

UNIVERSITY OF TARTU
Institute of Computer Science
Master in ITM curriculum

Veronika Vlasova
**An analysis of the impact of the process improvement at
LHV Bank**
Master's Thesis

Supervisors:
Fredrik Milani, PhD
Ezequiel Scott, PhD

Tartu 2020

An analysis of the impact of the process improvement at LHV Bank

Abstract: The financial institutions nowadays have big development departments and are trying to adjust all the processes towards the use of agile software development. Before they mostly relied on the waterfall methods of development because it was more reliable and brought overall more value. But the demand outgoing from the rapidly changing world made companies consider Agile software methods as well. Companies usually spend additional resources on the changes and software process improvement to become more Agile, therefore the ways to evaluate change results are relevant for them. In this thesis, the case study on the LHV bank which applied 8 proposals of Agile Software Process Improvement was done. The proposals introduction of the agile management culture in the whole company, organization of the relevant training, assigning the concrete development team for the bank products, setting the common objective to the autonomous teams, and review of the agile development methods, delivery, and moving system architecture from the legacy system to the microservices. In light of its context, this thesis addresses the research questions of how the proposals were implemented, what was the impact of the implemented changes on the company, and what can be done to further improve the impact? In order to answer the research questions, the interviews with the employees of the company took place and the qualitative data were analyzed. The analysis was done to allow us to understand how the proposals were applied and which further improvements can be done in the future in LHV Bank.

Keywords: Software improvement process, improvement implementation, improvement evaluation, financial institutions, LHV

CERCS: P170

Tarkvara arendusprotsessi parendamise mõju analüüs LHV Panga näitel

Lühikokkuvõte: Tänapäeva finantsasutuste juurde kuuluvad suured arendusosakonnad, mis püüavad kohandada kõiki protsesse välk-meetodil toimiva tarkvaraarenduse suunas. Kui eelnevalt tuginesid firmade tarkvaraarenduse osakonnad peamiselt arendusprotsessis kosk-mudelile eeskätt selle usaldusväarsuse ja töökindluse tõttu, siis kiiresti muutuva maailma esitatud nõudmistele jääb selline meetod paraku üha enam jalgu. See on pannud firmade arendusosakonnad kaaluma välk-meetodil põhinevat tarkvaraarendust. Kuna enamasti kulutavad firmad ülejäävad ressursid progressiivsetele muutustele ja muuhulgas püüavad nad tarkvara arendusprotsessi muuta üha enam välk-meetodile tuginevaks, on nende jaoks oluline saada ülevaade ning hindamisviisid tulemuste mõõtmise jaoks. Käesoleva magistr töö raames on analüüsitud kaheksat LHV Panga välk-meetodil põhinenud tarkvara arendusprotsessi kiirendamise ettepanekut. Välk-meetodil põhineva juhtimise tutvustamine tervele firmale ning selle laialdane kasutuselevõtt terve organisatsiooni baasil. Selle teoks saamiseks tuleb asjakohaste koolituste korraldamise kõrval määrata panga toodete arendamiseks kindel meeskond, meeskondade autonoomsuse tõstmine ja ühise eesmärgi seadmine. Kasutusesoleva arendusprotsessi kriitiline analüüs ning selle täiendamine vastavalt välk-meetodi uusimatele arengutele. Tarkvara kasutusele võtmise protsess lähtestada samuti välk-meetodile tuginedes ning eelistada väiksematest osadest koosnevat infosüsteemi ülesehitust. Antud konteksti valguses on magistr töö eesmärgiks leida vastus küsimusele kuidas on arendusprotsessi muutmise ettepanekud kasutusele võetud, milline oli ellu viidud muutuste kasu firmale ning kuidas edaspidi veelgi täiendada muutustest tulnud mõjutusi. Uurimisküsimusele vastamiseks viidi firma töötajatega läbi intervjuud, samuti analüüsiti kvalitatiivseid andmeid. Tehtud analüüs võimaldas mõista, kuidas arendust hõlmavad ettepanekud ellu viidi ning milliseid järgmisi täiendusi saab LHV Pank tulevikus teha.

Võtmesõnad: tarkvara arendusprotsessi parendamine, arenduse rakendamine, arenduse hindamine, finantsinstitutsioonid, LHV

CERCS: P170

Content

1.Introduction	4
2. Background	6
2.1 Software development process and development methodologies	6
2.2 Software process improvement	7
2.3 Software measurement in Agile Software Development	8
3. Related work	9
4. Case study design	11
4.1 Case study methodology	11
4.2 Research questions	11
4.3 Case and subject selection	12
4.4 Data collection	12
4.5 Data Analysis	13
5. Results	15
5.1 Introduce agile management culture organization-wide.	15
5.2 Organise relevant training	16
5.3 Assemble concrete teams for products	17
5.4 Set common objectives for team members	20
5.5 Give more autonomy to the teams	21
5.6 Review the agile development method used with its components	22
5.7 Automate the release process	22
5.8 Use more modular system architecture and microservices	23
5.9 Summary	25
6. Discussion	27
6.1 Action done to make the changes followed by proposals	27
6.2 The impact of the changes on the company	30
6.3 The possible future improvements	33
7. Conclusions	36
Acknowledgments	37
References	38
Appendices	40
I.Interview questions	40
II. License	43

1.Introduction

Product design and software development, configuration management, quality assurance, project management, and delivery are just a part of activities that are included in the software development process (SDP) and which should be done in order to create the software. There is no fixed software development process that companies for sure should follow. According to Keenan, “projects differ in their scale, scope, and technical challenge, and the same process will not suit all circumstances” [1]. Therefore, Digital organizations for decades are in active search for new methods and tactics which allow them to stay more competitive with the market by improving their SDP. Based on the number of articles on this topic starting from 1991, the research industry and research institutions showed their interest in Software Process Improvement (SPI). [2]

Software Process Improvement allows to enhance the satisfaction of customers and employees, product quality, and add some financial value in the case of profit and return on investments. That also refers to the financial institutions which nowadays have big development departments and are trying to adjust all the process. Before they mostly relied on the waterfall methods of development because it was more reliable and brought overall more value. But the demand outgoing from the rapidly changing world made companies consider Agile software methods as well.

Agile software development methods were introduced in 2001 with the creation of a Manifesto for Agile Software Development. The principles of agile include putting the “individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, responding to change over following a plan.” [3]. There is also no strict way for the company to become agile and provide SPI, every model can be adapted taking into account the regulation and specific of the organization. From the position of different stakeholders, the influence of the improvements and changes can differ.

Companies usually spend additional resources on SPI. Therefore the ways to evaluate change results is relevant. Moreover, the impact of changes can not influence company

performance at all. In this thesis, the case study on the LHV bank which is moving to Agile and applied 8 proposals to improve software development was conducted. At a high level the proposals were about the introduction of the agile management culture in the whole company, organization of the relevant training, assigning the concrete development team for the bank products, setting the common objective to the autonomous teams, and review of the agile development methods, delivery and moving system architecture from the legacy system to the microservices. To examine how proposals were implemented and which impact they had 3 research questions for this work were formulated:

- *RQ1: How were the proposals implemented?*
- *RQ2: What was the impact of the implemented changes?*
- *RQ3: What can be done to further improve the impact?*

Motivation to investigate those questions is in the need to see whether the changes actually deliver value to the company and in which way they do it. The result can be interesting for all who work with improving software development processes and mostly connected with financial institutions.

Chapter 2 provides the background of the agile practices and Software Process Improvement. In chapter 3 the related works are reviewed. Chapter 4 presents the case study design which was used in the current case. The results from the interview and related documentation are structured based on the changes proposals and given in chapter 5. The changes, their impact, and suggestion after the literature review and interviews are provided in chapter 6.

2. Background

The background part of the thesis presents a brief overview of the software development process and its connection to software methodologies. The development methodologies are described in order to show how differently applied for methodology works and which aspect characterize them. The software process improvement frameworks are introduced by the most used frameworks.

