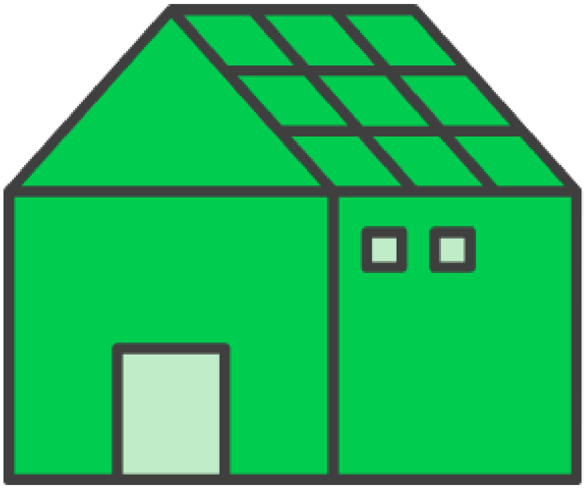


IMPLEMENTATION PLAN

Group 4

Anna Chernova, Izabella Bogdanova, Gijs Wijngaards, Tan Quang Nguyen, Yazan Fattal, Sjoerd Heijmann

17/01/2022

TABLE OF CONTENTS

1. INTRODUCTION

1.1. PURPOSE

This implementation plan's goal is to show a step-by-step list of tasks that will keep the Carleon project on track. Despite this, it will also supply the important stages and information that will be used to evaluate using the dashboard. (*Implementation plan* 2021)

1.2. SYSTEM OVERVIEW

The system overview will include both a description of the system and its additions. The dashboard will serve as our major technological solution for Carleon's project, displaying the entire system outcome.

1.2.1. SYSTEM DESCRIPTION

The dashboard is intended to highlight a better understanding of clients' accounts payable and current days sales outstanding as well as changes in these financial KPI's. The system will make use of the currently existing database that is available in the Exact globe next software. The system is designed to aid in visualizing and following up on open invoices and to streamline the monthly payment process.

A dashboard is a tool that allows to convey valuable data through a visualization housed in a platform; in our instance, that platform will be Power BI. Reports are used to create dashboard graphics, and each report is based on a dataset. The data comes from Exact globe software, which allows us to make a series of technological adjustments before displaying the results on a dashboard.

We can look at tools and documentation that will help us examine the system from an elevated level.

1.2.2. TOOLS

In terms of the tools, they will assist us in viewing what is required for us to use the system, referred to as the dashboard. As a result, under this approach, our tools will be assessed as software and materials that we will utilize to submit a solution that will help us complete our main objective of constructing the dashboard.

a. Software

The tools' software stage consists of a computer program that develop or support a program or application. As a result, we store the data for this project in a Microsoft SQL Server database. To access the data and acquire an overview of current trends, we initially used the software Exact. We next used SQL to import the data into R, where we used the tidyverse and ggplot2 packages to create illuminating graphics and validate our results. The final product will be a PowerBI dashboard with live data fetched from a database using SQL and edited with DAX. We keep track of our progress by storing it in Microsoft Teams, which gives us access to versioning.

As a result, our primary solution is placed on the *Power BI program*, with the rest of the applications, such as SQL, Exact, and RStudio, serving as supporting programs that supply us with a database or serve as a validation component for evaluating our work. Another group of applications, such as Microsoft Teams and WhatsApp, will be used as backup and deadline reminders. That is a complete collection of applications that will be used to create the dashboard.

b. Materials

Materials are the supplementary resources that will be used in the system evaluation. As an example, the materials for the system were taken from a case provided by Sique with access to Exact software and further database documentation, where we obtained an understanding of the data presented to us. As a result of that analysis, we were able to identify trends that we needed to include in our dashboard. Another set of sources are online resources, which were used in a wide range of ways, such as video lessons on Power Bi, supplementary articles on RStudio validation, and additional presentations on SQL queries, and so on. All these sources were used to support our data, dashboard creation, and validation research.

Finally, we will have a set of documentation for the system overview.

1.2.3. DOCUMENTS

We will provide a variety of documents that go under the category of system overview, all of which will aid in the solution's advice for our client and its testing to ensure that it runs smoothly.

Dashboard Documentation – is about the techniques we applied to create the dashboard. This document will be created at the start of implementing the dashboard.

Dashboard Test Report – is containing the validation of the results of the dashboard. The report will be made after the dashboard documentation.

Dashboard User Guide – is created to help the client to install and use the dashboard in its own company. This guide will be done at the end of the final dashboard.

After establishing the essential documentation for our system, we can move on to determining what potential hazards our system may face, as well as how we may monitor it.

2. RISK ASSESSMENT

Risks are a potential that may be categorized into a few key elements of this dashboard, such as time, database, visualization, and software. During the project, risk analysis is one of the most important parts. Therefore, risks which are identified in this project are categorized into a few key elements of this dashboard, such as time, database, visualization, and software.

The following are some of the risks that can arise because of dashboard development and use:

- **Timing risk** - explains the potential for missing beneficial acts. (*Timing risk* 2021).
  Example: Unexpected loss of time and tasks due to mistiming in the group.
- **Database risk** – shows that the data in the database could be missing or unverified, posing a risk to the database tidying process in the future.
  Example: Database observations in the graph were changing due to misunderstanding of how database working.

- **Visualization risk** – define the risks about unclear visuals, incorrect labels, etc.

  Example: Unclear visuals in the dashboard may result in the misconception of reading the graphs and getting the information right.
- **Software risk** - the possibility of a defective piece of code triggering an occasion that negatively influences the system or data.

  Example: This dashboard must be created with Power BI, therefore we did not study this language and started learning it this semester.

Table 1. Risk Assessment

Risks	**Probability**	**Impact on the dashboard**	**Monitor**	**Conclusion**
Timing	Medium	High	We employ the schedule program to track the progress of each deliverable, which includes the start and end dates as well as the name of the person in charge of it.	One of the team members sent daily notifications in "WhatsApp" group about the tasks and deadlines. In addition, we held 2-3 meetings per week, either in person or online, where we discussed the duties and assisted one another.
Database	Medium	High	If a code is incorrect, it will generate an automatic error in the power bi program, and if the code is correct, but there is a questionable outcome, a manual check will be performed to verify each measure that has been contributed through the code.	We began searching into the database early because it was large and required a lot of tidying; as a result, we discovered given SQL queries and used Power BI to modify them up for usage for our dashboard. Another method of risk management is validation in RStudio, which allows us to detect and prevent database issues as well as double-check data with our Power BI solution. For example, we experienced an issue where our database was updating without our involvement, but

				we obtained assistance and were able to fix this error.
Visualization	Low	High	Follow the standards for clear data visibility, approachable layout, and no crowdedness when it comes to data representations. In addition, situations and goals are clearly displayed. Personal re-check and accessible testing are also recommended to reduce the risk of occurrence.	We must consider the fact that there are users with disabilities who require suitable visualisation aid, such as appropriate colour use, correct label sizing, and straightforward navigating around the page. Then, in Power BI, there is a split with pages to prevent congestion with visuals. In addition, we can also include textboxes with explanations and create independent variables with text.
Software	Medium	High	Regarding PowerBI performance, pay attention to the supplied lessons, individual research, and practical. To avoid the risk, testing and quality checks will be performed afterward.	We learnt how to perform calculations in Power BI and R, to prevent software risk, we included the validation in RStudio, therefore it will be more practical to track the bugs or find mistakes.

The above table has led to the conclusion that risks are one of the possibilities that this development may encounter, and there are several of them that can also hurt the product. However, with the right response to our offered monitoring, we wish to prevent or address such delivery projections.

3. MANAGEMENT OVERVIEW

The measuring and identification approach to the tasks that will be established inside the dashboard development will be discussed in this section of the implementation plan.

3.1. ROLES

Our major solution for this case is a dashboard, hence a defined process for its development, measures, and layout as a fundamental constraint will be based on this delivery.

As a result, a role interpretation of the managing on dashboard development will be required to generate such an output. Each participant will be construed in the context of dashboard evaluation. It will include the role, the name of the person who will fill that role, and the contact information for the binder (see table below).

Table.2 Role distribution

Role	Name	Contact
Client	Raymond Kok	Raymond.kok@quercis.nl
Product owner	Marco	marco.hormes@fontys.nl
Data advisor	Gijs	g.wijngaards@student.fontys.nl
System developer	Quang	tan.nguyen@student.fontys.nl
Documentation specialists.	Anna, Izabella, Yazan	c.anet@student.fontys.nl, i.bogdanova@student.fontys.nl, y.alfattal@student.fontys.nl
Dashboard designer	Anna	-----
Testing consultants	All team	-----

Based on the table above, each project participant is assigned a dashboard development role, except for a stakeholder and a project manager, who will only be committed to a prototype form review and a definitive version grading of the generated product.

Each solid solution may contribute to the completion of major tasks, which are assessed depending on how the solution is implemented in a step-by-step manner.

3.2. MAJOR TASKS

This part will include a description of the activities that will be required for the definition of future steps of dashboard development including hardware, software, and databases, as well as data tidying and visualization.

The activities that will be used to create a dashboard are summarized in the following table:

Table 3. Tasks

Task	Will do
Review given data	Prior to the production of a main product, a review of available data is required to determine what is necessary of development and what is missing for us to address it.
Load relevant data	Loading the data into our software for future mutations after we locate it.
Transform loaded data	Checking data for missing values, evaluating extraneous columns, joining essential tables together.
Perform necessary calculations	Following the necessary cleaning, appropriate computations are performed to emphasize the information required by the client.
Validation of the results	Validation will oversee checking the data results against Exact software. That is, any calculations that are required because of using Exact will be included in PowerBI measurements.
Layout determination	Following the necessary calculations, a layout will be determined, which will aid in the development of a suitable concept for visualization.
Visualization	The dashboard's visualization will be used as a visual approach for a better presentation of the data.
Prototype	There will be a step of a prototype, which is a try out version for testing, once all the development and visualization has been completed.
Testing check	Testing will oversee running a manual check using a dashboard to see how it works and look for any flaws.
Quality check	Check final dashboard against previously made requirements
Resolve found issues	After testing and quality check, any found issues need to be fixed before finalizing the product definitively.
Final delivery	The dashboard's definitive version, which will be ready for submission.

Each significant task has an impact on the dashboard development process; each one contributes to the process. Most of the tasks are developed by the data advisor and system developer, but they are completed and finalized by the entire group.

With so many jobs, there is a need for approachable monitoring, which will be discussed in the following chapter.

4. PERFORMANCE MONITORING

This chapter contains an effective countermeasure to the monitoring and implementation schedule. Having properly created schedules and allocated tasks has an impact on system development.

4.1. MONITOR THE PROGRESS

It is critical to be able to monitor and evaluate the progress of tasks as well as to be aware that the system remains within its estimated scope. The monitoring aspects that will be taken in account are:

Table 4. Monitoring the progress

Method	Definition	Timing
Weekly personal meetings	To keep track of how each teammate is doing, what progress each of them has made on specified duties, and whether if it is necessary to contribute of require assistance.	Each meeting will require an average of three hours of group work (the timing of this can be expended or limited depending on the workload).
Review sections.	To make progress as a team we are giving reviews to each other after deliverables of the project to understand how we can improve our work in the future.	The review section is updated every six weeks based on provided results, and it includes a part with personal input from each group member.
Feedback from mentor and client.	Mentor gives us feedback about our progress, after that we follow up with a discussion about it. Then filling in the *FeedPulse* application to get an overview of the section.	Feedback is also provided by a mentor, who is assigned weekly, and by the client, who is assigned every seven weeks.
Often reminders via WhatsApp	The use of reminders benefits the team and their development deadlines. Having a reminder aid in jogging a team member's memory regarding their work.	A reminder done on everyday basis.

