Agile improvements to a requirement handling processes

INF5181

Arselan Sultani - arselans@ulrik.uio.no
Tanusan Rajmohan - tanusanr@ulrik.uio.no
Johan Paramanathan - johanpar@ulrik.uio.no
Birashanthan Tharmakulasingam - birashat@ulrik.uio.no

UNIVERSITY OF OSLO

Autumn 2017

Contents

1 Introduction **3**

1.1 Assumptions 3

2 Improvement context **4**

2.1 Company 4

2.2 Actors and stakeholders 5

2.3 Old BPMN model 6

2.4 Problem 7

2.5 Improvement Goals 8

3 Suggested changes **9**

3.1 Database 9

3.2 Communication with the customer 9

3.3 Prioritize requirements 9

4 Implementation of process changes **10**

4.1 Implementing database 10

4.2 Communication with the customer 11

4.3 Prioritize the requirements before JIRA 12

4.4 Improved BPMN model 14

4.4.1 14

4.4.2 14

4.4.3 14

4.4.4 14

5 Monitoring and Control **15**

5.1 Reduce time-to-market for new software products by 20%. 15

5.2 Reduce the necessary resources to initialize a project by 50%. 15

5.3 Improve process of creating requirements by 50%. 16

5.4 Data quality and validity 17
5.5 Technical debt 18
5.5.1 Definition 18
5.5.2 Comparison 18

6 Discussion **19**
6.1 Applied methods and Further work on the database 19
6.2 Rationale for Changes 19
6.3 Risks 20
6.3.1 Slow start 20
6.3.2 The risk of technical dept 20
6.3.3 Silver lining between old and new requirements 21

7 Conclusion **22**

8 Improved BPMN **23**

References **24**

1 Introduction

The general purpose of this report is to describe a realistic Software Process Improvement (*SPI*) plan for a system development organization. This report will be based on a fictional company with characteristics and trademarks of a real company.
The company primarily works with system development, however it in particular maintain software for a product line of access control. With systems with no formalized process for new requirement handling.

The **goal** of this report is to implement a new requirement handling process which will make the process more formalized and effective. The important aspects of this report will be, to make sure that all possible endpoints are improved. This will make the process more seamless and effective and hopefully lead to business benefits.

1.1 Assumptions

- The customer can send several use-cases together.
- The company has no formalized requirement handling in the beginning.
- The company does not prioritize tasks in Jira.
- The company does not verify with the customer about the new requirements.
- Only project manager and project specialist can access the database, but the specialist is the only one who can edit.

2 Improvement context

The company in this regard will be some type of business, which only works with customers whom already would have a system, up and running. This assures that the flow of the BPMN only is possible to be used by an existing customer of the company. This is what we are focusing on, however we're not excluding further possibilities for expansion. The product we would have in mind in this context, would be to add new functionalities and other upgradeable components to an existing system. The process starts with one of the customers providing one or more use cases to the company. The company then starts discussing it with the product specialist and deploys it just as is.

The problem with this method of handling requirements, is that the customer never reviews the newly developed and deployed functionalities. The new functionalities may not correspond to the true wishes of the customer. This can i.e. lead to retracting the deployed system and make alterations, which is time and cost inefficient. This makes the resource heavy project completely pointless and unnecessary, because it simply not what the customer wants nor needs.

2.1 Company

We have chosen a fictional company with the name "Stratton Oakmont". So in the report we will describe the changes that happens to the company. First of all the company is based in Silicon Valley, and has around 200 employees. And it mainly specialize in technology, development and consulting. With consulting they mainly work with preexisting systems and works on implementing new features on to them. Thus making them to a company which strictly works through a retainer scheme. With them being the consulting team of e.g. a small start-up, with that in mind, they will keep in touch with start-up companies. The company in this case will listen and improve on the given problems/features they want to remove/implemented on the existing system. They also create custom systems to new and old clients which is handled by different divisions, but in this report we will keep a distance from those cases and only focus on the use case and requirement handling improvement.

2.2 Actors and stakeholders

To make the structure as generalized as possible we have chosen to merge all the possible actors from the customer side into the actor "Customer", this to prevent all the different roles a potential customer might have. In the company there may be a lot of different stakeholders which may vary from stakeholders(business) to the company manager, but the ones the report would focus on are the product manager, product specialist and developers.