2.1 Software development process and development methodologies

The software development process can be denoted as the systematic set of activities that are related between themselves and lead to the creation of software products. The more detailed definition of the software development process is “a coherent set of policies, organizational structures, technologies, procedures, and artifacts that are needed to conceive, develop, deploy, and maintain a software product” [7]. The software process methodologies are the way to describe the software processes, and they also can be called as software development lifecycle. There are the main parts of the software process development lifecycle such as requirements engineering, software design, and implementation, software verification and validation, and software maintenance.

The traditional software development methodology is a waterfall. A waterfall model is a sequential approach, where each fundamental activity of a process is represented as a separate phase, arranged in linear order. Since the new environment in 90th was an environment where they need to respond faster and think differently about what best motivates and drives great performance in organizations the waterfall type of thinking was not considered as appropriate[8]. During the 1990th lots of lightweight software, theories have appeared as a counterweight to heavyweight waterfall theory. In 2001, the Agile Manifesto was introduced to increase the speed of the delivery of the code, have closer collaboration with the customer, support changes to requirements, and improve the systems continuously. But in counterpoint to the Agile advantages, the methods have been criticized

for lack of focus on architecture, the little scientific support for claims in the agile community, and for being applicable only to small teams [21]

There are the most common Agile-based methodologies for software development: Scrum, Kanban, and Lean. Scrum is mostly characterized by its roles and activities. There are three different roles that should be in the team: Product Owner who is supposed to be the main link between the team and the customer, Scrum Master who mostly gets rid of the road blocked and track whether the team is motivated enough and cross-functional and self-organized Dev Team. [30] There are also 4 main activities in the Scrum process: Sprint Planning Meeting, Daily Stand Up, Sprint Review, and Sprint Retrospective. Kanban was created for Toyota System and its main goal is the minimization of the waste. The main principle of Kanban is the visualization of the process achievement and small cards that are moved from one point to another. In Kanban, the amount of work that can be in progress is strictly limited. Lean as well as Kanban takes the root in Lean manufacturing where the principles of methodology work in the direction of waste elimination. Some companies use mixed methodologies in their companies, therefore the terms as Scrumban appear [31].

2.2 Software process improvement

The goal of Agile methodologies is to allow the teams to provide the maximum value to the customer in the minimum time and to decrease the cost of changes in the late phases of the development. The software process improvement (SPI) is a set of changes to develop higher quality, decrease the number of defects, and finally reduce the costs and time of development. [13] It is not only about adopting some techniques or tools usage, but it is also about the broad current process analysis and changes required in a specific context. The SPI process mainly consists of 4 steps: Current process evaluation, Improvement Planning, Improvement Implementation, and Improvement Evaluation [8].

The Improvement Evaluation is a comparison of measurement before the improvement and after. There are different methods to define a set of metrics for some specific conclusions. For example, the GQM represents a systematic approach for tailoring and integrating goals to models of the software processes, products, and quality perspectives of interest, based upon the specific needs of the project and the organization [10].

2.3 Software measurement in Agile Software Development

As Lord Kelvin said, *“If you cannot measure it, you cannot improve it”*[19]. DeMarco in his work supported this statement for the software projects *“you cannot control what you cannot measure”* [20]. According to the [21], the main motivation to use measurements are:

- Project planning and estimation
- Project management and tracking
- Understanding quality and business objectives
- Improved software development communication, processes, and tools

The measurement is the process when based on some defined rules some symbols or numbers are assigned to some attributes of entities of the real world [22]. By Norman Fenton, there are such possible entities of interest in software measures: processes, as activities related to software over time, products as artifacts that appear during the process, and resources as the inputs which play the role of inputs to the process. The measurement and metrics in many types of research are synonyms.

There are two types of metrics to estimate the impact of change: qualitative and quantitative. Qualitative data is the output of the interviews, observations, questionnaires with open questions. In this case, the data after collection is encoded and analyzed. Quantitative data mostly relies on numbers and qualitative data which are already available or especially collected for the estimations.

In summary, in the Background the importance for the current thesis steps of the SPI were discussed.

3. Related work

The related work part of the thesis is aimed to review papers that describe the changes implemented in different organizations to scale the agile principles across the company and the ways of measuring the impact.

Several studies aimed to investigate the factors which contribute to the success of the Software Process Improvement in companies or projects. For example, Niazi in [16] identified the real opinion of the software practitioners about software process improvement project success based on the data of 27 software development companies in Turkey which were experiencing SPI. The employees participated in a survey online and the questions were generalized for all the companies and the main goal of the research was to identify the factors which make the SPI more successful based on the responses of employees. The results of the study were identified and a prioritized list of factors that contribute to the success of SPI was presented. As a result, they concluded that factors such as professional growth, professional recognition, technical support, new technologies, and strong leadership are the factors that influence the success of the implementation of SPI. Another case study related to key success factors for implementing software process improvement of Rainer[18] concluded on the main factors which are the term of process, people, skills, and leadership. The data for analysis was collected through the questions with the predefined answers therefore some valuable factors of the success of the SPI could be missed.

Also, studies were conducted to find the best ways of pointing out the bottlenecks. For example, Jasmine and Vasantha [13] used a graph visualization to identify the weak places in the reuse of software components and the reasons for the failure of the component communication path. More precision and concentration on the monitoring of bottlenecks were shown in the study of Caivano [14]. The study presents a system which allows the bottleneck monitoring and identifies a point to focus on while designing measurement systems. This study also related to the work of Petersen and Wohlin [15] who identified the bottleneck while visualizing lean software development flow. The paper presents a number of formulas to visualize the main lean metric, but that study is difficult to use and apply to the real data taken from the company.

The work of Chris Fry and Steve Greene [6] describes the changes done for a CRM and on-demand application development company to roll out agile. It presents a detailed explanation of the process including the creation of the cross-functional teams, automation of processes, and training implication. The learning outcomes are very well-structured, but the changes were done more based on the assumptions of the research on the need of the company from the qualitative and quantitative data. Moreover, the work was written in the year 2007 and some technical limitations of that time should be considered before taking the outputs of the research into use. Another example of Agile related improvements given in [7] describes the adopting Scrum in large software development projects. The main aspects of the change were accomplished by the creation of Area Products Owners, common sprint planning, and common retrospective. The limitation of the study is in the scope of the interviews because moreover, the status of the study was ongoing.

In summary, there were listed the ways how the SPI was implemented in the company, different points of view on the success of the SPI in different companies, and ways of the identification of the bottleneck in the processes. However, there were no studies that reviewed the activities for improvement, the impact of improvement, and future improvement in one study. Therefore this work differs from the studies done before which considered all 3 perceptions of improvement in the real case of financial institutions.

4. Case study design

4.1 Case study methodology

One of the definitions of the case study in the software engineering field is “an empirical inquiry that draws on multiple sources of evidence to investigate one instance (or a small number of instances) of a contemporary software engineering phenomenon within its real-life context, especially when the boundary between phenomenon and context cannot be clearly specified” [25]. A case study differs from the experiment in a way it is an observational study when the experiment is a controlled one [26]. The main objectives of the case studies can be to investigate the reasons for the process working and find the field for improvement. However, the post-event studies also can be named as a case study when it studies the effects of change. Since the case study includes the context it can be suitable to use for the research in software engineering since software engineering phenomena are difficult to study in isolation. It is applicable to the software engineering processes in specific types of institutions, such as financial institutions, where the context plays a significant role. In the case discussed in the thesis, the context such as bank restriction, security reasons, or compliance influence of the implementation of the software development changes in the IT department. Therefore the case study method is suitable for the current work.

The main five steps of the case study are the case study design, preparation for data collections, collection of data, analysis of collected data, and reporting. Basically, it is similar to the research process of other empirical studies, however, the case study is more flexible and the numbers of iterations between the steps are acceptable. [27]

The case studies can be holistic ones or the embedded ones. Holistic case studies are based on one unit of analysis and the embedded ones or research a few units of analysis. The current case study is holistic since it considers only one unit of one company.

4.2 Research questions

The objective of this thesis is to investigate how changes proposed were implemented and what their impact has been.

