UNIVERSITY OF TWENTE.

Improved customer relationship by self-service

Martijn Koopman
Bachelor Industrial Engineering and Management
1st supervisor: prof. dr. M.E. Iacob
2nd supervisor: dr. L.O. Meertens
Supervisor from company: P. Verkroost
10-12-2017
University of Twente

Management Summary

Motivation of research

This report describes the investigation that is conducted at the company A in Enschede.
In this investigation, problems around the digital interaction with the customers of company A have been mapped. The customers of company A lack insight and overview of the projects they assigned to company a. As a result of this problem, the following research question has been stated:
How can the interaction process with customers be made more efficient with the help of an online portal?

To develop a portal which makes the customer interaction more efficient, the methodology of Peffers will be used. (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007)
These steps have been followed throughout the report in developing the online portal.

Definition of solution objectives

The portal should solve the lack of overview and insight in the projects of the customers. Currently, customers receive a PowerPoint where the progress of their project is shown. This PowerPoint often lacks information and it takes some time to build the PowerPoint itself. Objectives for a solution were defined by conducting research on CRM, BPR and web self-service. To make the customer interaction more efficient, the back office, the information stored in internal system, must be transferred and displayed to the portal. This increases the overview and insight in projects for the customers.

Demonstration

The portal has been developed around another external application. This has many advantages such as the ability to login in with the same credentials and all the functionalities of the other system. The portal has been built in Mendix. Information about the project will be send from the internal system, through eMagiz, a program which allows applications to interact with each other, to the portal. Each message consists of, one project, with *project lines* (phases within a project), with *hour registration lines* (lines that are filled by employees to justify their hours) and with the employees who filled in these *hour registration lines.*
Once the message is ready in the portal, the customer receives a notification by e-mail. By clicking a link in the e-mail, the customer will be re-directed to the portal.

Conclusion & Evaluation

For this application, the focus was on the insight of project, but there is more information in the internal system that can be displayed on the portal. Examples of this are invoices and contracts. (Geevers, 2017). It would be valuable if this functionality could be implemented in the portal. Another valuable functionality could be the insight into invoices. With this information, in combination with the project overview, the customer gets a clear image why they need to pay that amount of money.

With the help of this portal, the interaction process with the customers has been improved. Customers are well informed, since they can monitor their project more detailed than before. The portal is easy to use as well. The project history is visible in the portal, so that the customer is able to track their project development. Furthermore, the project manager does not need to make the PowerPoint anymore, he simply clicks on a button which sends the message to the portal. The portal itself could gain many more functionalities so that more information from the back office of company A can be shown in the portal. This investigation and prototype only contains a small part of the information, other functionalities could not be implemented in the time given for this investigation.

Preface

In front of you lies my bachelor thesis that I have written during my period at company A. I look happily back to research that I conducted during this period. I can state that I learned a lot from this period. I gained a lot of knowledge about Mendix, eMagiz and customer relationship management. Furthermore, I learned a lot from working at a company, especially taking initiative in getting answers to my questions. At the university, things are handed to you, but when you are conducting research at a company, you need to take initiative, otherwise you get no responses and results. These experiences were very valuable for my personal development.

I would like to thank company A, my supervisor Pieter Verkroost and all the employees for the help they provided during my period there. I could not have done this without their help. If I had any questions, there was always a person willing to help me even if they were busy with other things. I would also thank my supervisor from the University of Twente, Maria Iacob, for her advices and feedback concerning my investigation and this thesis.

Martijn Koopman
10-12-2017

List of Abbreviations

Full Name	Abbreviation
Business Process Re-engineering	BPR
Customer Portal	CP
Information System	IS
Service Point	SP
Customer Knowledge Management	CKM
Customer Relationship Management	CRM
Enterprise Architecture	EA
Frequently Asked Questions	FAQ
Key Performance Indicator	KPI
Knowledge Management	KM
Request For Change	RFC
Self-Service Technology	SST
Service Level Agreement	SLA
Web of Science	WoS

Table of Contents

Chapter 1: Introduction, Problem identification and motivation

This investigational report starts with an introductory chapter. In this chapter, a general introduction to the company and the subject of my investigation are given. Furthermore, the structure of this report is elaborated in this chapter. Lastly, the current situation and the problems associated to this situation are identified.

1.1 Brief introduction of the company

Company A is a fast-growing company which provides IT-solutions for other companies. Company A tries to identify the real problem within the organization, by spending time at the customer. They try to map the process within the organization and with that overview they are able to identify the core problems within the organization. They try to solve these core problems by building integrated model driven software solutions. (such as Mendix or eMagiz)

1.2 Research method and structure of the report

The methodology of Peffers has been chosen (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007), to structure this report. This methodology is especially created for the development of information systems. This methodology was chosen over the MPSM (Managerial Problem-Solving Method) (Heerkens & Winden, 2012), because this approach focuses on solving a problem by doing investigation. In my case, it is necessary to solve the problems by making a prototype, so developing something visible. This part is not elaborated in the MPSM.

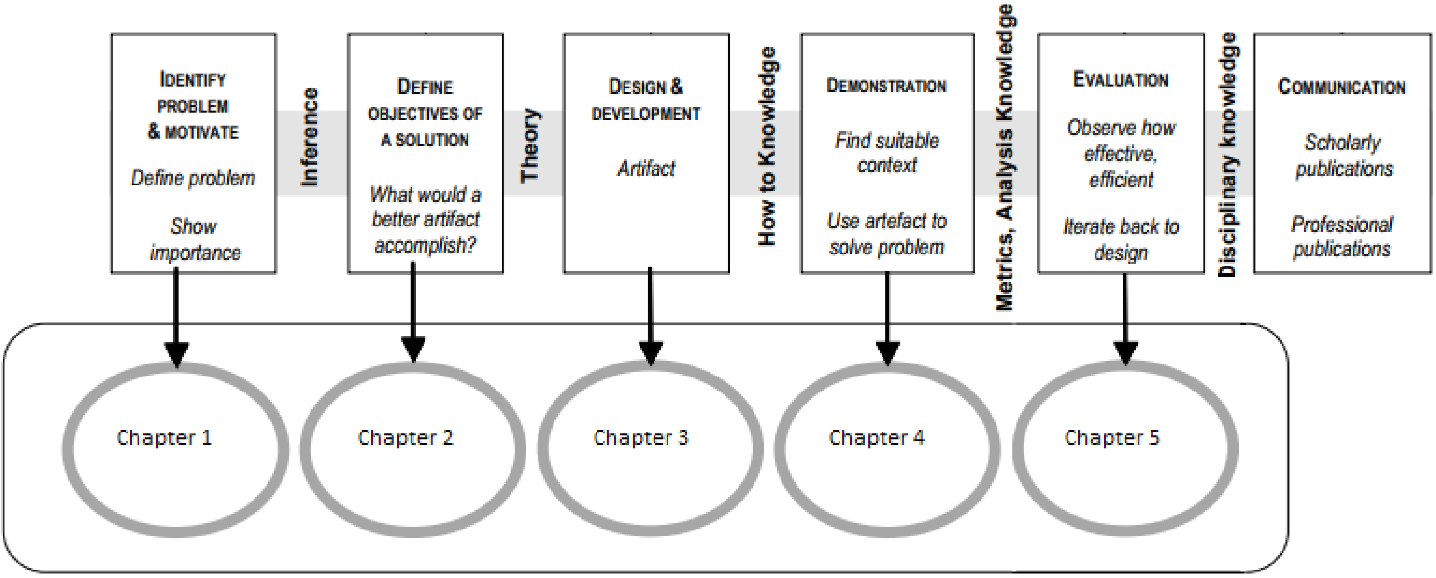


Figure 1: Schematic overview Peffers methodology

The methodology of Peffers consists of six parts as shown in figure 2. Firstly, problems are identified and a motivation for research is elaborated (chapter 1). After that, the defines objectives for a solution. (chapter 2) This is done with the help of existing theories such as literature or other information sources. Subsequently, an artefact will be developed according to the objectives set in the previous chapter. (chapter 3) Next, the artefact will be demonstrated with screenshots. (chapter 4) In the end, the prototype will be tested by users and these users will also be a source of feedback for further improvements. (chapter 5) The last block in the figure, communication, is the report that is written about the research.
As mentioned above, this chapter will continue with the explanation of the cause of research and the problems that are associated with that.

1.3 Motivation of research

As mentioned before, company A is a fast-growing IT company. They attract more and more customers due to the fact that they become more prominent amongst big companies. Within this increase of clientele lies a challenge for company A. Company A feels like they can do more for their customers when it comes to relationship management, since company A has a lot of information for, about and from their customers.

This exchange of digital information flows is currently not efficient. Things are being executed manually or being done twice and has become a bigger problem, since company A is attracting more customers. This phenomenon occurs almost in every contact moment (when company A needs approval or data from the customer or vice versa) with the customer throughout the process. The possible improvements associated to this inefficiency differ from the contact moment with the customer in the process. Company A knows that there must be a way to make these digital contact moments more efficient and easier for the customer.

All the problems associated to all the contact moments with the customers in the process must be identified, before company A can look for solutions for this problem. That is why this investigation will be about the problems associated to this process, potential solutions for this and at last the development of an artefact (application) which solves these problems.

1.4 Investigation model

In this part, the model that is being used for this investigation is elaborated. A visualization of the model will be shown and elaborated.

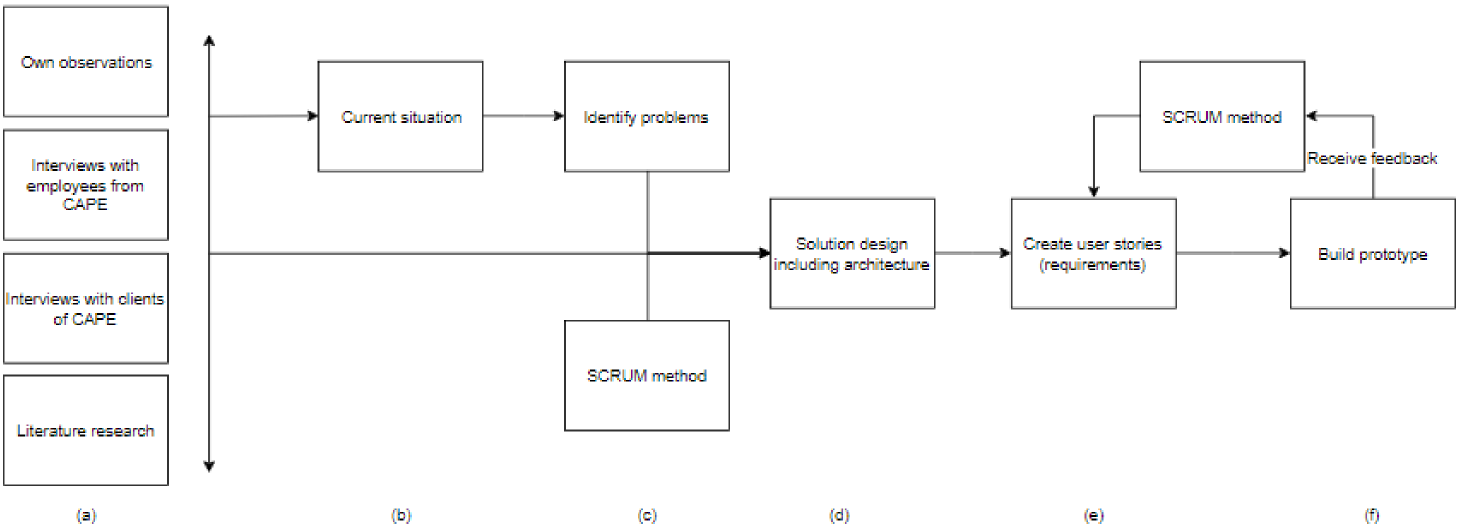


Figure 2: Investigation model

The investigation model is visualized below. A rectangle to the right of a horizontal arrow indicates knowledge that is gained from the rectangles on the left side of the arrow. A visual interpretation about the model is given in figure 3.

(a) An analysis of the available data within company A and from their customers. These data include interviews with employees from company A, interviews with customers from company A, literature research and my own view on the situation.

(b) All these data will be combined into a sketch of the current situation.

(c) Problems will be identified from this current situation.

(d) Combining the SCRUM method with the core problem, a solution design is made including an architecture.

(e) User stories or requirements are created in cooperation with the SCRUM method, the demands from company A and their customers.

(f) By selecting a few of these requirements, a working prototype is created. If this prototype is finished, it will be tested by potential users. These users will give feedback which can be translated to new user stories for expansion of the prototype. This process will repeat itself until there are no more user stories.

1.5 Deliverables

This research will result in the following deliverables:

- A bachelor thesis with my findings including recommendations for further research.
- A prototype of a customer information system solving the problem(s) mentioned throughout this thesis.

1.6 Problem identification

As mentioned in the previous paragraph, the current customer interaction process at company A must be elaborated. From this current situation, contact moments with a customer can be deduced. After this, potential problems that can occur during these contact moments are identified. In order to get an overview of these problems, a problem cluster will be made to identify the core problem of the situation.

As mentioned above, the current interaction process with a customer must be elaborated. The best way to do this is by making a flow chart, where all the events are shown in the right order. This flow chart is shown in figure 4.

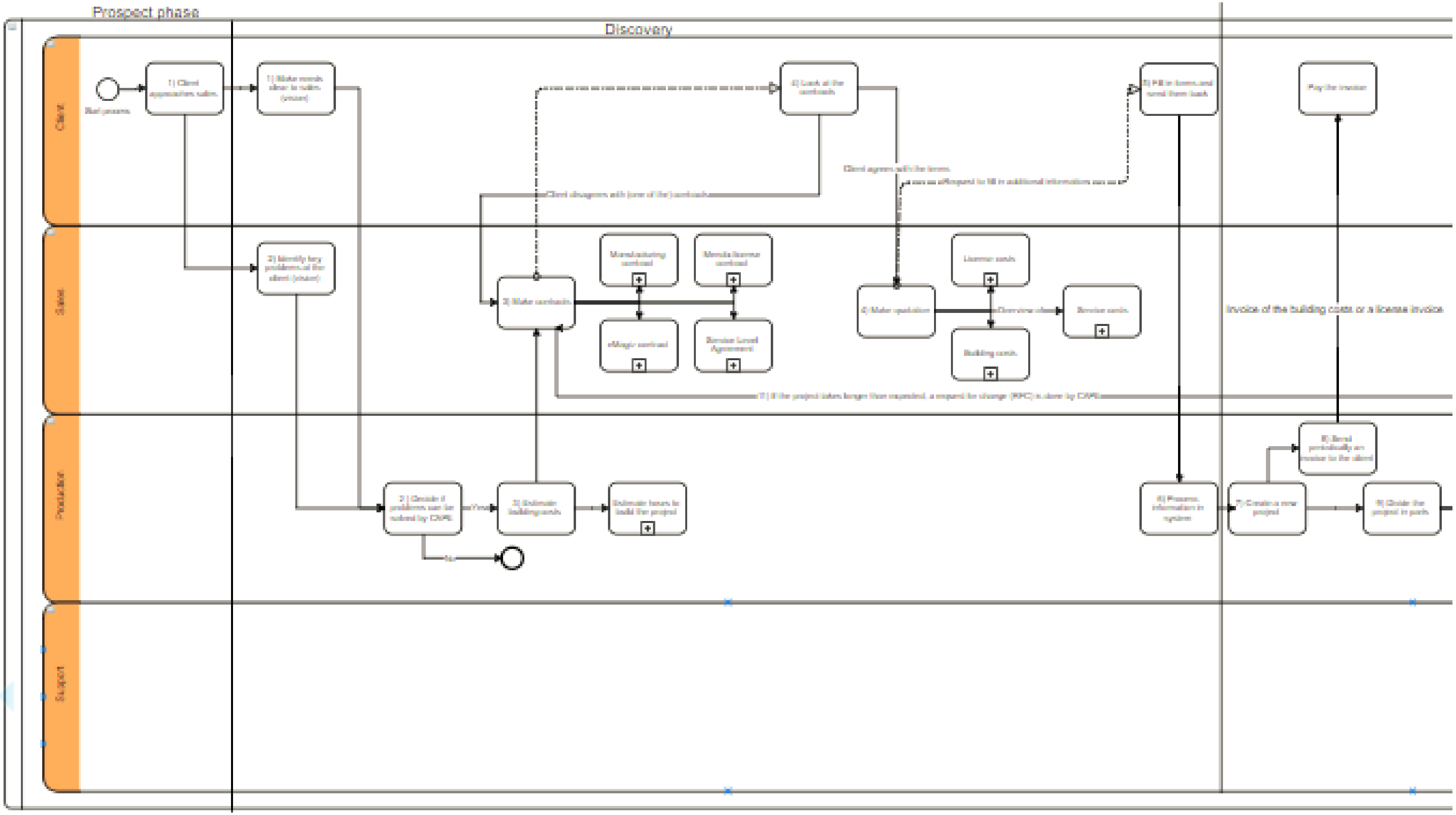

Prospect phase
Discovery
Client
Sales
Production
Support
Start process
1) Client approaches sales.
1) Make needs clear to sales (vision)
2) Identify key problems of the client (vision)
2) Decide if problems can be solved by CAPE
Yes
No
3) Estimate building costs
Estimate hours to build the project
3) Make contracts
Manufacturing contract
Morxis license contract
eMagic contract
Service Level Agreement
4) Look at the contracts
Client disagrees with (one of the) contracts
Client agrees with the terms
Request to fill in additional information
4) Make quotation
License costs
Building costs
Overview of costs
Service costs
7) If the project takes longer than expected, a request for change (RFC) is done by CAPE
5) Fill in forms and send them back
6) Process information in system
7) Create a new project
8) Send periodically an invoice to the client
9) Divide the project in parts
Pay the invoice
Invoice of the building costs or a license invoice

Figure 3: Current interaction process (1)

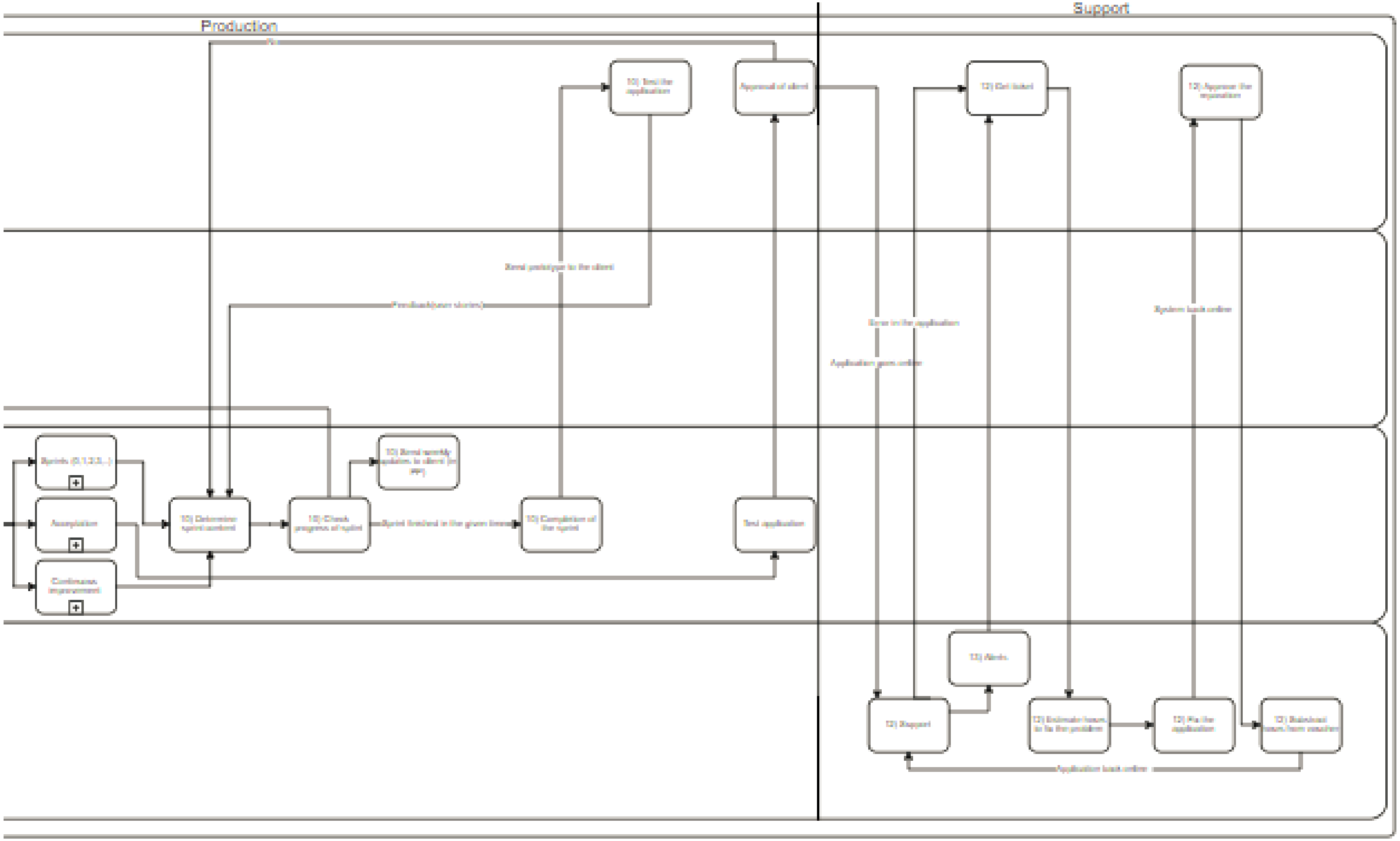


Figure 4: Current interaction process (2)

1.6.1 Current situation

Some clarification on how the process flow works:

1) Sales approaches or is being approached by a potential customer (or prospect)
2) A discovery is done by company A to assure that company A can help the customer. If it turns out that company A can help the potential customer, a message will be send to people within company A who have knowledge about setting up projects.
3) Sales and a project manager makes a contract for the customer with the terms and the estimated costs for the project. This is done by estimating the number of hours' employees of company A work on the project. At the beginning of the project the customer buys the license to use the created application.
4) When the customer agrees with the contract, a quotation can be made by the sales employee of company A. The contract is printed out from a PDF or PowerPoint file. After that it is sent by mail to the customer.
5) The customer signs the contract and fills in some general information that is needed for further communication.
6) This general information is processed manually by company A in their system to contact the customer when there are any difficulties.
7) When all this is done, and the customer agrees to it all, a project is created in the internal system of company A.
8) Invoices are send periodically to the customer. Normally this is monthly.
9) It is possible to divide the project into parts with the estimation the required man hours needed for the project. A project usually consists of 3 parts. (discovery, sprints and acceptation) A schedule is made with an estimation of the expected working hours on each part of the project.
10) After the planning is made, company A will execute the project. The project manager keeps track of the spend hours on every phase of the process. If the manager notices that certain phases take longer than expected and it is needed to revise the contract, a message is send to the customer. Currently this is done by sending a new contract to the customer. This process is somewhat time consuming, because it takes a couple of days before the contract finds his way back to company A.
11) As can be seen in the process flow, when there is a request for change (RFC), the process starts again at the creation of a new contract. A new tracking number will be allocated to the contract. All the other steps can be taken rather fast, since these steps have already been taken in the previous loop.
12) At last, company A has a support team which keeps all the applications of the customer up and running. The customer can buy a ticket at the support site of company A when their application breaks down. The urgency and a description of the problem can be filled in on this ticket. The ticket is send to company A and company A determines how much time (work hours) they need to fix the problem. If the customer agrees, the reparation of the application can begin. After the reparation, a message is send to the customer which states that their application is working again. The payment of this reparation can be done in two ways: company A sends an invoice to the customer or the working hours are subtracted from a voucher. This voucher can be bought by a customer after the application is launched for the first time. A customer can buy a voucher for 100 working hours for example. When reparation costs 10 working hours, then these hours are subtracted from the 100 on the voucher, which leaves 90 working hours for improvements or reparations in the feature. If certain errors occur very often company A

can decide in consultation with the customer, that the customer does not have to pay for the reparation.