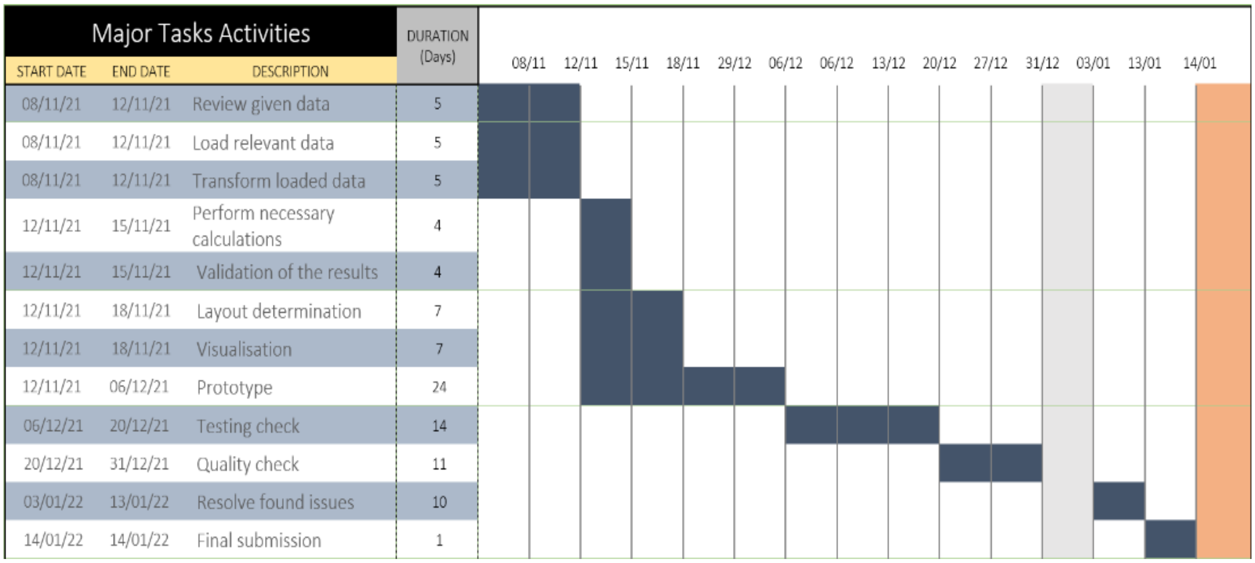
Weekly personal meetings, review sections, feedbacks, and reminders will be useful for the participants to work through the major tasks and in a way of a struggle or unmanageable issues will be found or discussed using these methods to monitor the major tasks and overall process of dashboard development.

The timetable is the next phase in our planning; in addition to proper monitoring, we can use the schedule to implement our primary tasks in the process, allowing us to keep track of their costs.

4.2. SCHEDULE

This section will outline the principal job schedule, with start and finish dates written in chronological order, followed by a work-breakdown structure that defines the phases and a timeframe.

4.2.1. GANTT CHART



Major Tasks Activities			DURATION (Days)
START DATE	END DATE	DESCRIPTION	
08/11/21	12/11/21	Review given data	5
08/11/21	12/11/21	Load relevant data	5
08/11/21	12/11/21	Transform loaded data	5
12/11/21	15/11/21	Perform necessary calculations	4
12/11/21	15/11/21	Validation of the results	4
12/11/21	18/11/21	Layout determination	7
12/11/21	18/11/21	Visualisation	7
12/11/21	06/12/21	Prototype	24
06/12/21	20/12/21	Testing check	14
20/12/21	31/12/21	Quality check	11
03/01/22	13/01/22	Resolve found issues	10
14/01/22	14/01/22	Final submission	1

Figure 1. Gantt chart

Gantt Chart is important in our implementation approach because it assists us in planning and scheduling (see figure 1). The participants have set up several activities to tackle the major tasks of each period. Each date that has been indicated represents our timeline with the duration of each task.

4.2.2. WORK BREAKDOWN STRUCTURE

The work breakdown structure is a hierarchical outline of the tasks that must be completed in order for our development to be finished. Breaking it down into smaller steps allows one or more team members to complete the task, resulting in superior results.

Each assignment is divided into three categories: business advice, dashboard, and implementation plan. After that, we divide down the work for these categories into smaller chunks to finish. Then we put in place the activities that will be used to fill in the gaps in these areas.

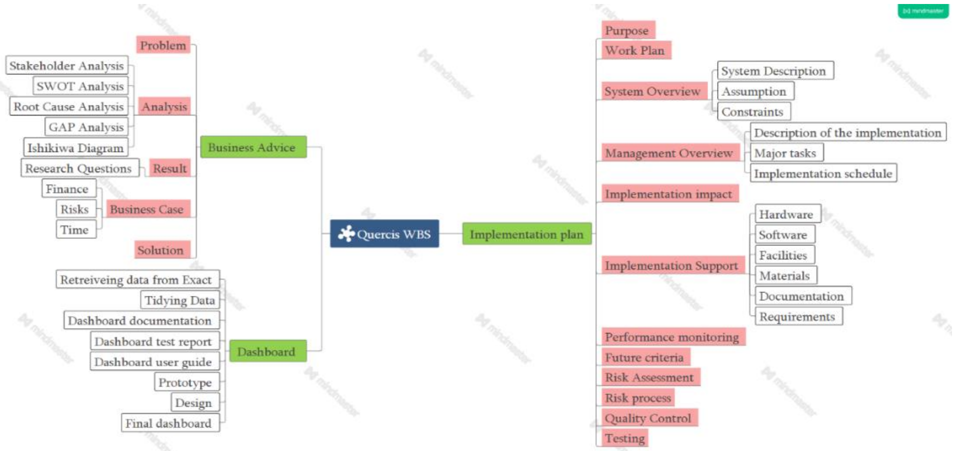


Figure 2. Work breakdown structure

4.2.3. TIMEFRAME

To summarize this chapter, we may assess the overall timeline for our growth, which will be 9 weeks in total. We can move on to the system's features when we've developed the primary tasks and their scheduling.

5. TECHNICAL SUPPORT

This chapter will give an overview of the technological aspects of our dashboard; we will then review the key performance indicators, requirements, use case diagram, and use case specifications to have a better understanding of the functional aspects of our solution.

5.1. KEY PERFORMANCE INDICATORS

Key performance indicator is a quantifiable measure of overall performance over time for a particular objective. KPIs offer goals for teams to shoot for, milestones to gauge progress, and insights that assist people from the company to make better decisions. (*What is a key performance indicator (KPI)? guide & examples*)

Table 5. Key performance indicator

Key performance indicator	Brief explanation	Importance	Period
Progression in total paid and unpaid invoices	Demonstrate the quantity of paid and unpaid invoices of customers	Keeping track of paid and unpaid invoices is a first sign that the company owns money from customers, allowing them to make any subsequent decisions (reminders for unpaid invoices, or any further analysis, etc...)	Per quarter, per month, per year.
Division of the amount of money that is still outstanding (by city)	This shows where in the customer pool the money is caught up	It is important to know which customers groups owe the most money, to	All time, per year, per month.

		know where to focus collecting energy	
Days sales outstanding	Represents the average number of days between sending an invoice to a customer and getting paid by the customer.	Measures the number of days Carleon has debt on its books, as well as the influence on the company's cash flow. Which are critically crucial aspects of any firm, particularly in terms of client collaboration.	Per month, per year.

These key performance indicators are a basic set of goals that the dashboard will present to give Carleon's financial status a clearer image. Getting closer to the impact that its implementation will have on the organization. After stating these, we can delve deeper into the functional and non-functional requirements that the system will meet for a client.

5.2. REQUIREMENTS

5.2.1. USER REQUIREMENTS

User requirements are a crucial part of our dashboard development since it allows us to understand what the client wants from the system on a high personal level.

ID 1 The financial director will be able to view the progress of current and future invoices

ID 2 The financial director will be able to observe the division of the amount that is still outstanding.

ID 3 The financial director will be able to view trends of DSO.

ID 4 The financial director will be able to move from page to page on a dashboard report.

ID 5 The financial director will be able to see the data tables loaded to PowerBI tool.

ID 6 The financial director will be able to observe the debt amounts from his customers.

ID 7 The financial director will be able to manipulate with filters on years and months of the dashboard.

ID 8 The financial director will be able to observe the top five cities including customers with highest debts.

ID 9 The financial director will be able to approach a decision regarding paid and unpaid invoices.

ID 10 The financial director will be able to decide regarding DSO trends.

ID 11 The financial director will be able to approach a decision regarding money that is still outstanding.

ID 12 The financial director will be able to interact with a dashboard at any time of the day.

ID 13 The financial director will be able to share the dashboard through shared link.

ID 14 The financial director will be able to access the dashboard through Power BI tool.

5.2.2. FUNCTIONAL

ID 15 The dashboard will be able to display the current and the future progress of invoices.

ID 16 The dashboard will be able to display the division of money that is still outstanding.

ID 17 The dashboard will be able to display the DSO.

ID 18 The dashboard will be able to get data from SQL Server.

ID 19 The dashboard will be able to update the data from SQL Server updated data there.

ID 20 The dashboard will be able to provide filters for displayed graphs.

ID 21 The dashboard will be able to provide multiple graphic formats.

ID 22 The dashboard will be interactive through the Power BI application.

5.2.3. NON-FUNCTIONAL

Our dashboard characteristics, such as usability, reliability, performance, and maintainability, are defined in stages for non-functional requirements.

Table 6. Non-Functional requirements

Factor	Example
Usability	**ID 23** The dashboard will be accessible through a shared link. **ID 24** The dashboard will have an appropriate color display for visual-disabilities personnel.
Reliability	**ID 25** The dashboard will be able to take up to 24 different subscriptions per dashboard. **ID 26** The dashboard file will request permission to access before entering it. **ID 27** On a dashboard, there will be filtering of years and months that can rely on all the visual formats per page. **ID 28** A dashboard will display the usage of change over-time values as part of the visuals. **ID 29** A dashboard will have a time-dependent display, meaning that the visuals will have a limitation period of dates for some of them.
Performance	**ID 30** Due to the large amount of data, a dashboard interaction will take 3 to 5 seconds to respond per filtering.

	ID 31 A dashboard will apply distinctive colors for the visuals.
Maintainability	**ID 32** A dashboard’s measures should be placed on a separate table. **ID 33** A dashboard will be non-stop available for access. **ID 34** A dashboard will be up to date with the data in the SQL Server.

Now that we've completed our dashboard's functionalities, we can move on to a more approachable evaluation between the client and our system, which will be based on a use case diagram.

5.3. USE CASE DIAGRAM

This diagram assesses the extent of the functions within our dashboard, as well as the client interactions within the dashboard. The flow is between the dashboard and the dashboard's actor, which can be either a client or a developer. It also shows the context and requirements from a dashboard, in addition to offering a scope of connections. (*Use-case diagrams, n.d.*)

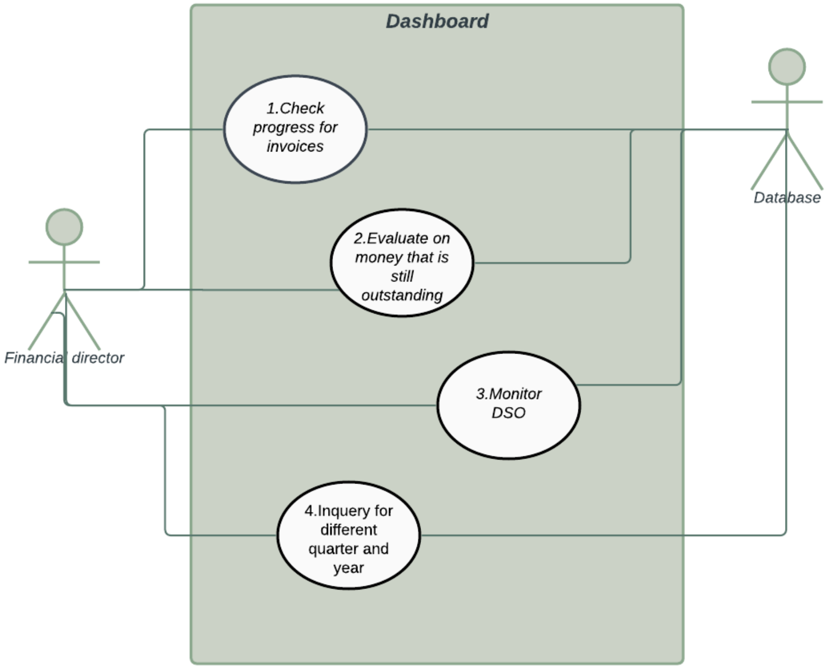

Dashboard
1.Check progress for invoices
2.Evaluate on money that is still outstanding
3.Monitor DSO
4.Inquery for different quarter and year
Financial director
Database

Figure 3. Use Case Diagram

We can see the interaction between our client and a dashboard in a simplified version in this diagram; a more detailed evaluation of it will be given in the use case specification.