In order to achieve the above-stated objective, three research questions are formulated:

- *RQ1: How were the proposals implemented?*
- *RQ2: What was the impact of the implemented changes?*
- *RQ3: What can be done to further improve the impact?*

4.3 Case and subject selection

The case study was conducted in the bank LHV located in Estonia. The LHV bank is one of the biggest banks in Estonia. For the last 2 years, the company grew from 300 employees to about 500 employees. The development division has used some Agile practices for the last 5 years. In the year 2018, the CEO of the company provided a list of 8 proposals for the change in the company taken from the [31] to make the company more Agile. From the company perspective, the intention of the changes is the need to fasten the development of the new features in order to stay competitive with Fintech companies. Every proposal described the different perspective of the work of the company. The case study provides a possibility to observe the change's impact on the outcome of the company as well as the structure of the work of the employees.

4.4 Data collection

The data collection type for the current study is an interview-based and document-based data collection. The interview questions were formulated on the 8 proposals and structured with respect to the research questions. The interview questions described in *Appendix 1*.

It was important to investigate the points of view about the changes from different roles as well as interview people from different teams in order to understand the effect of changes in their context and avoid the replications of similarities. For this reason, the employees from different positions and different areas were preferable to participate in the interviews. The roles of the people who participated in the interviews are:

	Position	Area	Office
P1	CEO	-	Tallinn
P2	Product Owner	Retail Credit Products	Tallinn

P3	Product Area Manager	Payment Products	Tallinn
P4	Product Area Manager	Investment Products	Tallinn
P5	Head of Software Engineering	-	Tartu
P6	Software Development Unit Manager	Payment Products	Tartu
P7	Software Development Unit Manager	Investment Products	Tartu
P8	Head of Technical Team	-	Tallinn

Table 1. The people to interview

The semi-structured interviews were conducted from 09.12.2019 to 04.02.2020 face-to-face. The interview lasted 1-1.5 hours and the language of the interview was English, all the interviewees were native Estonian speakers. The list of questions wasn't sent to the subject before and was led by the set of the question right during the interview. During the semi-structured interview questions were not asked in an order they were listed but the list of questions was used more like a check-list in order to allow the improvisation of the subject.

The interviews were recorded to the audio format and then the recordings were transcribed to the text with the Temi software. In total, 9 hours of the interview were transcribed. There were also notes taken during the interview, but they were used to support the main objectives discussed. The first faces of the analysis of the interviews were carried out in parallel with interviewing in order to improve the data collection and update the questionnaire accordingly. The interview of the CEO was also supported by the internal document that was provided.

4.5 Data Analysis

All the transcripts of the interviews were coded using the pre-coding, preliminary jottings, questions to consider as coding techniques from the book “The coding Manual for Qualitative researchers” by Johnny Saldana [3]. During the first cycle of coding the

structural, evaluation, hypothesis, process coding methodologies were used. The interview recordings were relistened and the code phrases were listed up. Then all the documents with the main code phrases were formed based on the person with whom the interview was conducted.

On the next phase, the unified table was created where the citations from the interviews of all the respondents and represented documents were separated and ordered in one documents based on the proposals and the main information this citation contains: “how it was before”, “what was proposed”, “what was expected”, “what was done”, “how did it impact”, “what is lacking” for further analysis.

For the qualitative data analysis approaches, we used a template approach and a quasi-statistical approach. [26] In the template approach, all the changes in the proposal were reviewed from the perspective of research questions and the appropriate tables were fulfilled. In the quasi-statistical approach, the calculation of the repetition of the concepts in a different context was made. The methodology was adopted during the study to present a clear chain of evidence that leads to the result received.

5. Results

In this chapter, the main data about the changes which were carried out in the LHV bank are listed. The statements are structured by the 8 proposals and in every proposal, the questions about the past state, current state, and future state are covered. The proposals include the introduction of the agile management culture in the whole company, organization of the relevant training, assigning the concrete development team for the bank products, setting the common objective to the autonomous teams, and review of the agile development methods, delivery, and moving system architecture from the legacy system to the microservices. Therewith, the goals of proposals expected results, and impact is also reviewed and proved by the quotations. It is done in order to review every proposal from all the perspectives and answer the research questions in the next chapter.

5.1 Introduce agile management culture organization-wide.

Before the IT departments used Agile practices in their work while the business side was more into the waterfall approach. The method by which the company works was detected and called Water-scrum-fall. The product development process is working sequentially and so is the delivery, but it makes more for the new features to be improved because of the bottleneck on both sides.



Image 1. The method detected in the bank

It was proposed to introduce the Agile methods and management culture to the management of the bank. In the end, it was expected that management will be less controlling in IT development and product development

On the high-level managerial level, it was identified that there are 3 main processes: Product Development Process, Development Process, and Delivery Process and the main changes had to be done on the Product Development and Delivery levels because they were considered to be the waterfall.

Also business side people started use Jira to request the development of the new features. For example, the SDUM of the Payment Development team pointed out that the business people less turn to her with the requests about the development and instead the business side goes to the Product Owners who outs the tasks for IT into JIRA: “I save a lot of time teaching them (business side) about things that are really non-value-adding, such as how to enter a task in JIRA. And then, of course, the quality is better because I don't have to get these descriptions of what they want, but rather I get the business requirements. And this is because they don't turn to me. Now they turn to the Product Owner” (*Software Development Unit Manager, Payment Products*)

Only one respondent out of 8 did tell what is the Agile means for him: “*Turning an organization to being more agile one that means basically delegating responsibility downward. Since the company is getting larger and larger, more and more products are developed and the hierarchies and responsibilities needed to be created.*” (*Product Area Manager, Payment Products*) “*In my opinion, the product owners are Agile. Okay. There's no problem. I think the development team is not that agile yet.*” (*Product Area Manager, Payment Products*)

5.2 Organise relevant training

During the last few years the number of employees, therefore the knowledge exchange between the team had to be conducted and the knowledge of the external practitioners had to be implemented. Before the company had a learning hour every Friday where the employees could share their knowledge.

The proposal contained the creation of the knowledge base and organizing of the lectures by the practitioners from the financial sector were proposed. And the main expectation of the change is the proper methodology of training for the whole organization

After the proposals were introduced the Product Owner Club was organized on a regular basis. The Product Owner Club that is the set of the regular meetings where the lecturers from external companies share the common practices of Product Ownership. The organizer of the club shared the “*I have asked people to come from who knows also the*

business side or like not only like the technical part “ (Product Owner, Retail Credit Products)

Also, the practice of a personal budget was applied, the team is not told where to spend a certain amount of money on the training they found interesting and useful for their work. *“So we expect that like every senior and then the lead developer at least has some, some expertise so we can expect them to be aware of what's going on with the subject in the world. They can be trained from it.” (Head of Software Engineering)* After the training was completed the employees can present the knowledge gained to their colleagues during Friday's study hour *“Then they can also like to provide this knowledge outside of their own team” (Head of Software Engineering)*

Every Friday the employees dedicate an hour to the lectures. Sometimes such lectures are presented by employees who have finished their personal budget training and want to share. Lectures involve the whole team, all roles in order to share the best practices. *“What we did is that we organized a kind of workshop for the IT teams and product management teams together. So we involved, uh, product managers, analysts and the lead developers about basically about agile product management development practices” (Product Area Manager, Payment Products)*

Another event connected to training is "Know your bank". *“If the team has finished a project the people who have been involved in the project present it. So everybody can get to know what is going in the bank. We have done it somewhere one and a half year ago” (Software Development Unit Manager, Payment Products)*

The only qualitative impact can be seen out of the changes connected to the training: *“It impacted the way we conduct things. I feel better about it that we are more dynamic in reprioritizing our roadmap and we need to probably improve the communication about these changes to the team because from the top it makes perfect sense then why don't we do this first and that first or that last? Well, why don't we do that at all? It was beneficial because they get to understand each other and actually started implementing different ways of doing the analysis” (Product Area Manager, Payment Products)*