13) Another function of this support system is that customers receive automatically a notification when the memory of their application runs out. This can have various reasons, there can be an error in the application or the memory is just full. (in that case, it is wise to buy more memory from company A) If the memory runs out because of an error, it might be wise for the customer to fill in a ticket, as mentioned above, to solve the problem.

1.6.2 Current systems and programs within company A

The systems that are currently being used by company A for this process are an Information System (IS) and Service Point (SP). Other programs that support these systems are Mendix, eMagiz, Sprintr and SharePoint. These systems will also be elaborated in this paragraph.

IS (Information System)
All the employees of company A fill in their hours on IS. This goes very specifically, because they need to justify this to their customers. IS can be compared to an ERP system. In this system, they fill in what they did during a day and how long they worked on a specific task. The data from all the employees who worked on the same project during the week will be gathered and an overview of the worked hours will be send to the customer at the end of the week. (in PowerPoint)
With the information from the employees and the set budget for a project, a comparison is made. This comparison shows how much hours were determined before the project and how much hours are actually spent on the project by employees of company A. At last, information about the customers from company A is being saved in IS.

SP (Service Point)
SP is the only external business service that company A is using currently. SP is a system which supports the application once it is online. Customers can request a ticket when they have problems with their application. An employee of company A gets a message that someone requested a ticket and he or she will look into it as soon as possible.

SharePoint
SharePoint is the database where all the documents are saved. Think about contracts, SLAs, licenses, employee data and invoices. Things can be added and retrieved from this database, if information is needed.

Mendix
Mendix is a platform that allows to innovate at web scale. This is done by enabling people to build, integrate and deploy web applications. Company A offers their customers solutions by using the functionalities of Mendix to improve business processes. An example for this is automatic an scanning process with the help of Mendix. When the device scans an item, Mendix will process the item and store these data in a table. Because company A uses Mendix for all of their projects, it will be used as well to develop the portal.

eMagiz
eMagiz is a program that is developed by company A. It allows consultants to make integrations between different models. So, when another Mendix application is needed when making your own, you can use eMagiz to extract the right data so that you can use it to make your own application. Knowledge about eMagiz is also needed in my project, because I need data from IS to make the portal functional.

Sprintr

The last system that company A uses is Sprintr. Sprintr is a digital SCRUM board. (see paragraph 2.2.6) When an employee of company A is working on a project, user stories are created by the product owner. These stories can be stored in Sprintr as the product backlog. The stories can be allocated to a sprint, when the sprint content is determined. A project member can indicate that they are working on a story by changing the status of the story from *to-do* to *in progress*. This will avoid the possibility that two employees are working on the same story. For this portal, Sprintr will be very useful. In Sprintr, a product backlog can be made (see appendix B), consequently the sprints can be filled from the product backlog.

1.6.3 Identification of contact moments

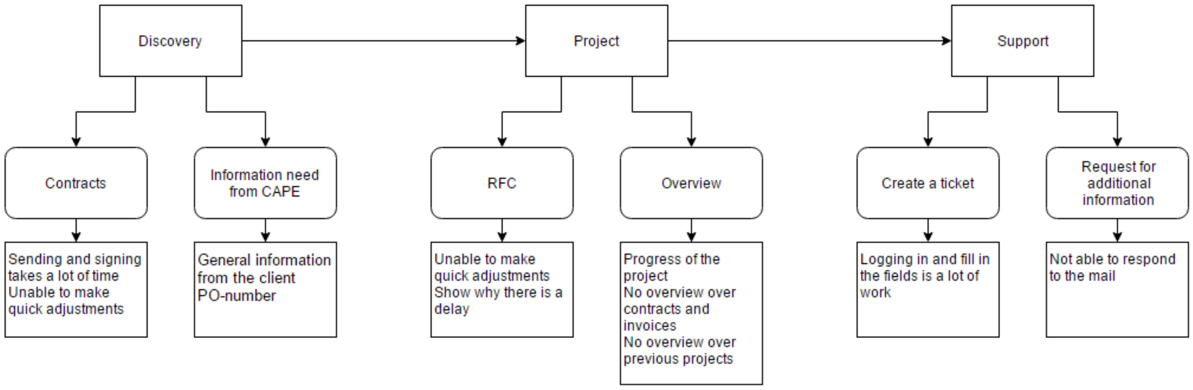

Discovery
Project
Support
Contracts
Information need from CAPE
RFC
Overview
Create a ticket
Request for additional information
Sending and signing takes a lot of time
Unable to make quick adjustments
General information from the client
PO-number
Unable to make quick adjustments
Show why there is a delay
Progress of the project
No overview over contracts and invoices
No overview over previous projects
Logging in and fill in the fields is a lot of work
Not able to respond to the mail

Figure 5: Touching points with the customer

With help from the process chart in figure 5, the digital contact moments or touching points which can be improved with the customer are identified. The first touching point is the process around contracts. Different contracts are being send to the customer. (e.g. a Mendix license, an eMagiz license and a SLA (service level agreement)) This process is currently done in a manual way, the contract is being send to the customer, if the customer agrees to the terms, they sign it, scan the contract and send the contract back to company A. If the customer does not agree to (one of the) contracts, they will send it back to company A with their side notes. If company A agrees to these side notes, a new contract will be made and send to the customer. In short, this process is very time consuming and is rather devious. The improvement of this touch point will not be done by me, because another bachelor student will try to tackle this problem. However, his solution could become part of my system, if that is possible.

The second touching point that can be improved is filling in the contract form by the customer. When company A starts collaborating with a new customer (or a new project starts), they send a contact from together with the contracts to the customer. The customer prints this form, fills it in, scans this form and sends it back to company A. When company A receives this form, they process the form by filling in the data in their own database (or customer register). The same thing is done with the unique PO-number (purchase order number). There are many possibilities to improve this process.

The third touching point is everything around a request for change (RFC). A request for change is a transformation of the terms of the contract. For example, a project does not take 80 hours to make but 100 hours. If this happens, the customer needs to re-approve the project with the proposed changes. In practise, approval is often given by the wrong employee of the customer.
This is something that can be avoided by better communication between company A and their customer.

Subsequently, the customer is often unaware of the progress of the project. The product owners from the customer are part of the process, so they have a good overview, but the people who call the shots, the CEO, or manager only attends a monthly meeting with the stakeholders in the project. Customers do receive a PowerPoint from the project manager with the progress of their project, but this takes some time to build and they often lack information. In short, an overview of previous invoices and projects could be something that can be improved.

The last touching point is when the application is online. When this happens, the application will be under the supervision of the support system from company A. (SP) In SP, a customer is able to request a ticket when their application goes down. Company A analyses the problem and estimates the workload to solve the problem. Company often asks additional information about the problem to the customer, but the customer cannot respond to this mail. There is a link to the application in the mail, but this link does not work. So, it is difficult for the customer to give this additional information to company A. This is something that can be improved. Furthermore, requesting a ticket is often experienced as time consuming by the customers. Some customers mentioned that a ticket should be requested by forwarding a mail they receive from their project team, where the problem is explained. Currently, this is done by filling in a form on SP.

1.7 Problem cluster

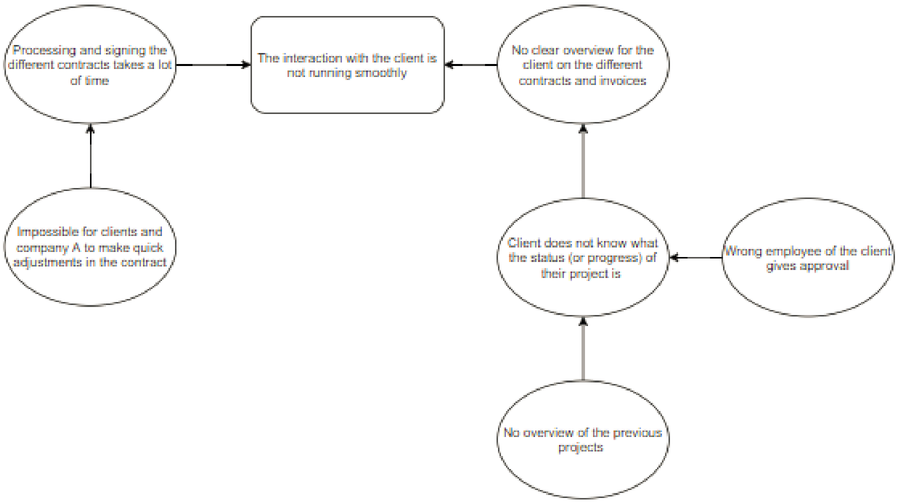

Processing and signing the different contracts takes a lot of time
The interaction with the client is not running smoothly
No clear overview for the client on the different contracts and invoices
Impossible for clients and company A to make quick adjustments in the contract
Client does not know what the status (or progress) of their project is
Wrong employee of the client gives approval
No overview of the previous projects

Figure 6: Problem cluster

Out of the problems mentioned at the touching point part of this report a problem cluster is made. Out of this problem cluster, a core problem is identified which will be the focus of the thesis and the prototype.

1.7.1 Choice of the core problem

As can be seen in the problem cluster, there are two main problems in this case. That is making the process of the contracts more efficient and the lack of overview for the customers. I have chosen for the lack of overview for the customers as my core problem. This has several reasons. The first one is that another bachelor student is going to solve this problem. But it is possible that is solution becomes part of the system I am going to build. The second reason is that it is very useful for the customer to see what company A does for them. Solving this problem could really improve the relation between company A and their customers.

1.8 Summary

In this chapter, a motivation of research, a description of the current situation and problems associated to the current situation are elaborated. The key findings in this chapter are:

- The motivation of research is: Company A thinks that the digital contact moments with their customers can become more efficient.
- Six digital contact moments are identified from the current situation.
- The core problem in the current situation is that customers have no clear overview of their project(s) at company A.

Chapter 2: Definition of solution objectives

In this chapter, a definition of a solution will be defined. This definition will be about the requirements concerning the artefact that will be demonstrated in chapter 4. Furthermore, this artefact must solve the problem concerning the insight in project(s) for the customers of COMPANY A. To come up with these requirements, research questions and theory about this subject are discussed in this chapter.

2.1 Research questions

The first step to define the solution objectives is to ask research questions. Objectives can be elaborated from the answers to these questions.

2.1.1 Main question

The answer to the main research question should solve the motivation of research. For clarification, the motivation of research is: Company A thinks that the digital contact moments with their customers can become more efficient. Therefore, the main research question will be:

How can the interaction process with customers made more efficient with the help of an online portal?

To answer this main question, I need to ask sub-questions to provide an answer this. This is done in the section below. These sub-questions are answered throughout this report.

2.1.2 Sub-questions

Questions that support this main question are:

- What are the problems in the current customer interaction process? (chapter 1)
- How can customer relationship management be applied in my project? (chapter 2)
- How can the artefact be developed in an agile way? (chapter 3)
- What does the architecture of my customer relationship model look like? (chapter 3)
- Does the artefact satisfy the requirements? (chapter 5)
- What are further improvements for the artefact? (chapter 5)

2.2 Theoretical framework

Within research, existing theories can be used to provide an answer to research questions. In this case, the research will be about, business process reengineering (BPR), because several activities in the whole interaction process can be executed differently. In cooperation with BPR, literature research about Customer Relationship Management (CRM), Knowledge Management (KM) and web self-service is elaborated in this chapter. The SCRUM method is explained at the end of this chapter. SCRUM is an agile software development method. For clarification, the build-up of this paragraph is visualized in figure 7.

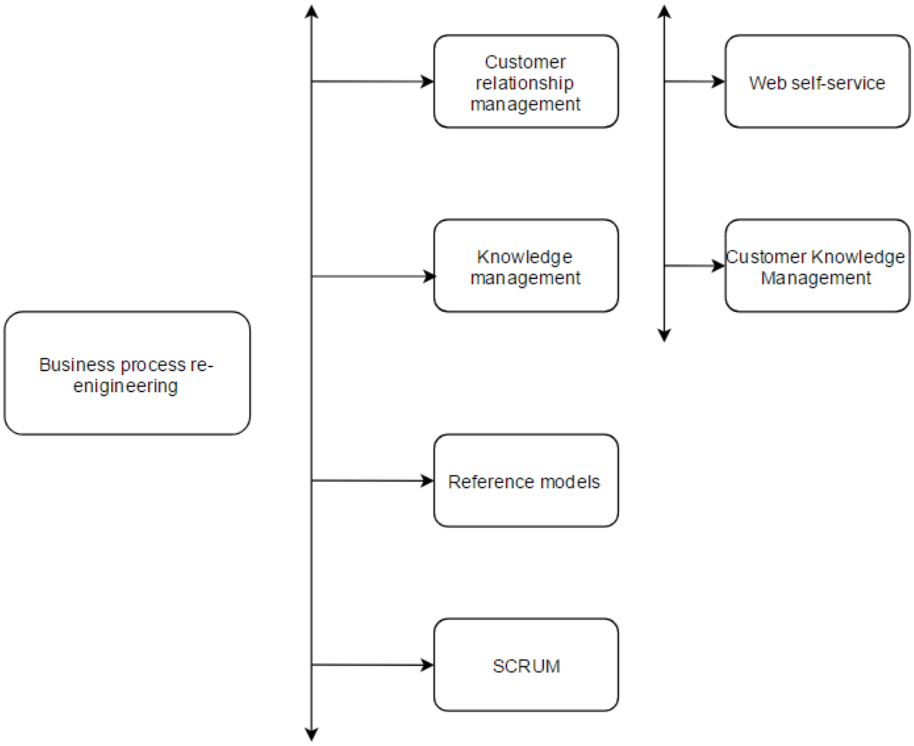

Customer relationship management
Web self-service
Knowledge management
Customer Knowledge Management
Business process re-enigineering
Reference models
SCRUM

Figure 7: Theoretical model

2.2.1 Business process re-engineering

Before one can re-engineer a business process, one must have a clear definition of what a business process is. There is no clear agreed definition of what a business process is in the literature. These definitions differ from each other, but they all have the same outline. A business process, is a sequence of activities or tasks to achieve an outcome. (Motwani, Kumar, Jiang, & Youssef, 1998) So, in this case, the business process is the interaction process with the customer as can be seen in figure 5.
But how can this process be re-engineered in a structural way? What elements lead to successful BPR? Which parts of the current process must be re-engineered and how this will be done is clarified later in this report.

Business process re-engineering (BPR) is all about rethinking and redesigning business processes to obtain dramatic and sustaining improvements in quality, costs, lead-times, outcomes flexibility and innovation. (Hammer & Champy, 1993) (Grover, Jeong, Kettinger, & Teng, 1995) So, in my case, BPR can be applied to rethink the current process and see which parts can be re-engineered to create more flexibility, higher quality of contact etcetera.

There are a lot of factors that influence a successful integration of BPR (Ahmad, Francis, & Zairi, 2007) (Al-Mashari & Zairi, 1999). In Al-Mashari's work, five dimensions of change are identified, which together cover all the opportunities and problems that one can occur during the implementation of BPR:

1) Change management, all the human- and social-related changes and cultural adjustment techniques that are needed to facilitate the insertion of newly designed processes.
2) Management competency and support, the management should understand that some beliefs and routines can't be compatible with BPR and must, therefore, be removed or replaced by other processes. This can be well expressed with strong leadership and risk management.
3) Organizational structure, understand the current organization and recognize possible improvements.

4) Project planning and management, adequate planning in a proper timeframe in cooperation with project management techniques should be able to finish the project on time.
5) IT infrastructure, understand the possibilities of IT and make use of the ones you need for the project.

With the help of these five dimensions and their sub-dimensions, I will be able to identify the choices that I need to make during the BPR process. This will be further elaborated after I have fully explained the current situation at the company and the possible improvements associated to that.

Furthermore, customer relationship management (CRM) and knowledge management (KM) are crucial in executing BPR in an enterprise. (Massey, Montoya-Weiss, & Holcom, 2001) How these methods can be used and what their implications are, is further elaborated in paragraph 2.3.

2.2.2 Customer Relationship Management (CRM)

The previous part of this proposal was all about re-engineering business processes. This is necessary, because current processes can be executed more efficiently. As described in the first chapter, some improvements around the contact with the customers can be implemented. Therefore, it is useful to get more knowledge about the relationship management with the customer and use relevant pinpoints in my own case. So, the question that I need to ask myself is, how can CRM be applied on my project? Before I will be able to apply this, I need to understand what CRM is and what the different views on CRM are, if there are any. At last I need to elaborate which view is most suitable for my project.
To answer these questions, literature research is needed. This research is done by following a series of steps. By selecting the right articles for my research, I will be able to answer this question and determine how I can use this in my own investigation. This systematic literature view is included in the appendix.

CRM is a combination of people, processes, and technology that seeks to understand a company's customers. (Chen & Popovich, 2003) It aims to retain current customers by getting to know their specific needs and adapt their selves to that.
The most common way to retain customers is currently done by the usage of relationship marketing instruments (RMIs), such as loyalty programs and direct mailing. (Verhoef, 2003) These RMIs are used to find out what customers perceive of the company. By indicating possible improvements, the company itself will be able to adjust to that. Another function of these RMIs is to increase customer share. Customer share is defined as the ratio of a customer's purchases of a particular category of products or services from supplier X to the customer's total purchases of that category of products or services from all suppliers (Peppers & Rogers, 1999). These two metrics are important in CRM (Hoekstra, Leeflang, & Wittink, 1999). So, a successful CRM is created by implementing it enterprise wide, make it customer driven, use technology to integrate CRM and link the different departments in a company with each other, e.g. the front- and back-office. (cross functional organisation)

There are different views on how customer retention can be integrated into the process of a company. In order to say something augmented about that, the different views on what a CRM system should do must be identified.

CRM system can be divided into three different segments: (Schwede, 2000)
Operational- a CRM system that improves the efficiency of CRM business processes and comprises solutions for sales force automation and call center/customer interaction center management.
Analytical- a CRM system that manages and evaluates knowledge about customers for a better understanding of each customer and their behavior. This can be done by data warehousing and data mining.

Collaborative- a CRM system that manages and synchronizes customer interaction points and communication channels.

A collaborative CRM system is most suitable in my case, because the system really needs to be customer oriented and make things easier for the customers. In other words, re-engineer the process to fit the need of the customer. For this collaborative system, good interaction with the customer is needed. Interaction management is the analysis to make these interactions as efficient as possible with the objective to increase the quality and value of these interactions. (Gebert, Geib, Kolbe, & Brenner, 2003) By enabling customers to serve themselves (e.g. Web-self-service) the interaction costs go down. Company A could serve the customer by making information available on a portal, if the customer is curious about certain things related to their project.
This concept has a lot of overlap with the customer-centric approach. When a company has a customer-centric approach, products and services are created that fit customer needs. There are five steps that one can take to make an organization customer centric: (Seybold, Marshak, & Lewis, 2001)

1) Make it easy for a customer to do business
2) Focus on the end customer
3) Redesign front office and examine information flows between front and back office
4) Foster customer loyalty by becoming proactive
5) Build in measurable checks to continually improve

As mentioned in the cause of research part of this report. The prototype which I need to deliver at the end of this period should be a portal, which is accessible for the customers via the internet, since company A works with such systems, this shouldn’t be a problem. In combination with the customer-centric approach a web self-service like system should be a good option.

2.2.3 Knowledge management

As mentioned in the previous part about CRM, knowledge about the customer and the current process is needed. This process is called knowledge management (KM).
KM is critical in CRM because a lot of knowledge is needed for a successful CRM.
The knowledge flows can be classified into three categories: (Gebert, Geib, Kolbe, & Brenner, 2003)

1) Knowledge for customers – knowledge that the customer needs, for example, knowledge on available products or services.
2) Knowledge about customers – knowledge that the company needs to know about their customers, for example, contact information such as a phone number.
3) Knowledge from customers – knowledge from the customer about their market segment, interests and problems. This knowledge is useful to sustain continuous improvement.

These knowledge flows are needed for a successful CRM. By using BPR and determining which parts of the process need to be re-designed, the need for the knowledge flows can be identified. For example, if the process of processing customer information must be re-engineered, an information need of the customer information is present. Furthermore, the choice of showing the outcome of this process to the customer is something the company itself needs to determine.

2.2.4 Combining CRM and KM (CKM)

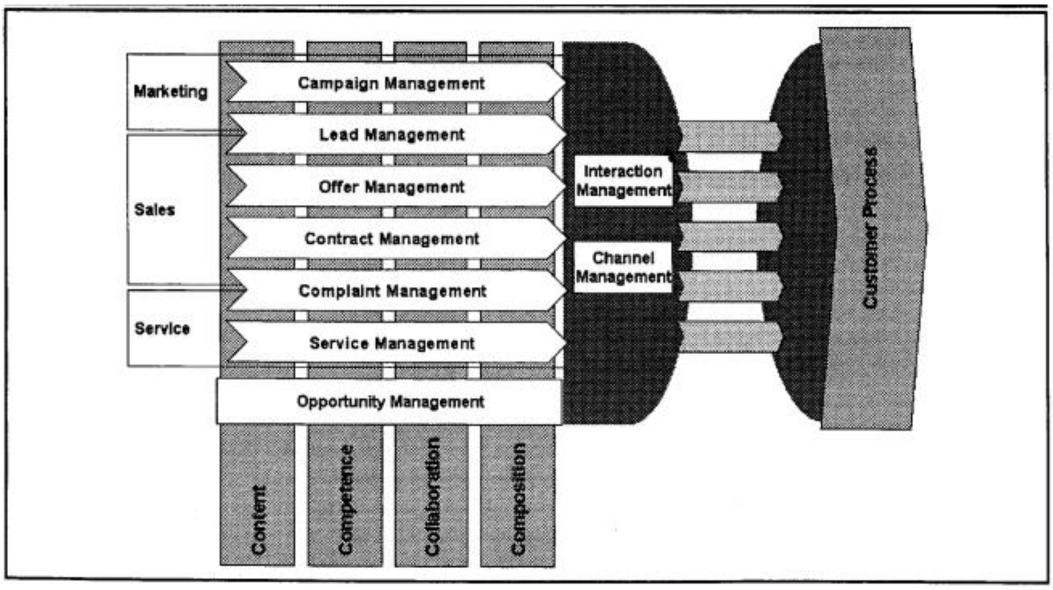


Figure 8: CKM-model (Gebert, Geib, Kolbe, & Brenner, 2003)

The model above shows the combination of CRM and knowledge management (KM). This combination is called customer knowledge management (CKM). (Gebert, Geib, Kolbe, & Brenner, 2003)
KM is about enabling process owners to create different options on which to focus when managing critical knowledge entities. (see paragraph 2.3.1)
KM and CRM knowledge is often captured in different departments within a company and is therefore separately handled. But these concepts can enhance each other perfectly. CRM is all about serving the customer, but without knowledge about the customer, a company can focus on the wrong aspects.
The combination of these two factors is called CKM and can be successful for an enterprise when conducting BPR. From a case study at IBM, the following things can be concluded: (Massey, Montoya-Weiss, & Holcom, 2001)

1) Focus efforts on critical issues that have high value and are aligned with organizational strategy – identify improvements that add the most value. This can be found out when talking to stakeholders. This makes sure that you keep the focus on the right things and that you stick to the set strategy.
2) Establish executive level support – establishing support for the customers, must also be accepted within the organization to become successful.
3) Treat knowledge as a process – knowledge is useless when it cannot be connected to customers or employees. So, make sure that knowledge ends up and the right person.
4) IT must be seen as a medium – IT can enable getting the right data from and to the customer. IT is a crucial success factor, but there is a complexity
5) Prototype fast – getting an early proof of concept to show to the customer. The SCRUM method can be used for this.
6) Recognize the potential of knowledge – make sure that the right information is accessible throughout the company.

