste.* LSP workshop is a method to improve and share the domain knowledge of the whole product team, which can *amplify learning.* An improved shared understanding and a good understanding of stakeholders created by LSP can mean better decisions and could helping to decide as Late as possible.

LSP can also help in *delivering as fast as possible.* It is debatable if an understanding of a business area has a straight correlation to reduced time spent coding a feature but if the point of view of speed of delivery is changed from features to goals, improvement might happen. Understanding the domain area better, should make it easier to choose the right features to reach a goal and this way reduce waste as fewer iterations are needed to reach the goal. Likewise, Understanding a business area could potentially correlate with better maintained code base which could reduce the actual time spent implementing the features.

LSP has very similar idea as *empowering the team.* It is the notion that the answer is already in the room. Besides being experts on their field, LSP acknowledges that in a highly educated organization is full of specialists, and those people understand a lot about matters outside their own field of expertise. LSP is aiming to harness that expertise to be used in decision making. The key for *building integrity in* is understanding the whole system (*see the whole*). LSP seems to support this principle as it effectively helps in creating a shared understanding. When the business area is understood well, the integrity of the product increases and the customer should also be able to feel it. Integrity can also be effectively built into code base in form of well maintained code which can reduce development times and that way reduce waste.

3 Case study

3.1 Research objective

The purpose of the case study is to elucidate the process of value creation in software development and to try to link company level business goals to actual product development. This study tries to improve product development planning by providing tools to visualize assumptions behind software development efforts to help in selecting the most important tasks and to improve value-creation by eliminating work that does not create value. This study evaluates three planning techniques that would support selecting the right backlog items at the right time. The items that contribute to reaching a certain business goals, for customers and for the company itself.

The study is a two step process:

Research question 1: How to use Lean canvas, Impact mapping and Lego serious play to support value creation in software product development?

The first two chapters of this thesis gave a background for the three methods that were chosen to be studied. The methods to be evaluated were chosen based on their potential to support software development planning in a context of a lean software organization. The methods are industry tested but there seems to exist no scientific literature to support them, except for studies on Lego serious play e.g. [RVS04, SO08, SG11, Swa11]. The study is about finding out whether these techniques have effect on the daily work of a product development team and how and when the methods should be used. This study is of qualitative nature.

Research question 2: How to create a value-oriented roadmap?

Based on the results of the first research question, give a proposal for how to move towards value-oriented roadmapping.

3.2 Scope and limitations

This study examines the problem of selecting backlog items from a viewpoint of a software development organization that is trying to adapt lean software development framework and evaluates techniques that could potentially aid in value creation. Focus of this study is on how a small to medium lean software organization, such as a start-up, an intrapreneurial project or a small business area of a company could benefit from the use of the evaluated planning techniques.

Because of the nature of this study, the results of this study are limited to giving impressions of how certain design techniques should be used and how they create value in software product development. Possible profit increases, customer satisfaction improvements and similar quantitative implications are outside of the scope of the study.

3.3 Context

Product development in medium and large enterprises is often done in small semi-autonomous teams concentrating on one product. The team might handle all the product related actions from planning and developing to deploying to the production environments, and taking care of customer service and software maintenance [VLR02]. It is easy to see that working in such way means that employees would need a wide range of know-how. This kind of working can be seen as a form of intrapreneurship, entrepreneurship inside an enterprise [Mau12a]. The context of this study is the planning stage of product development. This can happen in the beginning of a project, or nowadays more often, iteratively during a project.

3.4 Case companies

This study was conducted in two Finnish medium-sized software companies. The other company works solely in business-to-business (B2B) field while the other works mainly on B2B with some business-to-consumer (B2C) experiments. Software business in B2B differs from operating in a B2C environment. One big difference is that businesses are usually more logical and fact based while consumers tend to base their actions more on feelings. In B2B markets the user of a software and the person paying for it are often different people. Aspects such as usability and a great design might have a lower priority in B2B than in B2C, as in B2B the buyer often only cares about whether the product fulfills their defined needs. Although business users are starting to expect consumer level usability and user experience also from business applications.

The companies in this study are Steeri an Solita. Steeri is a medium-sized Finnish IT service company specializing in helping its customers to get more value out of customer data. Steeri offers services in customer relationship management, customer analytics, customer dialog process automation, customer data quality and customer data lifecycle management. Steeri's services are made possible by its own products and products by Salesforce, Oracle and SAP.

Steeri employs approximately 80 professionals from various fields of expertise. The company has multiple teams, each specializing in an area of customer relationship management. Product development in Steeri is done in small independent product-centric teams. A product development team participates in every stage of customer projects from planning to maintenance.

The company has recently started a process of lean transformation which means that it has made a decision to apply lean principles to all its work. The process includes monthly workshops which are used to deepen the understanding of lean principles. Lean agents are also named among the employees. Their job is to establish the lean principles in everyday work in the company. This study started as a part of the lean transformation. The author or this paper was one of the agents in the beginning of the study.

Solita is a digital business consulting and services company. They develop new business and digital services for corporate- and public sector customers. They combine business processes, contents and technology in a revolutionary manner for the customers' benefit. They consider themselves to be the customer's guide on the road to digitalization and change.

Both of the companies have a typical structure of an IT company where under a top management group exists multiple business areas that provide services for different customer needs. Some of the business areas base their services on self-built products while others rely on partner-, or open-source products. The focus of this thesis lies in these self-built software products.

The companies have identified several issues in their product development work. Operating in B2B makes a difference in how products can be developed. Businesses usually have strict privacy rules concerning their data. This often eliminates real usage-based statistics from product development decision making. Also continuous experimentation can be hard or impossible because businesses often want their tools to be stable and identical for each user [Ris14]. There are many ways to try to overcome this. One very simple method is user reviews but it also has its difficulties. User reviews in B2B require a permission and desire of a customer, who might not want to spend their expensive time to improve the already working product. Finding important improvements for a piece of software is hard if there are obstacles in studying the behaviour of a user. A major one being that the value of a current and future work can not be fully understood. This makes product development projects blind about their future direction and it also has an effect on the employee motivation.

Major challenges in enabling continuous delivery and continuous experimentation have also been identified [Ris14]. Furthermore, different types of issues include a lack of understanding of the companies products in another of the companies' sales team which reduces the sales success rate. A smaller but still very important problem is that planning meetings often get heavily out of topic. This reduces their effectiveness and time gets wasted.

3.5 Research method

The research method chosen for this study is action-oriented research which is a form of case study where the researcher participates in the research as an agent for change. This is quite different from a typical case study method where the researcher has more a role of an observer. The method is most often used in private industry and organizational development [Sma95].

The research method was chosen because there existed a need to try new ways to steer product development into right direction. The typical observing case study could not give hands-on knowledge of the methods. In action-oriented research the focus and the methodology can also possibly change during the research when new discoveries are made. The idea is that results of the study can be directly be used in benefit of the organization. Because of the small sample size and the nature of the method, the results of action-oriented research are usually qualitative. Because of this nature, style of the study is more narrative and descriptive than the more quantitative types of studies.

3.6 Subject and case selection

Subject of the study was originally to develop a new way to create value-oriented roadmaps. During the study it was noted that such a roadmap would not be very effective if even the value of a single software feature was not known. Because the current knowledge of value creation in the case projects was not on very high level, the focus gradually moved over to be more focused on finding software tasks that support achieving a certain goal.