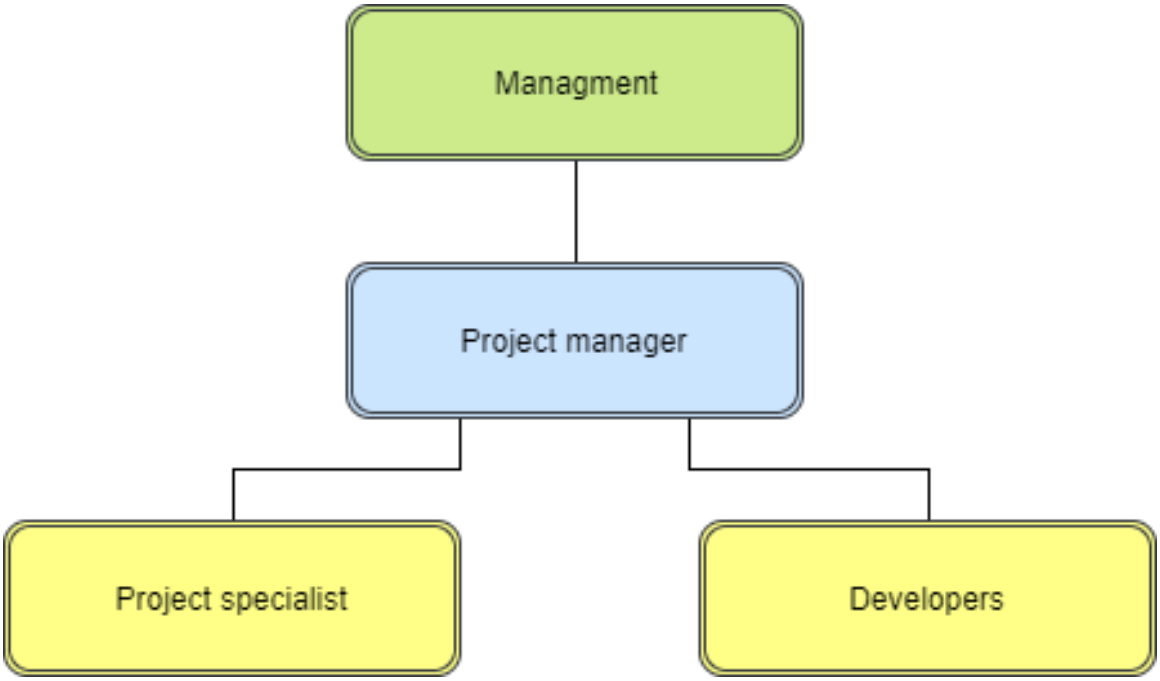

Figure 1: The company's role hierarchy

- **Management:** Would be the upper part of the company who would not have any direct relation to the customers.
- **Product Manager:** Is the actor whom has interactions with the customer.

- **Product Specialist:** Competent employee whom usually have a lot of options, regarding how the new and old structure would be after given changes.
- **Developer:** The one creating the actual new or revised changes to the system, or in some cases creating new systems.

2.3 Old BPMN model

The figure below describes how the work flow used to be in our fictive company. The work flow used to be very simple and we will describe each task below.