2.3.3 Web self-service portal

Following up the previous part about CRM, in the last sentence, it is mentioned that the system should be an online portal so that customers and employees can access it at any time.

The possibilities of this online portal should be investigated. The ideal form for this is an online portal, where the customers can help or serve themselves. Therefore, the term self-service portal. The possibilities and drawbacks of such a portal are explained below.

Self-service technologies (SSTs) are technological interfaces that enable customers to produce a service independent of direct service employee involvement. (Meuter, Ostrom, Roundtree, & Bitner, 2000) Some examples of SSTs are ATMs and pay-at-the-pump terminals. By replacing an employee with a machine, customers are able to serve themselves by giving instructions to the machine.
Self-service should satisfy the following requirements. (McGovern, 2014)

1) It should be simple. No humans are involved in the process, so everything should be clear on the portal itself, otherwise, customers will contact the help desk because they have questions.
2) Convenience is the next requirement. The self-service channel should be more convenient for the customer compared to other possible channels. E.g. call the helpdesk.
3) The third factor is control. The customer must have a sense of control when serving them. This can be done by enabling the customer to do things by themselves on the portal. For example, looking at the invoice from last month. Creating more control can have a trade-off with simplicity. The more features one adds to increase control, reduces the simplicity of the portal.
4) Speed is also an important determinant. The value created by an information system is faster response time than other means of communication. (Lu, Huang, & Heng, 2006)
5) Price is another requirement. If one serves his or herself, he or she expects to be rewarded for that. In other words, they do work that another should be doing, so they expect to get rewarded for the work they save for the organization. But, if the self-service provides a good overview (control) and works faster than the traditional approach, customers may be willing to pay more for it.
6) The last factor is continuous improvement. Things change all the time, also the interests of the customers. Furthermore, the system itself should be continuously improved, so that customers can be served faster. Also, monitoring the customers and ask them what they think could be improved is a crucial step in continuously improving the system (CRM).

This concept can also be applied in an internet environment. Instead of calling a customer service employee, a customer could be able to check a website or log on a portal to see what the problem is. By enabling the customer to do things themselves, interaction costs go down and if the information is shown correctly, the customer satisfaction goes up, because they feel that they are informed well about their projects. (Cooper, Lichtenstein, & Smith, 2006) This can for example be done by extending the capability to the customer for creating their own overview if they need one. This overview can be about the worked hours in a week or the invoices of a certain project.

2.2.5 Reference models

The second research question is about finding models that are comparable to my own case. From these models, useful functionalities can be identified. Functionalities that aren’t in my scope can also be identified and left out my final design. I looked at three different cases. All these cases were about web self-service portals and their functionalities.

The first case is from ServiceNow Express (Express, 2017). ServiceNow Express is enterprise software company which was founded in 2004. They developed a self-service portal for their customers with distinctive features. Firstly, customers are able to get in touch with different departments on the homepage and check the progress of their orders. Furthermore, customers are able to search for answers to their questions by themselves. This can be done by clicking on a help button and type in the thing you want to solve. A customer is also notified when an order gets late or something unexpected happens within the process. At last, the customer is able to ask assistance by filling in a short form on the site. This can also be done on a phone.

The second case is from Albelli. (van Vliet, 2014) Albelli is a company which offers photobooks and other products related to that. A company called Kana has developed web self-service for Albelli which relieves stress from the constantly growing help desk. This self-service system consists of the following elements: e-mail response management, live chat, call management. On the website, answers on frequent asked questions (FAQ) are shown. This is done to relieve stress from the help desk. The solution works well, less employees are needed in at the help desk, because customers can solve some of the problems by themselves by simply consulting the website.

The last case is from Randstad. (Teixeira-Pinto, 2014) Randstad is an employment agency. Randstad used the software of OutSystems to develop a new customer portal, because the old one was not feature proof. Given the expected growth of the company, they thought it was wise to make a brand-new system instead of repairing the current one. The system was designed for employees as well as the customer in order to stimulate the use of it by the employees. On the portal, employees are able to download their payslips or approve workflows delivered by colleagues. Customers are able to see the reports from the employees who are working for them at Randstad. Also, an overview of the different time-sheets is created to see what will be done each day. The customer is also able to approve the proposed workflows. By implementing this system, Randstad is able to train new employees faster and proposed changes take days instead of weeks. Furthermore, the customer has the ability to check his progress at any time. So, 24/7 availability.

The main points that I can use for my own project are, showing FAQs on the portal. By interviewing project managers from the customer and from company A, I will be able to identify questions that are often asked by the customers. If I can formulate an answer to that question, the customer will be able to help themselves by checking the portal. Another functionality that can be useful is showing a calendar where the customer can easily what will be done on what day, when a planning is determined. All the other points are also useful, but they will be mentioned in the improvement approach section.

2.2.6 SCRUM

Company A uses a framework for agile software development within their projects called SCRUM. This is, of course, vague and calls upon many questions. What is agile? What does it do exactly? Why does it work?
All these questions will be answered in this section.

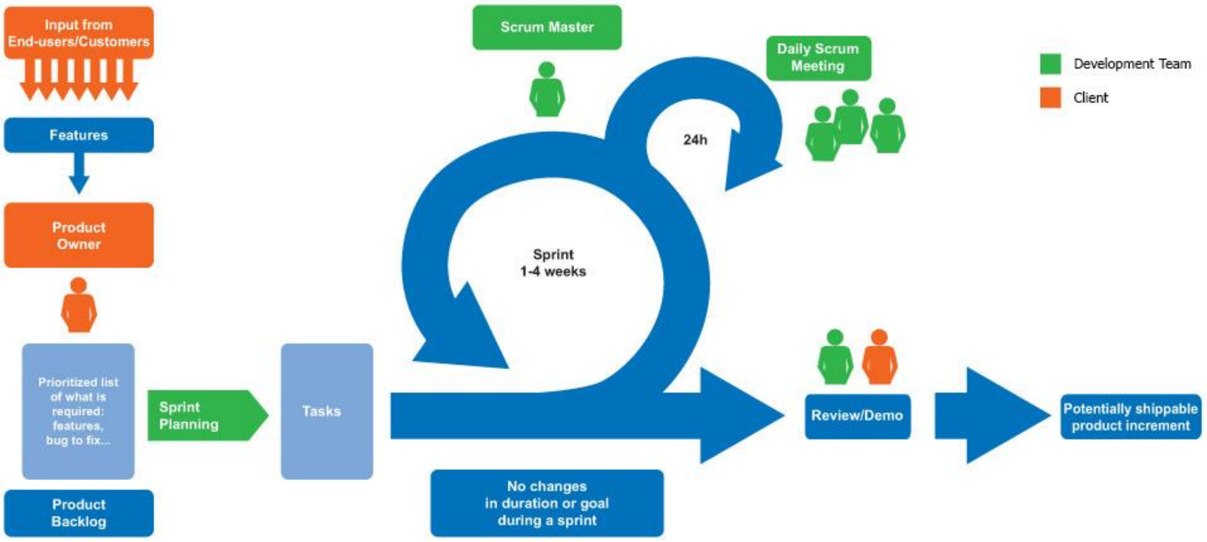

Input from End-users/Customers
Features
Product Owner
Prioritized list of what is required: features, bug to fix...
Product Backlog
Sprint Planning
Tasks
Scrum Master
Daily Scrum Meeting
24h
Sprint 1-4 weeks
No changes in duration or goal during a sprint
Review/Demo
Potentially shippable product increment
Development Team
Client

Figure 9: The process of SCRUM

It all starts with an idea or vision from a company or a **product owner**. The product owner has an idea that solves a certain problem for a group of people (**stakeholders or customer(s))**. But the owner itself does not really have an idea about how to execute this problem. The stakeholders and the product owner have ideas about what the product should do. These ideas are called **user stories**. These user stories must be built by a development team. The team is not going to build all the user stories at once, but they are going to build the three most important stories into a working prototype which is then send to the stakeholders. This process is called a **sprint**. A sprint usually takes 2-3 weeks. The stakeholders test this prototype and give feedback on this prototype (more user stories) All these user stories are kept in a **product backlog**. Before every sprint, a capacity is determined by the team. So, if there are 4 people working on the project during 3 weeks, the capacity of the team will be 4 (people)* 3 (weeks) *40 (hours per week) = 480 workable hours or points. After this, a workload per user story is determined. For example, a user story will take 10 hours to build, so this story is given 10 points. The list of user stories will be prioritized by the **MVP-rule** (Minimum Viable Product) This rule states that a user story must deliver a certain amount of value to the customers or stakeholders with the least amount of effort (Moogk, 2012), or in this case, story points. So, the development team will prioritize all the user stories and choose the stories which add the most value for the customer. The development team also makes sure that the amount of story points during a sprint falls within the capacity of the team. It can be the case that certain user stories will not be executed within a sprint, because the capacity has been reached. These stories will be kept in the backlog and can be executed in the next sprint.

After a sprint, the prototype will be send to the stakeholders. The team itself reviews at the end of the sprint if they did everything what they wanted to do at the beginning of the sprint. The team also reflects if everything went according to plan and if not, what could be done next time to make sure this does not happen again? This reflection of a sprint is called a **sprint review**. After this review, an increment of the product is delivered to the customer. The customer tests the increment and gives feedback on it. New user stories come up with this feedback. These stories than come into the product backlog with the other stories that have not been executed in the previous sprint. The development team prioritizes this list of stories and so the process repeats until all the user stories have been implemented in the product. (Sutherland, Scholingen, & van, 2011)

SCRUM is an agile way of developing products. The product developer starts with requirements in the beginning and after 2 months the product is finished, the customer may have other interests or things he would like to see in the product. The needs of the customer change over time and your development process should be able to change as well. (Schwaber & Beedle, 2002) There is no project manager in a project. The development team itself is responsible for the progress of the project. (Kniberg, 2012) With daily stand-up meetings, every member of the team explains what they have done since the last meeting, what they want to do today and which difficulties they have encountered since the last stand-up meeting. With these meetings, the team is able to keep track what everybody is doing (so avoid that two people are working on the same thing) (Cohen, Lindvall, & Costa, 2003)

How is SCRUM implemented within company A?

Company A works rather strictly to the SCRUM method. They spend most of their time at the customer and therefore they can tackle problems quickly. They also have sprint meetings, daily stand-ups and a backlog. They also work in sprints. A sprint often takes about 4 weeks and at the end of each sprint a sprint evaluation will be held, and a new sprint will be started. This sprint is started by selecting user stories from the backlog with story points and fill the sprint till the story points in the sprint are equal to the available story points in the sprint. This is repeated until the backlog is empty.

2.3 Summary

From this chapter, it can be concluded that a web self-service portal is very useful in this case since it enables the customers to check the progress within their project(s) by themselves. A web self-service portal should be: simple, convenient, controllable, fast, add value and be able to continuously improve. The portal can be developed with the help of SCRUM, an agile software development method. During the development of the application, one should bear several things in mind:

BPR is needed in my project, because I need to re-design a current process. BPR can be successfully integrated by managing human interactions, strong leadership, using risk management, understanding the organization, making an adequate planning and understanding the possibilities of IT. CRM can be increased by combining information from, about and for the customer to link the front- and back office with each other.

Out of the reference models, the following functionalities can be used for my portal: show FAQ's, show when an employee will work on your problem and approving workflows (or in the case of company A, RFC's) via the portal. This will not be the focus since the application will be built around the core problem, create insight in project(s) from the customers.

Chapter 3: Design and development

This chapter is about the creation of the application. As mentioned in the summary of chapter 2, the application will a self-service portal where the customers can log in and check the progress of their project. How this portal will be build is explained in this chapter. First, different scenarios are described, and the most suitable scenario will be chosen from that. From this scenario, a solution design and an architecture will be made.

3.1 Comparison of scenarios

The need of an information system for the customer has become clear in the previous part. A lot of processes can be improved by enabling the customer to do things by themselves. As mentioned in *1.6.2 Current systems and programs within company A*, there are two systems that company A currently uses that are suitable for the self-service portal. These systems are IS and CSP. Employees of company A fill in their worked hours in IS, the internal system of company A. CSP is the system where customers currently go to when they have issues with their application.
From this, three different scenarios emerge. These are all explained below.

Different scenarios
There are two scenarios that can be compared with each other:
The creation of a brand-new system versus expanding systems that are currently being used by company A.
Scenario 1: Build a new system.
In this scenario, a brand-new system will be created in Mendix. IS will be integrated in the system, so that information from IS can be used to inform the customer. SP has to be integrated as well, because the support part must also be available to the customer.
Scenario 2: Expanding a current system of company A.
2a) In this scenario, an existing system (SP) is being expanded with information from IS. Using this information, an overview of current projects and their progress can be created.
By combining these two systems and adding some features, a portal for customers can be created.
2b) Scenario 2b is very comparable to scenario 2a. This scenario states that IS will be expanded. This could also bring some risks, because not all the data within IS should be shared with the customers.

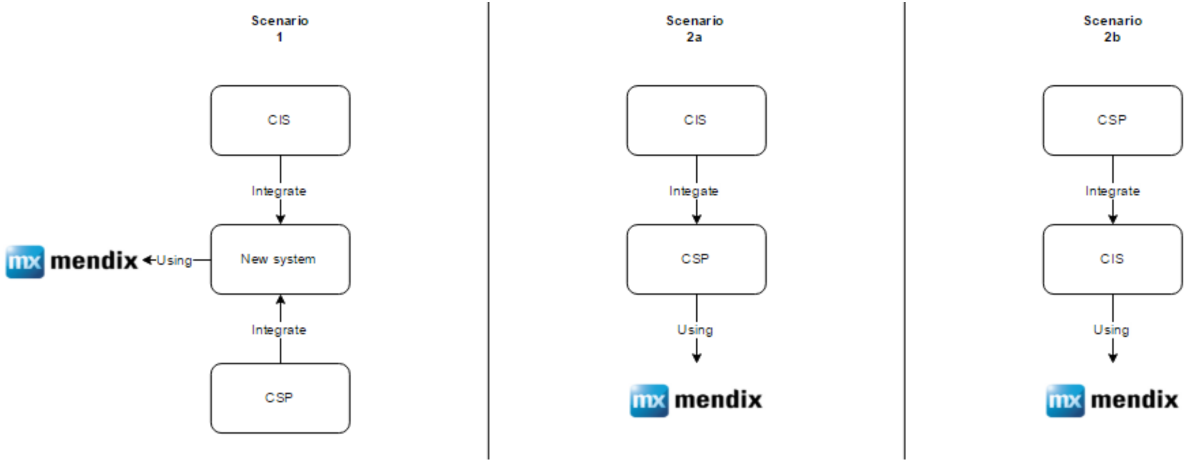


Figure 10: Visualization of scenarios

The comparison is shown in the table below:

Topic	Scenario 1: Make a new system	Scenario 2a: Expand CSP with IS	Scenario 2b: Expand IS
Workability for the developer	No restrictions (build from the ground up)	Restricted by the design of the current system (need to get familiar with the system and the possibilities)	Restricted by the design of the current system (need to get familiar with the system and the possibilities)
Impact of mistakes	System is not yet online, so if mistakes are made, no one will notice	System may go down when a mistake is made, but can be avoided when work is done locally.	System may go down when a mistake is made, but can be avoided when work is done locally.
Workability for the customer and employees of company A	Need to get familiar with the new system (may get lost in all the systems)	Customers of company A are already familiar with SP	Customers of company A are not familiar with IS
License costs	Extra license costs, because of creation of a new system	No extra license costs	No extra license costs
Information breach	No chance of information breach	Small chance that sensitive information reaches the customer	Bigger chance that sensitive information reaches the customer, since one is working in the internal system of company A
Sharing with the customer	The entire system can be shared with the right employee of the customer (employees with less rights can have access to a part of the system)	SP is already used by customers of company A. The message that will be send from IS to SP is controlled, so a small chance of sharing sensitive information with the customers.	Greater chance that confidential information reaches the customers, since it is all saved in IS.
Updates	Independent of the functionalities of another system	Dependent on the functionalities of another system	Independent of the functionalities of another system

Table 1: Scenario comparison table

3.1.1 Decision for scenario

Scenario 2a was chosen as the best scenario. It was augmented that with the uncertainty of the success of the project, it would be unwise to invest in a new license for a new system. (scenario 1)
The choice between scenario 2a and 2b was harder, because these scenarios have a lot in common. But, in the end, the choice was made to go with scenario 2a. The main reason for this was that SP is already an external business service for the customers of company A. Another reason for this choice is, when a part of IS becomes available for external parties, there will always be a chance that sensitive information reaches these parties. When SP is expanded, this is not really an issue since the information within the message from IS to SP can be controlled and therefore be filled with information which the external parties are allowed to see.

3.2 Architecture

After the decision from company A to expand SP with IS, an architecture for the customer portal can be made. This can be done in combination with the attention points indicated in the summary of chapter 2. (paragraph 2.3)

Architecture is defined as the “fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.” (IEEE, 2000) Enterprise Architecture (EA) therefore is understood as the fundamental organization of a company, as a whole or with their supply chain (so with suppliers, customers etc.) or in part (a department or a division). (OpenGroup, 2003) The architecture describes how the application will help the current process. (Weske, 2012)

On the next page, the architecture of the portal is displayed. The architecture provides a layered overview of the entire system and all its components. The best way to explain the system is to walkthrough the architecture from top to bottom.

External factors and actors: In this case, customers of company A need information about their project and want to report a problem concerning their application.
External business services: The business service that company A offers to satisfy the external process, is an online portal where the customer can see their project and report problems with their application.
Internal processes, roles and actors: The internal process behind the business service is the project flow. This flow is already explained in paragraph 1.6 and displayed briefly in this architecture. General information is needed from the client before a new project can be created. Once the new project is created in IS, the internal system of company A, the project monitoring starts as well. Once a week, the customer will be updated on the progress within their project. When the project ends, the application will still be under supervision of company A. The support service, formally CSP, will make sure that customers can report any problems with their application once it has gone live.
External application services: In order to provide the functionalities mentioned above, a number of services are needed. First a customer administration service is needed to save all data about customers. Also, a project monitoring service is needed to keep track of the progress in a project. Furthermore, a mail service is needed to alert the customers when there is an update available for them. At last the support service is needed enable customers to report issues with their application.
Application components and services: This layer in the architecture will be about the scenario chosen in paragraph 3.1. In this paragraph, the scenario to expand CSP with messages from IS has been chosen. In figure 13, creates a more detailed view on how this can be accomplished. For the expansion of CSP, or the development of CP information about customers, projects, budgets and time registration is needed. All this information is stored in IS.

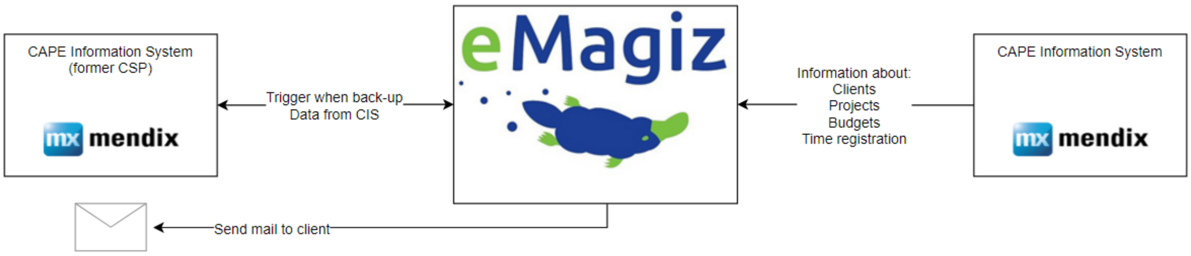


Figure 11: Application components and services

This data can be retrieved and send to CP by eMagiz. When this message arrives in CP, a mail will be send to the customer with a notification that there is an update available about their project. The customer is able to log in and see what has happened the last week in their project.

External infrastructure services: Data about customers, projects, hour registration from employees of company A is saved in IS. IS and SP, the two systems that need to cooperate in this scenario run on Mendix. The system that enables the link between the two systems is eMagiz. Therefore, Mendix and eMagiz are the two components of the *Infrastructure.*

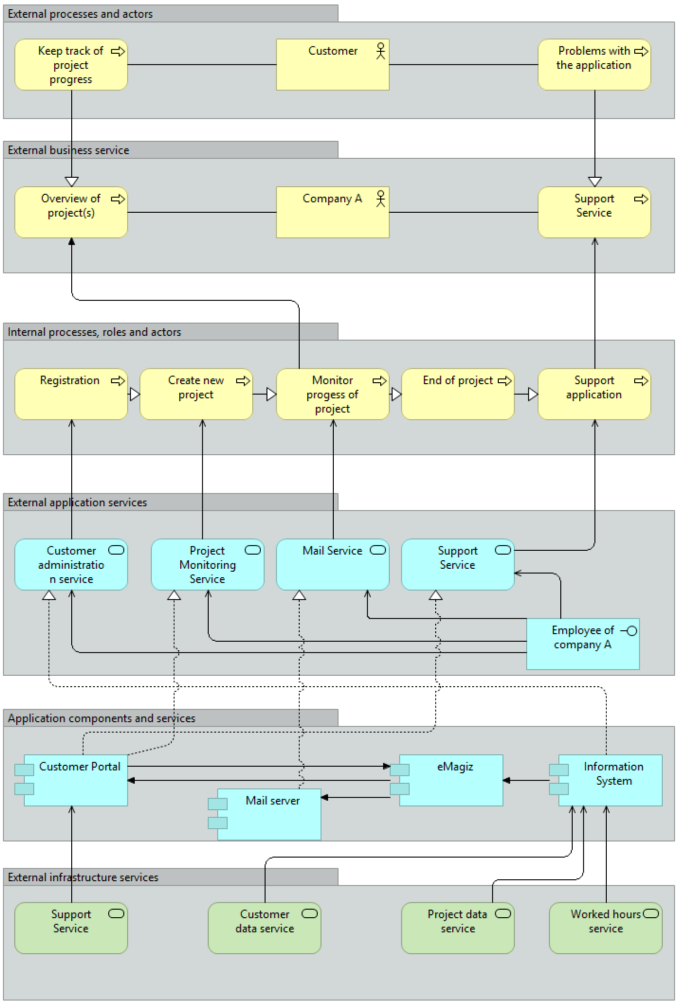

External processes and actors
Keep track of project progress
Customer
Problems with the application
External business service
Overview of project(s)
Company A
Support Service
Internal processes, roles and actors
Registration
Create new project
Monitor progess of project
End of project
Support application
External application services
Customer administratio n service
Project Monitoring Service
Mail Service
Support Service
Employee of company A
Application components and services
Customer Portal
eMagiz
Information System
Mail server
External infrastructure services
Support Service
Customer data service
Project data service
Worked hours service

Figure 12: Architecture of the application

3.3 Application of the SCRUM method

The SCRUM method can be well applied on this project. The project can be divided into roughly three parts: Building the model in CSP, add a mobile app and create the integration between the two applications. There around ten weeks to build the portal, so the sprints will last three weeks each. A backlog with user stories will be created for every sprint.
All the user stories that will be executed during these 10 weeks are available in appendix B.