The four case product teams were selected to represent different types of software projects to expand the sample size of inherently qualitative study. Two of the projects selected are intrapreneurship-like products, one product is intended only for internal use of a company and one product is Qlma, a young non-profit open-source startup sponsored by another of the companies.

3.7 Data collection

This study consists of studying the tested methods in multiple workshops for multiple product development teams for all the studied design methods, except for Lego serious play, for which, only two workshops were arranged. To get the overall tone of how the participants felt about the workshops, feedback was asked after every workshop. Also notions of how the participants reacted during the workshops were collected. Concrete actions following the workshops were also collected.

All the Lean canvas workshops used a commercial online tool [Spa15] to fill the canvas. The Impact maps were drawn on a whiteboard or on flip chart. The whiteboard maps were photographed and the flip chart maps collected. The Lego serious play workshop session artifacts were photographed. The plan was to follow the workshops lengths suggested by the original authors of the methods.

3.8 Execution

The study took place in a span of one and half a year and included numerous workshops. All the studied techniques were evaluated in their natural environment as a part of the product development process. All the methods were tested in a workshop-like manner. In some cases, all the participants were asked to spend around half an hour to study the methods before the first workshops. In some cases a short introduction to the idea behind the technique was given at the start of the first workshop, then the facilitator of the workshop led the discussion and helped in keeping the workshop in a proper form.

The facilitator for Lean canvas and Impact mapping workshops was the author of this thesis. The Lego serious play workshops were facilitated by two researchers specializing in service design.

3.8.1 Lean canvas workshop

For each product, multiple workshops were held to make the participants familiar with the method and to get an idea of how well the method works in actual use. There were three Lean canvas workshops for the Steeri's product A. They were done with a product owner, one project manager, and one software designer. The sessions were held in Finnish and the canvas was filled in Finnish. There were two canvas workshops for Steeri's product B with a product owner, product technical lead, a project manager and a software designer, and one session with only software developers present. The software developers only session was held in English, as one of the developers was a non-Finnish speaker. The other sessions were held in Finnish. Lean canvas was filled in English in all the workshops. The one product B session that was held with only product B was held with developers that did not participate in the sessions before, to elicit designers' knowledge compared to more heterogeneous group.

One Lean canvas workshops was held for Solita's internal product. It was done with two developers only. The same workshop included an immediate Impact mapping session. The workshops for Qlma were done with product owner/lead developer, a marketing director and a service designer.

In [Mau12a] Maurya states that a Lean canvas should be filled in 15 minutes. The idea is that this allows rapid experimenting with different kinds of business models. We did not strictly follow that instruction in all the workshop. Mainly because some times the participants were not familiar with the method yet and sometimes the discussion that arouse was seen as too valuable to cut it short.

The predefined order of sections to lead discussion was followed but the discussion sometimes returned to a previously discussed sections if something new and interesting came up. For example there were difficulties finding a precise high-level concept for a product but a couple of sections later one participant had a surprising intertextual idea and the high-level concept ended up being "One master to rule (them) all customer data.".

3.8.2 Impact mapping workshops

At Steeri, Impact mapping for products was done with product A team. Participants of the product A session were a product owner, the whole product A developer team and a newly recruited project manager who has a background in being a software developer. The session lasted for one hour. The participants were asked to read a small introduction to Impact mapping before the session. An other workshop was done with one Steeri business area team with a research purpose of getting understanding of how well the method works with and without coaching. The workshop also had a real business goal not related to the study set. The participants were not told about the research agenda of the workshopå. The participants were given a short introduction to Lean canvas method and then they were divided into two groups. The other group got support to create their impact map and the other did their work without support. Impact mapping was also tested with human resources (HR) related goals with administrative people and with another business goal with lean agents in a lean agent meeting.

A workshop for Solita's internal product was held immediately after the Lean canvas workshop for the same product was done. The session started with a brief introduction to the subject and then a goal for the Impact map was chosen. The author of this thesis worked as a discussion leader and drew the map the way the team felt was right.

The Impact mapping sessions for Qlma were done with product owner/lead developer. The first session tested how well the unique value proposition from Lean canvas works as a goal for an Impact map when creating minimum viable product. The second workshop tested using the solution section of a Lean canvas as a goal.

3.8.3 Lego serious play workshops

As the Lego serious play already has more research to support its use, it was only tested in two workshops that were held for Steeri products. The workshops were facilitated by two researchers who specialize in service design. They had self learned the LSP method and were not LSP certified professionals. They followed the structure presented in [LEG10].

The participants of the workshops included the product owner, a team leader, developers, salesperson and product support person. The duration of the workshops was 6 hours. There was a lunch break and two short coffee breaks during the sessions. The goal of both of the workshop was "To create a shared vision of the Product".

The workshop consisted of the following challenges:

1. Build a tower
2. Build your ideal neighbour
3. Build your typical Monday
4. Build a representation of a user of the Product.
5. Build an important challenge of your user
6. Build a solution to that challenge.
7. Build your vision of the future 'Product' in an ideal world.
8. Now, as a team... build an answer to the previous challenge

All the models built during the workshop were photographed with a description note next to the model. After the workshops, the facilitators produced a booklet which summarized the workshops from their point of view. The booklet was distributed to all the participants of the workshop.

3.9 Analysis procedures

The results of the workshops were collected and observed thorough the research. New ways to improve the workshops were introduced when the methods got understood better. The artifacts from workshops were compared to others to find similarities and to see how the quality of the artifacts increased when the participants became more familiar with the methods. The actions that each workshop caused in the product development, were listed when they happened. The proposition for value-oriented roadmapping approach is based on the knowledge and experience gathered during the research.

3.10 Validity procedures

To avoid bias towards any single group, multiple teams were studied. The research was conducted on multiple projects with different kinds of products. The projects included two commercial products, one internal product and one non-profit open-source project. This allows data triangulation. With most of the teams, the research was conducted for a prolonged time period. Depending on the project, this period ranged from one month to one year. Because of the nature of the study, the results of the study are not statistically significant but generalizations can be drawn to similar situations. A care was taken to not generalize the results too much. The study should be repeatable in similar organizations with similar teams.

4 Findings

4.1 Defining the business model with Lean canvas

Lean canvas proved to be a promising way to lead discussion about the business model of a product. The sections of the canvas are numbered in a way that prepares the discussion of the next section. This seems to improve the amount of focus participants have in the discussion. Discussing the customer segment first prepares for discussing the problems the product aims to solve. In a way it creates a frame for rest of the business model, as aiming to solve problems of a different customer groups can lead to different kind of business model and product even though the initial idea of the product was the same. The customer-centric approach leads to a perspective where the discussion happens from quite a problem- and value centric view. Efforts are put into comprehending the problems the customer is facing.

The workshops revealed that some concepts are very hard to define even if people have been working with the product for many years. For example, a task as simple as describing the main problems of the customers was not unambiguous. Also cost- and revenue structure were more unclear than anyone had expected before the workshops. Finding the unique value proposition also proved to be quite hard, the reason being mainly that each person had their own view of what is valuable. This often stirred discussion where each attendant was trying to understand each others point of view and try to adapt it to their own.