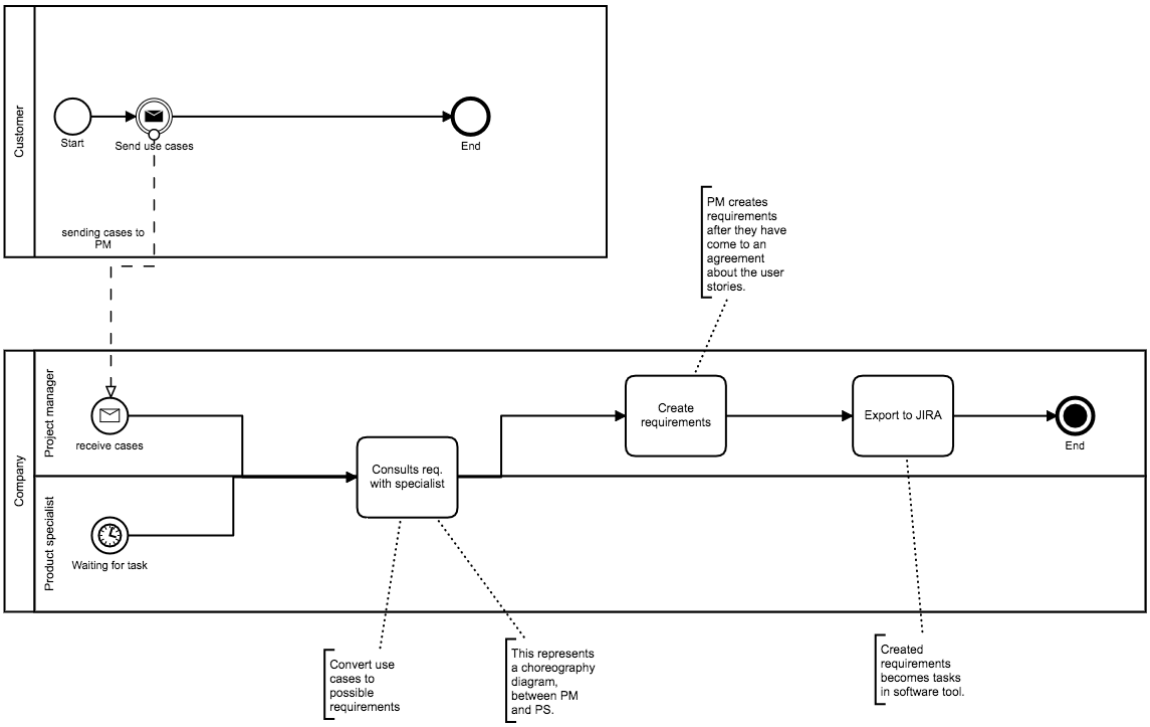

Figure 2: BPMN model

1. **Consults requirements with specialist**

   After the company receives use cases from their client, the project manager consults the use cases with product specialist to discuss the possibility of converting the use cases to requirements. Whether it is possible to create requirements out of the use cases or not, their client

does receive a feedback about it. Even if the use case can not be converted. However if they received feedback on the inconvertible use cases, they would have the possibility to reformulate their use case and make it better so that they might be possible to be converted.

2. **Create requirements**

   Use cases that are possible to convert to requirements gets converted in this step of the process. It is the project manager that converts them into requirements. It is only after the project manager and product specialist come to an agreement about user stories that the project manager gets to do this.

3. **Export to JIRA**

   All requirement that has been created, gets exported over to JIRA. JIRA is an issue and project tracking software. In JIRA the requirements become tasks for the developers to develop. The project manager can get track of every task and which ones that are in line for being developed, which tasks are in progress, which ones are in testing phase and which tasks are done. Since the tasks does not get prioritized after they get created, the developers does not know whether a task is essential or not. That makes it harder for developers to know whether if the task must be implemented first, or wait as they might not be essential part of the whole system.

2.4 Problem

The problem was discovered when they hired a new employee, who claimed that their former employers had a more formalized require management. Which made the former employers more competitive and productive. Since the company have tough competition from other companies; they're looking for ways to improve.

The process model that the company use for their requirement handling is an unstructured and unformalized process with a lot of potential. It is unformalized because every time a new project is started, they will start the whole process over. The previous documenting and requirement handling, have been made for either same or other customers will not be reused. But if there are some people in the current project that have been involved in similar projects before, then they might discuss

their experience. The new requirements might not necessarily be the same, because they create requirements from scratch every time. In other words, the requirements have to come straight from the horse's mouth.

2.5 Improvement Goals

From what the new employee discovered, regarding the problem with formalization; Is that the company have a really ineffective way of the creations of new requirements in their requirement handling process. However there are a couple of issues that are identified within the company that needs to be addressed and improved. Some of the issues are:

- Non structured work flow which leads to potential irregularities that causes more delay and overuse of resources.
- Lack of communication and verification between customer and company.
- Cannot reuse requirements.
- Each developer has a monopoly on knowledge of previous requirements, since there are no process the other actors can access this.

The **goals** need to be formulated in a way that they address the issues. The goals are:

- Reduce time-to-market for new software products by 20%.
- Reduce the necessary resources to initialize a project by 50%.
- Effectivize process of creating requirements by 50%.
- Create a system/storage to save previous requirements.

These are the main problems that the company faces today, we have listed some goals that should fix the problems. The company also mentions how big of difference should this new model bring to the process by percent. These goals will be implemented as soon as possible and we will, later in this report discuss how, when, by who and why.

3 Suggested changes

3.1 Database

Firstly the company needs a way to store previous requirements. A database or a storage would be suitable for this change. The database will be a good way to store the previous requirements, which would be a valuable asset in the long run for the company. The new requirements created will also be stored in this database, if they are not already in the database. With only storing requirements that are valid, we make the process faster and the flow smoother. This is to make sure that all the information stored in the database would be accepted by customers and in turn gain some trust to the specialist. Some of the time spent on formulating a new suggested requirements, should be spent on finding similar requirements from the database so that it can save time on similar new requirements.

3.2 Communication with the customer

One big issue in software development is the miscommunication between the customer when making requirements. Empirical data show requirements are often misinterpreted by the developing company. To solve this problem, the company should consult back with customer to make sure the right requirement is being developed. Even though this is excessive to some extent; it's a way of quality assuring, which would lead to a better company reputation. [5]

3.3 Prioritize requirements

Another issue the company has with requirement handling is the lack of prioritizing tasks. Without a proper system for prioritizing the tasks, the customer will not get the most important features first. The main functionality of the software and security is most likely the most essential parts of any software, thus highest prioritized. While the color of a certain button insignificant feature, compared security at least, so that it would be prioritized lower. [2]

4 Implementation of process changes

4.1 Implementing database

- When: Importing requirements from the database will be done in the beginning of the process, of creating new requirements. Exporting to a database will be done in a last period of the process, and more specifically right before exporting to JIRA. This assures the company that the input would be formalized.