5.3 Assemble concrete teams for products

The bank has 24 different products and 7 development teams. Before the changes while some products had a specific development team assigned, others shared the

development resources of the different teams, Every product had a specific PO and sometimes one team had to deal with a few PO to whom their product belongs. It caused the difficulties with the prioritization of the development. The CEO of the company commented on the situation like this *“Each team is having in average 3.5 products to develop which brings the conflict on POs level about the priorities” (CEO)*

It was proposed to decrease the number of the product by merging some of them and to review the organizational structure should be reviewed and to reduce the number of business divisions. In the end, it was expected that teams will be responsible for the product from the beginning of the development until the very end

First of all, the creation of 4 Product Areas: Clients and Channels, Investment Products, Financing Products, Payment Products was done. The change of the organizational structure led to the situation when instead of 7 different division 4 product areas were created. *“We are more specialized in just focusing on customer segments. We have a cross-group called projects, but that's because the structure of the IT system supporting it is still a bit old. So we have to have it. But you're moving towards having, even more, let's say clusters or in separate departments. And at the same time, you have certain divisions that are such as support to connect that it's, it supports different customer support areas. There's still one group. And we are not sure if you want to make that more specialized or if you want to keep it the more. There are operational challenges. And we don't even know whether we want to overcome those challenges. It might be also cost-related, human resource related.” (Software Development Unit Manager, Investment Products)*

Also, the new position Product Area Manager (PAM) was launched. The Product Area Manager has full responsibility towards their products The PO (Product Owners) are still appointed to the specific products and still keep their responsibilities but instead of direct communication with the SDUM and development team directly the PO communicate to the development team through the PAM who is responsible for the prioritization of the development in the group of products under the specific area. The PAM role is like coordination and link between the SDUM and PO. *“Every area had its own manager. So that idea was that if the Product Owner under that area can't link and figure out themselves was the priority list for, for the team, not the product is then that product area manager is the one who has to decide or give the correct priority list. So this, this is what we did” (Rainer Tik)*

The product Area Manager has to assign the resources toward the product in the Area: *“I manage an area which is dealing with acquiring. And all in all, I'm going to have five product managers in these areas. Now the product managers are effectively product owners. It's the same thing. The area manager here is that we are all riding on the same limited resource, which is development teams. My product area is handling basically what we have at our disposal is two teams: Bravo and Mike. The largest is Bravo which is about 20 people and Mike's team is supposed to be seven, but actually is six people. Plus there are the IT services part and so on. But it is if we strictly talk about software development. Plus we are using one outsourcing partner who is developing bits and pieces in the issuing side.”* (Product Area Manager, Payment Products)

As an impact, the transparency in domains appeared *“The clarity of our domain (the same as the business area) has been increased.”* (Software Development Unit Manager, Investment Products)

For future improvements, the gap was found in the PAM's understanding of the technical part of their domain. One product owner who got the good feedback from the team considers herself a technical Product Owner (notice that the employer came to the maternity leave before the PAM position was introduced) *“I didn't even think about it that I had already grown into the role and they were actually satisfied and the team was actually satisfied the way we were working together”* (Product Owner, Retail Credit Products) considers that the very important part of the business side people is to be more technical PO or PAM *“So as it has happened with the Delta and alpha, uh, we are more like technical product owners.”* (Product Owner, Retail Credit Products)

One of the PAMs also expressed the wish to understand the technical background and issues more: *“The main thing for me is to understand the problems that IT faces because they solve it themselves. But, but I feel that it's some situations perhaps I could assist them in the solution at an earlier stage and perhaps. It will just be more efficient if you are a bit more closely and you know. But you know just some of the areas I haven't really gone in and gotten a grasp yet, let's say. So their IT analyst knows more about your business than you know the domain”* (Product Area Manager, Investment Products)

5.4 Set common objectives for team members

Before the changes, the roadmap was less concentrated on business areas. The preliminary roadmap creation took place in the autumn, but every quarter or month it was adjusted considering the current situation, the compliance, or other necessary activities. “Roadmap creation meetings are not regular. We start working on the 12-month next year map roadmap in September. Starting from September the work is quite intensive, it can be held every week. At the end of December, we have a fairly good idea (fixed version) what's going to happen next year and we go on the rolling roadmap for the upcoming 2 months. So that means that during the year this roadmap is actually adapted. We also have meetings every month during the year to do some retrospective of the last 2 months and to plan further for the next 2 months in comparison with the already fixed roadmap we wrote. The roadmap is documented in Excel and this simple tool works very well. *“The road map has always been in place. But now we are putting more effort into the roadmap planning.”* (Software Development Unit Manager, Investment Products)

It was proposed to set the quantifiable business objectives and qualitative development objectives to the team members and the situation when the basic business objectives are set for each team member was expected.

After the changes were presented more effort has been put into the roadmap planning which is now done on the business area level and is the output of the collaboration of board with PAMs, POs, and SDUMs. *“We had this approach for the road mapping earlier, but this year we started doing it on the business area level. If before we just wrote the development plan for the team now we are doing this in collaboration with the product owners and the business managers.”* (Software Development Unit Manager, Investment Products)

“The IT department is also aware of these goals. Is it somehow, yes. This is what we call already the quarterly meetings with one team and monthly meetings with another team. So with the bigger team, we did quarterly business reviews and with the smaller team monthly meeting where we look at how many new customers we have on, uh, connect, uh, where we look at various indicators that there we look at all sorts of things like uptime.” (Product Area Manager, Payment Products)

Based on the interviews and technical documentation the main goal of this Proposal “set IT and business objectives to the development team” has not been reached. The

Technical Manager considered that expectations about quantitative and qualitative business objectives were not met: *“Currently I don't see and I don't feel that IT would have the goals of the business. We have our own goals, technical goals. But to get this involvement on that level the IT person feels that “this business goal is mine”. (Head of Software Engineering).* The possible solution which was proposed by the Head of Software Engineering is the ownership of the business goals by IT people in order to feel more responsibility to achieve those business goals in parallel with IT goals. *“My point is that, if you want to make a goal for someone that he or she owns, yeah. Then, then he or she has to have some like a word on it as well, like a like if the goals are created somewhere in the torque and then they're presented to you, then you can't have it. Like it's not your goal. You don't have ownership.” (Head of Software Engineering)*

But the change in the roadmap planning approach increased the flexibility in reprioritizing *“Also where I feel better about is that we are more dynamic in reprioritizing our roadmap.”*

5.5 Give more autonomy to the teams

Before the release the responsible people had to give their approval on the tasks: *“And before we knew responsible business people, they had to agree to release tasks. But the problem was that they often, they didn't know. But when you had like 20 tasks you had to do, it was like more just a formal thing for them.” (Software Development Unit Manager, Investment Products)* For instance, the sign-off was mostly done through chatting and emails. *“Before I had to gather a sign-off from all the stakeholders” (Software Development Unit Manager, Payment Products)*

It was proposed to increase the decision power of the teams over their products. That at the end team will have a power of deciding what products or features to develop, in which order and at what time

After the changes were presented the sign-off was redesigned in a way that the improvement of every responsible person is not required anymore *“We fixed the sign-of this was basic. The first thing we did that was easiest to do” (Head of Software Engineering)*

As a result, the speed of releasing increased because there is no waiting time anymore for sign-off from the authorities.

5.6 Review the agile development method used with its components

The usual consistency of the team: Analyst, one Lead Software Engineer, 3-6 Software Engineers, and 1-3 Quality Assurance Engineers. The average size of the team is 8-12 people. Usually, one SDUM leads 2 teams, therefore there is in average 16-24 people to lead

The proposals included combining components from different agile methods in order to achieve the best result for the financial institution, use refactoring and regression testing and a review of the communication tools should be reviewed. Continuing to use Slack is reasonable, but in addition to that, Zoom should be taken into use for teleconferences between the team members locating in different places. It was expected that the team will create its own efficient way of development based on every team on known practices

After the proposals were presented the number of automated tests was increased, two development teams in the Payment Area were into one in order to cover all the products of this particular area. Also to ease the communication between the parties Zoom was taken into use.