5.4. USE CASE SPECIFICATION

More information than is presented in a use case diagram can be evaluated using the use case specifications. By focusing on the use cases, we can provide more information about presenting the criteria in a logical order and providing a clearer context for how a client interacts with a dashboard. (*Use case specification guideline - best tips & guidance for 2021* 2021).

Use case:	“Check progress for invoices”
ID:	1
Scope:	Dashboard
Summary:	Based on given data, the financial director shall evaluate the paid and unpaid invoice placed on a dashboard.
Primary actor:	Financial Director
Supporting Actor:	Database
Stakeholders:	The financial director wants his invoices to be paid on time.
Level:	User
Include:	----
Extend:	----
Precondition:	The company's finance director assigns data to the dashboard, including invoice data.
Trigger:	Paid and unpaid Invoices have been updated through SQL Server.
Normal Flow:	1. Invoices from customers are uploaded to the database. 2. The invoices are distributed to paid or unpaid. 3. Financial directors check the progress of the invoices. 4. Financial director made a conclusion from paid or unpaid invoices.
Sub Flow:	---
Alternate Flows/ Exceptions:	a) Invoices from customers are uploaded to the database. b) The invoices were distributed incorrectly. c) Financial director manually distributes the invoices. d) Financial director fails to make a conclusion from paid or unpaid invoices.
Post-condition:	The finance director came to a judgment about paid and unpaid invoices.

Use case:	“Evaluate on money that it’s still outstanding”
ID:	2
Scope:	Dashboard
Summary:	Amounts that are still owed have been allocated, and the finance director requests a review.
Primary actor:	Financial Director
Supporting Actor:	Database
Stakeholders:	The company's financial director wants to know which consumers owe the company the most money.
Level:	User
Include:	---
Extend:	---
Precondition:	The information about the clients whose money is still owed to them is gathered.
Trigger:	The data about money outstanding has been updated to the SQL Server.

Normal Flow:	1. The money has been entered to the dashboard. 2. The financial director displays the money that is still outstanding. 3. The financial director classifies outstanding money.
Sub Flow:	---
Alternate Flows/ Exceptions:	a) An incorrect amount of money has been entered to the dashboard. b) The financial director displays an incorrect graph of money that is still outstanding. c) The financial director manually checks with the invoices the amount of money. d) The financial director cannot determine the money which is till outstanding.
Post-condition:	The financial director has viewed the financial aspect of the money that is still outstanding.

Use case:	"Monitor DSO"
ID:	3
Scope:	Dashboard
Summary:	DSO data has been stored and there a need of future allocation of it by financial director to approach a critical decision regarding it.
Primary actor:	Financial Director
Supporting Actor:	Database
Stakeholders:	The financial director wants the DSO standings to be no more than 4 weeks.
Level:	User
Include:	---
Extend:	---
Precondition:	The financial director must view the report on the DSO standings and give the data about DSO.
Trigger:	The data information about DSO has been updated through SQL Server.
Normal Flow:	1. DSO data has been stored. 2. The financial director reveals the information about the DSO. 3. The financial director creates a report about the DSO of the company. 4. The financial director provides possible decisions for the future.
Sub Flow:	---
Alternate Flows/ Exceptions:	a) Stored DSO data is invalid. b) The graph of DSO information cannot be viewed. c) The financial director cannot create the report regarding DSO. d) The decisions cannot be provided for the future steps.
Post-condition:	The financial director viewed DSO standings and evaluated the decision regarding it.

Use case:	“Inquiry for different quarter and year”
ID:	4
Scope:	Dashboard
Summary:	To be able to analyze the process in more debts, financial director evaluates the dashboard by filtering out the quarters and years his interest.
Primary actor:	Financial Director
Supporting Actor:	Dashboard
Stakeholders:	The financial director wants inquiry more information about Carleon’s financial state through various years and months.
Level:	User
Include:	----
Extend:	----
Precondition:	The financial director filters out wanted quarters and years.
Trigger:	The dates have been updated through SQL Server.
Normal Flow:	1. The dashboard gives filtering for different years and months. 2. The financial director chooses specific dates. 3. The dashboard updating the graphs based on the date. 4. The financial director plans a report based on dates.
Sub Flow:	---
Alternate Flows/ Exceptions:	a) The dashboard provided invalid dates. b) The financial director chooses invalid date. c) The dashboard cannot update the graphs of the invalid date. d) The financial director cannot make a report based on dates.
Post-condition:	The financial director analyzed the filtered data on the dashboard that was given.

We may go on to testing our system after defining the technological aspects and client-system interaction. Testing is a vital aspect of our planning since without it, the system may crush and cause us and our client's constant discomfort, which we will aim to minimize through testing.

6. TESTING

As previously stated, testing is an important aspect of our dashboard implementation, and in this chapter, we will discuss the testing plan and its components, as well as a set of testing techniques that will assist us in determining the most appropriate one for our dashboard, and finally, we will discuss user adaptation strategy.

6.1. TEST APPROACH

6.1.1. TESTING

Before delving into testing, we must first comprehend what is being tested. As a result, testing is a method for guaranteeing that the actual software or function is defect-free and matches the intended requirements. Its purpose is to uncover flaws, faults, or unmet needs in the current system, in this case in the dashboard. (Hamilton, 2021)

As a result, we will be able to deduce a set of problems and errors in the system if we are able to undertake testing. Because of the flexibility of the procedure, we can save time and money for our client by detecting problems or defects early on. Another key part of testing is the client's satisfaction with the system's up-to-date functionality and lack of severe problems.

However, no documentation, research, or testing can take place without proper planning; this is where it will be organized into steps and stored for future evaluation.

6.1.2. TEST PLANNING

A test plan is a blueprint for all the testing that will be carried out and delivered. The focus of this type of documentation is on describing what to test, what not to test, how to test it, and other related topics. (*Test plan*)

Another part of test planning is the scope around which the planning revolves, which in our instance is *assessing the dashboard*. Additional criteria will be built up based on that scope and will resolve on it.

As a result, we will use a three-stage method for our testing: acceptance criteria, test strategy, and test techniques. We will begin by examining the acceptance criteria from the stages.

6.2. ACCEPTANCE CRITERIA

Before we can evaluate it, we must first define what it is. As a result, acceptance criteria are the specifications that a test object must follow to satisfy the client, in this case, the financial director. The acceptance criteria are defined per use case (see Figure 3), starting with a description of a test object that serves as a collection of requirements, followed by norms that evaluate the acceptability defects. (*Acceptance criteria: DevOps*)

Acceptance criteria must be a set of assertions with a clear pass/fail conclusion, which are measured using functional and non-functional requirements, as illustrated by a use case diagram.

Acceptance criteria for testing stage provide the boundaries for use cases, allowing the project owner to highlight the outcome, and driving the analysis, among other things. Another important aspect of acceptance criteria is error flaws. While no system or software can guarantee 100 percent correctness in its job, faults can be recognized to prevent them from occurring or at the at least to know how to deal with them.

However, just like with any project, ownership or review of the elements placed on is required, and the same is true for the acceptance criteria for this document.

6.2.1. ACCEPTANCE PERSON

The project owner evaluates on acceptance criteria to accept or reject the outputs given, as stated in the original abbreviation. (*Acceptance criteria: DevOps*) However, because of the personalization of the testing and its capacity, the entire group will be assigned for acceptance criteria. Finally, the client's acceptance or rejection of the criteria might have an impact on the testing and overall planning, thus its value is critical.

Table 7. Acceptance person

Name	**Function**
Raymond Kok	Financial Director
Marco Hermes	Product Owner
Anna Chernova	Documentation specialist / Designer
Izabella Bogdanova	Documentation specialist
Gijs Wijngaards	Data advisor
Tan Quang Nguyen	System developer

When it comes to meeting approval conditions or making a conclusion, the product owner gets the word, while the rest of the group members get cooperation or less evaluation conditions.

Despite fulfilling the acceptance personnel, the next step is to dive into the acceptance criteria.

6.2.2. ACCEPTANCE TABLE

The defect reporting table is divided into multiple fields where various details can be submitted to gather useful information. (*Performance testing*)

Our test objects will be assessed against the system's functional and non-functional requirements, and defects will be assessed accordingly.

Table 8. Acceptance table

Use Case	**Test object / use case description**	**Norm (number of accepted defects)**
1.Check progress for invoices	Assuring that the dashboard will be able to show both current and future invoice progress. The key aim for this use case is to test the data tidying, display, and functionality of the invoices information.	A=0 B=1 C=2 D=2
2.Evaluate on money that is still outstanding	Assuring that the dashboard will be able to show the amount of money that is still owed. The data output, visualization interpretation, and filtering functionality will be the focus of the testing.	A=0 B=1 C=2 D=3
3.Monitor DSO	In this use scenario, ensuring that the dashboard will be able to provide DSO performance. Its testing should concentrate on DSO dates, data updates, visualization, and date filtering.	A=0 B=1 C=3 D=2

4.Inquiry for different year and month	Assuring that the dashboard's filters can be displayed on years and months. As a result, testing will concentrate on the functionality of dates and their appearance on other visuals related to filtering.	A=0 B=0 C=2 D=2

Faults are discovered during the testing process, and those flaws are divided into severity classes. (*Acceptance criteria: DevOps*) The following of classes:

Severity A = Defect is a show stopper and will obstruct production

Severity B = Defect is severe, but can be solved with less cost and some rework

Severity C = Defect is disruptive, and can be solved with little costs (user-friendliness)

Severity D = Defect is cosmetic (wrong layout)

Following the identification of acceptance criteria, a test strategy is necessary to aid in future acceptance criteria testing and implementation.

6.3. TEST STRATEGY

A test strategy is just a guidance for testers on how to approach testing and what actions will be tested as a result of it. (Hamilton, 2021) The concept of it will rely on the test goals and product risks analysis which are the inputs of the test base.

For a test strategy, we also need to allocate the supervision, the same as the acceptance criteria, and in this example our supervision will rely on the product owner.

The relevance of test strategy is also vital to the testing process; first, owing to a time constraint, the entire dashboard will not be tested; therefore, we will be able to focus on more significant areas of the dashboard that provide a larger risk of harm or damage. In the following chapter, we will use a product risk analysis to assess such hazards.

6.3.1. PRODUCT RISK ANALYSE

The PRA aids the test in allocating limited resources. It all starts with the risks that come with the task at hand. We want to evaluate the areas where the risk is the highest in the system operation, which involves the appraisal of money that is still outstanding and DSO monitoring. The PRA aids in the identification and prediction of these threats. Acceptance criteria and product risks are inextricably linked. As a result, the criteria are critical to the product's success. (*Product risk analysis (PRA)*)

Before we get into the risks and damages, we need to assess the test objectives. The rationale for introducing them before the risks is that it will help us to make high-quality decisions and benefit the process from various perspectives with a clear aim that will then be evaluated based on the risks and damages.