What seems to happen is that when taking the developers to the planning workshops where they are usually not invited, the people from different roles start to understand each others perspective better. The developers had to consider the customer perspective more than they are used to and the less technical people had to consider technical feasibility of their ideas. It often happened that a developer started to move the discussion towards very precise technical details when the discussion even touched a notion of a feature. Having a manager present prevented these discussions going too deep too early. On the other hand, less technical people had ideas that would not be feasible with current resources and the developers then gave them a reality check. Sometimes it happened that a manager did not know that something he was suggesting as a new idea, was already possible with the current product.

Lean canvas seems to work differently for different types of problem scopes and products of different maturity levels. As it was earlier noted, Maurya states in [Mau12a] that the canvas should be filled in fifteen minutes. Time spent filling a Lean canvas varied from twenty minutes to one and half a hour. The longest workshops happened with products that already were in production. This is interesting as it shows how the team might know exactly what the product does but they can not always describe why the features are valuable. The shortest time spent on the Lean canvas was with the Solita's internal product. At the time of first workshop for the product, about half of the expected features of their minimum viable product were implemented. The scope of the product was still quite small which seems to imply that understanding value of a small or young product is considerably easier.

Longest durations of filling a single a Lean canvas happened with the products that were the oldest and most mature. They took around an hour and a half to complete. Spending more time to fill a canvas might not be a bad thing. Lean canvas is aimed at startups that are still trying to find their business model. Therefore it seems reasonable that for a more mature product, that already have a huge set of features, filling the canvas from scratch would take more time. On the more mature products, there exists a wast knowledge about the product and accommodating it into the format of Lean canvas requires a lot of discussion. The longer time spent allows participants to truly concentrate on the current status of the product and to find precise bullet points for each section. Most of the time was therefore spent discussing the product and sharing each other views on the current status of each part of the business model.

A long time was also spent creating a canvas for Qlma, but that seemed to happen for a different reason. Qlma is the youngest of the products in the study and it is the that most resembles a startup. Therefore it is the product that is closest to the ones Maurya is aiming the Lean canvas for. The initial idea for the workshop was to tighten their vision about the product that was still in its infancy. As the product was still very young, it was not a surprise when the workshop soon derailed into questioning some very basic concepts of the product that would be vital for its success. One of the participants made the same notion as Maurya in [Mau12a], that multiple canvases would have to be created to elicit how the product changes when these main concepts change. These findings seem to suggest that Lean canvas is fit for products of different maturity level but the way it is used should be accommodated for the product needs.

Besides providing value for the product development efforts, Lean canvas was found to be useful also for sales purposes. In another of the companies there existed a known lack of understanding of company's products in the sales team. Lean canvas came to help when the sales director of another of the companies pointed out that the Unique value proposition-, Unfair advantage- and Problem-solution fit- sections of the canvas were very useful for sales. They provide a compact definition of why a customer should be interested about the product. Later a white paper about the product was written based on the Lean canvas to help a customer make a decision whether the product was an answer to their problems.

Putting the Lean canvas into practice did not need very high efforts as all the projects themselves knew that their knowledge of their own business could be improved. Some projects were also aware that their product lacked a clear long term vision. This made the workshops very welcome for most people in the projects. Only one manager was reluctant to join a workshop because he could not understand why his presence would be of benefit. In another of the companies, a few weeks after creating the first canvases, other teams in the company started to ask the researcher to help them have Lean canvas workshops. After good experiences, one product team started quarterly updating Lean canvas with whole team present to discuss the status of the product. This seems to imply that there really is a need for this kind of method.

Especially developers commented that the workshops gave them a clear picture of what their product is about and why they are working on it. Managers typically commented that they feel that the meetings are valuable and workshops like these should be kept regularly. Only one manager commented that he did not find any use for Lean canvas workshops, but he did no consider them waste either. From a lean perspective, the feedback from the workshops looks promising. Improving shared understanding empowers the team and the better understanding of the business model and the problem domain should improve their ability to build integrity into the product. Understanding what effect their work has, is a big motivational factor. The workshops seem to validate the claim that Lean canvas improves understanding of the business model and having a better shared understanding is clearly of benefit for the product.

4.2 Finding value-creating tasks with Impact mapping

Similarly to Lean canvas, Impact mapping was very good at leading the discussion of a workshop. One level of the map was mapped all at once before moving to the next level. The well defined structure of the map was very effective at keeping the discussion on topic. Impact mapping session lengths varied from half an hour to one hour. For simple goals, the sessions could be kept very short, while for bigger scopes one hour was usually not enough. The session lengths were limited to around one hour to maintain focus and to let ideas mature between sessions. Based on how the participants were able to focus, the maximum length of session was not tested but supposedly it is not very much above one hour, without breaks at least.

The way the goal is expressed makes a huge difference in the mapping results. When the goal is clear, for example, "improve stability of the product" or "reach one million users", it is easy to map the stakeholders. In these cases, for each stakeholder, the potential ways they could make impact on the goal were also found in a straightforward manner, and finally the actionable tasks that the development team could start implementing were easy to add.

When the goal was not clear, for example, the scope of the goal was too big, it was really burdensome to find tasks that have a clear links to the goal. Oftentimes the reason for difficulty of finding the tasks was actually the difficulty of expressing an idea in written form that fits the goal description. The most difficult aspect of the map seemed to be to understand the difference between the How and What levels. As the How is how one stakeholder group can help the team to reach the goal and the What part is about what can the team can do to help them do the How action. It is very easy to mix these two levels and considerable amount of time is spent forming the right way to express an idea. During almost all the workshops, finding the difference was an issue.

Based on the difficulty of understanding the difference between the How and What levels, it seems that Impact mapping needs some practice and/or active coaching to be effective. One session was held to test this assumption. The experiment was to evaluate the quality of the tasks that are mapped by two different fairly heterogeneous groups. The quality in this context being how actionable the tasks are. The tasks should be so clear that it is easy to simply hand them over to someone to execute. This does not mean that a tasks has to be small but that the tasks should have a clear intention so that work on it can be started right away. A group was given an introduction to Lean canvas and then the group was split half. Both groups had to create a canvas for the same goal. Another of the groups got active help from the researcher. The team with no coaching built a much bigger map with more What items but the quality of the items was lower. When there was no coaching, the tasks often were of form "improve this action" while when the workshop facilitator was helping with the process the tasks were of form "Do this action" or they were clearly features or epics.

Impact mapping was tested with different group sizes. The biggest workshop included six participants and a workshop leader. That group size was found to be too big for optimal team work but the resulting impact map was of good quality. When there was that many attendants, the quieter ones could not always get their ideas out. With smaller groups the discussion was more dynamic and reacting to each others ideas felt more natural. Around three person seem to be a good group size for impact mapping to be most effective. If the participants are very active or, for example, high level substance knowledge is required, a bigger group size could be beneficial. Also the bigger group size increases the shared understanding of the team but on the other hand the time spent in the session reduces the time that can be used for other tasks. Having a whole team Impact mapping be seen to support lean principle of empowering the team. Here, a balance between the maximum shared understanding and doing other work has to be found.

With Qlma, Impact mapping was used to define the features of a minimum viable product. To build on the knowledge from Lean canvas session, the Unique value proposition from Lean canvas was set as the Impact map goal. Even though the unique value proposition was first thought to be an excellent goal, it was surprisingly hard to find proper tasks with. The reason is presumably, the fact that the unique value proposition is actually a What type of goal, it is a task of very big scope, a product. It is something that should maybe exist on the lowest level of company level Impact map that maps ways to reach a company level business goal. Still, having the unique value proposition as a goal seemed to give a great context for finding valuable tasks.