- Who: For everyone involved in the project, but especially for the product specialist, since their job would revolve around this process. The product manager will also have access to this database, but only the specialist will be able to add requirements and tasks regarding the database. Although everyone will benefit from this process change, the client will get better feedbacks at a faster rate, which can make the process better.

- How: After receiving use cases from a client, the product specialist and project manager meet up to discuss the use cases. If the use case is realistic it gets passed, but if some of the use cases are not realistic then the database would be checked for suggested alternatives. And if they are not realistic at all, they will be voided. The requirements are then exported and saved in server so it can be used later. Although in the beginning, the change might not be noticeable and might cause some extra work, but the more it gets used, and the more requirements that get stored in database, the more effective the process will become. An example of how the data is stored in the database is shown in table 4.1. Here the first column shows when the requirement get stored in database, second column describing the requirement and third showing which specialist worked on that requirement and sent it to database.

Date	Requirement	Specialist(s)
19.10.2017	The system shall provide the ability to sort all contacts in ascending order	Johan
20.10.2017	The system shall provide the ability to use Facebook as login	Brish
21.10.2017	The system shall provide the ability to search for Instagram users in the app.	Tanu

Table 1: Description of how the data is stored

4.2 Communication with the customer

- When: Communicating with client will be spread throughout the process. In the current process it starts with clients sending their use case for new functionalities, they want to be added to their system. Through the iteration it would be wise to check if any of the use cases are unreasonable, and check with the customer for a revised version of that use case or in some cases discard that case. And to make sure the new requirements are accepted by the customer it would be wise to roll a full on iteration; which guarantees the acceptance of the customer, before the company does any kind of alteration.

- Who: Communication of this process is mostly between the client and the project manager. This will now happen in a more frequent way, in which will guarantee a more acceptable result to the customer minimizing wasteful work.

- How: If a use case from client is not realistic, then the company will communicate with the client to suggest a more reachable use case. As per agreed the client may get their feedback. If the client agrees, then the use case will move further in process and be made into requirements. But if the client does not agree, then that use case will be discarded, but the company will still get notified.

  Also after creating requirements from use cases that the client approved, they will be sent to clients to make sure that the clients agree to the requirements, so that it is no misunderstanding between the clients and the company. If they are approved, then they get prioritized

and exported to JIRA. If the clients do not approve, they get reformulated and sent to the clients for approval again. Figure 3 shows the part of the updated BPMN model, where the company communicates with the client.

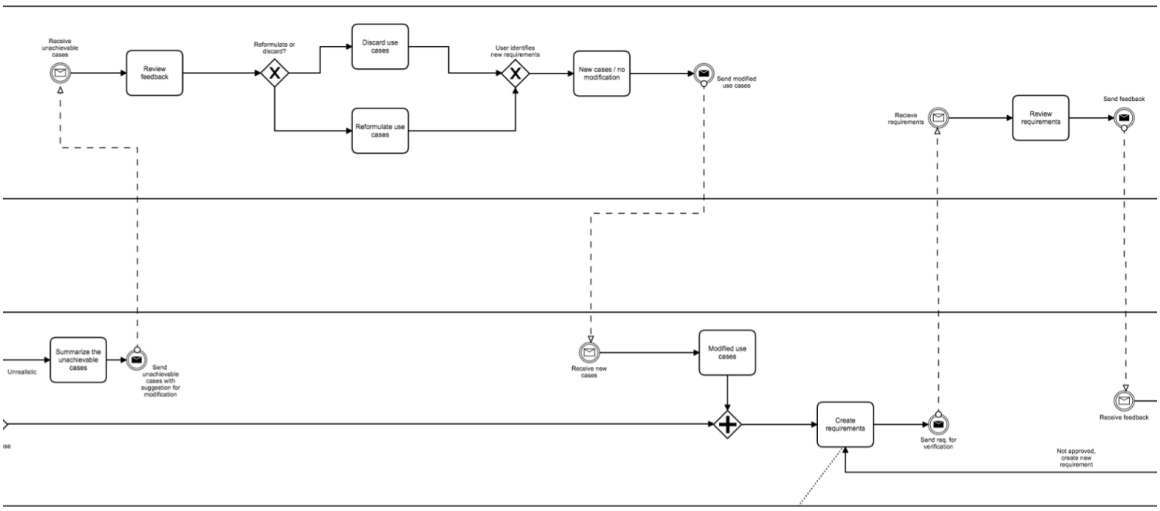

Figure 3: Screenshot of a section of the updated BPMN-model. See figure

4.3 Prioritize the requirements before JIRA

- When: This step happens almost at end of the process, right before exporting to JIRA which is the last step before the end state. It cannot be done before the the requirements are approved by the clients. It is important for the developers to know which of the features that needs more attention.