6.3.2. TEST GOALS

Table 9. Test goals

Test goal	**Description**	**Characteristic**

1.Check progress for invoices	All invoices which were uploaded, are checked, and confirmed.	Suitability
2.Evaluate on money that is still outstanding	The evaluation on money provides information which money is still outstanding and what must be reviewed.	Performance
3.Monitor DSO	DSO monitoring provides a real-time snapshot of customer payments dates, allowing you to assess them and make decisions about them.	Performance
4.Inquiry for different quarter and year	The inquiry for different years and quarters is found easily and made quickly.	Functionality

Characteristics identification:

- *Functionality* how the system can perform or complete a task.
- *Performance* how the system processes a task.
- *Suitability* how the system qualifies and gives an output of the task.

The risks must be assessed after the test goals have been defined. Its goal is to determine which needs are critical and which are not, depending on the test goals listed above (see Table 9). Despite this, risks must be assessed because they are concerned with the test effort and its relationship to failure to get the anticipated results.

6.3.3. RISK TABLE

Table 10. Risk Table

Test goals	Characteristic	Risk	Damage	RC	Argumentation
1.Check progress for invoices	Suitability	2	2	4	The failure in the checking progress of the invoice may result in loss of time, but the chances are small.
2.Evaluate on money that is still outstanding	Performance	5	5	25	Wrong evaluation on money that is still outstanding can lead to loss of unnecessary money and time.
3.Monitor DSO	Performance	4	5	20	The false decision of the DSO can lead the company to loss of money, customers and suppliers.

4.Inquiry for different quarter and year	Functionality	2	2	4	An incorrect inquiry of the year or month may result in redevelopment of the report or incorrect finding.

The risk and damage levels scale:

- Low: 1
- Slightly low: 2
- Moderate: 3
- Slightly high: 4
- High: 5

Besides risks and damage scaling, there is an RC, which is a meeting point of damage and failure which is also measured from 1 to 5 scale. The formula to determine the risk class:

Risk Class = Risk * Damage

We can draw conclusions about the results after evaluating the table above (see Table 10).

6.3.4. CONCLUSION

As a result, from risk table, the use cases “Evaluate on money that is still outstanding” and “Monitor DSO” are the highest, because analysis showed that both have risk class 20 and 25, comparing with other two use cases those risks are quite high. Therefore, these two use cases are critical in the company performance, because with bad evaluation on money can lead to a huge loss of money and time, then if company made an incorrect decision about DSO, it also would lead the company to huge loss of money and consumers.

We may contribute to the correct test approaches to provide proof and practical testing for our dashboard after assessing the results for dangers.

6.4. TEST TECHNIQUE

Test approaches are based on how functional and non-functional criteria can be used to test an application. (Hamilton, 2021) We will provide testing for our dashboard in our situation. The use of test procedures helps us to identify a variety of problems that can be applied to a dashboard and its operation.

Path testing, for example, is one of the techniques that we will augment in this document:

6.5. PATH TESTING

By allocating the dashboard behavior, path testing assists with decision points and path. The path testing is carried out using a flow chart that goes from start to finish and depicts the various paths that can be taken from decision points denoted by numbers. It also aspires to define each possible path. (*Paths,(n.d.)*)

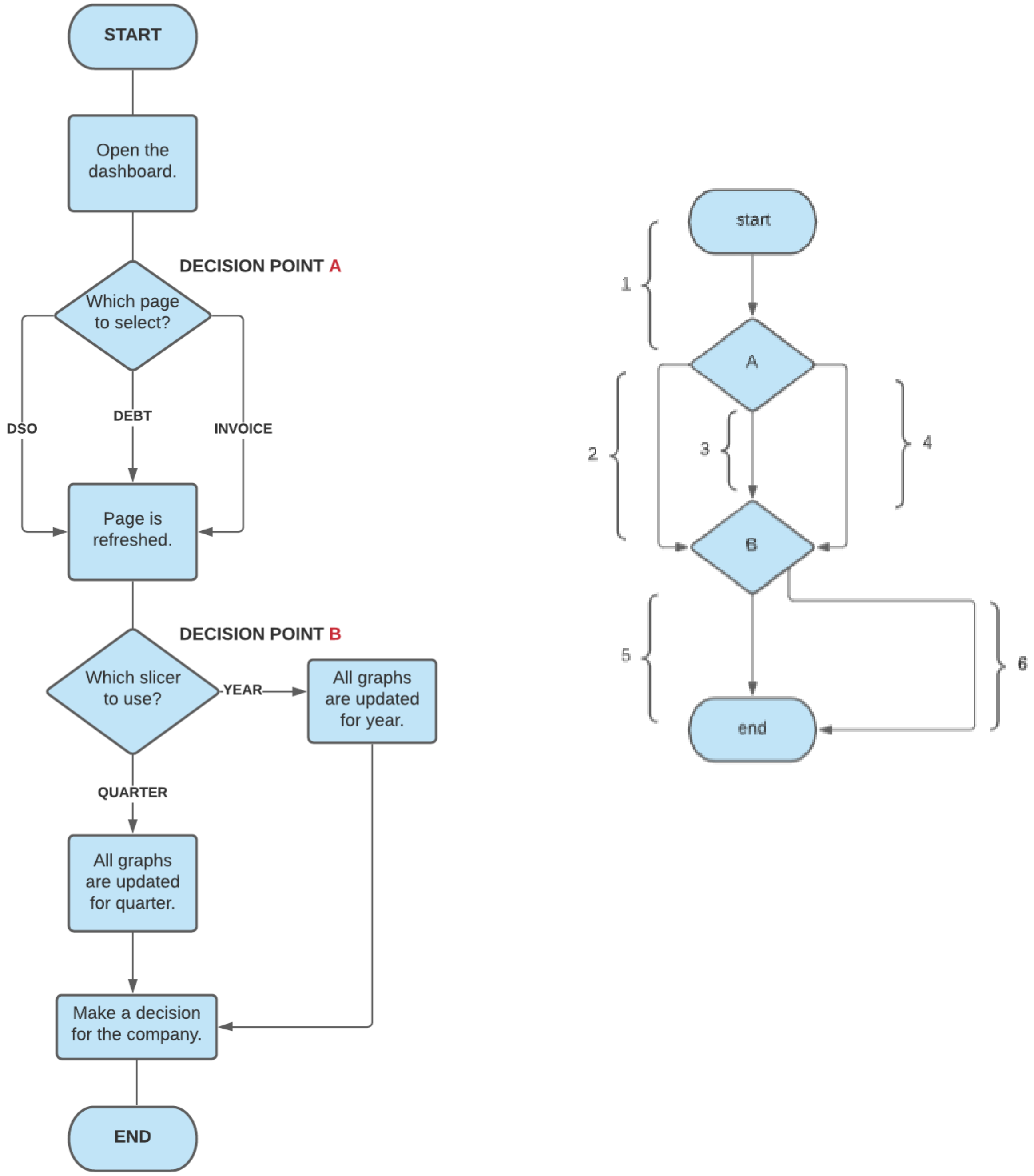
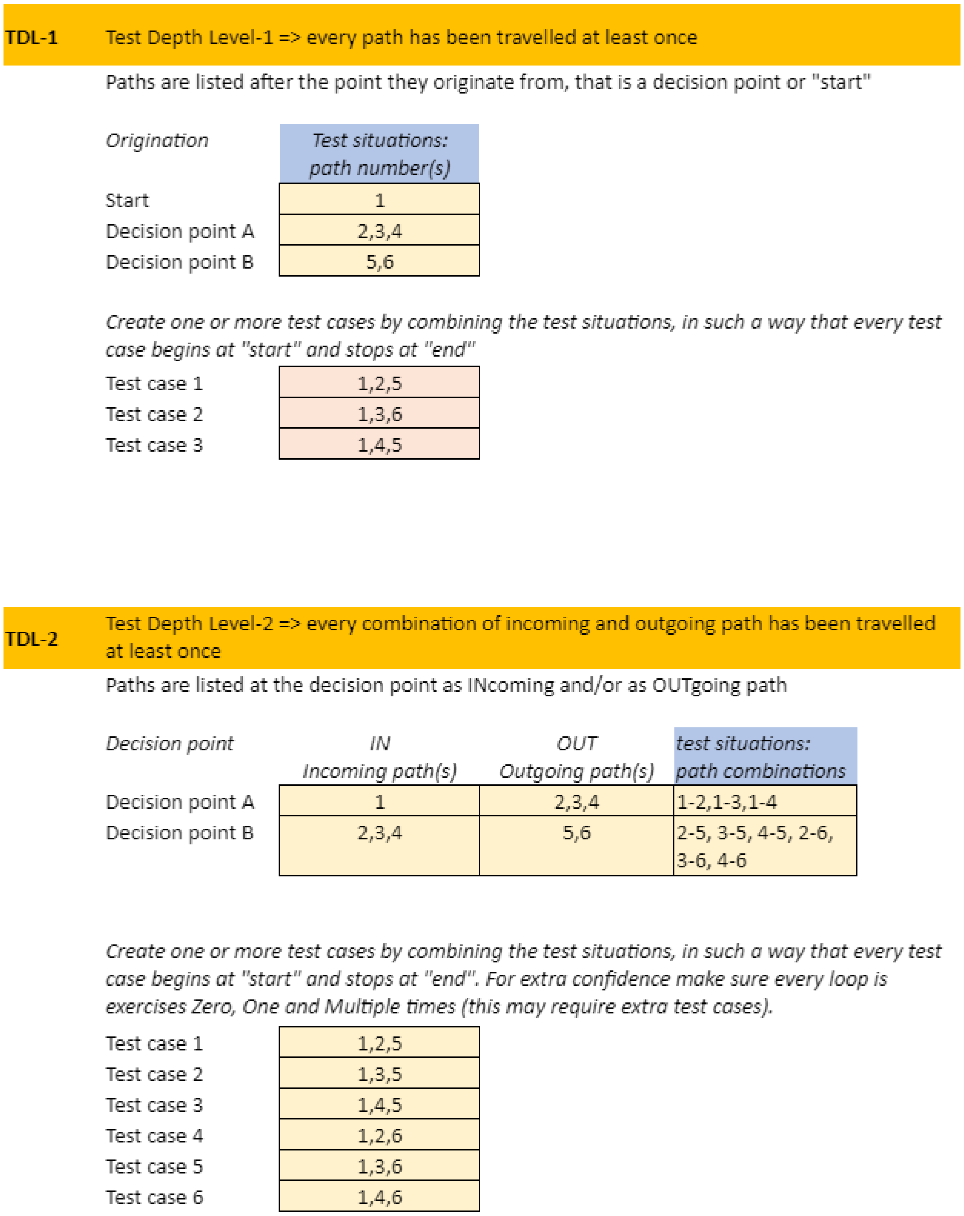


Figure 4. Path testing

To be able to assess the above-mentioned paths (see Figure 4), we will use a supporting table with test depth levels (see Figure 5).



TDL-1 Test Depth Level-1 => every path has been travelled at least once

Paths are listed after the point they originate from, that is a decision point or "start"

Origination	*Test situations: path number(s)*
Start	1
Decision point A	2,3,4
Decision point B	5,6

Create one or more test cases by combining the test situations, in such a way that every test case begins at "start" and stops at "end"

Test case 1	1,2,5
Test case 2	1,3,6
Test case 3	1,4,5

TDL-2 Test Depth Level-2 => every combination of incoming and outgoing path has been travelled at least once

Paths are listed at the decision point as INcoming and/or as OUTgoing path

Decision point	*IN Incoming path(s)*	*OUT Outgoing path(s)*	*test situations: path combinations*
Decision point A	1	2,3,4	1-2,1-3,1-4
Decision point B	2,3,4	5,6	2-5, 3-5, 4-5, 2-6, 3-6, 4-6

Create one or more test cases by combining the test situations, in such a way that every test case begins at "start" and stops at "end". For extra confidence make sure every loop is exercises Zero, One and Multiple times (this may require extra test cases).