Using the experience from the previous session, a new session was held that had the solutions from Lean canvas as goals for three separate Impact maps. After mapping only one goal, it was noticed that the previously tested Unique value proposition was better fit for a goal. The session had similar difficulties to the one with Unique value proposition as the goal but the How and What levels felt even harder to define in a proper form. It was later noted that the solution parts are exactly the kind of goals what Adzic advices to avoid in [AB12]. With such goals, it was not possible to build an effective Impact map

Based on this experience, Impact mapping is not optimal for finding the features of a minimum lovable product. But if the goals are, for example, improvements over existing products, such as, "Do something 30% faster", Impact mapping is be very effective. Still, when the goal was not optimal, Impact mapping provided a structured view to different use cases of the product, and when there existed the shared knowledge of the Lean canvas session, the structured view seemed to visualize the assumptions behind each deliverable.

Adzic suggests in [AB12] that features that can not be fitted to any Impact map should be removed and should not be implemented. This eases the task of prioritizing tasks. The impact maps allowed deriving of real tasks systematically from the resulting map. Overall, Impact mapping was found to be an effective way to find actionable tasks. It made the assumptions of how the suggested features would create value very clear. Impact mapping was seen as lightweight and promising technique that would be utilized more later.

4.3 Creating shared understanding with Lego serious play

The external facilitator seems to improve participation in a Lego serious play workshop. The attendants stayed focused on the tasks and even the more quiet ones were eagerly participating. Having an external facilitator present also made all the attendants equal. When the product owner was just one of the participants, his ideas were judged as equal to everyone else. Besides the external facilitator, the Lego bricks as tools to present ideas seemed to activate the participants. Most people are familiar with the bricks and can very quickly create structures out of them. Compared to for example drawing or clay, Lego bricks have the advantage of the participant not needing to be skillful in those more traditional handcraft methods. Also compared to clay, Lego bricks are quite easy to clean after the workshop ends.

The workshops brought forth multiple silently acknowledged issues that were usually not discussed with the team. An interesting example is how during the sharing turn of the task "Build an important challenge of your user", two developers who were sitting at opposite sides of the table, considered exactly the same challenge important. Even the Lego metaphor build to represent it was almost identical. Both developers also introduced almost identical solution for the problem. Overall many problems in the products and in the development process were found. The findings from the workshops were utilized when deciding where the focus of the product development should be in future. For another of the products, a prototype project was launched to test if the new direction was feasible.

Apart from finding issues, the workshops proved to be effective in finding hidden stakeholder groups. Finding stakeholders happened even thought the emphasis of the workshops was not on finding the stakeholders. The different stakeholder groups became very clear during the last challenge where an unified vision of the product had to be created. In the final unified model, different stakeholder groups and their relation to the business got presented. What seems to happen is that the stakeholders' power gets represented with easily understood metaphors. For example, customers could be sheep and CEO could be a skeleton with a whip standing on a pile of coins. This can be seen in Figure 12, which presents the final Lego model of a workshop with another of the teams. Likewise, participants could refer to different stakeholders during their sharing turn of the preceding challenges. In another of the workshops, one new stakeholder group was found. Thoughts of how this group could be taken into account during product development were expressed.

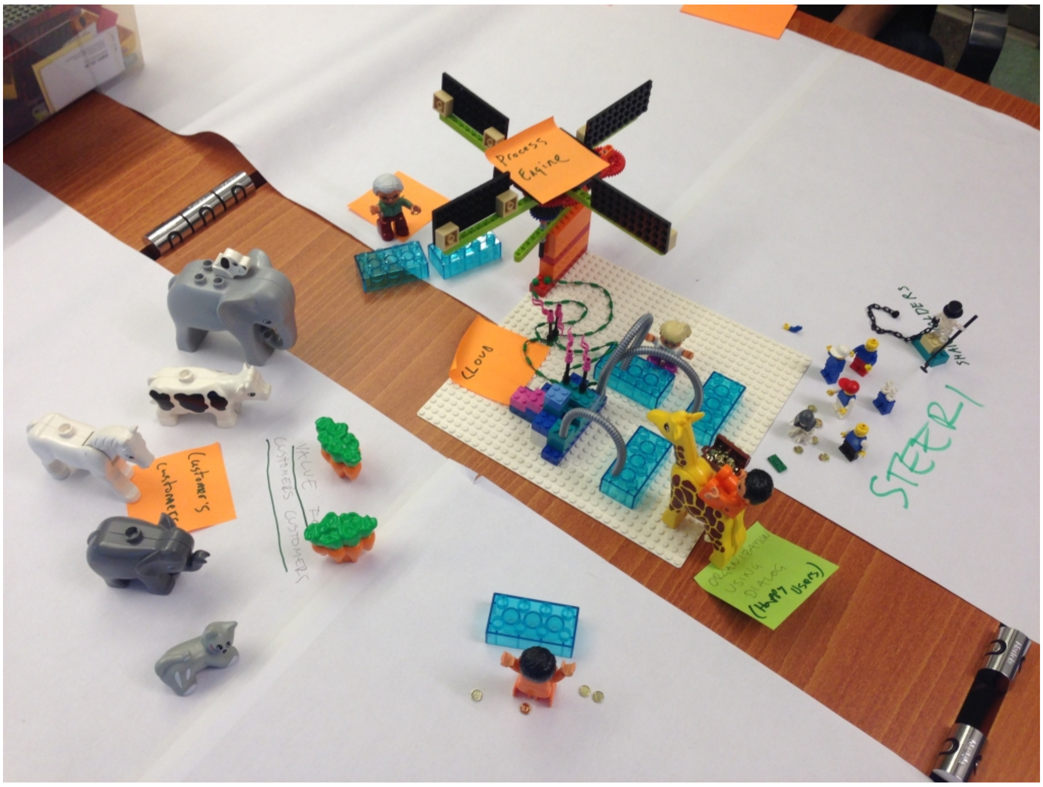

Figure 12: The shared vision of the future.

Lego serious play bases on the presumption that a participant builds the idea he or she has and this way makes his or her idea concrete. This gives them opportunity to explain themselves with more than just words. The concept and environment of the workshop is very playful which also makes participants very open to discussion. Even opinions that participants would not dare to say elsewhere can be spoken out. This helps in sharing the understanding of the products, the issues, the stakeholders and the business in overall. The Lego model, storytelling and the external facilitator seem to create a safe environment where there forms a high amount of shared understanding. This was also noted by the participants of the workshops. They all felt that after the workshop they understand better what their product is about and where it is going towards.

Participants were mostly excited about the new method. One participant of LSP workshop stated after the workshop that he did not like the idea of using Legos at all. All the others gave very positive feedback. The Lego bricks also gathered a lot of interest inside the company. An another workshop for another product development team was ordered immediately after the first workshop.

LSP and Lean canvas both seem to help in building shared understanding, but the kind of understanding they build differs in some ways. The Lean canvas stays on quite high abstraction level as it is trying to describe the business model of a the product. It has sections for problems, solutions and existing alternatives for the product, but apart from the existing alternatives, the problem descriptions have to be quite abstract as the canvas forces the user to fit the answers into a tight space. LSP on the other hand does not force the users to focus on certain level of questions. It is a more free-form method with a high level of visualization and the level of detail depends on restrictions given in the challenges. Depending on the way challenges are described, the understanding built using the method can be about very fine details or about very abstract visions. In the studied workshops, understanding was built from smaller issues towards high level vision. With this kind of approach, the low level challenges were not mapped very systematically and only the ones that developers felt were most important were found. The benefit of this approach was that it was teaching the participants to use the method and to think with metaphors which seemed to be helpful when the final unified vision of the product was built.