3.4 Summary

In this chapter, the scenario of expanding SP with IS was chosen. The name of SP will change to Customer Portal or CP. At the end of every week, a message with information about the project will be send to CP. When the message is received in CP, an e-mail will be send to the customer to notify that there is an update available for them. To visualize the system, an architecture is made. For the development of the application, the SCRUM method will be used. During the development, three sprints will take place where distinct parts of the application will be build.

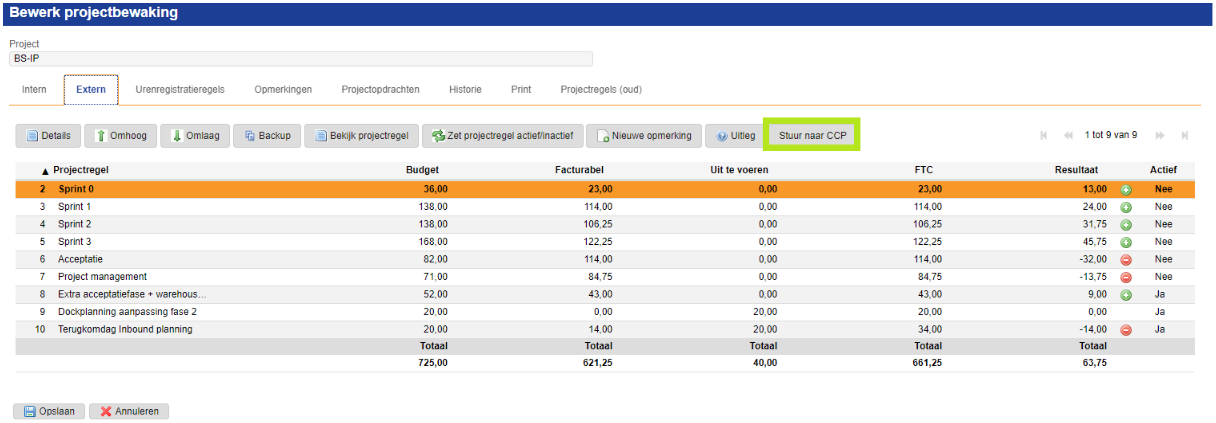
Chapter 4: Demonstration

In this chapter, the Customer Portal will be demonstrated. The functionalities of the app will be covered in this chapter. The desktop application will be discussed before the mobile app. How the prototype is built in Mendix can be seen in Appendix C.

As mentioned in the previous chapter, the process starts at IS. In IS, relevant data needs to be gathered for the message that is send to eMagiz. Since one project is send per message, one needs the following information:

- One project
- The project code (an abbreviation that is set for every project, every project code is unique)
- The status of the project (active or closed)
- The customer related to that project
- The project lines that are related to the project. Project lines are the phases within the project, these phases are often called sprints but can have other names as well.
- The characteristics of the project lines. Every project line has a name, number, budget, billable hours, hours to do, forecast to completion (FTC), a result and an enumeration of the result (positive, negative). This enumeration is used to stress the positive or negative result of a project line.
- Beneath the project lines, hour registration lines are attached. These lines are filled in by employees of company A in IS. These lines need to be in the message as well.
- An hour registration line contains the following attributes: date, name of employee, description and number of hours.

Now that we have our message, we need to gather the right information from IS. This is done with a microflow. This microflow must be triggered somewhere. A microflow can be triggered by a button or another event. In this case, the choice has been made to trigger the microflow by clicking on a button. In figure 14, a project overview is shown of a project with code *BS-IP.* When the project manager approves everything from the previous week, he clicks on the button *Send to CP* (button in the green box) to trigger the message.



Project
BS-IP

	▲ Projectregel	Budget	Facturabel	Uit te voeren	FTC	Resultaat		Actief
2	Sprint 0	36,00	23,00	0,00	23,00	13,00	⊕	Nee
3	Sprint 1	138,00	114,00	0,00	114,00	24,00	⊕	Nee
4	Sprint 2	138,00	106,25	0,00	106,25	31,75	⊕	Nee
5	Sprint 3	168,00	122,25	0,00	122,25	45,75	⊕	Nee
6	Acceptatie	82,00	114,00	0,00	114,00	-32,00	⊖	Nee
7	Project management	71,00	84,75	0,00	84,75	-13,75	⊖	Nee
8	Extra acceptatiefase + warehous...	52,00	43,00	0,00	43,00	9,00	⊕	Ja
9	Dockplanning aanpassing fase 2	20,00	0,00	20,00	20,00	0,00		Ja
10	Terugkomdag Inbound planning	20,00	14,00	20,00	34,00	-14,00	⊖	Ja
		Totaal	Totaal	Totaal	Totaal	Totaal		
		725,00	621,25	40,00	661,25	63,75		

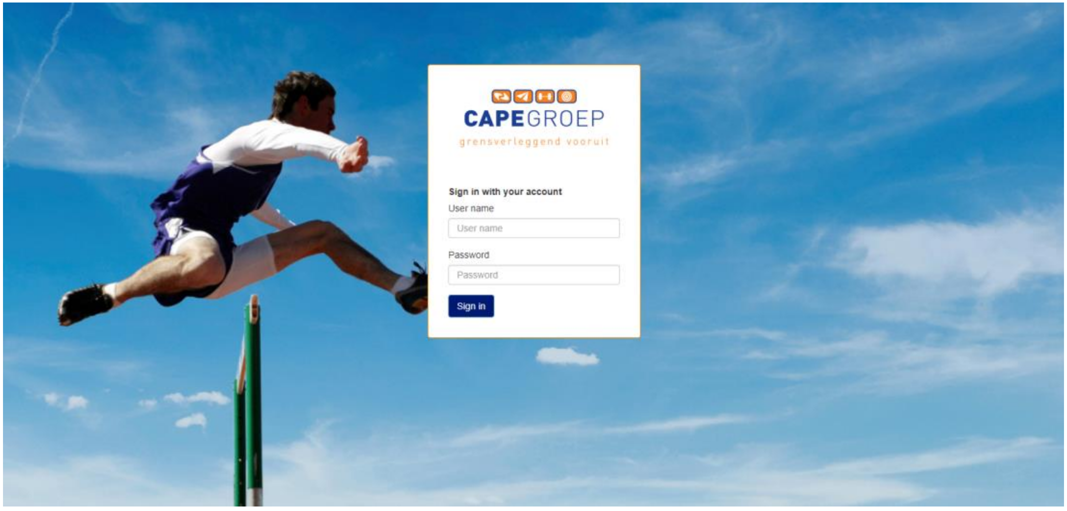
Figure 13: Project Overview page in IS with button Send to CP

The message will go from IS, through eMagiz to CP. The logic behind this process is elaborated in appendix C.

4.1 Desktop version

Once the message is received in CP, the information within this message can be displayed. This can be done with the help of tables and pages in Mendix. To demonstrate the functionalities of the application, a fake customer account is made. For this demo, the user will be an employee of *Post-NL*, a customer of company A.

When the application is up-and-running this is the first screen the user sees.



CAPEGROEP
grensverleggend vooruit

Sign in with your account

User name

User name

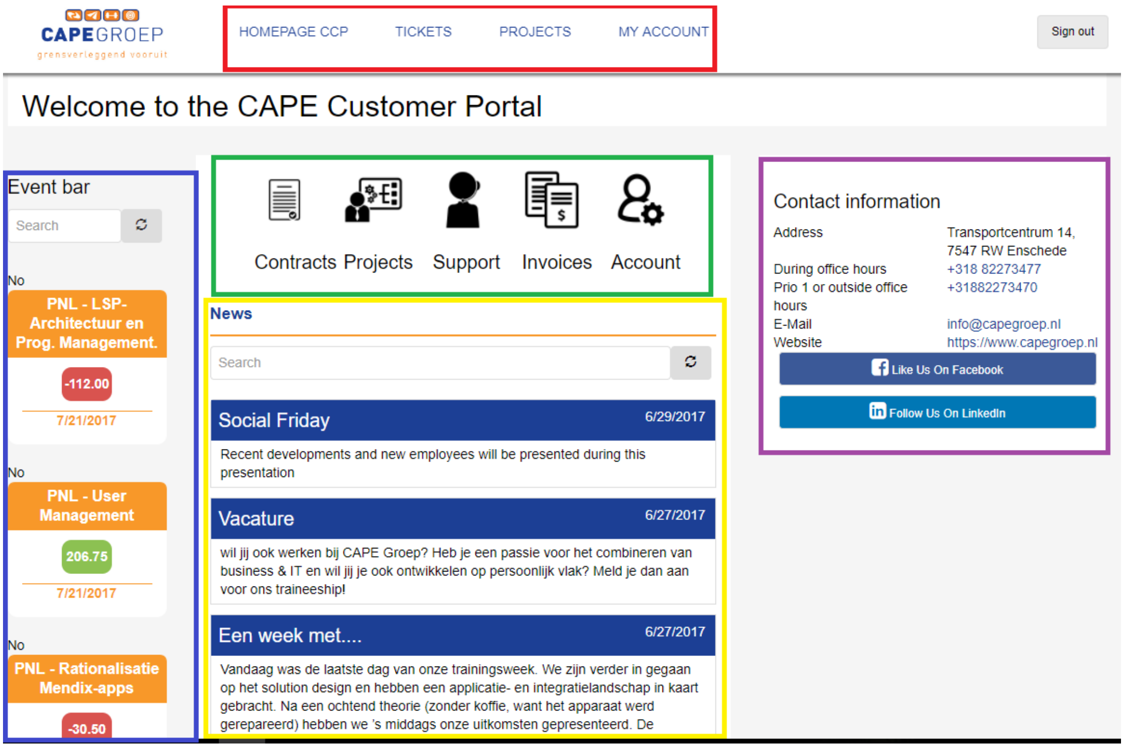
Password

Password

Sign in

Figure 14: Log in screen

The user can log in with the same credentials he or she uses when logging in to CSP.



CAPEGROEP
grensverleggend vooruit

HOMEPAGE CCP TICKETS PROJECTS MY ACCOUNT

Sign out

Welcome to the CAPE Customer Portal

Event bar

Search

No

PNL - LSP- Architectuur en Prog. Management.

-112.00

7/21/2017

No

PNL - User Management

206.75

7/21/2017

No

PNL - Rationalisatie Mendix-apps

-30.50

Contracts Projects Support Invoices Account

News

Search

Social Friday 6/29/2017

Recent developments and new employees will be presented during this presentation

Vacature 6/27/2017

wil jij ook werken bij CAPE Groep? Heb je een passie voor het combineren van business & IT en wil jij je ook ontwikkelen op persoonlijk vlak? Meld je dan aan voor ons traineeship!

Een week met.... 6/27/2017

Vandaag was de laatste dag van onze trainingsweek. We zijn verder in gegaan op het solution design en hebben een applicatie- en integratielandschap in kaart gebracht. Na een ochtend theorie (zonder koffie, want het apparaat werd gerepareerd) hebben we 's middags onze uitkomsten gepresenteerd. De

Contact information

Address	Transportcentrum 14, 7547 RW Enschede
During office hours	+318 82273477
Prio 1 or outside office hours	+31882273470
E-Mail	info@capegroep.nl
Website	https://www.capegroep.nl

Like Us On Facebook

Follow Us On LinkedIn

Figure 15: Homepage of CP

When the user has logged in successfully, the homepage of CP appears.

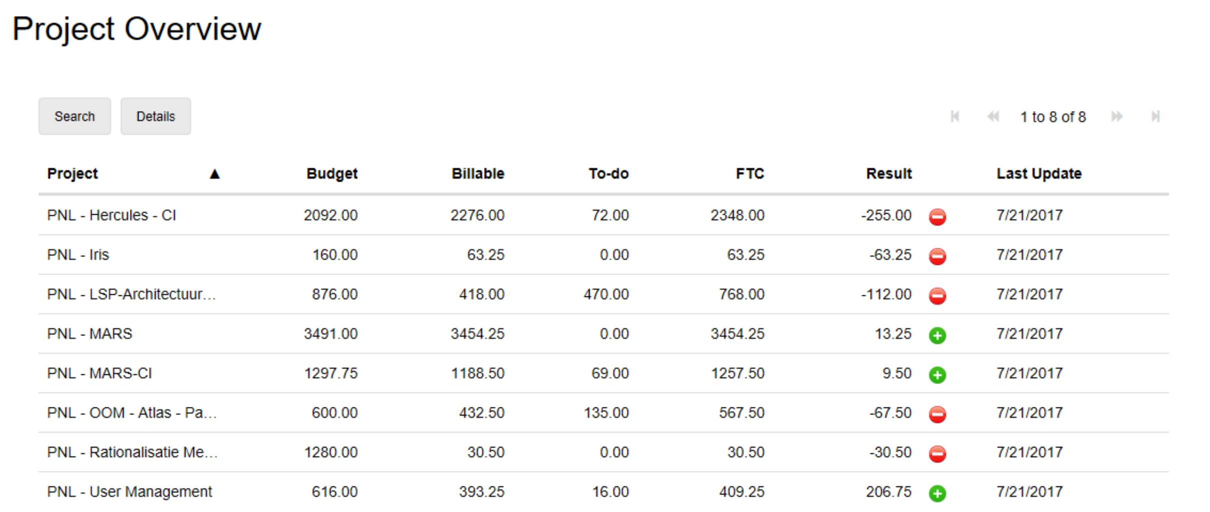
In figure 15, different sections of the homepage are placed in boxes of assorted colours. The homepage consists of five parts:

- Red- These are navigation buttons with different options. In this menu, the user is able to go back to the homepage, check his tickets, go to his projects or look at his account.
- Blue- The event bar. In this section project updates are shown. Every time the project manager presses the button *Send to CP* in IS. A message appears in the event bar. Every element in the event bar shows the name of the corresponding project, the result and the date of the update.

The element of result is red when the result is negative and green when it is positive. The user can click on an element in the event bar to check the details of his project. When the user does this, he will end up at the *Project Detail* page. (figure 17)

- Green: Another navigation menu. Five icons are shown, *contracts, projects, support, invoices* and *account.* Due to the scope of the prototype, the icons *contracts* and *invoices* are not clickable. This could be a functionality in the feature.
- Yellow: Latest news about company A is shown in this section. When the user clicks on an item in this section he will redirected to the corresponding website.
- Purple: In this section contact information of company A is displayed.
- On this screenshot, the calendar is left out. Events where company A is attending are shown in this section.

When the user clicks on *Projects* in the green box, the *Project Overview* page will appear. (figure 16)



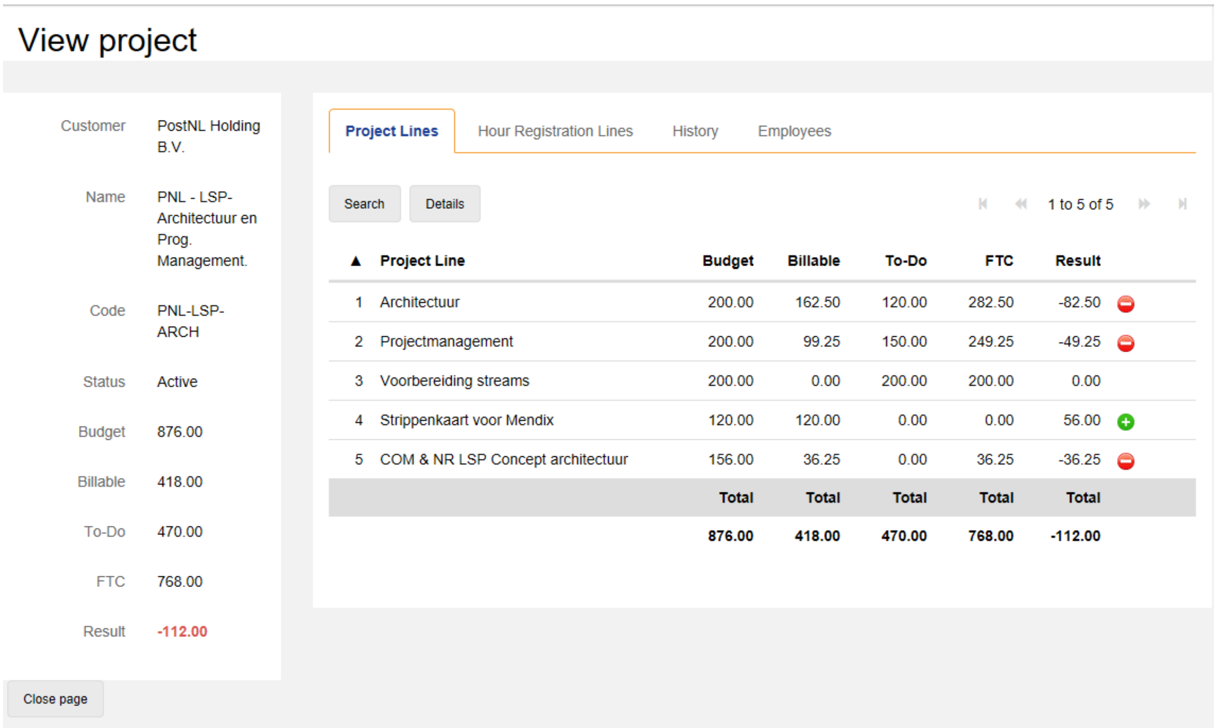
Project Overview

Project ▲	Budget	Billable	To-do	FTC	Result		Last Update
PNL - Hercules - CI	2092.00	2276.00	72.00	2348.00	-255.00	⊖	7/21/2017
PNL - Iris	160.00	63.25	0.00	63.25	-63.25	⊖	7/21/2017
PNL - LSP-Architectuur...	876.00	418.00	470.00	768.00	-112.00	⊖	7/21/2017
PNL - MARS	3491.00	3454.25	0.00	3454.25	13.25	⊕	7/21/2017
PNL - MARS-CI	1297.75	1188.50	69.00	1257.50	9.50	⊕	7/21/2017
PNL - OOM - Atlas - Pa...	600.00	432.50	135.00	567.50	-67.50	⊖	7/21/2017
PNL - Rationalisatie Me...	1280.00	30.50	0.00	30.50	-30.50	⊖	7/21/2017
PNL - User Management	616.00	393.25	16.00	409.25	206.75	⊕	7/21/2017

Figure 16: Project Overview Page

This screen gives the user an overview of his projects. In this case, the user has multiple projects at company A. Information about *budget, billable hours, hours to-do,* FTC and *result* at project level are also shown on this page. For the demo, the user is curious about the progress in project *PNL-LSP-Architectuur* since this project has a negative result of *-112.*

Therefore, the user clicks on this rule in the table.



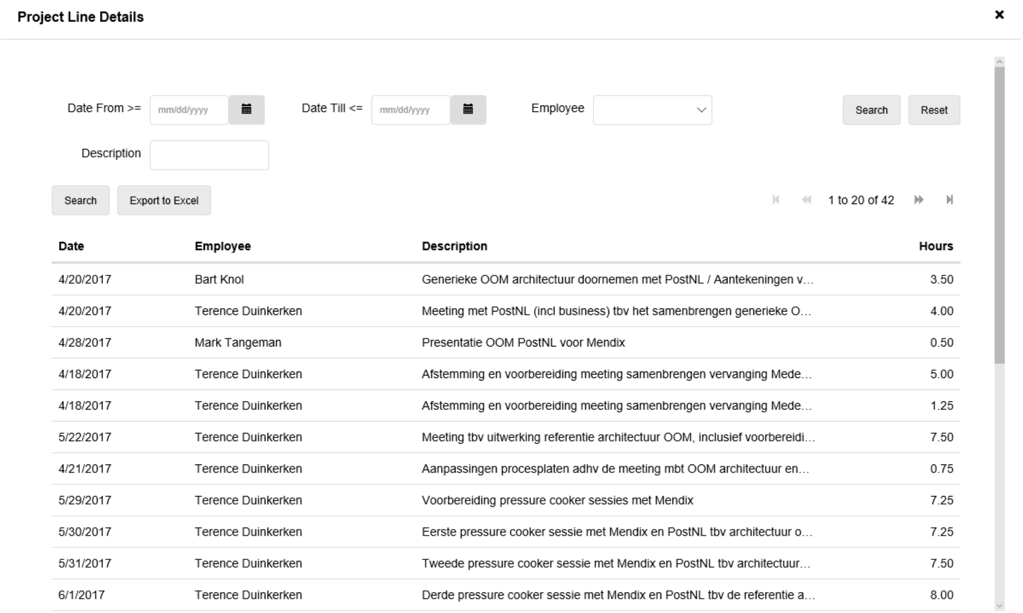
View project

Customer	PostNL Holding B.V.
Name	PNL - LSP-Architectuur en Prog. Management.
Code	PNL-LSP-ARCH
Status	Active
Budget	876.00
Billable	418.00
To-Do	470.00
FTC	768.00
Result	-112.00

▲	Project Line	Budget	Billable	To-Do	FTC	Result	
1	Architectuur	200.00	162.50	120.00	282.50	-82.50	⛔
2	Projectmanagement	200.00	99.25	150.00	249.25	-49.25	⛔
3	Voorbereiding streams	200.00	0.00	200.00	200.00	0.00	
4	Strippenkaart voor Mendix	120.00	120.00	0.00	0.00	56.00	⊕
5	COM & NR LSP Concept architectuur	156.00	36.25	0.00	36.25	-36.25	⛔
		Total	Total	Total	Total	Total	
		876.00	418.00	470.00	768.00	-112.00	

Figure 17: Project detail page

On this page, the user sees are more detailed table about the project. On the left general information about the project is shown. These are the same numbers as on the previous page. In the right section are four tab pages. In the opened tab page, data from the separate phases in the project are shown. To see even more details in a phase, the user can click on a row in the table. In this case the user is curious about the *Architectuur* phase



Project Line Details

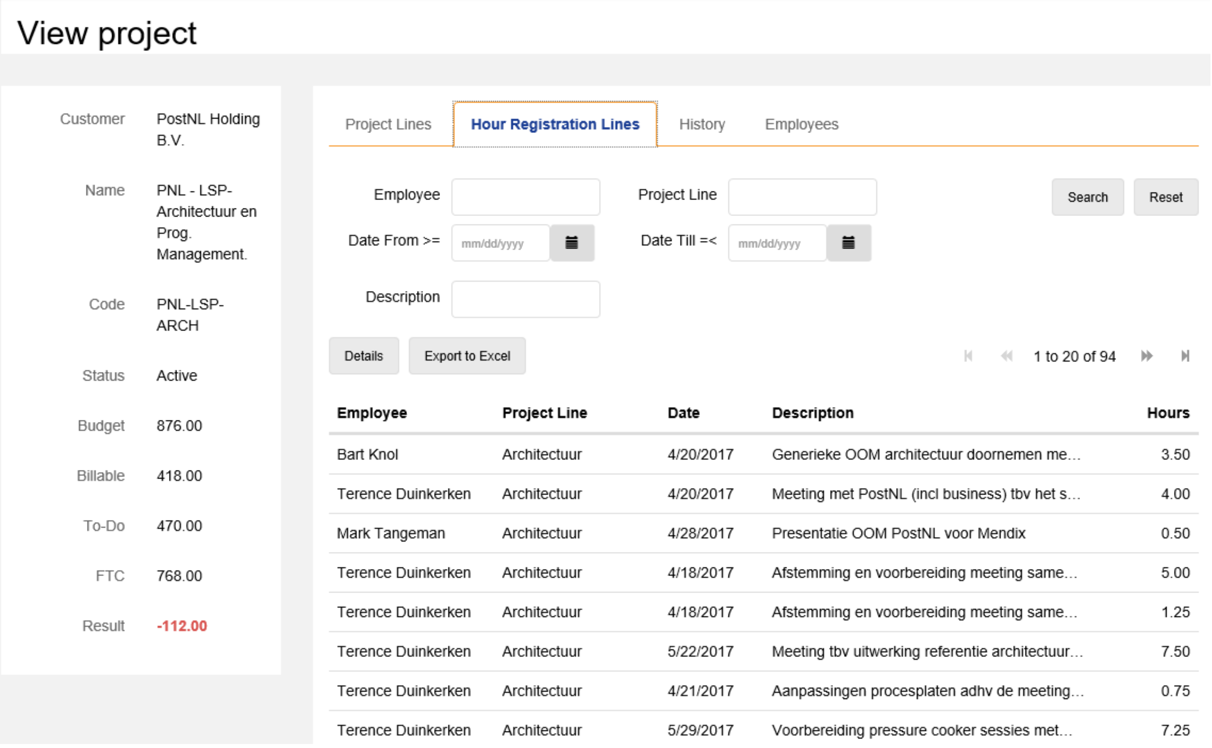
Date From >= mm/dd/yyyy
Date Till <= mm/dd/yyyy
Employee
Description