- Who: The project manager prioritizes the requirements before exporting to JIRA for developers to start developing the system.

- How: After the requirements that has been approved by clients are sent back to project manager. The project manager, who knows which features are essential for the clients system, makes sure that essential features are the highest prioritized requirements, while features that are not important to the clients are prioritized lower. Depending on the prioritization of the requirements, the developers will first focus on the higher ones as one would expect. The prioritizing will be done by color coding the highly critical tasks as red, critical as yellow and not critical as green. Figure 4 shows the backlog of JIRA with color coding.

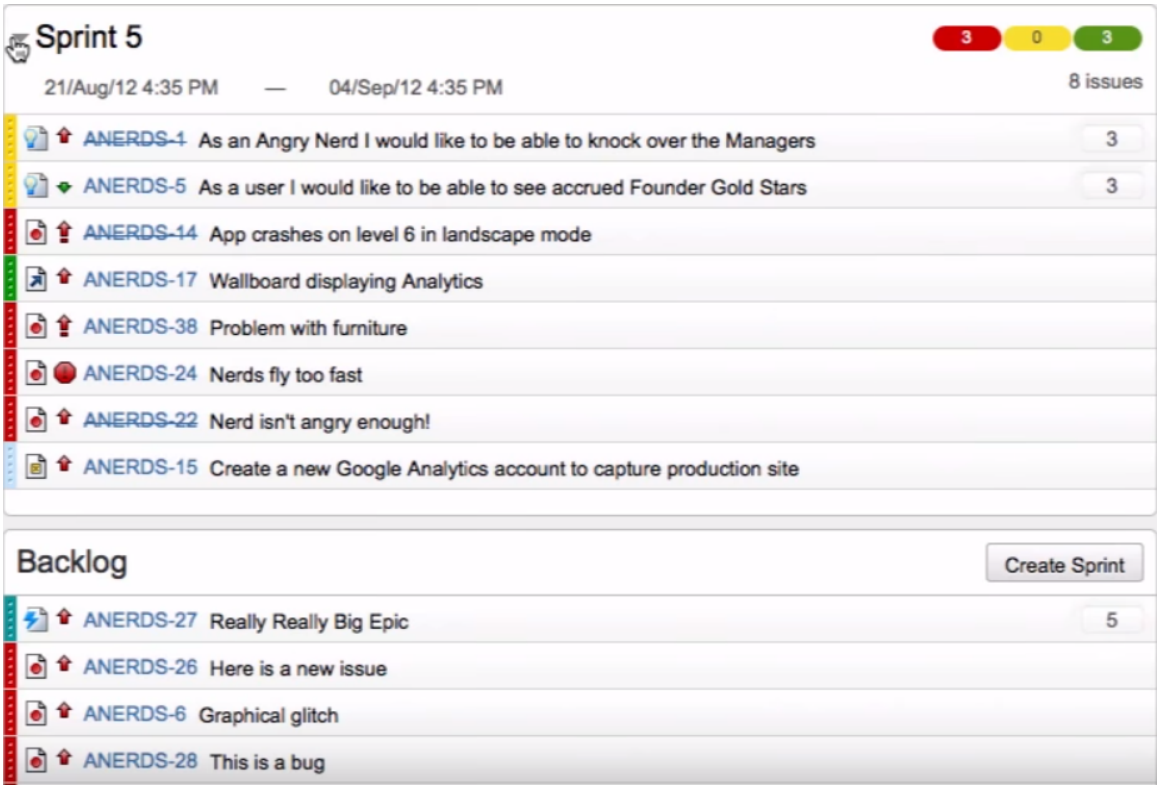

Figure 4: JIRA Scrum backlog [6]

4.4 Improved BPMN model

In this part we are going step by step to tell how the new BPMN model is setup.

4.4.1

This part is quite the same as the old BPMN model, with the difference being that the product manager can now read from a requirement database, and the product specialist can write and read from it.

4.4.2

Checks if the realistic use case, if then go to the next step. If not the product manager will send the unrealistic/unachievable to the customer and wait for the customer response.
So the process when meeting one or more unattainable cases is that the customer will get a full summarization of the unachievable cases and they will review it and check if any use case is not attainable. Afterwards they go through the process of eliminating or reformulate those cases, and sends now the reformulated set of cases. And this would be sent to the product manager back again.

4.4.3

When one of those two cases are fulfilled, the requirements would be made. With that done, the product manager would send the result to the customer for verification again. When the customer receives it, they then proceeds to evaluate this to check if this is a what they envisioned or something they can accept in that matter. After the evaluation is completed the customer sends back their evaluation. When the product manager receives this, he will first check if it's approved, if so we can proceed to the next step. If not the product manager would go through and reformulate the requirements and send them back and forth, till the resulting requirements are approved by both parts.