Despite the fact that the number of automated tests has increased, there is still a lot of manual testing left. ColdFusion which doesn't provide the ability of the automated tests the testing is very slow and not always efficient *"All the testing is actually very slow because yeah if you want to release something you have to do all the all the tests we have automated"* (Head of Technical Team)

5.7 Automate the release process

The releases usually were rather big and took place on Wednesday at night at 23. The teams prepared their releases until that time.

It was proposed to redesign the whole release process and take the relevant information into use. Although the priority of automating the release process is low in this list, it can actually be done as one of the first things in parallel with the previous proposals. The expectations were: autonomous release, shorter releasing phase, the faster reaction on the bugs from the team side.

As a result, a separate delivery team was created. The Ops team has been assigned for the delivery *“Team that is right now there are five people” (Head of Technical Team)*. Also, The teams have introduced the bouncer which is the role of the person who is responsible for the delivery and hotfix in the features developed. *“We have this bouncer role or introduced basically, and it's different by the team, but most teams have the bouncer. So the bouncer is the one who actually deals with all the stuff coming in unexpectedly for a week. Then they switch. One person works with normal development and, but he's always available for, yeah. Well again, it depends on the team. But uh, with bigger teams, they basically don't do anything normal” (Head of Software Engineering)*

All the 8 respondents agree that the ability of the automated release is still lacking and introduced their wishes to have the automated tests in order to make their work more efficient.

“When I know once the team gets to the point where we are really able to release, whenever a feature gets ready, this is my end goal. The team is not there yet, partly because they're still dependent on these legacy parts and partly because of the quite frankly development methods which have not adjusted to the new model of working. So I mean to have continuous, uh, delivery in place, that means your team is organized in a different way as well. And I think probably the team has not caught it. Mike's team has one product which is the open packing API, which is delivery.” (Product Area Manager, Payment Products)

5.8 Use more modular system architecture and microservices

The first core system of the bank accounts was developed already around 20 years ago and it has remained the basis for most of the main banking services. Although the new modules of the system have already been built separately, these are still relatively large and do not enable different development teams to work on the same component at the same time

It was suggested to move towards smaller components and microservices and separate the core system to smaller components using Java and not ColdFusion. As a result, it was expected that bank infrastructure is in smaller components and microservices, the speed of development of components increased.

As the first step of the changes on the company level, it was acknowledged that the old system is a legacy system. As the next step, it was decided to rewrite the system to the four smaller independent smaller systems such as payment products, investment products,

clients and on-boarding process, internet bank (*Meetings notes*). The SDUM of the Payment Products proved that her team already rewrote their part starting from the year 2018, but there are still some parts left to be rewritten. *"We were the first team writing the core system from the Cold Fusion to Java. There is still something left in a core system. "* (Software Development Unit Manager, Payment Products)

Also, authorization and authentication parts were taken out from the ColdFusion. *"I wouldn't say that we moved to the microservices because we are in the banking domain, which means we can't make all our services very small. But yeah, we did want to have a mapping where there is one system for one team that there wouldn't be like a, a huge system where our four, four teams are developing and they have to communicate and uh, yeah, it makes things slow"* (Head of Technical Team)

After the changes started taking place *"The developers feel the ownership in the system. In their new system, they can do automated tests, in the old system in Cold Fusion they cannot do it"* (Software Development Unit Manager) There was the advantage to create the automation tests, which decreased the number of bugs found. *"The Cold Fusion is like spaghetti. If somebody is doing some feature it very affects and creates many bugs"* (Software Development Unit Manager) The increase of speed was also mentioned as an advantage after moving towards microservices *"I think it's common sense that with the new system where you have quite good quality code and a well-designed system, then there is obviously much faster to develop stuff. Then all the resistance. This is just like a feeling or commonsense. Okay. So speed and quality improvement, and happiness among the developer's course"* (Head of Software Engineering).

"Actually, we got lucky, it took about nine months and we scoped and cut some parts out. Now we are rewriting another part and there are bits and pieces left. The majority of the part of the payment has been actually rewritten and it has had a very good effect on the team's release autonomy and flexibility cause, uh, they don't need to release some things in the night because previously they literally took a lot back down on Wednesday at 11 in the evening. Now a lot of releases happen during the day and it's just more bearable for many people and of course, the infrastructure has developed." (Product Area Manager, Payment Products)

There is still a lot of work and other technical prerequisites before rewriting the system to the containers which the teams should complete *"Then after that another extra thing to do for the teams is their applications running in the container. These are the*

prerequisites for the actual automatic pipeline” (Head of Software Engineering). The technical debt was mentioned by 4 respondents: “Because every next thing, every next thing we’ll, uh, we’ll take slightly longer if you don’t, you know, clean the house every now and then. So for this year, we haven’t had an agreement with, uh, the IT teams that they can set aside about 20% of their time to deal with technical debt as they see fit. So, uh, this is the part where they kind of go around with no questions asked from me.” (Product Area Manager, Payment Products)

5.9 Summary

Based on the results, the actions connected to proposals were done in the company and had an impact on it. All the proposals contain some information and citations about the actions that were done in order to make the improvement in the company. However, not all the proposals contain information about the impact of lacking points because during the interviews they were not found. Therefore those statements were skipped in the result subchapters.

There was a concept calculation in the interviews considering the research questions to which it could be referred to. The table presents the number of people who mentioned the factor in the interviews and can be used later on in the discussion.

Several concepts were identified from the interviews. The concepts are “Agile practices”, “Role of the Product Area Manager”, “Automated deployment”, “ColdFusion”, “Sign-off process”, “Automated tests”, “Team of 20 people”. “Compliance”. The Agile practices groups mean the definition of Agile mentioned in the interview during the conversation. The Role of Product Area Manager groups means the responsibilities or role interactions of the PAM. Compliance groups mean the regulatory, control, or other bank compliance tasks mentioned in the interviews. The Automated deployment, ColdFusion, Sign-off process, and Automated test groups are rather straightforward as well as their meaning. The number of respondents mentioning the concepts were counted. Table 2 shows these results.

	RQ1: Actions done	RQ2: Impact	RQ3: Future improvements
	№ people	№ people	№ people
Agile practices	2	2	-
Role of the Product Area Manager	4	3	2
Automated deployment	2	4	8
ColdFusion	5	3	4
Sign-off process	8	4	-
Automated tests	4	3	2
Compliance	-	3	-

Table 2. Concept calculation

6. Discussion

In this chapter, the results are discussed in order to answer the research questions. The reflection about improvements that were done is ordered based on the proposal. Also, further wishes towards future changes or future states are provided based on the review of the existing literature. Based on the results, the actions connected to proposals were done in the company and had an impact on it. All the proposals contain some information about the actions that were done in order to make an improvement in the company. However, not all the proposals contain information about the impact of lacking points because during the interviews they were not found.

6.1 Action done to make the changes followed by proposals

In this subchapter the RQ1 “How were the proposals implemented?” is discussed. The proposals prepared for the Agile improvements in the LHV bank are different based on their sphere of influence, affected department, and the steps which should be done in order to fulfill them. Therefore, it is difficult to separate them from each other or compare them and the discussion on the research question will have different forms. While some of the proposals cannot be completed without previous finishing of the others, activities from the proposals lead to a disagreement in the company. For example, in order to complete the parts “automated testing” of the *Proposal 6* and “automated release” of *Proposal 7* are dependent on the activities of the *Proposal 8* such as “rewriting the legacy system from the Cold Fusion to Java” should be done beforehand. The set of actions which were done to fulfill every proposal or the impact are also different from the quantitative and qualitative side because they are related to the proposal's nature.

After the Proposals for the change were published, there was a big meeting in the company where all the points were discussed. Also, the CEO of the company had a personal interview with some managers in order to review the specific bottlenecks. Based on Table 2, four out of eight respondents pointed out that it is difficult to identify the specific start of the changes and actions which were done which influence the set of changes. The employees could read the description of the changes, but the separate document with the step-by-step

structure or milestones for every change was not presented. The changes were also impacted by the external environment. Since the study was conducted on the financial institutions the compliance, regulation, and security could be considered as external factors.