Date	Employee	Description	Hours
4/20/2017	Bart Knol	Generieke OOM architectuur doornemen met PostNL / Aantekeningen v…	3.50
4/20/2017	Terence Duinkerken	Meeting met PostNL (incl business) tbv het samenbrengen generieke O…	4.00
4/28/2017	Mark Tangeman	Presentatie OOM PostNL voor Mendix	0.50
4/18/2017	Terence Duinkerken	Afstemming en voorbereiding meeting samenbrengen vervanging Mede…	5.00
4/18/2017	Terence Duinkerken	Afstemming en voorbereiding meeting samenbrengen vervanging Mede…	1.25
5/22/2017	Terence Duinkerken	Meeting tbv uitwerking referentie architectuur OOM, inclusief voorbereidi…	7.50
4/21/2017	Terence Duinkerken	Aanpassingen procesplaten adhv de meeting mbt OOM architectuur en…	0.75
5/29/2017	Terence Duinkerken	Voorbereiding pressure cooker sessies met Mendix	7.25
5/30/2017	Terence Duinkerken	Eerste pressure cooker sessie met Mendix en PostNL tbv architectuur o…	7.25
5/31/2017	Terence Duinkerken	Tweede pressure cooker sessie met Mendix en PostNL tbv architectuur…	7.50
6/1/2017	Terence Duinkerken	Derde pressure cooker sessie met Mendix en PostNL tbv de referentie a…	8.00

Figure 18: Project Line Detail page

This page shows all the *hour registration lines* that are written on this phase by employees of company A. In this page, the user is able to search on date, employee and description of the *hour registration line.*

The user closes this page and selects the tab page *Hour Registration Lines.*



View project

Customer	PostNL Holding B.V.
Name	PNL - LSP-Architectuur en Prog. Management.
Code	PNL-LSP-ARCH
Status	Active
Budget	876.00
Billable	418.00
To-Do	470.00
FTC	768.00
Result	-112.00

Employee

Project Line

Search Reset

Date From >= mm/dd/yyyy

Date Till =< mm/dd/yyyy

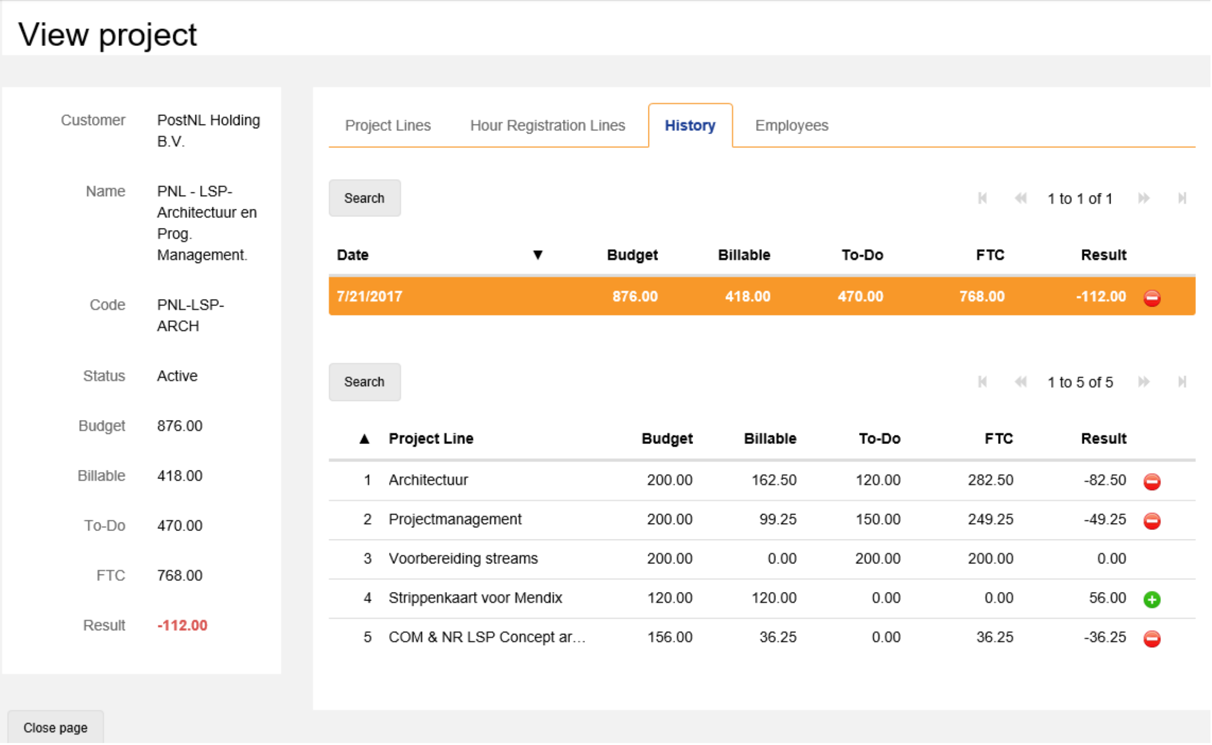
Description

1 to 20 of 94

Employee	Project Line	Date	Description	Hours
Bart Knol	Architectuur	4/20/2017	Generieke OOM architectuur doornemen me...	3.50
Terence Duinkerken	Architectuur	4/20/2017	Meeting met PostNL (incl business) tbv het s...	4.00
Mark Tangeman	Architectuur	4/28/2017	Presentatie OOM PostNL voor Mendix	0.50
Terence Duinkerken	Architectuur	4/18/2017	Afstemming en voorbereiding meeting same...	5.00
Terence Duinkerken	Architectuur	4/18/2017	Afstemming en voorbereiding meeting same...	1.25
Terence Duinkerken	Architectuur	5/22/2017	Meeting tbv uitwerking referentie architectuur...	7.50
Terence Duinkerken	Architectuur	4/21/2017	Aanpassingen procesplaten adhv de meeting...	0.75
Terence Duinkerken	Architectuur	5/29/2017	Voorbereiding pressure cooker sessies met...	7.25

Figure 19: Hour Registration Line page

On this page, all the *hour registration lines* of the project are shown. The user is able to search on employee, project line, date and description.



View project

Customer	PostNL Holding B.V.
Name	PNL - LSP-Architectuur en Prog. Management.
Code	PNL-LSP-ARCH
Status	Active
Budget	876.00
Billable	418.00
To-Do	470.00
FTC	768.00
Result	-112.00

1 to 1 of 1

Date ▼	Budget	Billable	To-Do	FTC	Result	
7/21/2017	876.00	418.00	470.00	768.00	-112.00	⊖

1 to 5 of 5

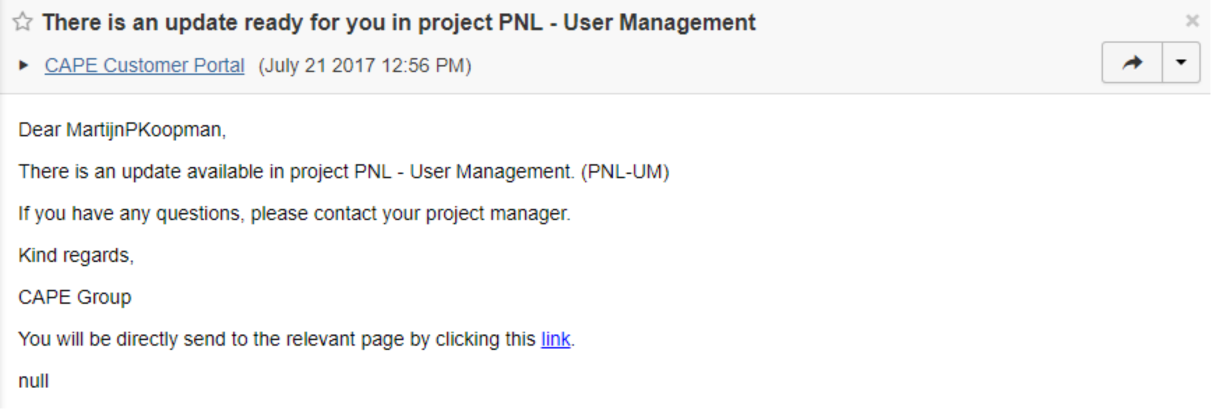
▲	Project Line	Budget	Billable	To-Do	FTC	Result	
1	Architectuur	200.00	162.50	120.00	282.50	-82.50	⊖
2	Projectmanagement	200.00	99.25	150.00	249.25	-49.25	⊖
3	Voorbereiding streams	200.00	0.00	200.00	200.00	0.00	
4	Strippenkaart voor Mendix	120.00	120.00	0.00	0.00	56.00	⊕
5	COM & NR LSP Concept ar...	156.00	36.25	0.00	36.25	-36.25	⊖

Figure 20: Project history page

The third tab page shows the history of the project. During the project, multiple updates will be send to CP. On this page, the user is able to keep track of the history of his project.

When the user clicks on a row in the upper table, the bottom table will be activated. This table shows the *Project Line* data on the 21st of July.
On the last tab page, employees of company A who are working on this project are displayed.

The last functionality of the portal is a notification e-mail when there is an update about a project.



There is an update ready for you in project PNL - User Management

CAPE Customer Portal (July 21 2017 12:56 PM)

Dear MartijnPKoopman,

There is an update available in project PNL - User Management. (PNL-UM)

If you have any questions, please contact your project manager.

Kind regards,

CAPE Group

You will be directly send to the relevant page by clicking this link.

null

Figure 21: Mail of CP

By clicking on the link, the user should fill in his credentials, before he will be re-directed to the *Project Overview* page (figure 16) The project that is on top of the table was updated the latest.

4.2 Mobile app

As mentioned in the introduction of this chapter, a mobile app has been developed as well. The app has the same functionalities as the desktop application. All the screenshots from the mobile app are on the bottom of this section.

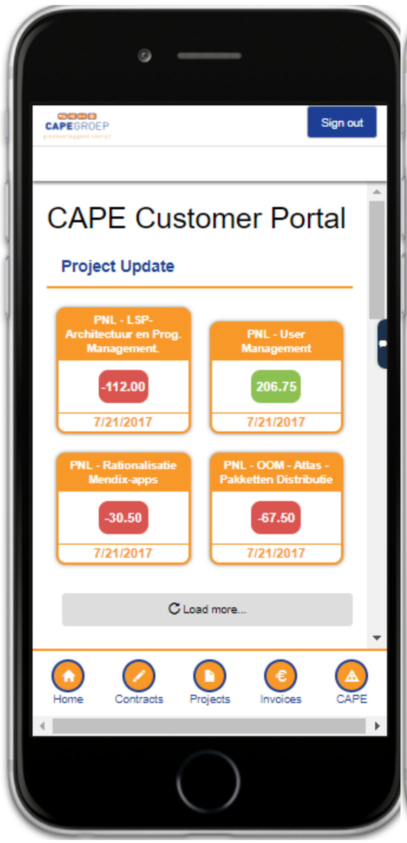
In the beginning, the user sees the same log in page as on the desktop. By filling in his credentials, the user comes on the mobile homepage. (figure 23) The homepage is somewhat different than the desktop version. On the bottom of the screen, there is a navigation menu where the user can zoom into projects or their tickets. When the user scrolls down in the page, news about company A, general information and the calendar will appear.

When the user clicks on *Projects* on the bottom of the page, the *Project Overview* page appears. (figure 24). This page has the same functionalities as the desktop page. When the user wants to see more details about a project, he simply clicks on the corresponding card.

Subsequently, the user ends up at the *Project Detail* page. (figure 25) Just as the desktop application, four tab pages are displayed where the user can look more closely to *project lines, hour registration lines* or the history of their project.

4.3 Summary

In this chapter, the functionalities of the application have been explained. A desktop- and a mobile application have been made to enable the customers to serve themselves to gain insight in their projects. In appendix C, the logic behind the app is explained.

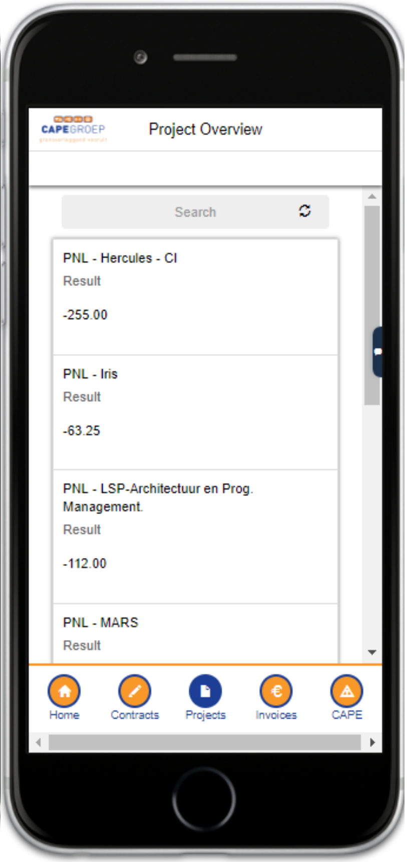


CAPE Customer Portal

Project Update

PNL - LSP-Architectuur en Prog. Management. -112.00 7/21/2017	PNL - User Management 206.75 7/21/2017
PNL - Rationalisatie Mendix-apps -30.50 7/21/2017	PNL - OOM - Atlas - Pakketten Distributie -67.50 7/21/2017

Figure 22: Mobile homepage



Project Overview

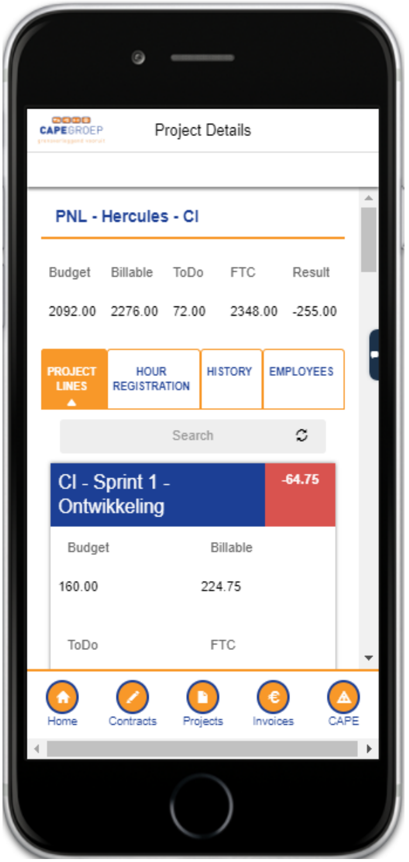
PNL - Hercules - CI
Result
-255.00

PNL - Iris
Result
-63.25

PNL - LSP-Architectuur en Prog. Management.
Result
-112.00

PNL - MARS
Result

Figure 23: Project Overview page mobile



Project Details

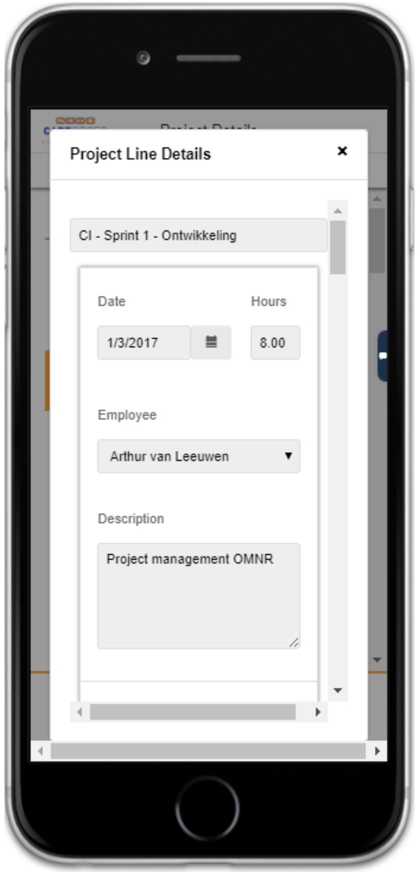
PNL - Hercules - CI

Budget	Billable	ToDo	FTC	Result
2092.00	2276.00	72.00	2348.00	-255.00

CI - Sprint 1 - Ontwikkeling -64.75

Budget	Billable
160.00	224.75
ToDo	FTC

Figure 24: Project Detail page mobile



Project Line Details

CI - Sprint 1 - Ontwikkeling

Date 1/3/2017

Hours 8.00

Employee Arthur van Leeuwen

Description Project management OMNR

Figure 25: Project Line Detail Page

Chapter 5: Evaluation

Now that the prototype is developed and explained, the question raises whether the application improves the current situation. Next to this, the application will be tested with the customers of company A. At last, the main question can be answered and from that conclusions can be drawn.

5.1 Testing the application

The application has been tested by two customers of company A, company B and company C. During this session, a brief introduction was given to the users and a quick demonstration of the functionalities of the application. After that, the user was free to explore the application by themselves and ask questions about it. At the end of the session, the users filled were asked to fill in a questionnaire about the portal and the current situation. The user was able to choose between the options: *strongly disagree, disagree, neutral, agree* and *strongly* agree when answering the questions. The questions were based upon the non-functionals needed for a web self-service portal. These non-functionals are:

1) Simple- the portal should be straightforward in the usage.
2) Convenient- it should be easier than other communicational means.
3) Controllable- customers should have a sense of control when using the portal.
4) Speed- it should be faster than other communicational means.
5) Continuous improvement- the portal should be able to continuously improve, since interests change over time.

The exact results and questions of this questionnaire can be found in appendix D.

The feedback that was received during these sessions and from the questionnaire was very useful for the further development of the portal. As can be deduced from the results of the questionnaire, the customers saw potential in the portal. They liked the fact that they could monitor their project in such detail and see the history of their project. The structure of the pages was also clear and appreciated. (so, from project to *project lines* to *hour registration lines*) Overall, they appreciated the portal, but they indicated that there is room for improvement.

Both customers missed the link with their invoice. They get a monthly invoice from company A where all the *hour registration lines* are shown, but they do not know from which phase or sprint these lines are. (More than one sprint can be executed in a month) They asked if there could be a way to show this in the invoice as well. They also asked whether the invoices could be displayed in the portal as well.

The last remark was that a customer could see where his project took longer than expected, but he does still not know why his project was delayed during that period. He hoped this could be indicated in the feature.

5.2 Improvement of the current situation

In chapter one, the core problem of the digital customer interaction was the lack of overview of their projects. In order to solve this problem, an application was designed and developed in the previous chapter. This application can improve the current situation, for example the project manager does not have to make a PowerPoint about the progress of the project every week, he simply clicks on a button when he approves the week. This click triggers the transfer of the message from IS to CP.

Furthermore, in the current situation customers receive this PowerPoint from the project manager. With the portal, the customer receives an e-mail with a notification. The customer can decide by themselves if they want to check the progress of their project.

When the customer decides to do that, he is able to follow his project in much more detail than before. It also becomes clearer when a project is delayed, by the overview that is created per *Project Line.* (figure 18)

Another feature of the portal is the ability to check the history of the project. In the old situation, the customer needed to look up the PowerPoint on their computer. In the portal, this information is gathered and displayed clearly. As mentioned in the previous paragraph, customers liked the fact that they could monitor their project in such detail, but there is still room for improvement. The portal itself could become clearer and

Chapter 6: Conclusion

The main question will be answered in this chapter. Furthermore, limitations and recommendations are elaborated as well.

6.1 Answering the main question

As mentioned in the introduction of this chapter, the answers to the sub-questions must be elaborated before the main question can be answered. The sub-questions and a brief answer to them are shown below.

What are the problems in the current customer interaction process? (chapter 1)
In paragraph 1.7 a problem cluster was made, and the core problem of the current situation was the lack of overview of the projects for the customer.

How can customer relationship management be applied to this case? (chapter 2)
In this case, CRM can be applied by to link the front- and back office with each other. In the back office, information about projects is present. A link is needed between the back- and front office to enable the customers to gain insight in their project. A way to do that is web self-service.

How can the artefact be developed in an agile way? (chapter 3)
The artefact or application can be developed by following the SCRUM method. SCRUM is a method which works in short development cycles to adapt quickly to any changes.

What does the architecture of my customer relationship model look like? (chapter 3)
The visual representation of the model is shown in figure 13. The model will link the back office, IS, with the front office, CSP. This is done with the help of eMagiz.

Does the artefact solve the problem? (chapter 5)
The artefact or application does solve the problem that is mentioned in the first sub-question. From the test sessions, positive results were collected. Testers indicated that the portal was an improvement of the current process and they liked the amount of detail they were able to see in the portal.

What are further improvements for the artefact? (chapter 5)
The portal could be improved with displaying the contracts and invoices of customers. These contracts could also be signed on the portal with a digital signature. Also, reasons why the project has been delayed could be shown in the portal as well.

Now that all the sub-questions are briefly answered above, the main research question can be answered. The main research question as stated in paragraph 1.7 is:
How can the interaction process with customers made more efficient with the help of an online portal?
In the current situation, customers do not have sufficient overview of their projects. PowerPoints are send by the project manager to the customer to show the progress of their project. It costs the project manager some time to make this PowerPoint and the customer is not always able to find the relevant information in the PowerPoint. One of the key points to increase CRM is to link the front- and the back office with each other. In the current situation, this link is present but insufficient since the customer lacks information in the PowerPoint.
Therefore, an online portal is developed to enable the customer to monitor the progress of their project in detail. A lot of information from the back office, so internal systems, is transferred to the portal, the front office.
This results in a more efficient interaction process. The project manager does not have to spend time on making a PowerPoint, he simply clicks on a button and all the information from the previous week is transferred to the portal. Furthermore, the customer is able to monitor their project in detail. The

current situation and history about the project is present in the portal. Customers can serve themselves and do not have to call company A to get certain information.

6.3 Limitations

It was not possible to develop the portal fully in the given time. As already came forward in paragraph 5.1 there are still some improvements and additions that can be made. A deliberate choice was made halfway in the process to focus on the project insight. This was developed during the remaining time of the investigation. It was also not possible to introduce the portal in an online environment. The portal runs locally on a laptop, it was not possible to realise this in the time given.

6.4 Recommendations

Customers liked the amount of overview the portal created, but they missed the link to their invoice. Making invoices visible in the portal is a feature that is strongly recommend for future development. In combination with the project overview that is already created, customers are able to see what was developed for the money that they are paying. Furthermore, enabling the customer to digitally sign their contracts on the portal could be an important functionality. (Geevers, 2017) Subsequently, indicating why a project takes longer than expected can increase the insight in their projects as well. But most of all, the portal should be introduced in an online environment. Company A may want to choose between introducing it with the current interface and functionalities or wait until more functionalities are built in. But it should come online so that customers can test the application live.