Test case 1	1,2,5
Test case 2	1,3,5
Test case 3	1,4,5
Test case 4	1,2,6
Test case 5	1,3,6
Test case 6	1,4,6

Figure 5. Excel Path testing

The dashboard can take a variety of paths based on the path testing results, and the test depth levels aided in the knowledge of each path's traveling behaviour. Even though the dashboard is used to evaluate the path, it has its own impact on the technology and personnel factors.

7. IMPLEMENTATION IMPACT

This section discusses the dashboard's implementation impact, or how it will affect Carleon's organization in terms of technology and employees.

7.1. TECHNOLOGICAL IMPACT

The impact that will follow from the implementation of the dashboard will rely on a company’s technological evaluation on aspects like g*etting more value out of presented data, consolidate and automate multiple data points, performance of more accurate business analysis.*

An ability to evaluate in terms of the technology component of a dashboard implementation is the process of extracting additional value from supplied data. That data taken from Carleon's local database will have a set of clear goals and strong context included through the visualization and transformation on interactive dashboard, and that technically, it will be a way of representing given data and the possibility of getting a better awareness of values within.

Numerous data points will be consolidated and automated, allowing the dashboard to display multiple metrics in the same interface. This will have an impact on research time, sharing results within Carleon, effort required for issue areas, and so on. Such technical advantages are beneficial to a large corporation such as Carleon and its financial predicaments.

Impacts include more accurate business analysis, which means the process will be useful in implementing and managing changes based on data from a dashboard. The problems that will be revealed may be a significant help when it comes to this strategy; they can lead to a development of identifying a probable cause that can be related to customers or any other component, as well as a re-building of the organization based on Carleon's decision.

7.2. PERSONNEL IMPACT

Like the technological impact, there will be an implementation that will rely on company workers who will approach the dashboard from a given perspective. If we are talking about product influence, the evaluation will be based on a variety of possibilities, *such as aligning teams and departments, becoming more aware of the financial structure, and the ability to make meaningful decisions about the compan*y.

All the above-mentioned impacts of the dashboard implementation are dependent on how the dashboard can change the overview on the company. Aligning teams and departments can allow multiple departments of the company to have an objective view of current performance and serve as a prevalent group discussion and idea sharing concerning the company's financial state. This type of discussion can go not only for the financial department, but also to customer service and other areas of the organization, shedding additional light on what is going on and how it might be remedied in the future.

Another effect of the integration will be a greater understanding of the company's financial structure. Carleon's employees will be able to evaluate where they stand within the firm based on the results that will be displayed on a supplied dashboard, as well as what changes may be done in areas such as reminders and so on. Employees that are aware of the company's financial position have a better understanding of what steps to take in the future.

Finally, the ability to make important business decisions is a major possibility for a large corporation like Carleon. Having the chance to develop a decision regarding their customer base for reminders or imposing restrictions on daily sales outstanding are examples of potential evaluations that Carleon may face in their work and being able to decide whether to accept a new custom reminder or switch their policy to DSO can come from a data analysis.

8. FUTURE CRITERIA

Here, we will assess the dashboard's administration features as well as our client's future directions. The management coverage is significant because it provides knowledge that will allow our customer to utilize the dashboard without our assistance in the future, and future instructions are a collection of recommendations that may be followed to complete the system's task.

8.1. MANAGEMENT OF THE DASHBOARD

One of the first features of this dashboard is its interactivity, allowing Carleon employees to continuously change the dashboard's filtering results. Such a feature will be important for the dashboard's management component, as being able to track changes within opens control opportunities.

Another feature is an up-to-date approach, which means that the dashboard will display up to date taken information. Anytime, when necessary, the dashboard can be refreshed with up-to-date information from the database. During development there was limited access to this database, so afterwards the data will need to be refreshed to show actual representative graphics and numbers. Because the data is easily refreshed, a financial employee can view relevant information at any time required.

If we look at appendix one and two in more details, we can see that there is additional examination of the criteria and support that can be followed by using the dashboard. But the only changes that possibly need to be made in the future could result from a change of database structure of the switch between complete systems. In this case, a Power BI expert could integrate these new back-end solutions in this same dashboard, by linking the new variables to the DirectQuery tables.

9. GLOSSARY

KPI – Key performance indicator is a quantifiable measure of overall performance over time for a particular objective.

WBS – Work breakdown structure is a hierarchical outline of the tasks required to complete a project.

SQL – Structured Query Language. SQL is used to communicate with a database.

DAX – Data Analysis Expressions. Is library of functions which are used in Power BI.

ID – Identity document. It is numeral that is used for identification.

DSO – Days sales outstanding is a measure of the average number of days that it takes a company to collect payment for a sale. The DSO is calculated with the following formula:

$$\frac{AccountReceivable}{CreditsSales} * number\ of\ Days$$

PRA – Product Risk Analysis. It aims to identify high risks.

10. CITATION

Table 1. Risk Assessment. It shows the risks analysis based on the dashboard.

Table 2. Role distribution. It displays the position distribution based on the team members.

Table 3. Tasks. This table shows the major tasks for the implementing the dashboard.

Table 4. Monitoring the progress. It shows the monitoring of how we will keep track of deadlines and progress within the dashboard.

Figure 1. Gantt chart. Explains the schedule of primary tasks.

Figure 2. Work breakdown structure*.* It shows the outline of the tasks required to complete a project.

Figure 3. Use Case. It displays the use case diagram of the dashboard.

Figure 4. Path testing. It shows the path testing for the dashboard and distribution of the paths.

Figure 5. Excel Path testing. Show the analysis of the path testing.

Table 5. Key performance indicator. It shows KPI's which will be implemented in the dashboard.

Table 6. Non-Functional requirements. It shows the dashboard's characteristics, such as usability, reliability, performance, and maintainability.

Table 7. Acceptance person. It shows who is responsible for accepting the criteria.

Table 8. Acceptance table. It displays the use cases with number of accepted defects.

Table 9. Test goals. It describes the test goals and their characteristics.

Table 10. Risk table. This table describes the risk class of the test goals.

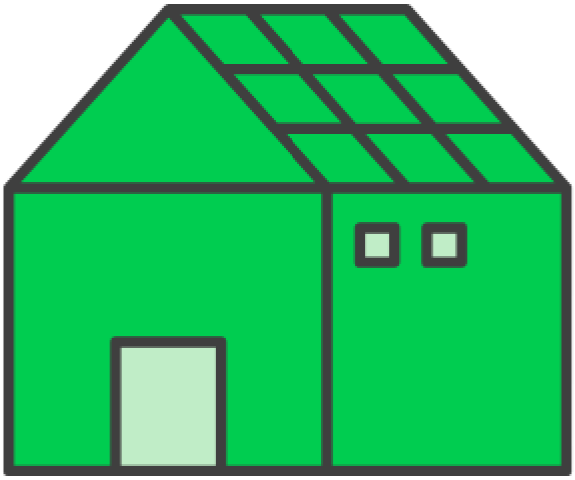
11. REFERENCES

- Smith, B. (2021, August 17). *What is an interactive dashboard? features & how to use them*. iDashboards. Retrieved November 18, 2021, from https://www.idashboards.com/blog/2016/12/19/4-ways-to-use-an-interactive-dashboard/.
- *What is a dashboard? how adjust displays your data*. Adjust. (n.d.). Retrieved November 18, 2021, from https://www.adjust.com/glossary/dashboard/.
- *6 pitfalls of marketing dashboards and how to leap over them*. Inbound marketing for B2B technology companies. (n.d.). Retrieved November 18, 2021, from https://www.articulatemarketing.com/blog/pitfalls-of-marketing-dashboards.
- *8 examples of maintainability*. Simplicable. (n.d.). Retrieved November 18, 2021, from https://simplicable.com/new/maintainability.
- *How to monitor a project (with tips)*. Indeed Career Guide. (n.d.). Retrieved November 18, 2021, from https://www.indeed.com/career-advice/career-development/monitoring-a-project.
- Author Dean Leffingwell -, -, D. L., & Leffingwell, D. (2021, August 21). *Nonfunctional requirements*. Scaled Agile Framework. Retrieved November 18, 2021, from https://www.scaledagileframework.com/nonfunctional-requirements/.
- Hayward, E. (2021, June 9). *The Starter Guide to dashboards*. Klipfolio.com. Retrieved November 18, 2021, from https://www.klipfolio.com/blog/starter-guide-to-dashboards#WhyDashboard.
- *What is the role of Dashboards in Business Intelligence*. ClicData. (2021, March 12). Retrieved November 18, 2021, from https://www.clicdata.com/blog/dashboards-useful-tools-business-organizations/.
- *Why is data analysis important in business? [video]*. GetSmarter Blog. (2021, September 8). Retrieved November 18, 2021, from https://www.getsmarter.com/blog/career-advice/data-analysis-important-business/.
- *Is there a limit to the number of people on a subscription distribution?* Home - Microsoft Power BI Community. (2020, July 8). Retrieved November 19, 2021, from https://community.powerbi.com/t5/Service/Is-there-a-limit-to-the-number-of-people-on-a-subscription/m-p/1204459.
- Chen, J. (2021, September 21). *Timing risk*. Investopedia. Retrieved November 19, 2021, from https://www.investopedia.com/terms/t/timingrisk.asp#:~:text=Timing%20risk%20is%20the%20speculation,based%20on%20future%20price%20predictions.&text=This%20could%20cause%20harm%20to,high%20or%20selling%20too%20low.
- *The benefits and limitations of a business intelligence dashboard*. Unleashed Software. (2021, April 20). Retrieved November 19, 2021, from https://www.unleashedsoftware.com/blog/benefits-limitations-business-intelligence-dashboard.
- *Implementation plan*. Colorado Department of Transportation. (2018, November 7). Retrieved November 19, 2021, from https://www.codot.gov/business/process-improvement/self-service/tools/solve-guide/part4/implementation-plan.
- *What is a key performance indicator (KPI)? guide & examples*. Qlik. (n.d.). Retrieved November 19, 2021, from

https://www.qlik.com/us/kpi#:~:text=KPI%20stands%20for%20key%20performance,the%20organization%20make%20better%20decisions.