4.4.4

In this part the requirements are prioritized and sent to the database for the product specialists to access, and the requirements are sent to JIRA.

5 Monitoring and Control

5.1 Reduce time-to-market for new software products by 20%.

Purpose: Decrease time to release.

Issue: Time inefficient

The person **whom** will collect/report data in our company will be the Project Manager, who has direct contact with the customers. The product specialist will also have data on all the requirements, but to reduce time we will look more into the data that the project manager will collect. The time will be checked after the project is done, so that we can see **when** the goal is achieved or if the time actually was decreased. The data will be corrected approximately each time the project takes longer or the same time as earlier projects. **How** this will be estimated and checked, is by comparing the time from earlier project with the old work structure, with new projects created by using the new improved process. The company has their own tool that the product manager will use to collect data, which will collect necessary data such as time when project started and ended. This estimation will not be 100% correct because we do not redo the old projects. We could do this but that will take time, which is time that we would like to save.

The solution mention in earlier sections, where we suggest to implement more validation and verification. This is one of the solution which could help the company to decrease their time-to-market, because they can try to avoid communication errors and actually see that the product they made matches the demand. The solutions mentioned will help to reduce the overall time used to build the expansions and fixes the problem the clients have encountered.

5.2 Reduce the necessary resources to initialize a project by 50%.

Purpose: Improve the way new tasks are started on.

Issue: Inefficient structure for starting

The persons **whom** this would affect is the project manager and the product specialist. Both the manager and specialist will be responsible for collecting data. The data will be collected each time the start process is complete, which in our case will be after the use case is given to the project

manager. Entities such as time, resources and employees used, etc. will be the data in this case. In this goal, the data might not be corrected as much because, the data should be minimized after redefining the process/model.

This would be in mind that the "bottleneck" of the current process currently is the interactions between these actors(Project Manager and the Client). What we then have in mind in this process is to improve the process, making it easier time and resource wise. This will mostly go about to cut the unneeded resources and make the process better by implementing a structure through lean thinking. Usually initiating these kind of tasks for the company have been highly ineffective, and created waste of time and effort which could rather be spent on handling another requirement. To illustrate this better, the old process was primarily built upon that the Project Manager and the product specialist casually talked around the project and finds ways to make their cases as fast as they seems fit. Which is the basis of why the process currently is unorganized, with correlation to that these actors may be replaces someday. Also, that in a way the same actors would not be in charge of the same structure each time, which results in different ways of things are done. This is based on that a Project Manager can communicate with different product specialists whom can work on different tasks at a time. So **how** we can do this is to make a formalized process which assures that both parties must go through certain steps to make sure the outcome is of better quality.

5.3 Improve process of creating requirements by 50%.

Purpose: Minimize effort and time regarding requirement creation.

Issue: Time and resource inefficient.

By having a system/storage for storing and retrieving older requirements, there are no longer need to create the same or similar requirement over and over again. Why make the same requirement again if it has been made already? It will only use more resources and time that the company may not afford, "Why reinvent the wheel?". The product specialists will be the ones **whom** will access this storage facility and work on it so that the creation may be done as fast as possible. The specialist will also be the one who is responsible for collecting and reporting data. How often this is corrected

will vary, because it will not always be data to be correceted. The goal will be accomplished **when** the requirements are set, so that we can see if they reused some of the old requirements, or need to make new ones. **How** this will be done has already been covered in the "Database" part(4.1), but the way it would complement the tasks is the thing we are trying point out.

5.4 Data quality and validity

The data quality and validity will be ensured by the product manager who will be the person to confirm the requirements with the customer. The approval will decide if the sets of characteristics of data fulfills the requirements and ensures that the data is complete, accurate, secure and consistent. This will be confirmed in a way described in section: 4.2. We make sure that the company has formulated a requirement that suits the customer; and that it has the specifications the customer actually wants.

The possible challenges with data collection and data validity, might be problems such as irrelevant or duplicate data. All the data collected might not be as important as the Project Manager and Product Specialist thinks. What if the data collected is misinterpreted or if the data collected isn't enough? Or that one of the staff may have documented the data poorly. These problems are one of the most common problems when we talk about data collection. Our company may not ensure the data collection, but the validity is more ensured because of the communication between the customer and the company.