References

ACM. (2015). *Software Engineering Code of Ethics and Professional Practice*. Retrieved April 21, 2017, from Association for Computing Machinery: http://www.acm.org/about/se-code

Ahmad, H., Francis, A., & Zairi, M. (2007). Business process reengineering: Critical success factors in higher education. *Business Processs Management Journal, 13*(3), 451-469.

Al-Mashari, M., & Zairi, M. (1999). BPR implementation process: An analysis of key success and failure factors. *Business Process Management Journal, 5*(1), 87-112.

Chen, J., & Popovich, K. (2003). Understanding Customer Relationship Management (CRM). *Business Process Management Journal*(9), 672-688. doi:10.1108/14637150310496758

Cohen, D., Lindvall, M., & Costa, P. (2003). Agile Software development. *DACS SOAR Report, 11*, pp. 1-63.

Cooper, D., & Schindler, P. (2011). *Business Research Methods* (11 ed.). New York: McCraw-Hill/Irwin.

Cooper, V., Lichtenstein, S., & Smith, R. (2006). Knowledge Transfer in enterprise information technology support using web-based self-service systems. *International Journal of Technology Marketing*(2), pp. 145-170.

Crossman, A. (2017, March 2). *The major theoretical perspectives in sociology*. Retrieved April 14, 2017, from ThoughtCo: https://www.thoughtco.com/theoretical-perspectives-3026716

Express, S. (2017). *Resolve Issues Fast with a Complete IT Service Management Solution*. Retrieved April 24, 2017, from LBI Software: https://www.lbisoftware.com/productguides/LBi-Employee%20Self%20Service%20Portal-Product%20Guide.pdf

Gebert, H., Geib, M., Kolbe, L., & Brenner, W. (2003). Knowledge-enabled customer relationship management: Integrating customer relationship management and knowledge management concepts. (5), pp. 107-123. doi:10.1108/13673270310505421

Geevers. (2017). HET OPTIMALISEREN VAN HET ORDERPROCES VAN CAPE GROEP.

Grover, V., Jeong, S., Kettinger, W., & Teng, J. (1995). The implementation of business process reengineering. *Journal of Management Information Systems, 12*(1), 109-144.

Hammer, M., & Champy, J. (1993). *Re-engineering the corporation: a manifesto for business revolution.* New York: Harper Business.

Heerkens, H., & Winden, A. v. (2012). *Geen Probleem.* Utrecht: Van Winden Communicatie.

Hoekstra, J., Leeflang, P., & Wittink, D. (1999). The Customer Concept: The Basis for a New Marketing Paradigm. *Journal of Market-Focused Management, 4*(1), 43-76.

IEEE. (2000). *IEEE Recommended Practice for Architectural Description of Software Intensive Systems.* New York: IEEE Computer Society.

Kniberg, H. (2012, October 12). *Agile Product Ownership In A Nutshell*. Retrieved from Youtube: https://www.youtube.com/watch?v=502ILHjX9EE

Lu, X., Huang, L., & Heng, M. (2006). Critical success factors of inter-organizational information systems - A case study of Cisco and Xiao Tong in China. *Information and Management, 43*(3), 395-408.

Massey, A., Montoya-Weiss, M., & Holcom, K. (2001, December). Re-engineering the customer relationship: Leveraging knowledge assets at IBM. *Decision Support Systems, 32*(2), 155-170.

McGovern, G. (2014). *Web self-service Management Principles and Business Case, Customer CareWords*. Retrieved April 13, 2017, from InfoSys: https://www.infosys.com/industries/utilities/case-studies/Pages/web-self-service.aspx

Mendix. (2017, July 24). *Mendix 6 Reference Guide*. Retrieved from Mendix: https://docs.mendix.com/refguide6/modules

Meuter, M., Ostrom, A., Roundtree, R., & Bitner, M. (2000). Self-Service Technologies: Understanding customer satisfaction with technology-based service encouters. *Journal of Marketing, 64*(3), 50-64.

Miller, L. F. (2017, February 7). Motivating and Ensuring an Ethical Research Design [college slides]. *University of Twente*.

Moogk, D. (2012, March). *Minimum Viable Product and the Importance of Expirimentation in Technology Startups*. Retrieved from Technology Innovation Management Review: https://timreview.ca/article/535

Motwani, J., Kumar, A., Jiang, J., & Youssef, M. (1998). Business process reengineering: A theoretical framework and an integrated model. *International Journal of Operations & Production Management, 18*(9/10), 964-977.

Newman, M., & Zhao, Y. (2008). The process of enterprise resource planning implementation and business process re-engineering: Tales from two Chinese small and medium-sized enterprises. *Information Systems Journal, 18*(4), 405-426.

OpenGroup. (2003). *OpenGroup*. Retrieved from TOGAF "Enterprise Edition" version 8.1 : http://www.opengroup.org/architecture/togaf8

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems, 24*(3), 45-77.

Peppers, D., & Rogers, M. (1999). *Enterprise One-to-One: Tools for Competing in the Interactive Age.* New York: Doubleday.

Schwaber, K. (1997). Scrum development process. *Business Object Design and Implementation*, 117-134.

Schwaber, K., & Beedle, M. (2002). *Agile Software development with Scrum* (Vol. 1). Upper saddle River: Prentice Hall.

Schwede, S. (2000). Vision and Reality of CRM. *Information Management & Consulting*(15), pp. 7-11.

Seybold, P., Marshak, R., & Lewis, J. (2001). *The Customer Revolution.* New York, NY: Crown Business.

Sutherland, J., Scholingen, R., & van, R. E. (2011). *The Power of Scrum.* North Charleston: CreateSpace.

Teixeira-Pinto, V. (2014). *Randstad builds customer portal in six months using OutSystems and three developers*. Retrieved April 13, 2017, from OutSystems: https://www.outsystems.com/case-studies/randstad/customer-portal/

van Vliet, R. (2014). *Albelli focust of zeer tevreden klanten*. Retrieved April 13, 2017, from Kana: http://nl.kana.com/kana-express/casestudy/kana-express-albelli-case-study-nl.pdf

Verhoef, P. (2003). Understanding the effect of customer relationship management efforts on customer retention and customer share development. *Journal of Marketing, 67*(4), 30-45.

Weske, M. (2012). *Business Process Management* (2nd ed.). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-642-28616-2

Appendix

Appendix A: Summaries of interviews with different stakeholders

Interview with the project manager from company A

The project manager explained to me how company A works (see the SCRUM part) and how the projects are divided into the different sprints. He indicated that some contact moments with the customer were not running smoothly within the project. He started with the PO-numbers, which the customer needed to write down with a pen in an e-mail. After that, he mentioned that customers (project managers) can get an overview of their project in a weekly update which is send by company A. But he also mentioned that this could become a part of self-service, where the customer will be able to download this update on the portal by themselves. Furthermore, he indicated that the customers may get lost in all the contracts they have signed. Since there are also a lot of licenses involved in the projects.

Interview with the sales manager from company A

At the start of the projects, customers are attracted and advised by the sales manager from company A. If they are new customers for company A, their information is processed manually in the internal system of company A (IS). He indicated that this could be done electronically. Furthermore, he mentioned that processing contracts takes a lot of time. If a customer does not agree with the terms, the contract is being send back. He mentioned that this could be done electronically if that is legally possible. Another bachelor student is looking into this subject.

Interview with the support manager from company A

The support manager would like to make improvements in his system, but he is busy with solving ticket requests from customers. Therefore, he mentioned a lot of possible improvements in the interaction between company A and the customers. For example, when a ticket is being filled in by the customer, additional information is often needed for company A to solve the problem. Company A asks this per mail, but the customer can’t respond to that mail. (no-reply mail address) He also mentioned that a link with the agenda of the person who is going to solve a problem would be a significant improvement. In other words, when an employee of company A accepts solving a problem and estimates how long solving this will take, a “meeting” in his agenda will be made which states that he is going to solve the problem on Monday from 10 till 1 for example. Following this, the meeting will be shared with the customer, so that they can see when company A is going to repair their application.

After the interviews with different employees of company A, I went to two customers of company A. Information of these interviews was gathered so that improvements can be presented in a structured way.

Interview with the project manager of company C

Company C is a wholesaler in all kinds of products, for example sustenance products, but also perfume and tobacco. I presented my overview of the possible improvements at some touchpoints between company A and a customer. After the presentation, I asked if he saw added value in such a portal and if he struggles with other contact moments. He indicated that he saw added value in such an application, but he wanted to get a clear overview in the changes made within the sprints. So, if something is left out of sprint 1, will it be moved to sprint to and why. This is also an idea to present in the application. He had also some other ideas, but they fall out of the scope of this investigation. Ideas about contracts and such.

Interview with the project manager of company D

Company D is a transportation company. The project manager of company D saw potential in my design, he mentioned that he often lacks overview of the signed contracts and would like to see this back in my application. He also mentioned that he thought it was too much effort to request a ticket when his application goes down. Logging in and fill in the required fields takes too much time, he would simply like to send an e-mail to company A, which he gets from a member in his team, where the problem is stated. Next to this he had no remarks and he was rather enthusiastic about the application.

By understanding the culture of the company and the country, one must be able to come up with changes that the employees are willing to accept. If one does not understand the culture of the people, the changes he suggests will be against the believes of the people who work at the company. Also, a good understanding of the structure of the organization and the current processes is needed to come up with relevant improvements. At last, the technology that can be used for these changes is business process architecture. With this technology, one is able to map the process within a company and see where re-engineering can take place. I will make an architecture for my own case later in this report.

Appendix B: User Stories

Name
Sprint 1: Basic CP and budget status
As a developer, I would like to add my own module in CSP and make the domain model
Check which entities are already in use
Add entities for the portal
Project
Name (string)
ID number
As a customer, I would like to be able to log in with my own account, so that I am able to see information about the activities that COMPANY A does for me.
Use accounts created in CSP
As a customer, I would like to get to a homepage, where all the possible functions are shown.
Link to projects
Link to my account
Event bar
Link to CSP
Support status
News about COMPANY A
Action bar
Link to social media
See consignation
As a developer, I would like to create different roles in the application, so that different roles in organizations can be distinguished
Use the existing roles in CSP!
As a customer, I would like to have an overview of all my projects
Create blocks where the different projects are named
Link to contracts
Link to licenses
Link to project information
As a customer, I would like to see the agreed budget of my project(s)
External budget
Billable hours
To-do (hours)/tasks

FTC
External result
As a customer, I would like to see the hour registration of the employees of COMPANY A who are working on my project
Export to Excel
Date
Employee
Description
Billable hours
Filter between sprints, hours, employees, activity, status, date etc.
Sprint 2: Mobile and other components
Add UI/UX
As a customer, I would like to see recent updates of COMPANY A on the homepage, so, I get up-to-date with the recent developments within COMPANY A
Add calendar
As a customer, I would like to have a mobile app, so that I can make quick changes
Create insight in projects
Add calendar
Show contracts, invoices, status of the app on the homepage
As a developer, I would like to gain knowledge about eMagiz, so that I will be able to make integrations between applications
As a developer, I would like to retrieve information from IS and use it in CP
This can be done by making integrations with the help of eMagiz
As a customer, I would like to see the latest Facebook developments of COMPANY A, so that I am up-to-date with the recent developments within COMPANY A
As a customer, I would like to see the latest LinkedIn developments of COMPANY A, so that I am up-to-date with the recent developments within COMPANY A
As a developer, I would like to show a calendar, so that my clients are able to see events where COMPANY A will be present
For mobile and desktop
Sprint 3: Other features
As a customer, I would like to receive an e-mail when there is an update.
Show the relevant page with a link when an e-mail is send to the client. The link shows the updated project
As a customer, I would like to see the history of my weekly reports on the portal, so that I am able to check projects in the past.
As a customer, I would like to see an action bar, where all the changes or updates are shown
Prepare test environment
Write report
As a customer, I would like to see the telephone number that I need do dile when I am having trouble with my application
Also on mobile

Appendix C: Explanation of the functionality

Mendix

A Mendix project consists of modules: one System module and one or more user-defined modules. Modules are a way to split the functionality of your application into separate parts.
A module always contains exactly one domain model. The domain model is a data model that describes the information in your application domain in an abstract way.

Within a module you can define module security via module roles and specify security settings of those module roles for pages, microflows, entities and data sets.
Furthermore, a module can contain several types of documents. Each type of document is described in its own domain-specific language (DSL). For example, user-interface forms are described by using a visual language with elements like text boxes, tables and grids. Below you see tables grouped by category of all the different kinds of documents you can create within a module. (Mendix, 2017)

Common

Document type	Typical elements	Description
Pages	Data view, Data grid, Table, Text box	Forms are used to create a user interface for the end user. They are composed of components that are called widgets.
Microflows	Activities, Sequence Flow	Microflows describe the logic of your application. They are composed of activities that manipulate objects, interact with the client etcetera.
Enumerations		An enumeration is a set of predefined values, for example: in a web shop, an enumeration called MemberType could have the values Gold and Silver.
Consumed App Services		App services are the preferred way of connecting Mendix applications to each other. An app service can be imported and its content can be used.

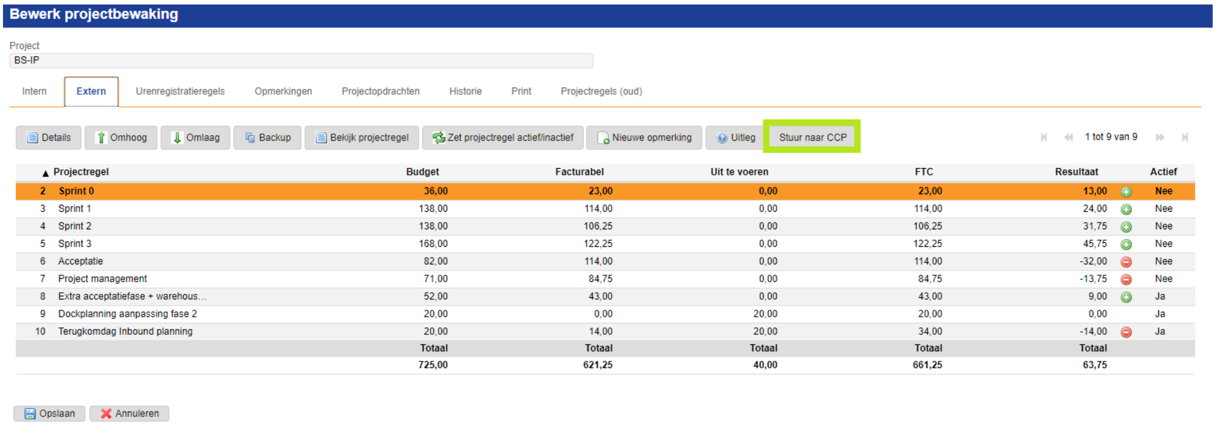
Table 2: Common Mendix functionalities (Mendix, 2017)

Information System (IS)

As mentioned in the introduction of this chapter, the process starts at IS. In IS, relevant data needs to be gathered for the message that is send to eMagiz. Since one project is send per message, one needs the following information:

- One project
- The project code (an abbreviation that is set for every project, every project code is unique)
- The status of the project (active or closed)
- The customer related to that project
- The project lines that are related to the project. Project lines are the phases within the project, these phases are often called sprints but can have other names as well.
- The characteristics of the project lines. Every project line has a name, number, budget, billable hours, hours to do, forecast to completion (FTC), a result and an enumeration of the result (positive, negative). This enumeration is used to stress the positive or negative result of a project line.
- Beneath the project lines, hour registration lines are attached. These lines are filled in by employees of company A in IS. These lines need to be in the message as well.
- An hour registration line contains the following attributes: date, name of employee, description and number of hours.

Now that we have our message, we need to gather the right information from IS. This is done with a microflow. This microflow must be triggered somewhere. A microflow can be triggered by a button or another event. In this case, the choice has been made to trigger the microflow by clicking on a button. In figure 14, a project overview is shown of a project with code *BS-IP.* When the project manager approves everything from the previous week, he clicks on the button *Send to CP* (button in the green box) to trigger the microflow.



Bewerk projectbewaking

Project
BS-IP

Intern | Extern | Urenregistratieregels | Opmerkingen | Projectopdrachten | Historie | Print | Projectregels (oud)

Details | Omhoog | Omlaag | Backup | Bekijk projectregel | Zet projectregel actief/inactief | Nieuwe opmerking | Uitleg | Stuur naar CCP

1 tot 9 van 9

	▲ Projectregel	Budget	Facturabel	Uit te voeren	FTC	Resultaat		Actief
2	Sprint 0	36,00	23,00	0,00	23,00	13,00	⊕	Nee
3	Sprint 1	138,00	114,00	0,00	114,00	24,00	⊕	Nee
4	Sprint 2	138,00	106,25	0,00	106,25	31,75	⊕	Nee
5	Sprint 3	168,00	122,25	0,00	122,25	45,75	⊕	Nee
6	Acceptatie	82,00	114,00	0,00	114,00	-32,00	⊖	Nee
7	Project management	71,00	84,75	0,00	84,75	-13,75	⊖	Nee
8	Extra acceptatiefase + warehous…	52,00	43,00	0,00	43,00	9,00	⊕	Ja
9	Dockplanning aanpassing fase 2	20,00	0,00	20,00	20,00	0,00		Ja
10	Terugkomdag Inbound planning	20,00	14,00	20,00	34,00	-14,00	⊖	Ja
		Totaal	Totaal	Totaal	Totaal	Totaal		
		725,00	621,25	40,00	661,25	63,75		

Opslaan | Annuleren

Figure 26: Project Overview page in IS

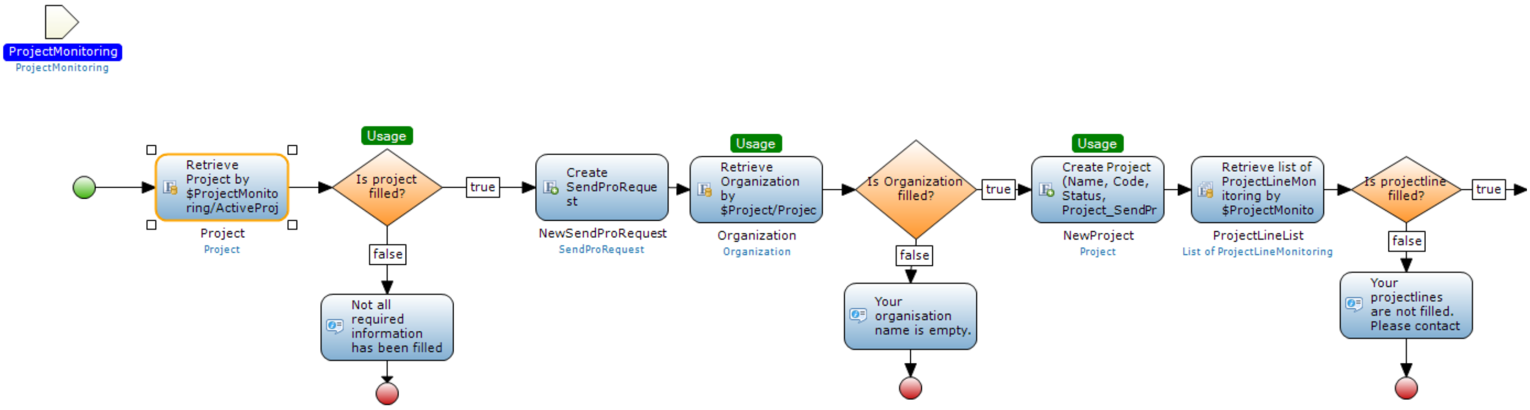
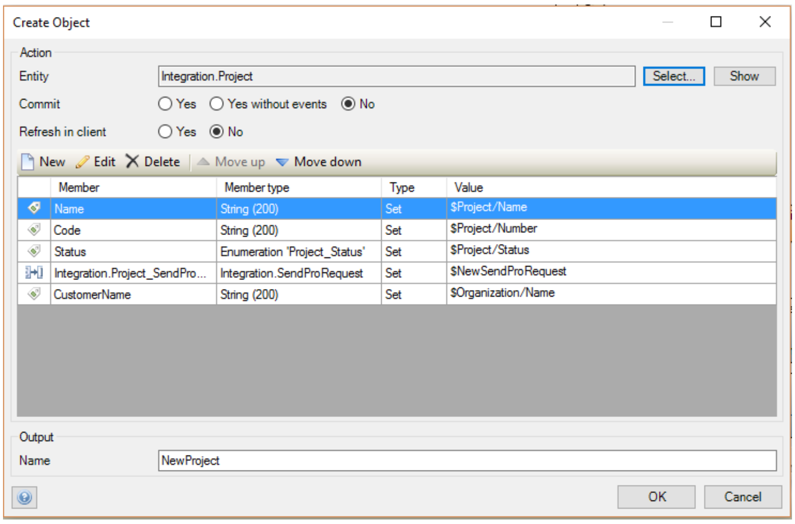

ProjectMonitoring
ProjectMonitoring
Retrieve Project by $ProjectMonitoring/ActiveProj
Project
Project
Usage
Is project filled?
true
false
Not all required information has been filled
Create SendProRequest
NewSendProRequest
SendProRequest
Usage
Retrieve Organization by $Project/Projec
Organization
Organization
Is Organization filled?
true
false
Your organisation name is empty.
Usage
Create Project (Name, Code, Status, Project_SendPr
NewProject
Project
Retrieve list of ProjectLineMonitoring by $ProjectMonito
ProjectLineList
List of ProjectLineMonitoring
Is projectline filled?
true
false
Your projectlines are not filled. Please contact

Figure 27: Microflow triggered by button Send to CCP (1)

The microflow will be explained block by block, from left to right.

- The microflow starts at the green dot on the left.
- Then a *Retrieve activity* is executed. A retrieve activity retrieves certain information from the database. In this case, this activity retrieves the *active* project, which means the project that the project manager tries to send to CP. (in this case the project with project code *BS-IP*)
- After that, an *exclusive split* is executed. An exclusive split makes a decision based on a condition and follows one and only one of the outgoing flows. (Mendix, 2017) In this case, the exclusive split checks if the active project is filled, if not the project manager gets an error message, if the project is filled, the flow will continue to the next activity.
- Then, a *create* activity is executed. A *create* activity creates a new object, in this case a new *sendprojectrequest*. This is necessary since Mendix needs to send a message to eMagiz. A request needs to be made to send this message.
- The customer, or in this case *organisation* is retrieved. This retrieve is done *by association* with project. This means that the customer connected to the selected project is retrieved. After the *retrieve*, an *exclusive split* is executed to check if the *organisation* is filled. This is done with the following code: $Organization/Name != **empty.** In this case, *organization* is the entity and *name* is the attribute. When the name is empty, or in other words, when no name is filled in, an error message will be shown to the project manager.



Create Object

Action

Entity: Integration.Project — Select... Show

Commit: ○ Yes ○ Yes without events ● No

Refresh in client: ○ Yes ● No

New | Edit | Delete | Move up | Move down

Member	Member type	Type	Value
Name	String (200)	Set	$Project/Name
Code	String (200)	Set	$Project/Number
Status	Enumeration 'Project_Status'	Set	$Project/Status
Integration.Project_SendPro...	Integration.SendProRequest	Set	$NewSendProRequest
CustomerName	String (200)	Set	$Organization/Name

Output

Name: NewProject

OK Cancel

Figure 28: Create new project activity

- Then a new project is created with the *create* activity.