Regarding *Proposal 1 “Introduce agile management culture organization-wide”* based on Section 5.1, the high-level management identified the 3 main process phases during the creation of the new features for Software: Product Development Process, Development Process, and Delivery Process. The phases were mostly identified based on the executors of the activities in every phase. Due to the meeting notes documentation the main changes had to be done on the Product Development and Delivery levels since they were considered to be a waterfall and were the bottlenecks during the development. Therefore the world Agile and Agile practices started spreading from the development unit to other units of the company. There are some units closely related to software development that are not agile and cross-functional enough such as customer support. Those units were taken into account yet, because of the structural change *“and we don't even know whether we want to overcome those challenges, it might be also cost-related, human resource related.”* (Software Development Unit Manager, Investment Products)

As for *Proposal 2 “Organise relevant training”* for the Product Owners, the Product Owner Club was organized on a regular basis. According to the results shown in Section 5.2, practitioners who have similar domains and are more technical were invited as speakers from Pipedrive, Vasco, Duarte, Avira, Nokia. On the development side, the Concilium with specific roles and Head of Software Engineering now takes place every 2 weeks to discuss the related topics.

In the organization, in general, the employees from the development unit got a personal budget they can spend on the domain training they wish and study hour also takes place in the Development Unit every Friday in order to share the knowledge. Moreover the activity “Know your bank where done” in order to spread the ideas about the project finished is executed.

The main change related to *Proposal 3 “Assemble concrete teams for products”* is that on the structural level, the 4 product areas were created. Therefore the 24 different

products are separated between the 4 areas and every area has a set of development teams assigned to the area. The creation of the areas led to the creation of the new position Product Area Managers (PAM). The role of the PAM and the creation of interaction between the IT and business side, therefore the Product Owners instead of heading to the Software Development Manager who is responsible for the development teams turn to the PAM. In the 5.4 presented the explanation of responsibilities by the one Product Area Manager, though not all the have the same view on the responsibilities

Concerning *Proposal 4 “Set common objectives for team members”* the roadmap creation process was improved considering the creation of the 4 product areas and is done on the product areas level. The long-term planning currently is created by the board in the collaboration with Product Area Managers, Product Owners, and Software Development Unit Managers. But the company did not build a standardized methodology of the communication of business objectives or objectives of reprioritization to the team. The activities related to the short-term planning and adoption of the roadmap to the factors caused by the environment are not specific enough yet.

With reference to *Proposal 5 “Give more autonomy to the teams”* the sign-off process was reviewed and considered to be a bottleneck. Therefore it was redesigned in a way that the team does not need the approval before the start of the work or before the release. The Software Development Units Manager mentioned that instead now she just sends the letter with the new features developed.

On the subject of *Proposal 6 “Review the agile development method used with its components”* two teams were merged into one and currently, the team was an average of 20 people. Based on the 5.6 it was a high risk to ignore the literature advice of the sufficient average size of the team of 6-7 people. However, the management of this team finds it sufficient to have it in this way for the current domain. Also, the number of automated tests increased for the parts of code that were rewritten from ColdFusion to Java. ColdFusion doesn't allow the creation of the automated tests, due to the bank constraints it is forbidden to have the exact copy of the Prod Environment on the test because it contains the data of bak users.

The main changes related to *Proposal 7 “Automate the release process”* the separate team that is responsible for the release process was created. The teams are dependent on the delivery team, which at the moment of the interview consisted of the 5 people mostly dealing with the communications issues and gathering all the archives with the new features together. The releases take place on Wednesday at 23.

In most interviews in 5.8.1, it was evident that the parts of the code which have been rewritten from ColdFusion to Java now have automated releases and therefore the further actions in this step are dependent on the fulfillment of the *Proposal 8*.

As for *Proposal 8 “Use more modular system architecture and microservices”*, the main changes are related to the system architecture. The old monolith system which was written in ColdFusion was considered a legacy system and it should be rewritten into 4 smaller independent systems. Based on the interviews 5.8.1 the Payment Products team rewrote the code from the core system. The team had a fixed percentage of time which was spent on it.

In summary, many changes were applied from organizational to system architecture. Some changes are still in the process such as refactoring the modular system, some have not been started yet such as the creation of common objectives for all the team members.

6.2 The impact of the changes on the company

Subchapter 6.2 aims to answer the RQ2 “What was the impact of the implemented changes?”. Considering the impact it is taken into account that not only changes based on proposals could have an impact on the spheres described in the next chapter, but some external or internal events could also influence the spheres discussed.

Concerning the *Proposal 1 “Introduce agile management culture organization-wide”* the term Agile spread from the development unit to the business units of the company. it seems that all the units are involved now in this Agile practice

Regarding *Proposal 2 “Organise relevant training”* the training implemented helped the PO to improve their knowledge about the product development process and use such practises in their work. The personal budget got to be used by employees, it made the work

in LHV for then more beneficial and brought the knowledge to the company which can be probably used.

Respecting *Proposal 3 “Assemble concrete teams for products”* the clarity of the business areas has been increased, moreover, the emergence of the PAM role in between the Product Owners and SDUM helped to decrease the time SDUM or team spent on the communication with the business people and focus o more on the development.

As for *Proposal 4 “Set common objectives for team members”* the company did not build a standardized methodology adjustment of the creation of the roadmap and reprioritization in one team lead to the situation when the team cannot finish one epic which is already on the ones of the finish stages and because of the reprioritization should switch to the next one. Therefore the time for arrival of some tasks increased rapidly just because of the need for the switch. From the 5.4 follows that one team member even left the company because of such dynamics. The

On the subject of *Proposal 5 “Give more autonomy to the teams”* the redesign of the sign-off process and more autonomy for the release of the features changed the speed of the release and made it faster. 8 out of 8 interviewees mentioned this change as an achievement they made after the proposals were released, because it removed the bottleneck. However, one interview mentioned that after the sign-off process has been changed the communications between the business and development side decreased because nobody knows what can be released next.

In some teams, there is still some dependency on the priorities of the features to develop made by PAM. Sometimes the priorities are changeable so often that the team doesn't have an opportunity to the Epics which had only a few days to finish due to the chewable priorities or not proposes ETAs and it extends the total speed of the development of the features. Therefore the results of the activity of the creation of PAM above the SDUM from the Proposal 3 and the goal of the reaching the autonomy of the teams prioritize the tasks and make the decision about the next feature to develop are conflict to each other in a way that they cannot be completed at the same time considering that the approaches of the interaction of the PAM with SDUM and the teams are different for the areas.

With reference to *Proposal 6 “Review the agile development method used with its components”* the teams are quite autonomous in the decision which practices or methodologies to use. Also, the tools which are required to use also were not identified. However, in some teams, there is still some dependency on the priorities of the features to develop made by PAM. Sometimes the priorities are changeable so often that the team doesn't have an opportunity to the Epics which had only a few days to finish due to the chewable priorities or not proposes ETAs and it extends the total speed of the development of the features. Therefore the results of the activity of the creation of PAM above the SDUM from the Proposal 3 and the goal of the reaching the autonomy of the teams prioritize the tasks and make the decision about the next feature to develop are conflict to each other in a way that they cannot be completed at the same time considering that the approaches of the interaction of the PAM with SDUM and the teams are different for the areas.

As for *Proposal 7 “Automate the release process”* the creation of automated releases for those parts of code that were rewritten from the microservices and moved to Java helped teams improve the quality of code and decrease the number of bugs and hotfixes in production. Given in 5.8.1 the team can release when it is needed and the work done and not to boost the work in order to release the part of work on Wednesday at night when it is possible to release the new features to the monolith. Therefore the team members can develop the new features in their tempo, knowing that they can release whenever needed and not finish quickly some parts just to finish everything until the common release on Wednesday. Moreover when teams release the new features on their own they feel responsible for the parts of code released and can react to bugs or need in hotfix more quickly.