LSP is not the best choice when systematical analysis must be made but it works great when an overview of a domain should be created. Besides the way LSP was used as a tool to find unified vision for a product, an example use case for LSP could be a project kick-off to create a mutual understanding of the project roles and project goals. As the method is quite time consuming, it can not be used as often as Lean canvas and Impact mapping. It seems that if the results of the LSP workshop could be documented well enough, improvement goals could potentially be used as goals for Impact mapping. The workshops take whole day, which means that in product development a workshop, for example, once a year to improve the unified vision of a product could be beneficial. When carefully choosing the right time and reason to use LSP, it could be considered to fit the lean software development organization.

4.4 Challenges

Introducing all the methods depend heavily on a competent workshop leader. In Lean canvas it is essential to compress the ideas to as small units as possible, and without a coach, it seems to not happen that well. The other side of coaching is that sometimes it was hard to find anything to say about a certain part of the canvas. In those cases, a competent discussion leader can lead the discussion to right direction by asking the correct questions. In Impact mapping, the most difficult part was for the participants to comprehend what kind of ideas were expected to each level of the map. Here the workshop leader was able to aid in forming the kind of answers that communicate value.

One challenge of Lego serious play is that there is a risk of moving away from the original task to focus too much on the Lego model, not the challenge in hand. Lego as a toy seems to be so engaging that participants of the workshop would often start focusing on finding certain special Lego bricks instead of focusing on the metaphor they are trying to express. LSP workshops can also take very long to complete, because of because of this, care should be taken to not use the method too often.

There exists no predefined way to document the actions from a Lego serious play workshop. Having tested way to document and describe the findings of a workshop would be helpful. Now it is left for the participants of a manager to choose what findings are important. The lack of instructions on documenting the process results could potentially lead to a situation where no concrete actions or results can be derived from the workshop. As only a part of Lego serious play is open-source, this lack of documentation could be solved when using the commercial version of Lego serious play.

5 Discussion

5.1 Metrics

Impact made it easier to create value through linking all actions to business goals. It did not explicitly offer actionable metrics which would move product development process towards continuous delivery and continuous experimentation, which would be beneficial to support the current trend in software development [Ris14]. To ensure that the software development is going towards the right direction and is creating value, a metrics system to support the development process can be implemented on top of Impact mapping.

Having a visualization of the expected value goal makes it easy to find the assumptions that could be measured. Important notion is how all the links between nodes of the map are assumptions. It is not always know if the chosen stakeholders are the ones who can make an impact on the goal. It is also an assumption that the chosen action that a stakeholder could do, would make an impact on the goal. It is also an assumption that the task of Why level would make an impact on the action that stakeholder could do. The assumptions are presented in Figure 13.

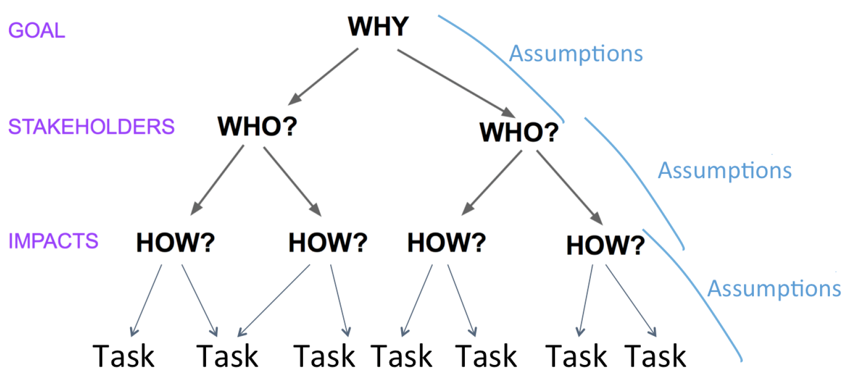

Figure 13: An impact map with assumptions hilighted.

When a task is selected to be implemented, meters should be attached to each assumption on the route from the task to the goal as is shown in Figure 14. Each meter should then have an interpretation model attached to them which is shown in Figure 15.

A way to use metrics would be to use them to eliminate subtrees or subsections of the map. The riskiest and most important assumptions should be validated first. For example, if an Impact map goal is to reduce customer

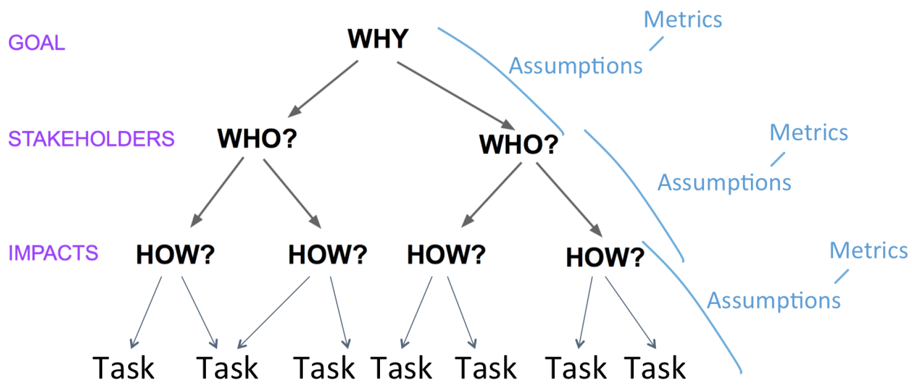

Figure 14: An impact map with with the placement for metrics hilighted.

churn, an assumption could be that by implementing a task to speed up customer service ticket handling rate the company would reduce customer churn. One of the features mapped, assumed to improve the customer service work speed, can then be implemented. Then it should be measured if the implemented feature is used and if the use of the feature really speeds up the the ticket handling rate. If it is noticed that the customer service personnel are faster at the task after the new feature is implemented, but if the improvement on ticket handling rate does not have an impact on reducing churn, features that are linked to improving the ticket handling rate should be eliminated. Instead, maybe focus should be on other actions of the customer service than ticket handling rate or maybe the customer service is already as good as it should be and the focus should be moved to other stakeholders.

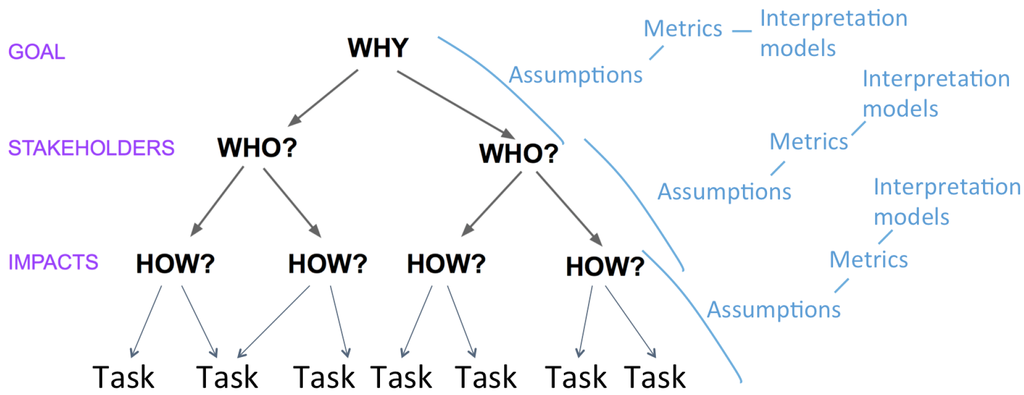

Figure 15: Interpretation models for metrics.