- nitesh.rns1@gmail.com. (2021, May 22). *What is test approach? different techniques and types*. Tutorials Hut. Retrieved November 19, 2021, from https://tutorialshut.com/test-approaches/.
- *The main types of software testing methodologies and testing based on business objectives*. SENLA Company. (n.d.). Retrieved November 19, 2021, from https://senlainc.com/blog/the-main-types-of-software-testing-methodologies-and-testing-based-on-business-objectives/.
- Vashishtha, S. K. (n.d.). *Agile Testing: Break Silos and build quality in from the start*. Xray. Retrieved November 19, 2021, from https://www.getxray.app/blog/agile-testing-build-quality-in-from-the-start?utm_term=&utm_campaign=DSA%2BSearch%2B-%2BXray%2BWW%2B-%2BBlog%2BPosts&utm_source=adwords&utm_medium=ppc&hsa_acc=9970092548&hsa_cam=13952027914&hsa_grp=121805042541&hsa_ad=535121461807&hsa_src=g&hsa_tgt=dsa-1392437111908&hsa_kw=&hsa_mt=b&hsa_net=adwords&hsa_ver=3&gclid=EAIaIQobChMIsvPIs8ak9AIVkeJ3Ch2H5wQLEAAYAiAAEgKc3_D_BwE.
- Chen, J. (2021, September 21). *Timing risk*. Investopedia. Retrieved November 19, 2021, from https://www.investopedia.com/terms/t/timingrisk.asp#:~:text=Timing%20risk%20is%20the%20speculation,based%20on%20future%20price%20predictions.&text=This%20could%20cause%20harm%20to,high%20or%20selling%20too%20low.
- Contributor, C. (2020, December 16). *Product adaptation strategy*. Small Business - Chron.com. Retrieved November 19, 2021, from https://smallbusiness.chron.com/product-adaptation-strategy-62189.html.
- *Use-case diagrams*. in UML modeling. (n.d.). Retrieved November 20, 2021, from https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case.
- Nicholas, J. (2021, October 13). *Use case specification guideline - best tips & guidance for 2021*. BusinessAnalystMentor.com. Retrieved November 20, 2021, from https://businessanalystmentor.com/use-case-specification-guidelines/.
- Hamilton, T. (2021, December 11). *What is software testing? definition, basics & types in software engineering*. Guru99. Retrieved January 17, 2022, from https://www.guru99.com/software-testing-introduction-importance.html
- *Test plan*. TMap. (n.d.). Retrieved January 17, 2022, from https://www.tmap.net/building-blocks/test-plan
- *Acceptance criteria: DevOps*. TMap. (n.d.). Retrieved January 17, 2022, from https://www.tmap.net/building-blocks/Acceptance-criteria-DevOps
- *Performance testing*. TMap. (n.d.). Retrieved January 17, 2022, from https://www.tmap.net/node/14
- Hamilton, T. (2021, December 18). *How to create Test Strategy Document (sample template)*. Guru99. Retrieved January 17, 2022, from https://www.guru99.com/how-to-create-test-strategy-document.html
- *Product risk analysis (PRA)*. TMap. (n.d.). Retrieved January 17, 2022, from https://www.tmap.net/building-blocks/product-risk-analysis-pra *Paths*. TMap. (n.d.). Retrieved January 9, 2022, from https://www.tmap.net/wiki/paths

12. APPENDIX



User guide

Carleon case – Dashboard

Group 4

Anna Chernova, Izabella Bogdanova, Yazan Fattal, Gijs Wijngaards, Sjoerd Heijmann, Tan Quang Nguyen

17/01/2022

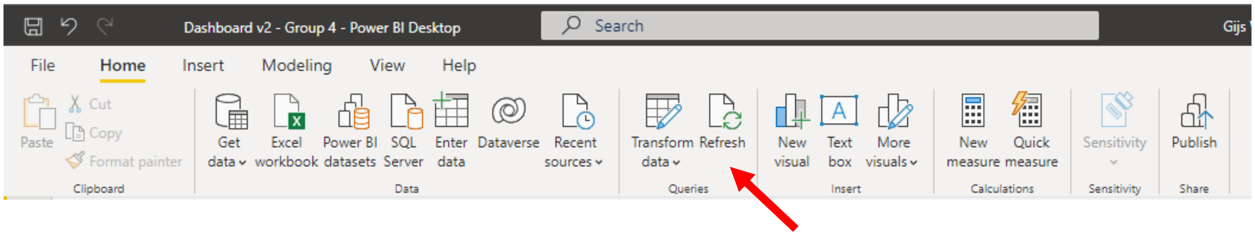
Table of Contents

1. Deployment

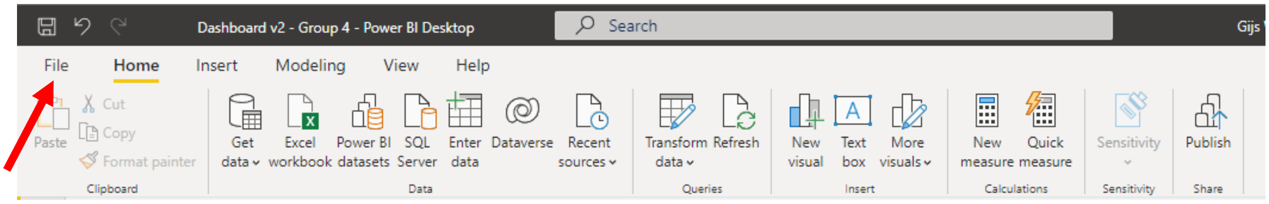
To deploy the dashboard, it is important to follow the steps below. Make sure the following prerequisites are met:

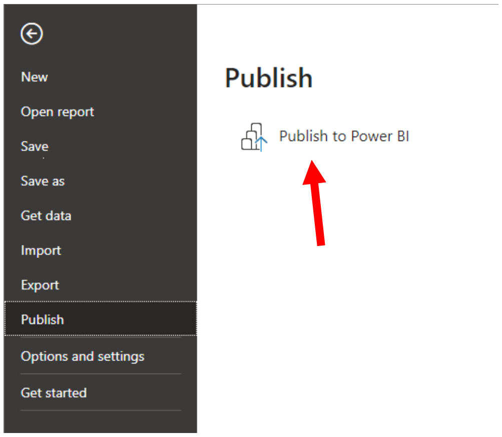
- Power BI Desktop is installed
- You have access to the database

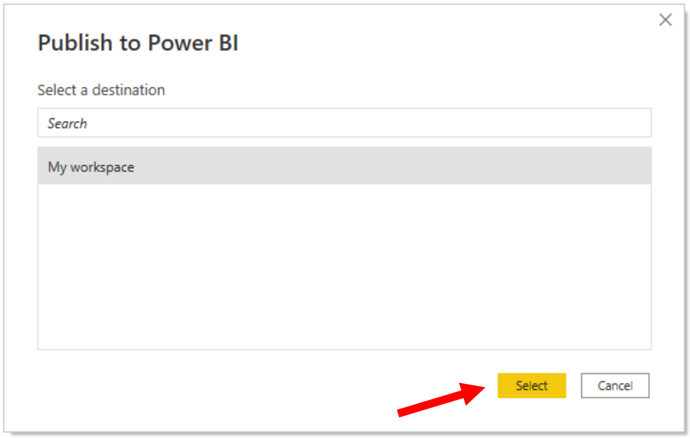
First the data used in the dashboard needs to be updated. This can easily be done by opening the dashboard in '*Power BI Desktop*' and clicking '*Refresh*', which should be in the *'Home'* tab under *'Queries'*.



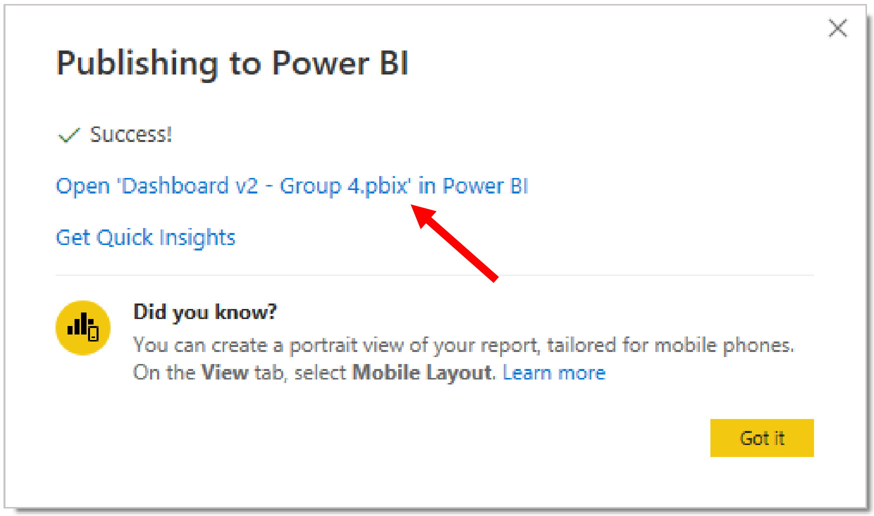
After the data is successfully updated, the dashboard needs to be published to make the update accessible. To do this, click *'File'* followed by *'Publish'*→ *'Publish to Power BI'*. This will show the option to choose what workspace you want the dashboard to be published in. Select the destination and click *'Select'*.







After you click select, Power BI Desktop will start publishing the refreshed dashboard. After it is done, you will see a popup similar to this one:



Publishing to Power BI

✓ Success!

Open 'Dashboard v2 - Group 4.pbix' in Power BI

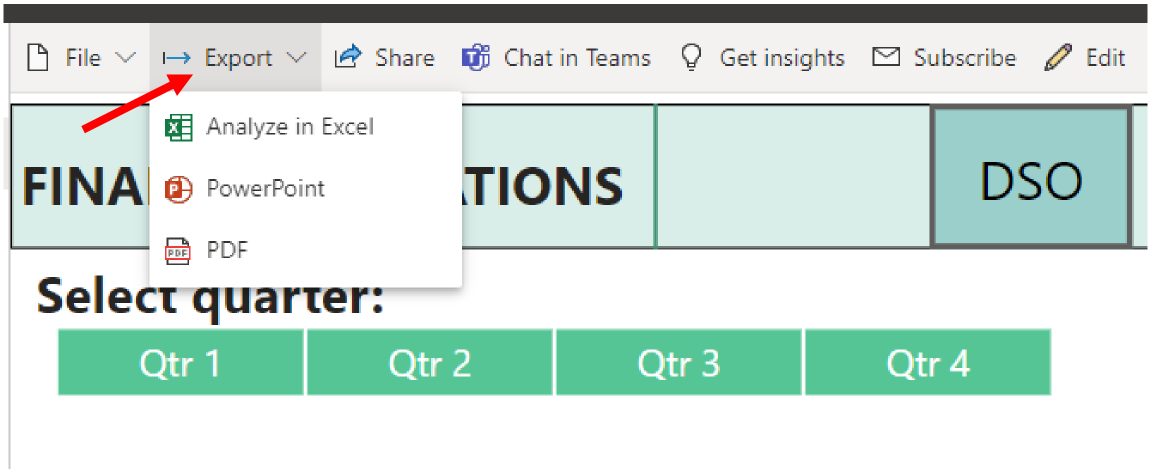
Get Quick Insights

Did you know?

You can create a portrait view of your report, tailored for mobile phones. On the **View** tab, select **Mobile Layout**. Learn more

Got it

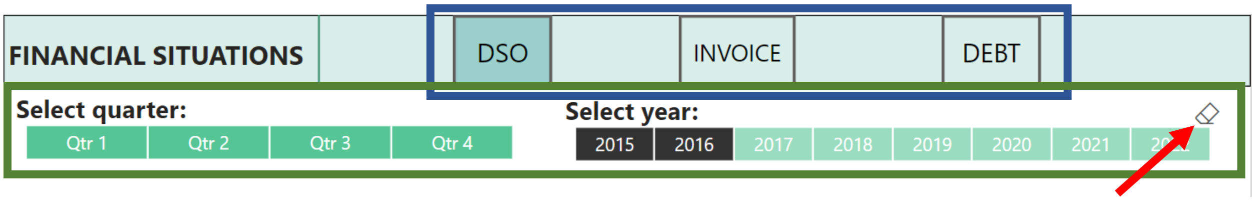
Clicking *'Open dashboard in Power BI'* you can open the new dashboard in Power BI, where you can easily send the dashboard to the recipients by sharing the dashboard through *'Share'* or *'Export'* to *'Analyze in Excel'*, *'PowerPoint'*, or *'PDF'*.