With reference to *Proposal 8 “Use more modular system architecture and microservices”* the impact of the code rewriting from ColdFusion to Java the technical side managers mostly described from the qualitative side. For example, based on *Table 2* people mentioned the happiness of the developers because it is easier and more pleasant to write code on Java and the feeling of ownership in the system when they take out pieces of code from the monolith. The rewriting of the code impacted the ability to create automated tests. Given in the interviews in 5.8.1 the rewriting of the code from the ColdFusion to Java allowed the creation of the automated and the impact of it described in the 6.2 for *Proposal 6*.

Moreover, the tests decreased the number of bugs and it is connected to the impact. Also the team and automated releases.

In summary, the changes basically had an impact on the company, however, it differs from the expectations presented in proposals. Also overall this impact was different in different teams. The factors which influenced the results are the clearly defined roles of the PAM, PO, SDUM, and their communication in different domains. In addition, the size of the team was an important factor, because the changes had a different impact on the team which consists of 20 people. Moreover, the time which the company spent on rewriting the services from the monolith was considered as important because it allowed them to complete other proposals.

6.3 The possible future improvements

The aim of the next subchapter is to discuss the RQ3 “*What can be done to further improve the impact?*”. Overall it seems that the company made improvements on the different levels of the organization, teams, processes, and technical side. But during the research, it was identified that some technical issues, room for improvement, and tensions between the different parties still take place. Therefore the further work on every proposal could be done.

The main changes related to *Proposal 1 “Introduce agile management culture organization-wide”* there was no common meaning what does Agile means for the whole company and this factor can lead to the misunderstanding of agile concepts. Masood et al. [29] describe how the agile practices are carried without the understanding of their purposes and sometimes that misunderstanding can lead to the “mini-waterfalls”. There is no exact definition of Agile and different practices can bring different results to the company, however, it is reasonable to unify the meaning in order to exclude the bias and ease the process of the future improvement or changes in the company. The possible suggestions taken from Masood et al. [29] define the agile values needed for the company, concentrate on those values, and communicate them clearly and intensively.

As for *Proposal 2 “Organise relevant training”* many actions were done in regards to the training organization and employees did not have any wishes for further improvement and based on the impact it seems it works well. However, from the research perspective, the main

expectation to build the proper methodology of training for the whole organization is not completely met and there is some room for improvement. The training, seminars, lectures, programs are chiseled, separate in their implementation, and don't have the collaboration or strategic plan to follow. For example, personal budgeting has an impact on the wellbeing and motivation of the employees [29], but it does not solve the problem of the spread of the common Agile understanding across the company. The same with PO club was external but the theory about the Product Ownership is missed. What can be reasonable is to implement the same types of training for the development and business side in order to diminish the “perception gap” between the parties and get more value from the training. The next questions can be helpful before the training creation.

- What are the main reasons for the training organization? Which problem do we want to solve?
- How can we measure that the problem is solved after the training takes place?

Regarding *Proposal 3 “Assemble concrete teams for products”* based on the interviews it was identified that there is still some tension on the level of communication between business and development sides and long-term prioritization of the tasks. Moreover, the clear set of responsibilities or practices for the Product Owner and PProducts Are Managers was not clearly stated yet. Also, it was identified that all the Products currently have the Product Owners. But there are lots of processes in the company which doesn't have any Process Owners: for example onboarding, leasing. The creation of Process Owners allows to create the matrix approach in the management and helps to speed the processes.

As for *Proposal 4 “Set common objectives for team members”* in Section 5.4.2, it was given a valuable point of view of the Head of IT who had an idea to create and goal ownership in the company. The business goals made in the board are reviewed by managers, segmentation, and moved to every employee. It can be done through the online systems

Possible future improvements related to *Proposal 5 “Give more autonomy to the teams”* are to review the roles in the teams and the team's sizes. Five out of 8 interviews mentioned the big team which consists of 20 people and difficulties which team members face. It is also proposed to give an opportunity for the team member to spend a day in another

team for 1 day to gain the knowledge from the other team and understand another domain more. This practice is called “Job shadowing” and it can be helpful to get the team members opportunity to see on the real workday which methodologies are used in another team and which roles have which responsibilities.

Respecting *Proposal 6 “Review the agile development method used with its components”* it is proposed to increase the number and quality of automated tests, but based on the 5.6 the completion of this point depends on *Proposal 8*.

As regards *Proposal 7 “Automate the release process”* the teams are still dependent on the delivery team and all the delivery tools and frameworks which were taken into account are created to make the visualization and communication between the parties more efficient. There should be the role of a bouncer in every team in order to be ready to react to the bug quickly and efficiently.

As for *Proposal 8 “Use more modular system architecture and microservices”* since the completion of the moving from the monolith to microservices showed very good results for the teams who already finished it is suggested to increase the technical debt time in every team in order to finish it.

Based on the results presented above proposals the implementation of changes was done and made an impact on the company on the organizational and technical level. However, the main goal of the fast development of the features is not reached yet. The main problems which were identified during the interviews are the lack of communication between the units and business, engineering, and delivery side. The proposal had a goal to introduce and apply Agile practices to every phase to get rid of the “Water-scrum-fall” method, however, the waterfall principles still can be observed in the work of the company. Other problems that were identified are communications, common values, priorities, and the spread of the information across the company. This indicates that the next steps to fasten the development process can be done in the direction of the creation of better communication links between the business, development, and delivery processes actors.

6.4 Threats to validity

From the perspective of external threats, first of all, it is difficult to find the starting point of the application of changes. Since the company is a dynamic subject for the observation there also was problematic to follow up on whether the changes in the company were done or impact on the company was observed due to the application of the changes or there were other phenomena that impacted it. Also, one may argue on the sample size and presentation of the results. Due to the time constraint and restricted collection of the data from the business side only 8 employees could participate in the interviews and only qualitative data was reviewed. To obtain the broader view on this topic more employees should be involved, the qualitative analysis has to be provided as well as more time from the point of presentation of proposals in the company should have passed to explore the results.

Another threat to mention is that the changes and impact on the company identified during the interviews could not be connected to the proposals and this could just be evidence of other conditions or events which have happened in the company. Therefore, the interview questions were structured in a semi-structured way to help the participant to find the relevant sequence of the actions for, its impact, and the current state.

The internal threat of validity was a bias and subjectivism of the researchers during the qualitative analysis. In order to diminish the impact of this threat the findings were weekly discussed in the research team, there were a few researchers presented in the interviews or meetings.

Finally, considering only one point of view is the internal threat of validity as well. The countermeasures which were taken: to get the information from the different views such as provide the interviews with different roles from different teams. It is also reasonable to review the terms used by interviews and their meaning for the analyst during the analysis process.

7. Conclusions

Software Process Improvement can lead to significant changes in the speed and quality of the development in different types of institutions. However, the measurement of the changes and impact on the company should be done at some point to review the changes

and adjust the direction of the movement if it is needed. In this thesis, a case study on the analysis of the impact of the process improvement in LHV bank was presented. The case study consisted of the next parts: 1) review of the proposals Kilu et al. [31], 2) identification of the actions done in order to fulfill the proposals, 3) identification of the impact of changes, 4) identification of the lacking points. The study was done based on the interviews with the employees.

After the review, it was identified that many actions in the company were done in order to optimize the processes. For instance, the sign-off process was removed, the 4 product areas Clients and Channels, Investment Products, Financing Products, Payment Products were created, the relative training for Product Owners was provided, and the work of getting rid of the legacy system and rewriting code to Java. However, when some changes such as sign-off or code rewriting optimized the speed of the product development from the end, others lead to some inconsistency and disagreements in others such as dynamic reprioritization of the roadmap. It was also identified that the impact of changes depends on the context of the team and domain. Also, some of the changes were not done because they are dependent on other changes that should be done previously are not a priority or need more time to be completed, such as automated tests and automated releases need to have code rewritten to Java before. The next list of questions can help to continue performing every change more accurately. What problem on the concrete level exists? How does it influence the work? What is the preferable future state? Does it create some discrepancy with other levels? Also, the possible steps which can be done based on the Dikert et al. [28]: concentrate on Agile values, communicate the change intensively, and make change transparent.