5.2 An approach for value-oriented roadmapping

As the methods tested in the case study proved to be effective in value-creation, a more structured way to use the methods will now be presented. When creating this proposition for value-oriented roadmapping, several requirements were considered. The first requirement was that it should visualize the assumptions behind each software development task and show the value linkage all the way from the task implementation to company level vision. Second one was to show alternatives for each task to allow experimentation. Third one was that the roadmapping method should not be centered only around software, non software actions should be considered equal. Also the method should adapt to appropriate planning levels.

The approach for value-oriented roadmapping is aimed for companies with similar company structure as the studied companies. It is inspired by GQM$^+$Strategies [BLR$^+$10], which is a method for measurement-based decision-making built on top of well know Goal Question Metric Approach [VGD94]. It provides a way to link business goals to the actual software development efforts and to measure the goals. Knowing how each activity is linked to a business goal can potentially increase the productivity of the company and decrease risks. This is possible because actions will have known assumptions behind them which then can be validated. If an assumption is proven wrong, the tasks related to it can be removed from the backlog. Proving assumptions wrong helps in quickly experimenting with a wide field of options. Having visibility of these assumptions would therefore improve the chances of finding the right efforts and investing in them.

GQM$^+$Strategies is consists of two different components. Goal$^+$Strategies elements and GQM graph. The strategies element contains a Goal which is realized by a set of Strategies. For each strategies element there exists a goal in the GQM graph which is made measurable through questions that are answered through metrics which have interpretation models attached to them. Each strategy element can lead to a new set of strategy elements. This creates a tree of strategy elements with higher level goals on top and lower level elements in the bottom. Figure 16 visualizes the GQM$^+$Strategies method.

The Strategies-elements of GQM$^+$Strategies are substituted with the use of Lean canvas and Lego serious play. Instead of substituting the elements, it would also be possible to use these methods to help creating the Strategies elements. Instead of the GQM Graph element, Impact maps are used to create adaptive value-graphs of multiple planning levels. To give structure for the proposition, a product development roadmapping checklist by Vähäniitty et al. [VLR02] is used as a base for the proposition. The checklist is presented

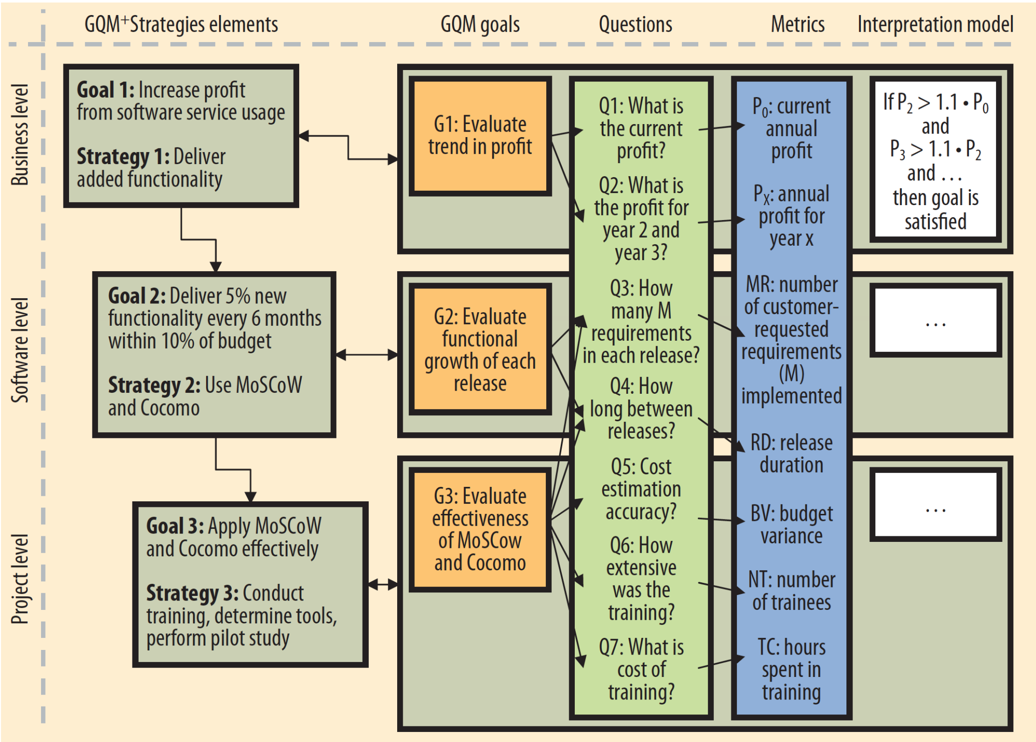

Figure 16: GQM+Strategies model [BLR+10].

in Figure 17. The approach is focused on roadmapping a product in small and medium companies and should fit into a lean software development company.

Step	Objective
Define strategic mission and vision. Outline product vision.	Clarify and communicate what business the company is in
Scan the environment	Choose position and focus, assess the realism of the product vision and examine what technologies should be used
Revise and distil the product vision as product roadmaps.	Establish release cycle, objectives for releases and allocate resources. Record decision rationale with business requirements
Estimate product life cycle and evaluate the mix of development efforts planned	Check sanity. Assess whether the planned development is parallel to the product vision

Figure 17: Checklist for creating roadmap [VLR02].

The checklist's first step is to "Define strategic mission and vision. Outline product vision.". The objective is to "Clarify and communicate what business the company is in". The mission and vision of a company is centered around its business model, which means that Lean canvas is great tool for clarifying the mission and vision. The unique value proposition in Lean canvas could be seen as the mission and vision for the product. Lego serious play can be used to deepen the understanding of a common vision.

The process of roadmapping can be started with a Lean canvas workshop for the company management group. This vision can further be deepened with Lego serious play. After the company mission and vision is clear, an Impact map with the company vision as a goal can be created. The Impact map should list, at least, all the business areas of the company as stakeholders. Then all the business areas and other stakeholder actions that could help in reaching the business goal are mapped. The final What level of the map is to map what the company could do to help the stakeholder groups do that something that would help the company reach its vision.

Next, each business area can have their own Lean canvas workshop to understand what the business model of their specific product or business area is. Like in the company level vision, Lego serious play can be utilized to improve the shared understanding. After the vision and goals are clear, Impact mapping workshops can be utilized to find the concrete actions to reach the chosen goal. Mapped tasks could be product development related, while others can be concrete actions to be executed by non-product development related stakeholders.

The second step is to "Scan the environment". The objective is to "Choose position and focus, assess the realism of the product vision and examine what technologies should be used". This is a business area specific task and can be supported with Lean canvas which makes it possible to iterate and try different business and value models really fast, while Lego serious play could be used to make a realism check for the product vision. Impact mapping could be used to find the best technologies.