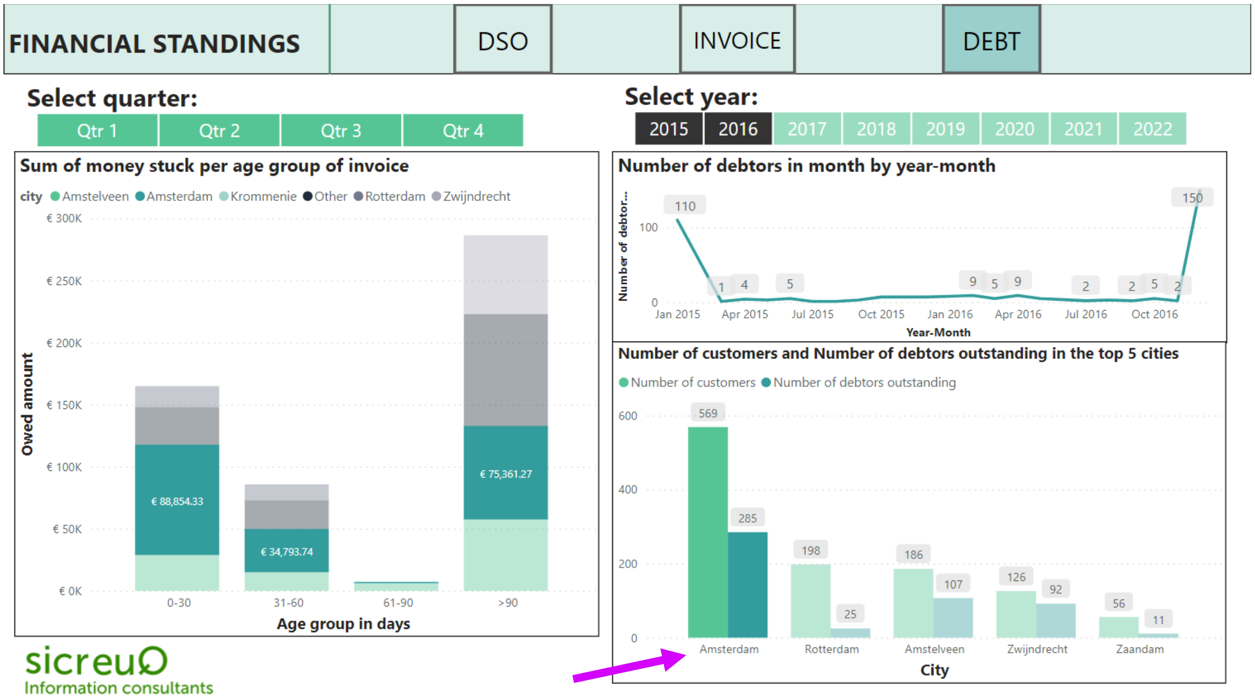


2. Usage

2.1. Dashboard basics

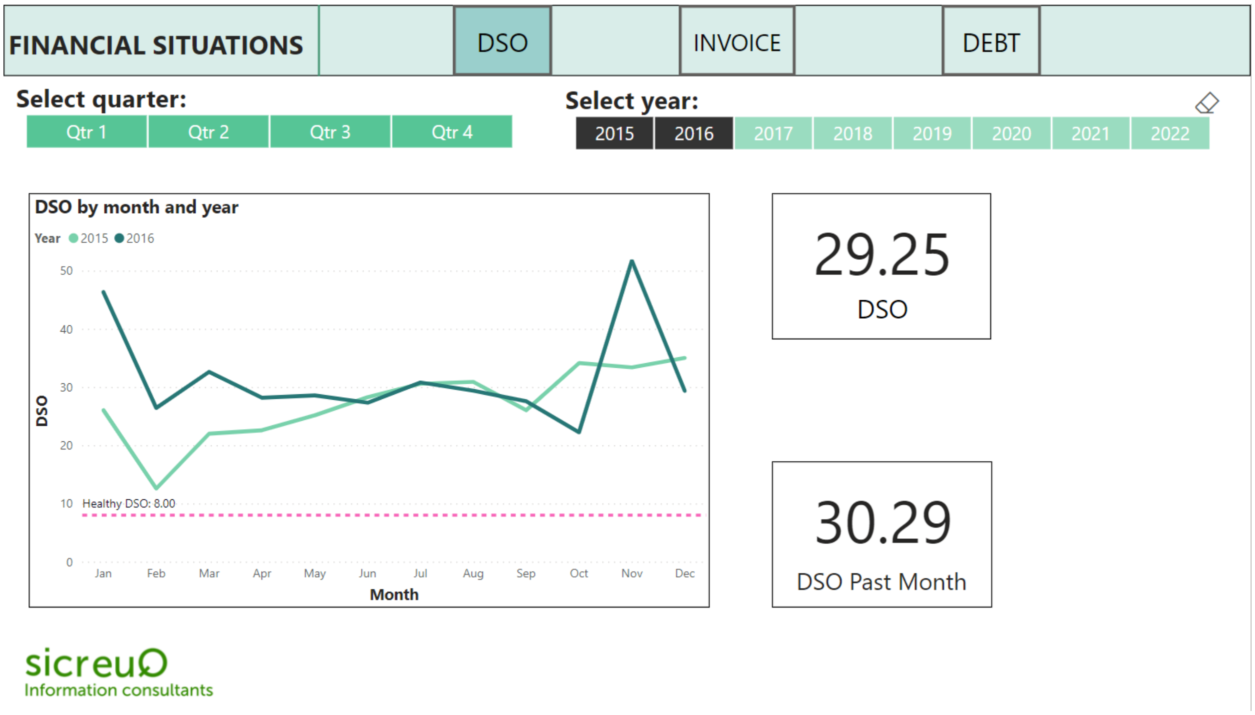
First, the basic dashboard controls are discussed that are available in all pages. At the top of the dashboard, you are able to navigate to the 3 pages of the dashboard; *'DSO'*, *'INVOICE'* and *'DEBT'* (highlighted in blue below). Each page gives the option to filter data per quarter and year with the buttons highlighted in green. Multiple years and quarters can be selected in any combination by clicking on them. If you want to reset the selection, you can click the eraser that shows when you hover over a selector (highlighted with red arrow). Within each page, you can also filter out information by clicking on a certain value. As an example, in the *'DEBT'* page below "Amsterdam" is clicked where the purple arrow is pointing. As you can see, this filters all other information on the page to highlight the data about Amsterdam.





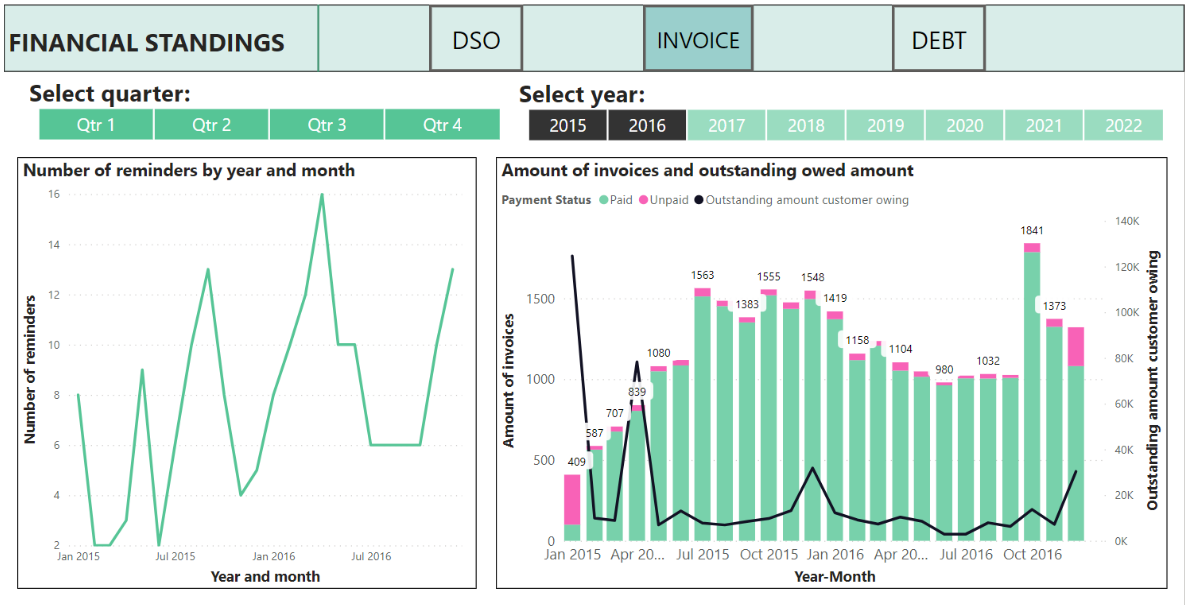
2.2. DSO page

When you open the dashboard, you will be greeted by the most valuable page, *'DSO'*. Here you can immediately see the DSO trend lines per year on the left, compared to a healthy DSO of 8 days. The top *'DSO'* value is the DSO for the selected time period. The *'DSO Past Month'* shows the DSO for the last 30 days.



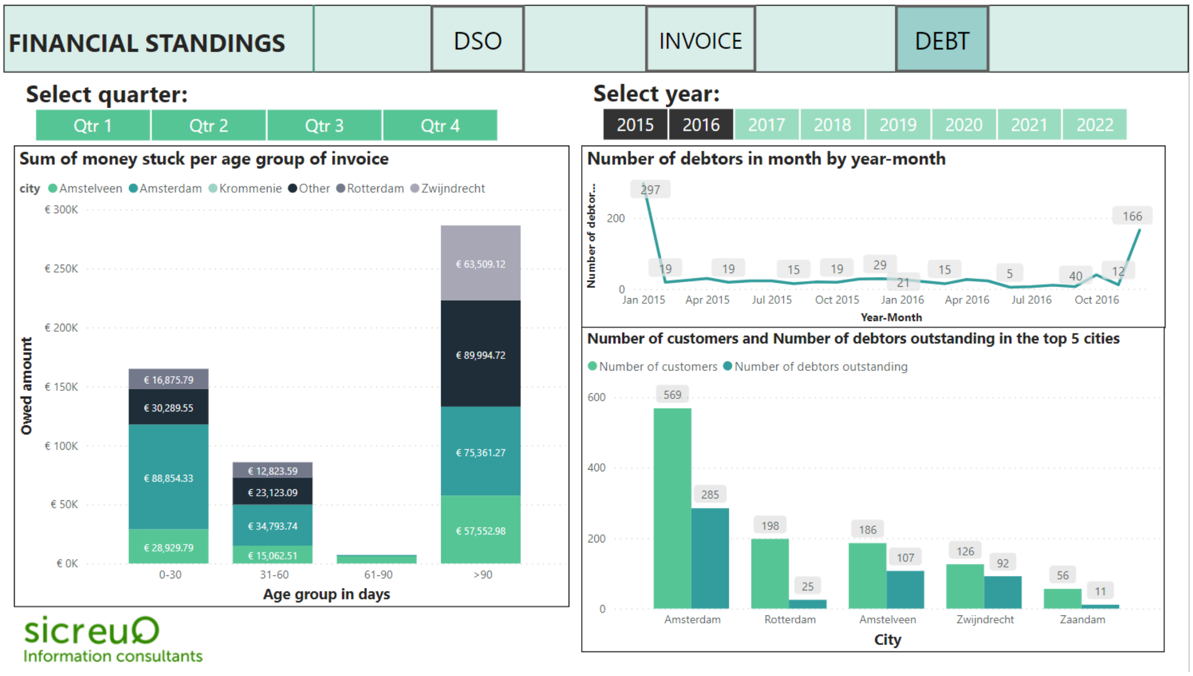
2.3. Invoice page

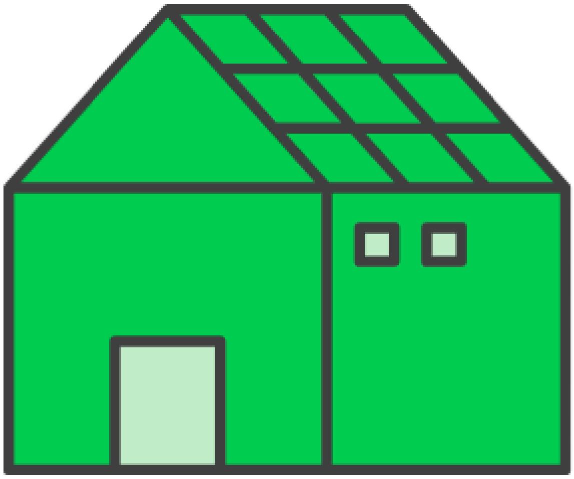
The next page is the *‘INVOICE’* page. This page shows the trends in all the invoices. The left graph shows how many reminders have been send to late-paying customers. The right graph shows 2 things: The black line shows how much money is owed based on each month. The bars show the number of paid invoices in green and the number of unpaid invoices in pink. At the top you can see the total invoices send that month.