Goal	**Question**	**Metric**
Goal # 5.1	How much time is being used, compared to before?	# Number of days
Goal # 5.2	What is the difference between initial resources and actual effort from before?	# Numbers of days
Goal # 5.3	How much time is saved after implementing the database?	# Number of hours

Table 2: GQM model

5.5 Technical debt

5.5.1 Definition

This refers to the negative consequences of making shortcuts, which will in the long run make complication and make the "saved" resources wasted. Which in turn might even worsen the original state.[4]

5.5.2 Comparison

With this in mind, the new process would exclude a lot of potential cases which would most likely prevent the deployment of an unfinished or unsatisfying requests. Which in turn would have been done anew or in worst case, would make it so that the customer loses confidence in them.

6 Discussion

The improved BPMN (figure 8) is the company's main baseline when it comes to requirement handling in a current or finished project. Doing this to every requirement in the start of the project, would inevitably lead to a failed project.

6.1 Applied methods and Further work on the database

Formalized requirement handling.

The database can be used to more than only requirements. Because of the format, it can easily be integrated to other applications such as HP Quality Center(Test application manager). This makes it easier for the testers, to get hold of the requirements and for the product manager to track the project. In addition we suggest adding more data to the database, such as how long time it took to complete the task. This is also valuable for the new employees and product manager, to estimate how long a task will take to finish (figure 8).

Date	**Requirement**	**Developer(s)**	Finished	Time
19.10.2017	The system shall provide the ability to sort all contacts in ascending order	Johan		5
20.10.2017	The system shall provide the ability to use Facebook as login	Brish		3
21.10.2017	The system shall provide the ability to search for instagram users in the app.	Tanu		4

Table 3: This is an example of the possible storage system the company can have in the future.

6.2 Rationale for Changes

The biggest issue the company had, was the way they handled the requirements. Requirements are the building blocks of a project, therefore it should be dealt with carefully. Summerville says *"We have not unequivocally demonstrated that RE process improvement leads directly to business benefits.This is extremely difficult in a short term project. However, the evidence that was produced* ***strongly suggested*** *that participation in a process improvement process resulted in* ***benefit*** *and it is likely that at least some of this benefit resulted from better requirements engineering practice."* [1].

This strongly implies the correlation between business and good requirement handling.

Another important reason for change is that, product manager and product specialist in the company now can look up previous requirements. Storing information such as test cases, time etc. In addition to requirements that can be crucial for the improvement of efficiency, for the company. In other words, the database can be used as to store historical data for the company. This is a good baseline for the company, and opens up a lot of possibilities and innovative options in the future.

6.3 Risks

6.3.1 Slow start

One of the major risk of this updated BPMN-model (see figure 8) is that the process is not more effective than previous model. Since it is more tasks to go through and communicate multiple times with the client, it makes the process less effective. However what it does, is that it gives the end product more quality, we risk efficiency over quality. But this is only the beginning, the more the process gets used, the more the requirements that gets stored on server, the faster and more effective the process will be. So after a while, the process will be more effective and add quality to software.

6.3.2 The risk of technical dept

In matter of technical dept our task fit the bill to the teeth, in short we can say that all the improvement of requirement handling shortens the processes in the long run. With it cutting down a lot of potential wasted time and resources trying to cope with cases or requirements, that either the company or the customer might not approve of. This is mostly because of how the new model spares nothing when it comes to verify that the customer is on board of the thoughts of the company, this vastly increases the success of the software development in the long run. This because of how ill mannered the old BPMN model was, where it potentially created a lot of cases the company had to remodel or in some cases throw away the initial request.

6.3.3 Silver lining between old and new requirements

A problem the company is going to deal with after some time, is the reuse of requirements. If the project manager is lazy and chooses to reuse the requirements. For example, a case where the requirement should have been made from scratch, the implications of this might be the wrong features being developed. If this is done multiple times, it can damage the company's reputation, because of lazy leadership. This also makes the company vulnerable for lacking of innovation, which is the opposite effect of what's wished upon.

7 Conclusion

The problem was that the company had a unformalized process for requirement handling from the customer. Which made the overall process a lot slower and in the end ineffective. And resulted in that they lost in terms of efficiency, compared to their competitors.

After applying all the changes, we believe the company would have a significant improvement in their requirement creation process. All the methods which were used, made the requirement handling better and made the company better positioned against their rivaled companies. Which we illustrated in the old and new BPMN model, where the given improvement could be made.

From what we have seen it would be safe to say that by utilizing lean working methods, and improve upon processes would improve a lot of processes. And that this should be done with proper analyzing and planning, so that the outcome would be as good as it can.