The third step is to "Revise and distill the product vision as product roadmaps." The objective is to "Establish release cycle, objectives for releases and allocate resources. Record decision rationale with business requirements". If the product is already in production, Impact mapping should be used to create value-oriented backlogs. If the product is not yet in production or is a completely new product, it could be helpful to build the *minimum viable product* (MVP) using the Unique value proposition from Lean canvas as the value goal in Impact map. This was tested during the case study but proved to not be a perfect solution. Still, it provided a great context for finding the requirements for MVP. Another potential, not yet tested, way would be to use the top three problems of customers from Lean canvas as goals for three Impact maps.

The final step in the checklist is to "Estimate product lifecycle and evaluate the mix of development efforts planned". The objective is to "Check sanity. Asses whether the planned development is parallel to the product vision". This is a stage where the features gathered using Impact mapping are checked against the business model that was built on earlier stages. At this step, there should exist Impact maps of at least two different abstraction levels. Higher level is the company vision level where each project is connected to the company vision. The lower level is the business area or product development level connecting software development tasks to chosen development goals. Figure 18 represents these two levels of mapping and shows how the product level map is connected to company level map. Other levels could also be added to support the base levels. One example could be a customer project level maps where customer's business goals would be used as Impact map goals. Project- and product level level mappings are closest to the way Adzic suggest Impact mapping should be used [AB12].

Roadmaps typically consists of spatial and temporal dimensions [KS01]. The spatial dimension in this context meaning the differentiation between, for example, product, technology and R&D projects and temporal dimension meaning that the roadmap gives approximations of when each feature or project is finished. This allows retro- and prospective studying of product development. The proposed approach for roadmapping does not have an opinion about these dimensions. If the spatial and temporal dimensions are needed, for example, for planning release schedules and HR needs, they should be derived from the Impact maps

When a task is selected to be implemented, the spatial and temporal plans can be created. One way to create them would be to use business value models, such as, the business value model by Heidenberg et al. [HWMP12] which was presented in Chapter two. A temporal value dimension could be added in a form of work estimates to the existing value model and this model could then be attached to each task node in the value map. To move the idea even further: if the dependencies of the tasks could be mapped, automatized calculation of the *critical path* [KW59] would be possible. But accordinding to Vähäniitty et al. [VLR02] "in practice it is not possible to specify a system using features only because they depend on each other in complex ways". Based on Kostoff's [KS01] claim that "a balanced combination of the expert- and computer-based approaches may prove to be the most effective and efficient approach to roadmap construction" finding the optimal level of automation would require iteration.

One thing missing from the checklist by Vähäniitty et al. is the creation of metrics. To allow experimentation and improve value creation, when a task is selected to be implemented, metrics should be attached to each node in the path from the task to the business business goal. The method described in

Metrics chapter can then be used to validate assumptions. This can be seen as a distinct fifth step or it can be included into the fourth step to emphasize a vision of having metrics as a part of implementing a feature.

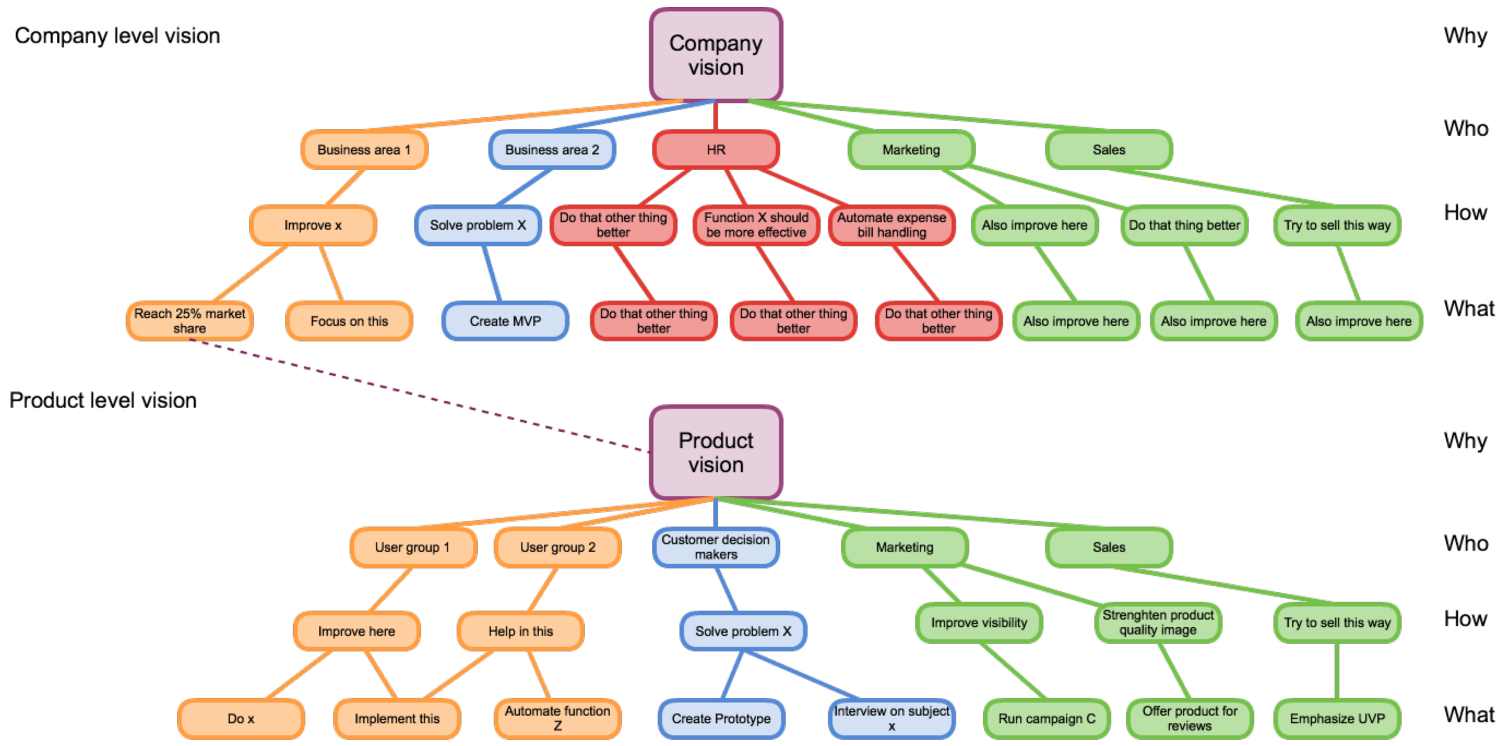

Figure 18: Company- and product level Impact maps.

6 Conclusions

Lean canvas was found to provide a great way of defining and clarifying a business model. It was found to be an useful method for compressing the business model and communicating it to the stakeholders. It has a feature of equalizing the stakeholder groups, which will help in product planning, and can improve the team's ability to communicate the value of a product.

Impact mapping was helpful in finding software development actions that create value, and therefore help in reaching the business goals. Impact mapping was found to be a very powerful technique for work planning as it immediately visualizes value assumptions. It can be used to effectively map the many routes to reach a certain goal, a route that might otherwise be missed because of a lack of communication between the members of the organization. It also helps in implementing measurement programs.

Lego serious play is an effective way to create shared understanding. It is useful for analyzing the business area of a product. LSP helps in finding issues with a product and unmapped parts of a business domain. It can be used to build and clarify a common vision for a product team, which can lead to more sound and valuable software systems.