  In this activity, a new project will be created with the help of the retrieved information in the previous activities of the microflow. In this case a new project is created with, a name, a code, a status and a customer name. Also, the association with *sendprorequest* is set here. This is done to send a project in the message to eMagiz.
- After that, a list of *project lines* associated to the active project are retrieved and checked if the list is filled. In the case of *BS-IP,* there should be nine *project lines.*

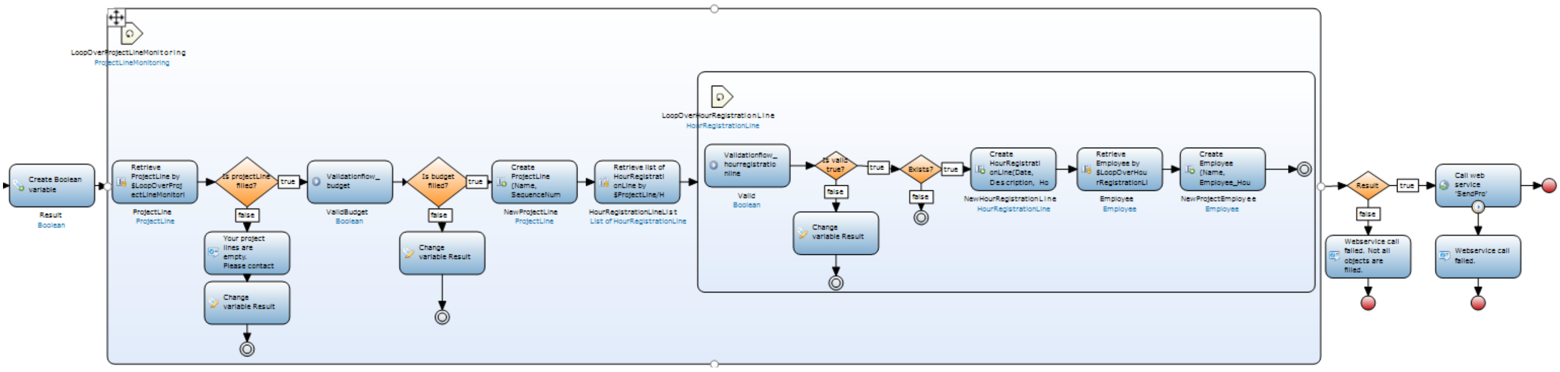
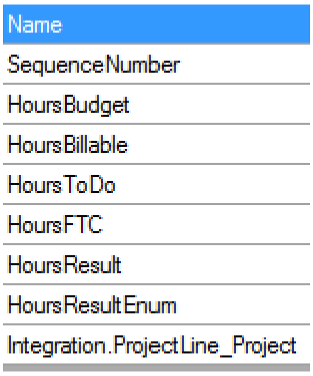

LoopOverProjectLineMonitoring
ProjectLineMonitoring
Create Boolean variable
Result
Boolean
Retrieve ProjectLine by $LoopOverProjectLineMonitori
ProjectLine
ProjectLine
Is projectLine filled?
true
false
Your project lines are empty. Please contact
Change variable Result
Validationflow_budget
ValidBudget
Boolean
Is budget filled?
true
false
Change variable Result
Create ProjectLine (Name, SequenceNum
NewProjectLine
ProjectLine
Retrieve list of HourRegistrationLine by $ProjectLine/H
HourRegistrationLineList
List of HourRegistrationLine
LoopOverHourRegistrationLine
HourRegistrationLine
Validationflow_hourregistrationline
Valid
Boolean
Is valid true?
true
false
Change variable Result
Exists?
true
false
Create HourRegistrationLine(Date, Description, Ho
NewHourRegistrationLine
HourRegistrationLine
Retrieve Employee by $LoopOverHourRegistrationLi
Employee
Employee
Create Employee (Name, Employee_Hou
NewProjectEmployee
Employee
Result
true
false
Webservice call failed. Not all objects are filled.
Call web service 'SendPro'
Webservice call failed.

Figure 29: Microflow triggered by Send to CP (2)

- When all the *project lines* are filled, a new *variable* will be created. This *variable* is used in the upcoming loops to check if everything is filled. The *variable* is a Boolean (true or false). The default value of this *variable* is true, but if there is anything in the loop that is not filled, a property of an hour registration line for example, then the *variable* will be changed to false. After the loops, the *variable* will be checked on his value, if the value is false, the project manager will see an error message on his screen.



Name
SequenceNumber
HoursBudget
HoursBillable
HoursToDo
HoursFTC
HoursResult
HoursResultEnum
Integration.ProjectLine_Project

Figure 30: Information needed for a project line

- First, a loop will be executed over the *project lines*. The first activity in the loop will retrieve one *project line.* After that a check will be executed to see if this *project line* is filled with a name and number.
- Subsequently, another microflow is called (*Validationflow_budget)* to check if the *budget, billable hours, hours to do, forecast to completion* and *result* are filled for this *project line.*
- Then, a new *project line* is created with the relevant information for a *project line.* See figure 18. The last row in figure 18 sets the association between *project* and *project line* so that Mendix knows that it needs to send *project lines* together with the *project.*
- The last activity before the other loop is the retrieval of a list of the *hour registration lines* associated with the current *project line.*
- The next loop will iterate over the *hour registration lines.*
- First, another microflow called *Validationflow_hourregistrationline* will be called to check if one *hour registration line* is filled with a date, description and billable hours.

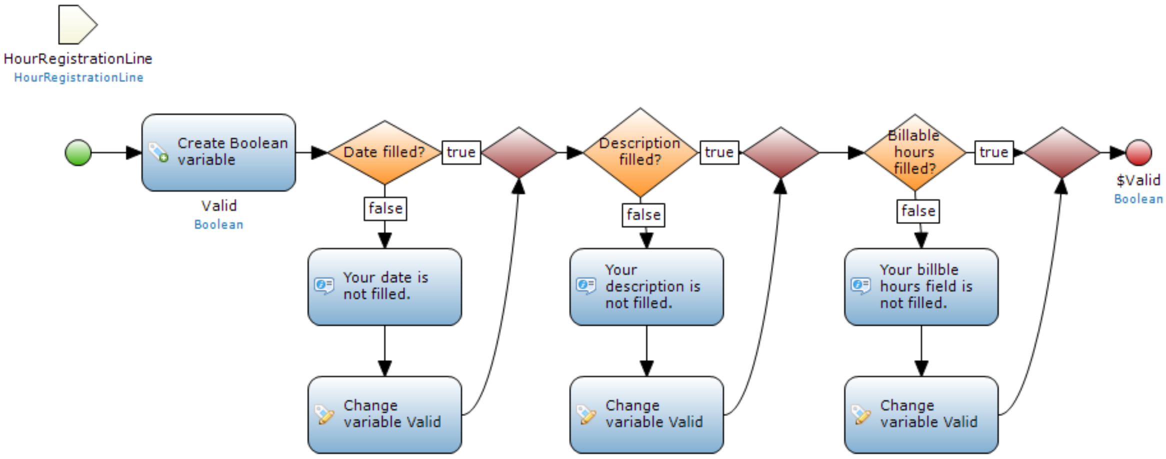


Figure 31: Microflow validationflow_hourrregistrionline

- When this is all filled, a new *hour registration line* will be made with a date, a description and a number of hours. Also, the association between *project line* and *hour registration line* is set so that *hour registration lines* are included in the message.
- The last thing we need for the message are the employee names. The employees of COMPANY A fill in a number of *hour registration lines* on a project.
- First, the employee associated to the *hour registration line* is needed. This is done by a retrieve activity.
- Then a new employee is created. This employee only needs a name. Also, the association between *hour registration line* and *employee* is set.
- The microflow will loop over all the *project lines* and *hour registration lines* in a project. When both the loops are finished a final check will be executed to see if the whole message is filled. This is done by checking the value of the Boolean *variable* that was introduced just before the loops. When the value of this *variable* is true, then everything is filled and the flow will continue. If not, an error message will be shown to the project manager.

Since there is a message filled with a *project, project lines, hour registration lines and employees,* the only thing that needs to be done is sending the message to eMagiz. This is done with a *consumed webservice.* Web services are a way of exposing or absorbing functions and/or data entities between systems. They can be used to enable applications to “talk” to each other through the network (or internet). Mendix supports the interaction between servers using SOAP. SOAP defines a standard way by which systems know how to communicate with each other. XML is used as the message format. This can either be Mendix-to-Mendix, Mendix-to-Third Party or third Party-to-Mendix. (Mendix, 2017)

In this case, a XML scheme is needed to send the message from IS (Mendix) to eMagiz. (Mendix-to-third-party) The XML scheme of Mendix must correspond to the *request handler* of eMagiz. A *request handler* is the message mapping of the incoming message to eMagiz. When these two mappings do not correspond, errors will be generated in Mendix, because eMagiz expects certain data which do not correspond to the message that is send from Mendix. This *request handler* is loaded into Mendix so that it can be mapped against the message send from IS.

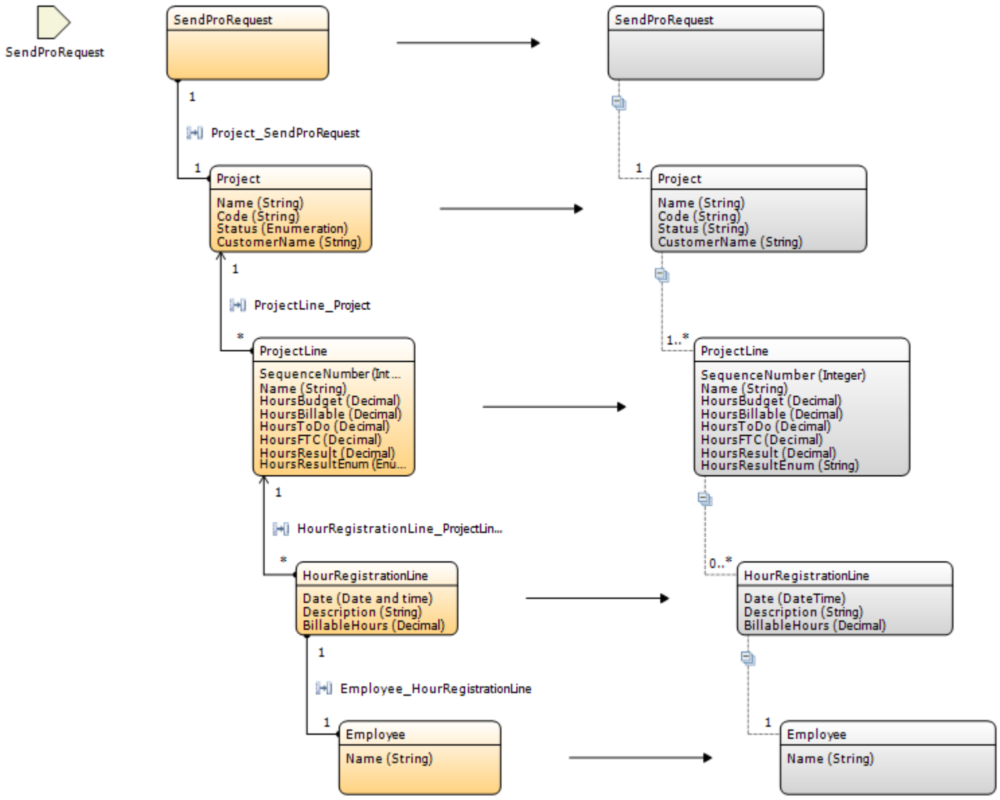
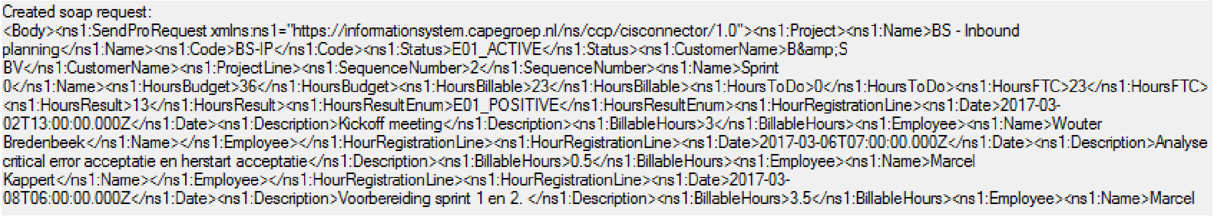


Figure 32: Message mapping in IS (left is XML scheme from IS, right is request handler of eMagiz)

As can be seen in figure 20, all the names of the mapping correspond to each other so the message can be received by eMagiz. An example of such a message is shown in figure 21.



```
Created soap request:
<Body><ns1:SendProRequest xmlns:ns1="https://informationsystem.capegroep.nl/ns/ccp/cisconnector/1.0"><ns1:Project><ns1:Name>BS - Inbound
planning</ns1:Name><ns1:Code>BS-IP</ns1:Code><ns1:Status>E01_ACTIVE</ns1:Status><ns1:CustomerName>B&amp;S
BV</ns1:CustomerName><ns1:ProjectLine><ns1:SequenceNumber>2</ns1:SequenceNumber><ns1:Name>Sprint
0</ns1:Name><ns1:HoursBudget>36</ns1:HoursBudget><ns1:HoursBillable>23</ns1:HoursBillable><ns1:HoursToDo>0</ns1:HoursToDo><ns1:HoursFTC>23</ns1:HoursFTC>
<ns1:HoursResult>13</ns1:HoursResult><ns1:HoursResultEnum>E01_POSITIVE</ns1:HoursResultEnum><ns1:HourRegistrationLine><ns1:Date>2017-03-
02T13:00:00.000Z</ns1:Date><ns1:Description>Kickoff meeting</ns1:Description><ns1:BillableHours>3</ns1:BillableHours><ns1:Employee><ns1:Name>Wouter
Bredenbeek</ns1:Name></ns1:Employee></ns1:HourRegistrationLine><ns1:HourRegistrationLine><ns1:Date>2017-03-06T07:00:00.000Z</ns1:Date><ns1:Description>Analyse
critical error acceptatie en herstart acceptatie</ns1:Description><ns1:BillableHours>0.5</ns1:BillableHours><ns1:Employee><ns1:Name>Marcel
Kappert</ns1:Name></ns1:Employee></ns1:HourRegistrationLine><ns1:HourRegistrationLine><ns1:Date>2017-03-
08T06:00:00.000Z</ns1:Date><ns1:Description>Voorbereiding sprint 1 en 2. </ns1:Description><ns1:BillableHours>3.5</ns1:BillableHours><ns1:Employee><ns1:Name>Marcel
```

Figure 33: Example of a message send by IS

eMagiz

eMagiz is a platform which enables consultants to make integrations between applications. In this case, an integration is needed between IS and CP. In the previous part, the message itself was explained and how this message is send to eMagiz. The message enters eMagiz through the *request handler*, then the message is transformed in eMagiz and send through to CP. In eMagiz, a mapping similar to the XML mapping in Mendix is made to structure the transformation. In this case, the message mapping in eMagiz corresponds to the mapping made in Mendix, therefore eMagiz only passes through the message to CP. eMagiz is able to do all kinds of adjustments to the message itself, but with this message that is not necessary. A lot more happens in eMagiz, but these details will not be discussed in this report.
As shown in figure 21, the message will go from IS, through eMagiz, to CP.

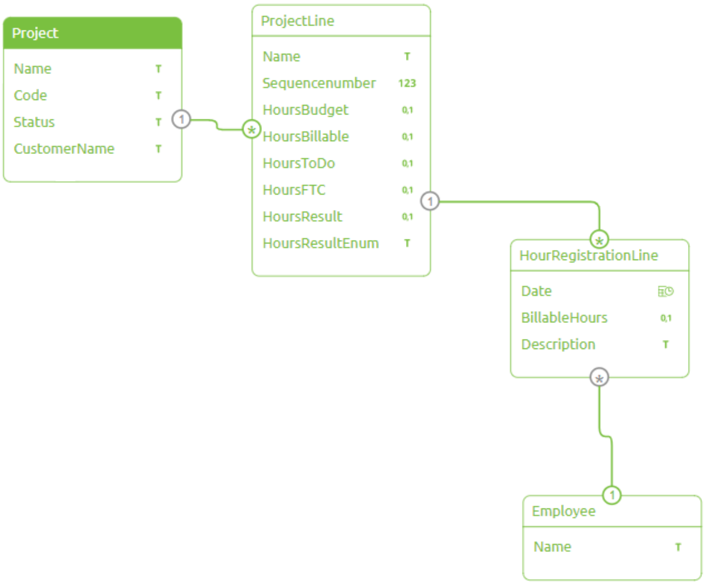


Figure 34: Message structure in eMagiz

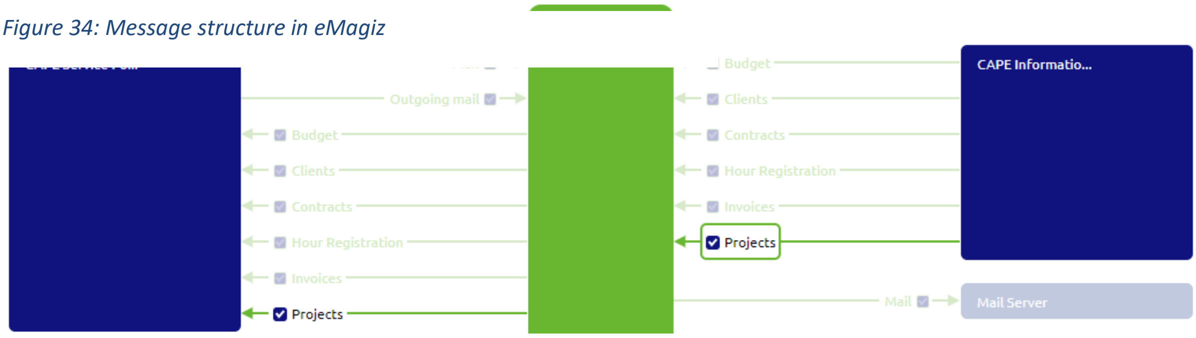


Figure 35: Project integration in eMagiz

CP

CP is the Mendix application where the message is send to. Before this message can be received a new module must be made in the former Mendix application CSP. Because of this, CP runs on an old database of CSP. (Mendix 6.9.1).
A domain model is a data model that describes the information in your application domain in an abstract way. It is central to the architecture of your application. The domain model consists of entities and their relations represented by associations. (Mendix, 2017) The domain model of the module is shown in figure 22.

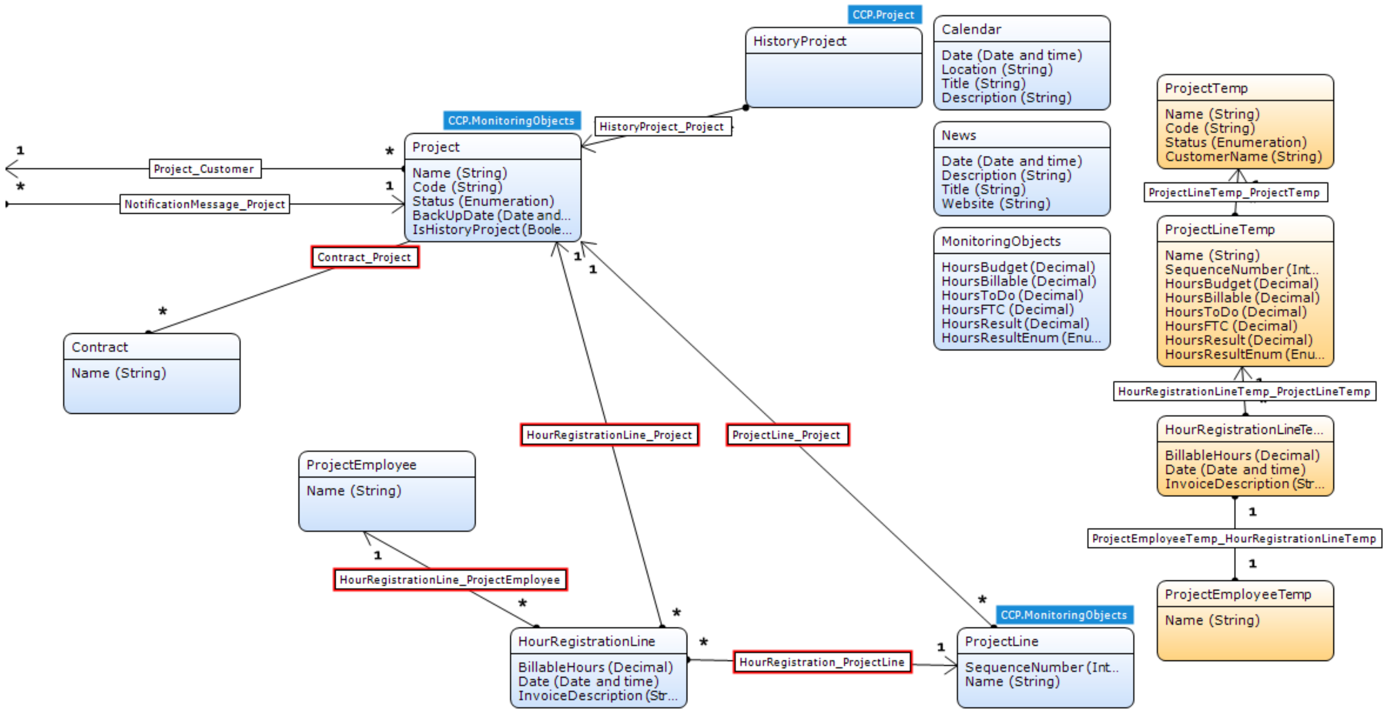
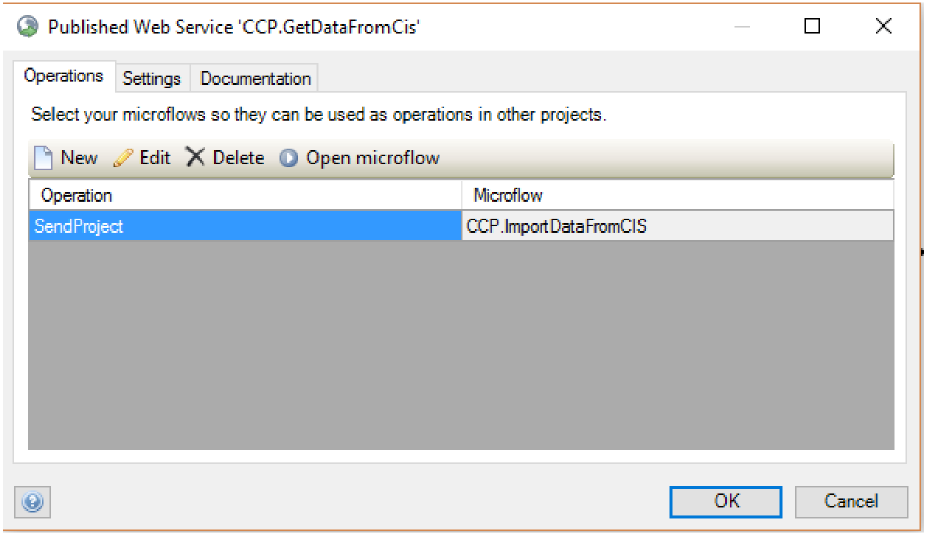


Figure 36: Domain model of CP

In the domain model, the blue boxes represent the *entities*. The lines between two entities are called *associations*. The text in the blue boxes are called *attributes.* For example. *BackUpDate* is an *attribute* of the *entity Project.* The difference between the blue and the orange boxes will be explained later.

The domain model can be explained by describing how the message enters CP. The message enters CP by a *published webservice.* The function of a *published webservice* is to receive the incoming message from a third party, in this case this third party is eMagiz.



Select your microflows so they can be used as operations in other projects.