8 Improved BPMN

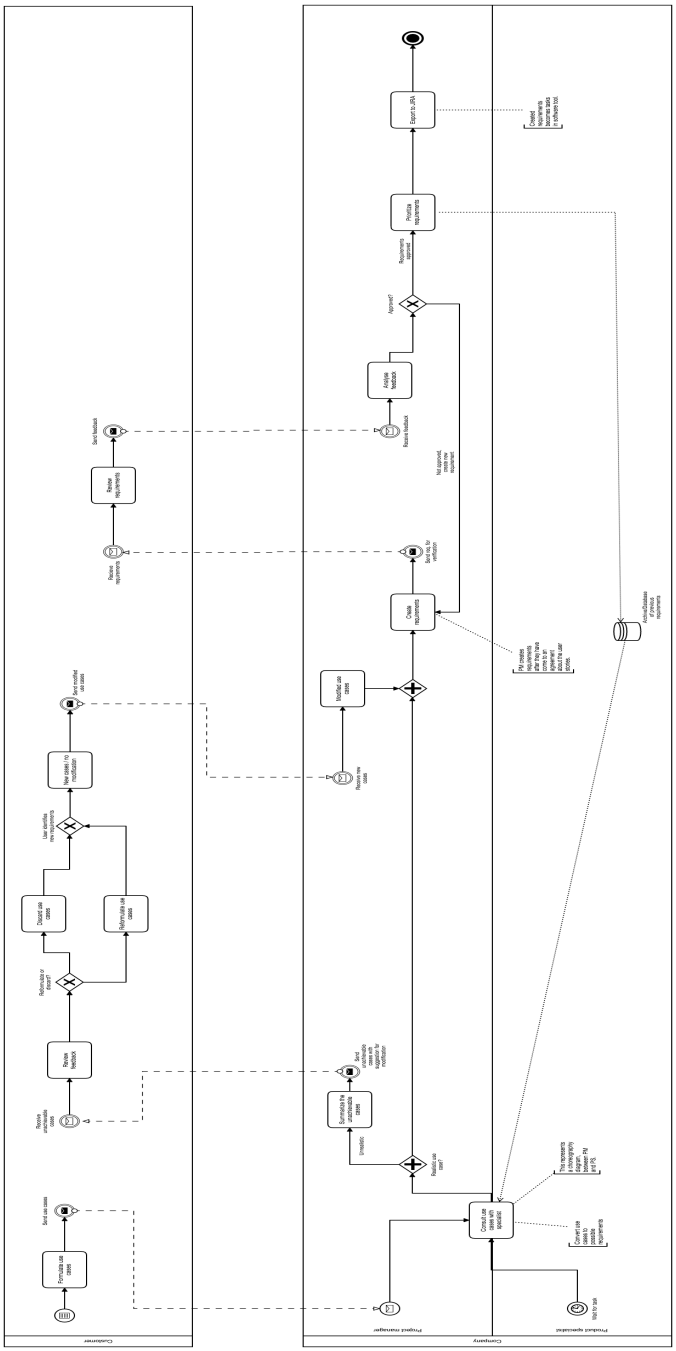

Figure 5: Improved BPMN model

References

[1] Ian Sommerville and Jane Ransom

An Empirical Study of Industrial Requirements Engineering Process Assessment and Improvement.

[*This article describes an empirical study in industry of requirements engineering process maturity assessment and improvement.*].

Published in: Journal - Transactions on Software Engineering and Methodology (TOSEM) (2005).

[2] Sarah Beecham, Tracy Hall And Austen Rainer

Defining a Requirements Process Improvement Model

[*This paper describes how the requirements engineering (RE) process is decomposed and prioritized in accordance with maturity goals set by the SEI's Software Capability Maturity Model (SW CMM).*]

Software Quality Journal: September 2005, Volume 13, Issue 3, pp 247–27

[3] Witold Pedrycz, Joana Iljazi, Alberto Sillitti and Giancarlo Succi

Prediction of the Successful Completion of Requirements in Software Development—An Initial Study

[*In this work we have presented how we can analyze the evolution of requirements to predict those that will fail, thus building a model that could to decide where to drop the effort of developers.*]

Conference paper - First Online: 01 June 2016

[4] Antonio Martini and Jan Bosch

The Danger of Architectural Techinal Debt: Contagious Debt and Vicious Circles

[*This paper shows dangers of gaining Technical Debt.*].

Published in: Martini, Antonio & Bosch, Jan. (2015). The Danger of Architectural Technical Debt: Contagious Debt and Vicious Circles. Proceedings - 12th Working IEEE/IFIP Conference on Software Architecture, WICSA 2015. . 10.1109/WICSA.2015.31.

[5] Kamsties E. Aurum A., Wohlin C.

Understanding Ambiguity in Requirements Engineering.

[*This chapter illustrates that ambiguity is a serious problem of natural language requirements documents...*]

Engineering and Managing Software Requirements. Springer, Berlin, Heidelberg(2005), pp 245-266

[6] Atlassian Corporation Plc

JIRA in a Nutshell demo video

[*An overview of JIRA: issue and project tracking for software teams.*]

url: `https://www.youtube.com/watch?v=xrCJv0fTyR8#t=2m58s` (*screenshot* from: 2:58 - 3:02)