In the future, it would be reasonable to review the quantitative side of the changes in order to support the statements taken from the personal interviews by the quantitative results.

Acknowledgments

We are thankful to the teams of the LHV bank who kindly agreed to provide us with relevant documentation and shared a valuable point of view during the interviews.

References

1. Keenan, F.: Agile process tailoring and problem analysis (APPLY). In: Proceedings of 26th International Conference on Software Engineering, ICSE, 2004
2. Hall, T., Rainer, A., & Baddoo, N. (2002). Implementing software process improvement: An empirical study. *Software Process: Improvement and Practice*, 7(1), 3–15, 2002
3. Manifesto for Agile, Electronical source, [retrieved from <https://pdfs.semanticscholar.org/3eda/bb96a07765704f9c6a1a5542e39ac2df640c.pdf> 10.10.2019]
4. Johnny Saldana, *The coding Manual for Qualitative researchers*, Sage, 2015
5. Large Scale Agile Transformation in an On-Demand World, Chris Fry and Steve Greene, IEEE, Computer Society, 0-7695-2872-4/07, 2007
6. A Maturity Model for Scaling Agile Development, Igor Stojanov, Oktay Turetken, Jos J.M. Trienekens, DOI 10.1109/SEAA.2015.29, 2015
7. Maria Paasivaara, Casper Lassenius, *Scaling Scrum in a Large Distributed Project*, DOI 10.1109/ESEM.2011.49, 2011
8. Fuggeffa A., *Software Process: A Roadmap*, 2000
9. Abrahamsson, P., Marchesi, M., & Maurer, F., *Agile Processes in Software Engineering and Extreme Programming*. In Pula, 2009
10. Kupiainen, E., Mäntylä, M. V., & Itkonen, J., *Using metrics in Agile and Lean software development - A systematic literature review of industrial studies*. *Information and Software Technology*, 2015
11. Basili et al, 1994a
12. Kupiainen E, Mäntylä M, Itkonen J *Using metrics in Agile and Lean software development - A systematic literature review of industrial studies*, 2015
13. Petersen Kai, Wohlin Claes, *Software process improvement through the Lean Measurement (SPI-LEAM) method*, 2010
14. Jasmine K.S, Dr. R. Vasantha, *Identification Of Software Performance Bottleneck Components In Reuse based software Products With The Application Of Acquaintanceship Graphs*, 2009
15. Danilo Caivano, Oivo M., Baldassarre M.T., Visaggio, *Monitoring Bottlenecks in Agile and Lean Software Development Projects – A Method and Its Industrial Use*. 2011
16. Petersen, K., Wohlin, C.: *Measuring the flow in lean software development*. *Software: Practice and Experience*, 2010
17. Niazi, Mahmood, Mishra, Alok, Gill, Asif Qumer, *What Do Software Practitioners Really Think About Software Process Improvement Project Success? An Exploratory Study*, 2018
18. Rainer, A.; Hall, T.: *Key success factors for implementing software process improvement: a maturity-based analysis*. *J. Syst. Softw.*, 2002

19. JD Burchfield, Lord Kelvin and the Age of the Earth, 1990
20. DeMarco, T. 1982. Controlling Software Projects: Management, Measurement and Estimation. Yourdon Press. New York. 296 p.
21. Dingsøy, T., Moe, N. B., Fægri, T. E., Seim, E. A., & Moe, N. B. (2018). Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. 490–520. <https://doi.org/10.1007/s10664-017-9524-2>
22. L. Finkelstein, “A review of the fundamental concepts of measurement,” Measurement, vol. 2, no. 1, pp. 25-34, 1984.
23. Norman Fenton, Software Measurement: A Necessary Scientific Basis, IEEE Transactions on Software Engineering. vol. 20, no. 3, March 1994, pp. 203
24. Egon Berghout and Rini van Solingen, The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development, McGraw Hill, 1999
25. Runeson, P., Host, M., Rainer, A.W., Regnell, B.: Case Study Research in Software .. Engineering. Guidelines and Examples. Wiley, Hoboken, 2012
26. Wohlin C., Runeson, P., Höst, M., Experimentation in Software Engineering, DOI 10.1007/978-3-642-29044-2
27. Andersson, C., Runeson, P.: A spiral process model for case studies on software quality monitoring – method and metrics. Softw. Process: Improv. Pract. 12(2), 125–140, 2007
28. Dikert K., Paasivaara M., Lassenius C., Challenges and success factors for large-scale agile transformations: A systematic literature review, <https://doi.org/10.1016/j.jss.2016.06.013>, 2016
29. Masood A., Impact of Motivation on Employee Performance with Effect of Training: Specific to Education Sector of Pakistan, ISSN 2250-3153, 2013
30. Matharu G., Singh H., Mishra A., Upadhyaya P., Empirical Study of Agile Software Development Methodologies: A Comparative Analysis, DOI:10.1145/2693208.2693233, 2015
31. Kilu E., Milani F., Scott E., Pfahl D., Agile Software Process Improvement by Learning from Financial and Fintech Companies: LHV Bank Case Study
32. Nikitina N., Kajko-Mattsson M., From Scrum to Scrumban: A Case Study of a Process Transition, IEEE, ICSSP, 2012

Appendices

I. Interview questions

P1 Introduce Agile Management organisation-wide based on the development process

1.1 Can you tell us a little bit about the changes as compared to the last year in regards to the process/ collaboration/ roles/

1.2 How do you see these changes expressed in the daily/ weekly/ monthly work?

1.3 What is the impact of these changes? Did it meet the expectations?

1.4 How were these changes implemented

1.5 What is your work that is better after the changes? Why is it better? How do you know it?

1.6 What is still needed/ can to be improved? Why and how?

P2 Organise the relevant training

2.1 How is training organized at your team?

- What are the types of training?
- Their frequency
- Topics of the training?
- Have you had any training during the last 2 years?

2.2 Has anything changed (topic or other aspects) of training at LHV in the past 2 years?

P3 Assemble concrete teams for products

3.1 Does every product have a concrete team?

3.2 How the changes impact on quality, speed, size?

3.3 Have the sizes of the team's changed in the past years?

3.4 What can be improved (why and how)?

P4 Set common business objectives for the team members

4.1 Can you tell us about the common business objectives?

- What kind of business objectives, who set them, how are they set?
- Do they change often, how do they change, why do they change?

4.2 How are the objects visible or expressed in the process?

4.3 What impact does it have?

4.4 Was it good?

4.5 What can be improved?

P5 Autonomy to the teams

5.1 In which way teams are more autonomous?

5.2 How does it impact?

5.3 Has it anyhow increased/decreased

5.4 Is it good for the team's daily work?

5.5 How was it implemented?

P6 Review agile development methods

6.1 What are the changes you see have been made in the past years to the development process?

- Code review, testing, deployment, communication

6.2 How are the changes visible in the process?

- Artifacts, new statuses, new ways

6.3 What is the impact of the changes in the process?

6.4 Is it good or not?

6.5 How the changes were implemented?

P7 Automate Release Process

7.1 Can you tell us about the release process (your role in the release process, the process, artifacts, etc.) and changes made for improvement during the last year?

7.2 What was the impact of release process automation?

7.3 Was it good?

7.4 How was it implemented?

P8 Modular system

8.1 How has the change to the modular system been done?

8.2 Which impact did it have?

8.3 Was it good?

8.4 How was it implemented?

General

9.1 What is the main challenge you see now?

9.2 What would you like to change and in which way?

9.3 Anything else you would like to mention?

II. License

Non-exclusive license to reproduce thesis and make thesis public

I, **Veronika Vlasova**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

An analysis of the impact of the process improvement at LHV Bank,

supervised by Fredrik Payman Milani, PhD and Ezequiel Scott, PhD.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Veronika Vlasova

16.05.2020