2.4. Debt division page

The third and final page shows more details about the outstanding debt customers have. The left graph gives insight of the owed money per age group. The age is the amount of days since the due date of an unpaid invoice. Within these bars, the top 5 cities are highlighted with their respective amounts of debt. On the right, we can see the trend line of new debtors per month at the top. At the bottom, the number of debtors is compared to the total number of customers of the top 5 cities.





Documentation

Carleon case – Dashboard

Group 4

Anna Chernova, Izabella Bogdanova, Yazan Fattal, Gijs Wijngaards, Sjoerd Heijmann, Tan Quang Nguyen

17/01/2022

Table of Contents

1. Overview

The dashboard's main goal is to let Carleon gain a better understanding of their present financial status by offering some crucial data.

The dashboard's underlying concepts and models, as well as the calculation formulas, will be explained in this document.

2. Dashboard

2.1 Import data

The data are retrieved from *SQL Server* (which is already provided) by using data connectivity mode: *DirectQuery* in the *PowerBI.*

Following the import of raw data. *PowerQuery* makes changes to all the data tables, such as renaming, removing, and unpivoting columns, among other things.

Before producing the graphic, we created the relationship between tables in the model tab, to ensure that our data has a connection and that we will be able to work on filtering.

2.2 General

The dashboard is split into 3 pages which are: *Unpaid invoice progression*, *Debtor division* and *DSO*

There are two types of graphs that can be used to examine how data changes over time: *line charts* and *bar charts*.

Two slicers are included on each page to assist users in filtering out a specific amount of time if necessary.

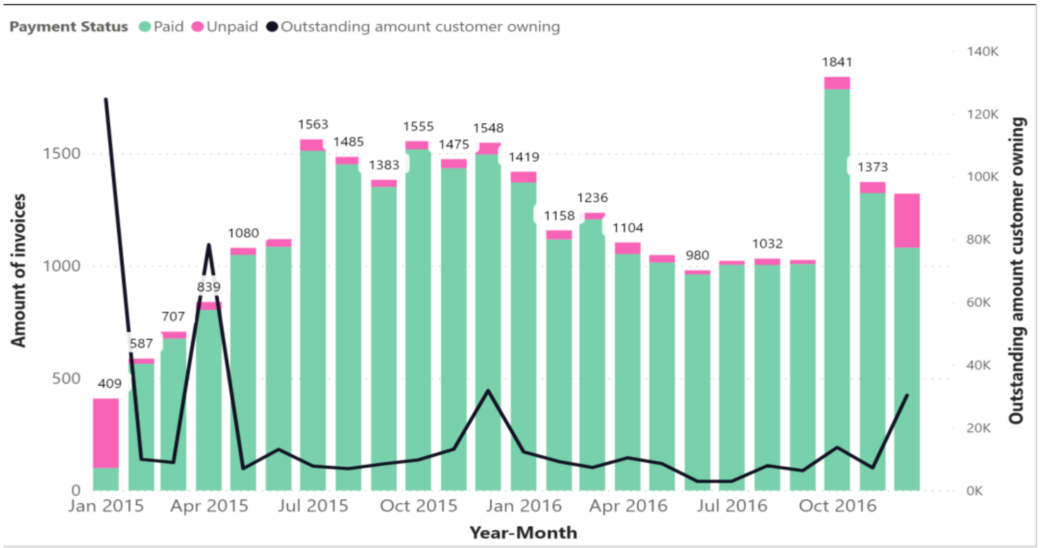
2.3 Scope

The dashboard's purpose is to display our KPIs, which are displayed in the charts below. We managed to emphasize the key performance indicators we pulled from the data that will be essential for the customer on the charts.

We used a variety of techniques and tools to create them, like Power Bi, which is the major one where the dashboard is displayed, and SQL Server, which is where the data is taken. However, to finish our job, we will contribute to RStudio, where validation will be included to ensure that our calculations are proper and may be used in result.

2.4 Charts

Model 1 : Amount of invoices and oustanding amount owning by year-month and payment status.



The stacked columns illustrate the number of paid (green part) and unpaid (pink part) invoices – measured in the right y-axis

The line graph shows the total of outstanding money owning – measured in the right y-axis

Calculations:

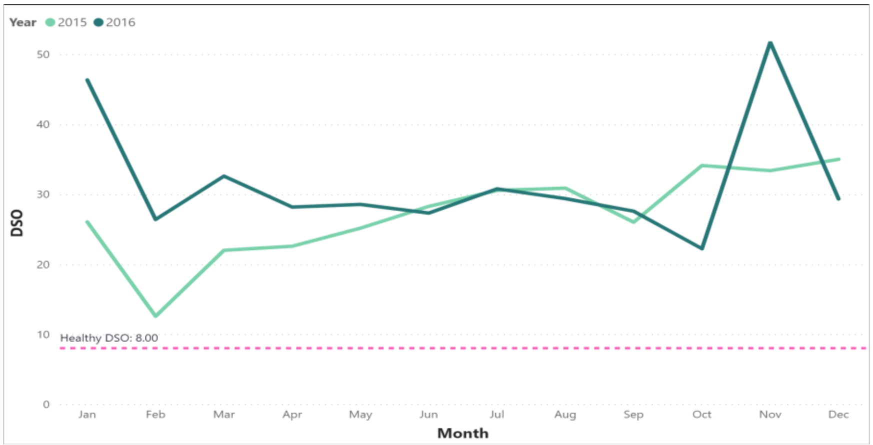
+) An additional column ('Payment Status') is created based on the 'Paid' column in powerbi_balancelist table : If Paid=0→ 'Unpaid', Paid=1→ 'Paid'

+) Using count function for the column 'Payment Status' (COUNT(powerBI_balancelist[Payment Status])) to count the number of invoices with different status

+) The total of outstanding money owning is calculated by summarise all the values greater than 0 in `Outstanding Amount` column in the 'GRV_DebtorOutstanding' table : CALCULATE(SUM(GRV_DebtorOutstanding[Outstanding Amount]),GRV_DebtorOutstanding[Outstanding Amount]>0)

In general, there is not many unpaid invoices in each month-year. It just fluctuates around between 30 and 60 invoices. In some like Jan 2015, Dec 2016, the number of unpaid invoices reached to more than 200. Normally, we will think that the outstanding money will increase if the unpaid invoice increases. It's true, but not entirely, for example the outstanding money still grow up even if the number of unpaid invoices drop from Jun to July in 2017. It is because the outstanding amount is depend on the value of each invoice, not the quantity.

Model 2 : Daily Sales Oustanding



The dark green line represents for the DSO in 2016

The light green line represents for the DSO in 2015

The dashed line is the safe level for DSO (assumption is 8)

Formula: DSO= (Account Receivable/Total Credit Sales) * number of days

Calculations:

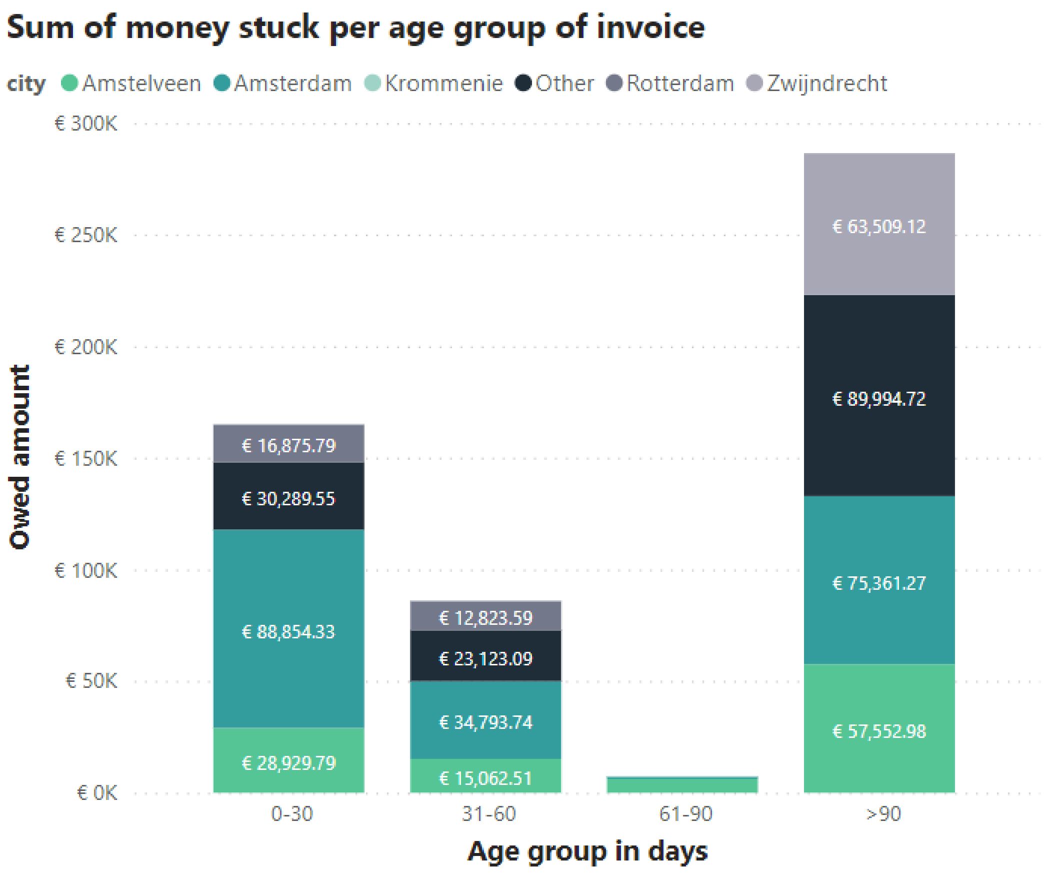
+) Total credit Sales is calculated by summarising the total invoice amount in in powerbi_balancelist table: Credit Sales = SUM(powerbi_balancelist[InvoiceAmount])

+) Account receivable from customers = total credit sales- amount that customer has paid:

[Credit Sales] - CALCULATE(SUM(powerbi_balancelist[ReceiptPaid]), powerbi_balancelist[Paid])

It is important for the company to keep track of the DSO-the average number of days to take the customers paying their bill after receiving invoice. By calculating DSO, the company is able to know how long they can get money from their customer on average, then if it suddenly increases, they can respond quickly before losing some unnecessary money for suppliers because of paying late as customers paying late. In general, the DSO fluctuates around 30 days which is not a big problem, but it should be supposed to decrease. Therefore, the company can get money earlier from customers to pay for suppliers

Model 3 : Sum of money per age group



This graph above shows the sum of outstanding money per age group with top5 city. As we can see, Amsterdam and Amstelveen are cities with the most oustanding money in all 4 groups, especially in the group >90. Most of the owed money also are stuck at this group which is more than 50% of total. It is really worrying as the company cannot use this money to pay for suppliers. It is understandable that bigger cities have more outstanding money, it might be because the company has more property and customer than smaller cities.

The measure “Top 5 and other” calculates the [Sum of money per age], by filtering out only the top 5 cities based on the sum of money measure. If a city is not in the top 5, its value gets added to “other”

Others:

Number of customers calculated by count the distinct code in cicmpy table: DISTINCTCOUNT(cicmpy[cmp_code])

Number debtors calculated by count the distinct code as for the debtors with outstanding amount >0

CALCULATE (DISTINCTCOUNT(GRV_DebtorOutstanding[Debtor code]),GRV_DebtorOutstanding[Outstanding Amount] > 0)

Number of reminders = COUNTROWS(powerbi_reminders)