All three studied planning methods were found fit into lean software development framework. All the methods visualized information that might otherwise easily stay hidden. They also gathered a lot of positive attention and the author of this thesis was asked to help other teams to have workshops, which implies that the methods could be landed on many similar organizations. Because of the nature of this study, no strong implications of the effect of Lean canvas, Impact mapping and Lego serious play, for example, on product profitability can be made. Still, according to Bowman and Ambrosini [BA00], improving value-creation in product development has a straight connection to the company profit margin.

Based on the results and the amount of positive reactions the studied techniques raised, it seems that these techniques truly have a positive effect on the organization and should be studied further. A proposal for how to use the studied methods to create value-oriented roadmaps was presented. Further studies are needed to verify the roadmapping approach and to find the best practices. The next research goal would be to test and refine the proposed roadmapping approach in practice. If the roadmapping approach proves to be effective, a software tool to aid in building hierarchical value-oriented roadmaps with support to metrics could be considered.

References

[AB12] Adzic, G. and Bisset, M.: *Impact Mapping: Making a Big Impact with Software Products and Projects.* Provoking Thoughts, 2012, ISBN 0955683645, 9780955683640.

[Adz11] Adzic, G.: *Agile product management using Effect Maps*, 2011. `http://gojko.net/papers/effect_maps.pdf`, Accessed: 12.11.2015.

[BA00] Bowman, C. and Ambrosini, V.: *Value creation versus value capture: towards a coherent definition of value in strategy.* British Journal of Management, 11(1):1–15, 2000.

[BAB+06] Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., and Grünbacher, P. (editors): *Value-Based Software Engineering.* Springer, Berlin, 2006.

[BLR+10] Basili, V., Lindvall, M., Regardie, M., Seaman, C., Heidrich, J., Munch, J., Rombach, D., and Trendowicz, A.: *Linking software development and business strategy through measurement.* Computer, 43:57 – 65, 2010.

[Bry04] Bryson, J. M.: *What to do when stakeholders matter: stakeholder identification and analysis techniques.* Public management review, 6(1):21–53, 2004.

[EA98] E., Colin and A., Fran: *Making Strategy: The Journey of Strategic Management.* SAGE Publications, 1998, ISBN 9780761952251.

[HWMP12] Heidenberg, J., Weijola, M., Mikkonen, K., and Porres, I.: *A model for business value in large-scale agile and lean software development.* In *19th European Conference, EuroSPI 2012, Vienna, Austria, June 25-27, 2012. Proceedings*. Springer, 2012.

[Hyv14a] Hyvönen, J.: *Creating shared understanding with lego serious play.* In Münch, Jürgen (editor): *Data- and Value-Driven Software Engineering with Deep Customer Insight*, pages 36–42, 2014.

[Hyv14b] Hyvönen, J.: *Lean canvas and impact mapping as tools for linking product development to business goals – a case study.* In Münch, Jürgen (editor): *Real-Time Value Delivery in Software Engineering*, pages 20–27, 2014.

[JS02] Johnson, G. and Scholes, K.: *Exploring Corporate Strategy: Text and Cases.* Financial Times Prentice Hall, 2002, ISBN 9780273651178.

[KS01] Kostoff, R. and Schaller, R.: *Science and technology roadmaps.* Engineering Management, IEEE Transactions on, 48(2):132–143, 2001.

[KW59] Kelley, J. and Walker, M.: *Critical-path planning and scheduling.* In *Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference*, pages 160–173. ACM, 1959.

[LEG10] LEGO: *Introduction to LEGO Serious Play.* 2010.

[LSM15] Lean Startup Machine, (Javelin): *The validation board*, 2015. `https://www.leanstartupmachine.com/validationboard/`, Accessed: 12.11.2015.

[Mau12a] Maurya, A.: *Running Lean - Iterate from Plan A to a Plan That Works, 2nd Edition.* The Lean Series. O'Reilly, 2012, ISBN 978-1-449-30517-8.

[Mau12b] Maurya, A.: *Why Lean Canvas vs Business Model Canvas?*, 2012. `http://practicetrumpstheory.com/2012/02/why-lean-canvas/`.

[MBH+10] Mandić, V., Basili, V., Harjumaa, L., Oivo, M., and Markkula, J.: *Utilizing gqm+ strategies for business value analysis: An approach for evaluating business goals.* In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, page 20. ACM, 2010.

[McC15] McCahill, L.: *Introducing the happy startup canvas*, 2015. `https://medium.com/the-happy-startup-school/introducing-the-happy-startup-canvas-76a71edc4af8`, Accessed: 12.11.2015.

[Moo95] Moore, J. B.: *Strategic Planning for Public and Nonprofit Organizations: A Guide to Strengthening and Sustaining Organizational Achievement.* Jossey-Bass nonprofit sector series. Jossey-Bass Publishers, 1995, ISBN 9780787901417.

[OB07] Ottersten, I. and Balic, M.: *Effect Managing IT.* Edition 1:1. Liber, 2007, ISBN 9788763001762.

[Ohn88] Ohno, T.: *Toyota Production System: Beyond Large-Scale Production.* Productivity, Cambridge, MA, 1988, ISBN 0-915299-14-3.

[OP10] Osterwalder, A. and Pigneur, Y.: *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers.* Wiley Desktop Editions. Wiley, 2010, ISBN 9780470901038.

[Pic15] Pichler, R.: *Roman's product canvas*, 2015. `http://www.romanpichler.com/tools/product-canvas/`, Accessed: 12.11.2015.

[PP03] Poppendieck, M. and Poppendieck, T.: *Lean Software Development: An Agile Toolkit.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003, ISBN 0321150783.

[RDS09] Racheva, Z., Daneva, M., and Sikkel, K.: *Value creation by agile projects: methodology or mystery?* In *Product-Focused Software Process Improvement*, pages 141–155. Springer, 2009.

[Rie11] Ries, E.: *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses.* The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business, 2011, ISBN 9780307887894.

[Ris14] Rissanen, O.: *Extending the development process towards continuous delivery and continuous experimentation in the b2b domain: A case study.* Master's thesis, University of Helsinki, 2014.

[RVS04] Roos, J., Victor, B., and Statler, M.: *Playing seriously with strategy.* Long Range Planning, 37(6):549–568, 2004.

[SG11] Schulz, K. and Geithner, S.: *The development of shared understandings and innovation through metaphorical methods such as lego serious play^{TM}.* 2011.

[Sma95] Small, S.: *Action-oriented research: Models and methods.* Journal of Marriage and the Family, 57:941–955, 1995.

[SO08] Statler, M. and Oliver, D.: *Facilitating serious play.* The Oxford Handbook on Organizational Decision-Making (Oxford University Press, Oxford), pages 475–494, 2008.

[Spa15] Spark59: *Leanstack*, 2015. `https://leanstack.com`, Accessed: 12.11.2015.

[Swa11] Swann, D.: *Nhs at home: Using lego serious play to capture service narratives and envision future healthcare products.* INCLUDE 2011 Proceedings, 2011.

[VGD94] V., Basili, G., Caldiera, and D., H. Rombach: *Goal question metric paradigm.* Encyclopedia of Software Engineering, 1:528–532, 1994.

[VLR02] Vähäniitty, J., Lassenius, C., and Rautiainen, K.: *An approach to product roadmapping in small software product businesses.* In *ECSQ2002), Conference Notes*, pages 12–13, 2002.

[Wer84] Wernerfelt, B.: *A resource-based view of the firm.* Strategic management journal, 5(2):171–180, 1984.