Operation	Microflow
SendProject	CCP.ImportDataFromCIS

Figure 37: Published webservice CP

When a message is received from eMagiz, a microflow will be called. (in this case the microflow is called *ImportDataFromCIS)*

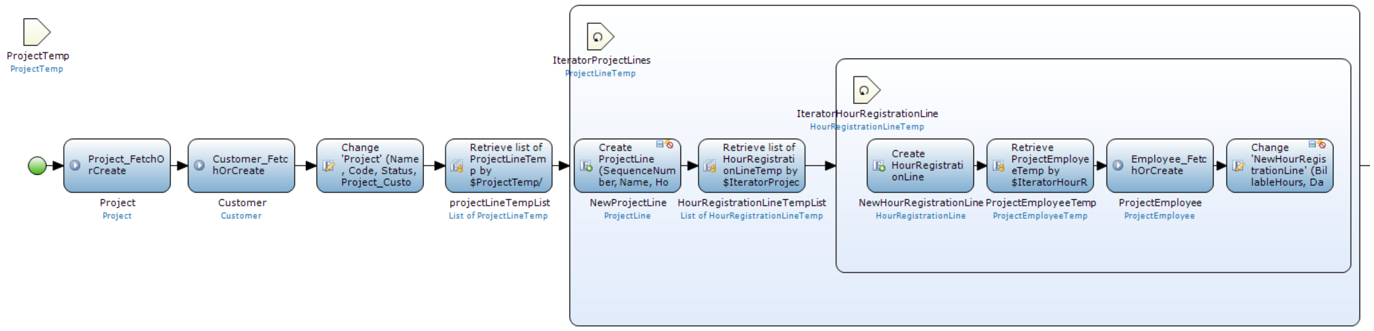

ProjectTemp
ProjectTemp
Project_FetchOrCreate
Project
Project
Customer_FetchOrCreate
Customer
Customer
Change 'Project' (Name, Code, Status, Project_Custo
Retrieve list of ProjectLineTemp by $ProjectTemp/
projectLineTempList
List of ProjectLineTemp
IteratorProjectLines
ProjectLineTemp
Create ProjectLine (SequenceNumber, Name, Ho
NewProjectLine
ProjectLine
Retrieve list of HourRegistrationLineTemp by $IteratorProjec
HourRegistrationLineTempList
List of HourRegistrationLineTemp
IteratorHourRegistrationLine
HourRegistrationLineTemp
Create HourRegistrationLine
NewHourRegistrationLine
HourRegistrationLine
Retrieve ProjectEmployeeTemp by $IteratorHourR
ProjectEmployeeTemp
ProjectEmployeeTemp
Employee_FetchOrCreate
ProjectEmployee
ProjectEmployee
Change 'NewHourRegistrationLine' (BillableHours, Da

Figure 38: ImportDataFromCIS (1)

Persistability

The difference between the blue and orange entities in figure 20 must become clear before the microflow can be explained. The blue entities are *persistable*. When an entity is declared *persistable*, a database table is created for the entity. Committing an instance of such an entity results in a row being inserted into the table. Attribute and association information stored in this instance is saved in the database as well. (Mendix, 2017)

The orange entities are so called *non-persistable entities. Non-persistable entities* cannot be stored in the database and hence have no associated database table. Committing non-persistable entities only stores the current attribute values and association values in memory, which allows for a rollback to revert to those values. (Mendix, 2017)

This functionality is needed, because certain changes need to be made to the incoming data before they are stored into the database. These *non-persistable entities* are suitable to make these changes before the data is stored into the database.

The microflow of figure 25 starts with a cross-reference to another microflow. This microflow is shown in figure 26.

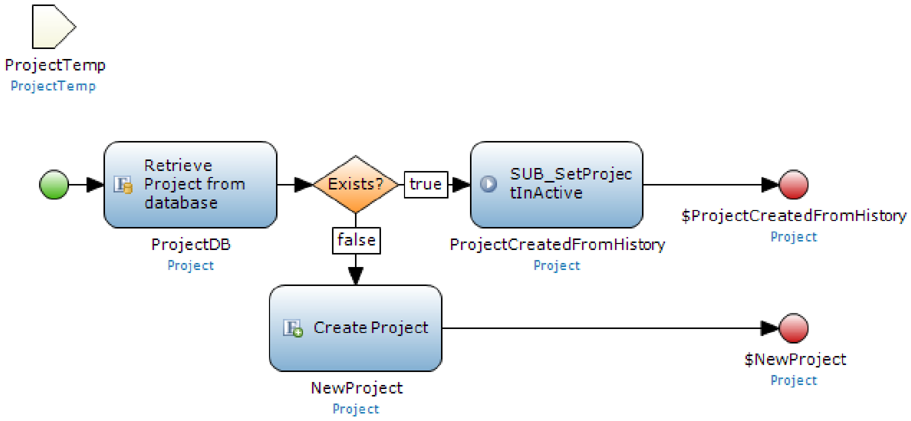


Figure 39: Microflow Project_FetchOrCreate

Since a lot of projects are going to be uploaded to CP, the need of avoiding duplicate projects is present. That is the idea behind *fetch or create.* In figure 26, all projects present in the database of CP are retrieved. With the help of the *exclusive split*, a check is done to see if the name of the project send in the incoming message corresponds with a project name that is already present in the database. A new project will be created if there is no project with such a name. (the lowest path) Another microflow will be called when there is already a project with the same name present in the database. This is the concept of *fetch or create, fetch* when the object already exists, *create* a new object if the object does not exist yet.

History

For now, the assumption is made that the project already exists, so the upper path will be followed. The upper path calls another microflow.

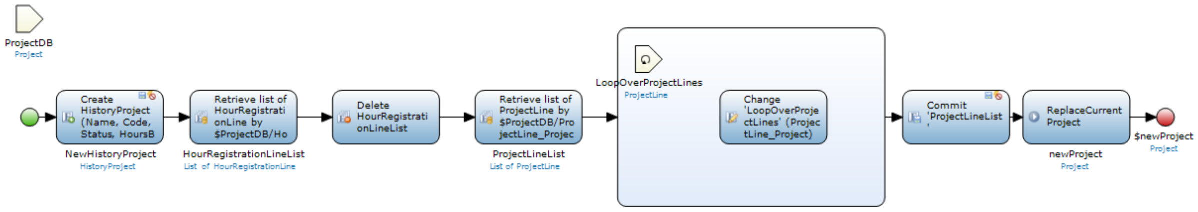
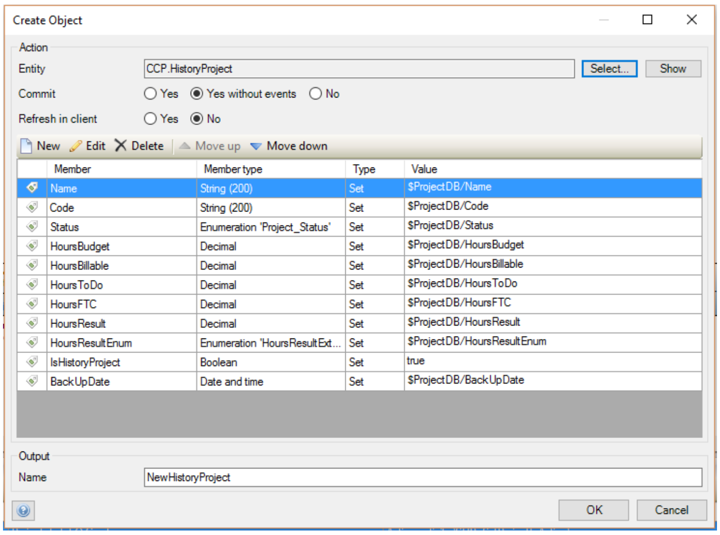

ProjectDB
Project
Create HistoryProject (Name, Code, Status, HoursB
NewHistoryProject
HistoryProject
Retrieve list of HourRegistrationLine by $ProjectDB/Ho
HourRegistrationLineList
List of HourRegistrationLine
Delete HourRegistrationLineList
Retrieve list of ProjectLine by $ProjectDB/ProjectLine_Projec
ProjectLineList
List of ProjectLine
LoopOverProjectLines
ProjectLine
Change 'LoopOverProjectLines' (ProjectLine_Project)
Commit 'ProjectLineList'
ReplaceCurrent Project
newProject
Project
$newProject
Project

Figure 40: Microflow SUB_SetProjectInActive

Being able to see the history of the progress is very important for the customers. Therefore, the distinction must be made between the last update and previous updates of a project.

The assumption was made that there was already an update present from the project, so the update that was already in the database must become a *History Project* and the latest project update should become a 'normal' project. This is done in the microflow from figure 27.

- First, a *NewHistoryProject* is created. In figure 28, all the attributes of a *history project* are shown.



Create Object

Action

Entity	CCP.HistoryProject
Commit	○ Yes ● Yes without events ○ No
Refresh in client	○ Yes ● No

Member	Member type	Type	Value
Name	String (200)	Set	$ProjectDB/Name
Code	String (200)	Set	$ProjectDB/Code
Status	Enumeration 'Project_Status'	Set	$ProjectDB/Status
HoursBudget	Decimal	Set	$ProjectDB/HoursBudget
HoursBillable	Decimal	Set	$ProjectDB/HoursBillable
HoursToDo	Decimal	Set	$ProjectDB/HoursToDo
HoursFTC	Decimal	Set	$ProjectDB/HoursFTC
HoursResult	Decimal	Set	$ProjectDB/HoursResult
HoursResultEnum	Enumeration 'HoursResultExt...	Set	$ProjectDB/HoursResultEnum
IsHistoryProject	Boolean	Set	true
BackUpDate	Date and time	Set	$ProjectDB/BackUpDate

Output

Name	NewHistoryProject

Figure 41: New History Project

- After that, all the *hour registration lines* of the current project are retrieved and deleted. All the *hour registration lines* of the project are in the incoming message, also from previous weeks, so there is no need to keep this data.
- Subsequently, all the *project lines* are retrieved from the project update that was already present. In the loop, all the *project lines* from the database will be connected to the *NewHistoryProject.* (the first activity of the microflow) This activity changes the 'status' of the *project lines* from 'normal' lines to 'history' *project lines.* This action is *committed*, so that the results will be directly visible in the application itself.
- At last, another microflow is called.

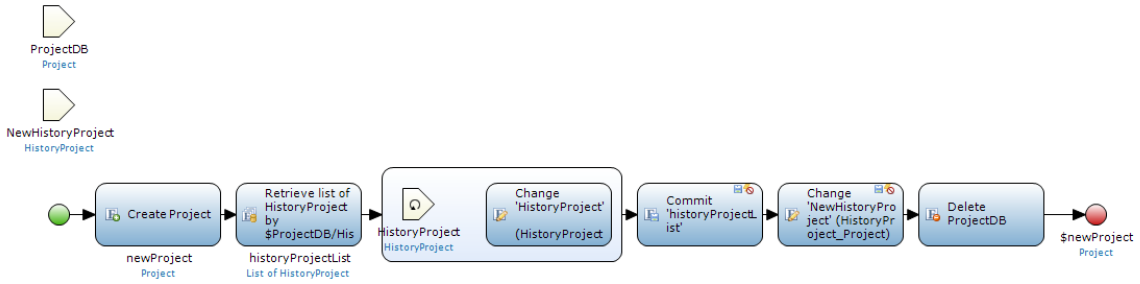


Figure 42: Microflow ReplaceCurrentProject

- Since the project that was already in the database has been converted to a *history project,* the new project, from the incoming message, can be created. This is done in the first activity of this microflow.
- After that, all the present *history projects* are retrieved as a list. In the loop, the association is made between project and *historyproject.* With this change, the new *historyproject* and the ones that were already a *historyproject* will be visible. (can be more than one since there can be more than one *history message* about a project).
- At last, the project in the database will be deleted, because this project has become needless. A project update is now a 'normal' project or a *historyproject.*

After a long detour, the second activity from figure 25 will take place.

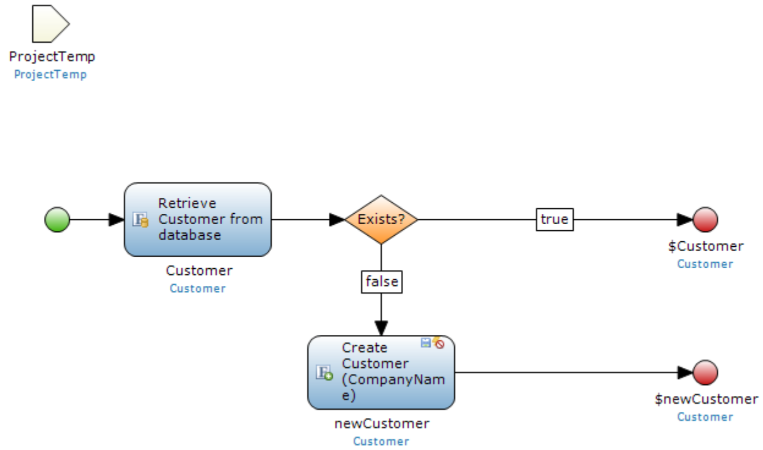
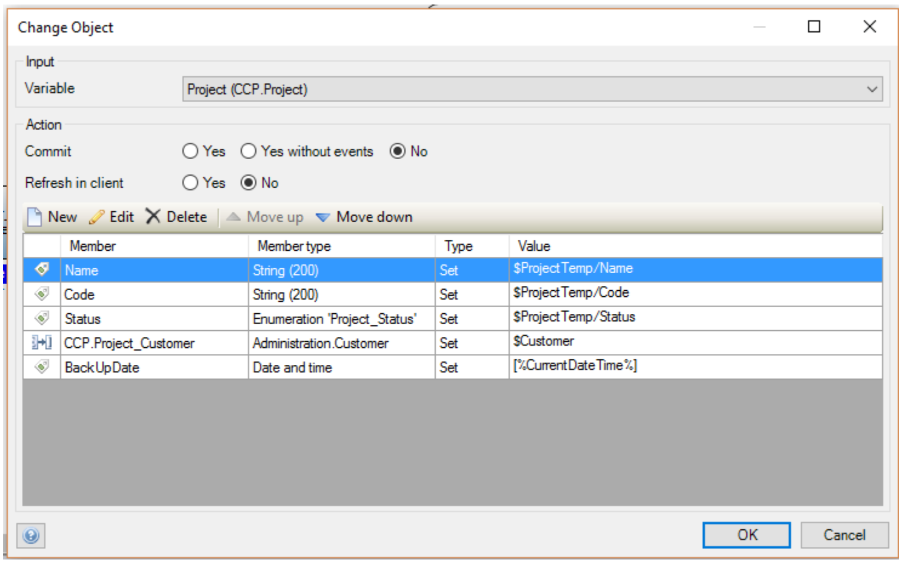


Figure 43: CustomerFetchOrCreate

The second activity also calls a microflow. This microflow works from the same principle as the previous *fetch or create.* A list of customers is retrieved from the database. When this customer is already in the database, the microflow will return that customer. A new customer will be created it this is not the case.

After this, the *persitable* object project is filled with the *non-persistable* project data.



Change Object

Input

Variable: Project (CCP.Project)

Action

Commit: ○ Yes ○ Yes without events ● No

Refresh in client: ○ Yes ● No

Member	Member type	Type	Value
Name	String (200)	Set	$ProjectTemp/Name
Code	String (200)	Set	$ProjectTemp/Code
Status	Enumeration 'Project_Status'	Set	$ProjectTemp/Status
CCP.Project_Customer	Administration.Customer	Set	$Customer
BackUpDate	Date and time	Set	[%CurrentDateTime%]

Figure 44: Change project

In the webservice, the *non-persistable* entity called *projecttemp* is filled with the information from the incoming message. With this change activity, this information is stored into a *persistable entity* and can therefore be stored into the database.
Subsequently, the *projectlinetemp* list is retrieved. Again, the *projectlinetemp,* is a *non-persistable* -entity which is filled with the incoming message.
The loop will iterate over all the *project lines.* With the help of the create activity, new *project lines* will be created from the data in the *projectlinetemp* entity. At last, for every *project line* the list of *hourregistrationlinetemp* will be retrieved.
The next loop will iterate over the *hour registration lines.* First, an *hour registration line* is created. After that, the list of *projectemployeetemp* is retrieved. The micro checks with *fetch or create* if the employee already exists in the data or a new employee needs to be created in the database. The last activity in this loop is the change object where the *hourregistrationlinetemp* entity is filled with date, billable hours, description and employee.

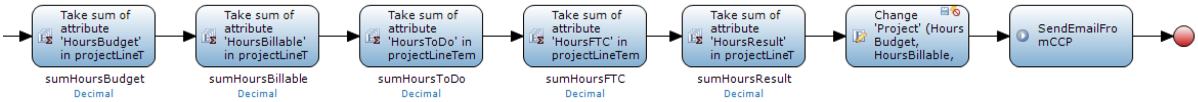

Take sum of attribute 'HoursBudget' in projectLineT
sumHoursBudget
Decimal
Take sum of attribute 'HoursBillable' in projectLineT
sumHoursBillable
Decimal
Take sum of attribute 'HoursToDo' in projectLineTem
sumHoursToDo
Decimal
Take sum of attribute 'HoursFTC' in projectLineTem
sumHoursFTC
Decimal
Take sum of attribute 'HoursResult' in projectLineT
sumHoursResult
Decimal
Change 'Project' (Hours Budget, HoursBillable,
SendEmailFromCCP

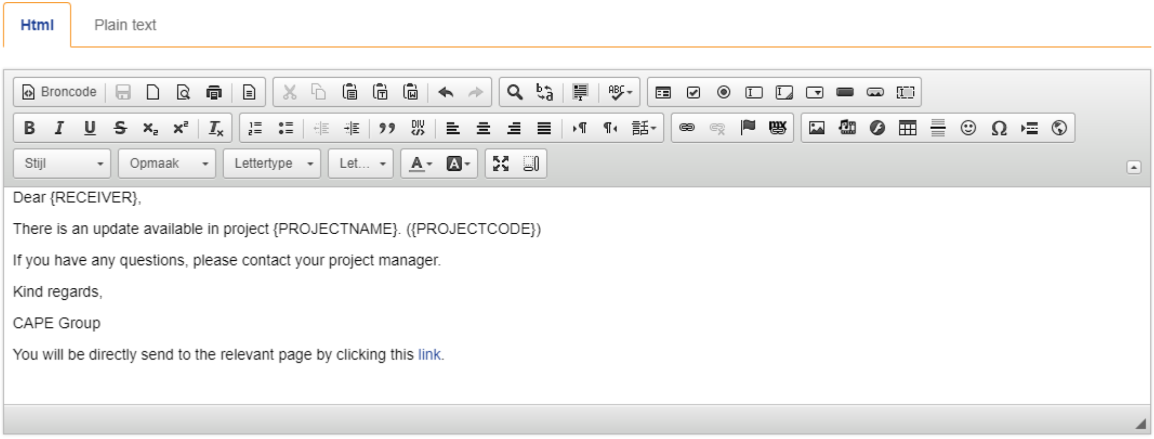
Figure 45:Microflow ImportDataFromCIS (2)

The last part of the microflow is rather straightforward. The first five activities have almost the same function. In the loop over the *project lines*, budget, billable hours, hours to do, forecast to completion and result have been imported into the project. These numbers are all on *project line* level. These data must also be visible on project level, for example the result of the project as a whole and not only the result of one *project line.* This can be done by accumulating all the values of result of the *project lines.* In the change activity after that, the values of, budget, billable hours, hours to do, forecast to completion and result are changed for project.

E-mail

In order to notify the customer of an update, an e-mail functionality has been built in as well.
There was already an e-mail functionality present in CSP which could be used to send these e-mails as well. Since I did not build this whole functionality myself, I will only explain briefly how this functionality was developed.

First, a new e-mail template needed to be made. This template exists of plain text and parameters. Parameters are needed to send the e-mail to the right customer. In this case the parameters are *project name, project code, a receiver* and *an URL.*



Dear {RECEIVER},

There is an update available in project {PROJECTNAME}. ({PROJECTCODE})

If you have any questions, please contact your project manager.

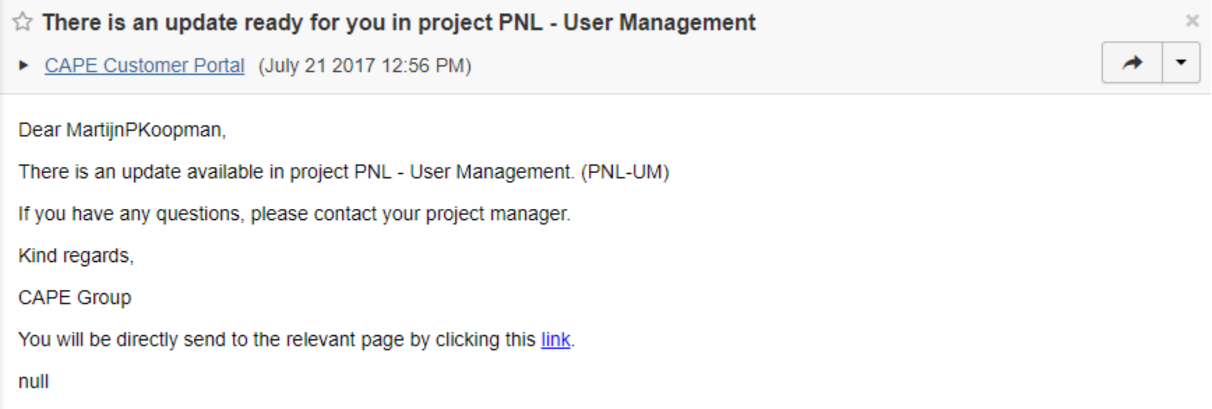
Kind regards,

CAPE Group

You will be directly send to the relevant page by clicking this link.

Figure 46: E-mail template

The microflow checks which project the mail must be about, which code is attached to that project, what the mail addresses are of the customers associated to this project and the URL to direct the customer to the relevant page.



There is an update ready for you in project PNL - User Management

CAPE Customer Portal (July 21 2017 12:56 PM)

Dear MartijnPKoopman,

There is an update available in project PNL - User Management. (PNL-UM)

If you have any questions, please contact your project manager.

Kind regards,

CAPE Group

You will be directly send to the relevant page by clicking this link.

null

Figure 47: Example of the e-mail

The mail is send from an external mail server. The e-mail goes from CP, through eMagiz and then to the mail server of company A. The e-mail goes through eMagiz, because Mendix is not able to talk directly with the mail server, whilst eMagiz is able to do that.

projects are shown. By creating a plain home-page, customers will be able to navigate easily to the section they are interested in.
All the user stories are prioritized by the groups that are presented in paragraph 4.3. After every story, the number of the corresponding group is given. So, story 3 corresponds to group 1, projects. This list is agile, that means that during the project, other stories can be added by the product owner or by myself. This list contains all the relevant things that are known currently, but this can change over the coming weeks.

Appendix D: Questionnaire responses

I look regularly at the updates that are currently send to me by company A	I think it's a good thing that company A is sending me updates concerning my project	The concept of self-service appeals to me	The portal is easy to use	The portal is visually attractive	I like the fact that the portal is accessible by phone as well	The portal creates a clear overview of the progress within my project	In the portal, I am able to find relevant information quickly	With the help of the portal, I am able to retrieve relevant information quicker than before	I like the fact that I can monitor the progress of my project in such detail	Do you have any improvements or recommendations for the portal?
Sometimes	Agree	Agree	Neutral	Neutral	Agree	Neutral	Neutral	Agree	Strongly agree	A link between the 'project' hours and the invoice will be useful.
Sometimes	Agree	Agree	Agree	Agree	Strongly Agree	Strongly Agree	Agree	Strongly agree	Strongly agree	A link to my invoice would be useful
Sometimes	Agree	Agree	Strongly Agree	Agree	Agree	Strongly Agree	Agree	Strongly agree	Strongly agree	Link to contracts and invoices would be useful