**Project Charter**

**PRJ4D – Group 9**

**Abstract**

The goal of this project is to find out why creating a Sales Product Analytics and Benchmarking website is best done using Scrum. Our small team benefits from Scrum's efficiency and adaptability, which enable us to work well together and respond rapidly to changes. Prioritizing key features and focusing on our goals are our main concerns to increase the satisfaction of the stakeholder and quality of the end-product. Setting up a database, integrating APIs, and putting features in place for efficient sales data visualization are important steps. Weekly meetings and code reviews are used to guarantee quality. Stakeholder satisfaction, data accuracy, visual aid effectiveness, and security compliance will all be used to make sure we stay on track for our goals. This project provides insightful information about implementing agile approaches in practical development scenarios.

**Contents**

[**Introduction** 1](#_Toc162365763)

[**Business Case** 2](#_Toc162365764)

[**Approach** 3](#_Toc162365765)

[**In Scope and out of Scope** 4](#_Toc162365766)

[**Deliverables** 6](#_Toc162365767)

[**Quality Management** 7](#_Toc162365768)

[**Prerequisite** 8](#_Toc162365769)

[**Success Criteria** 9](#_Toc162365770)

[**Work Break Down Structure** 10](#_Toc162365771)

# **Introduction**

# **Business Case**

# **Approach**

Why Scrum is the optimal choice for building our sales product analytics and benchmarking site.

This chapter talks about why the Scrum methodology is the most effective approach for developing our Sales Product Analytics and Benchmarking site. While waterfall and hybrid approaches have their advantages, Scrum is best on adaptability, continuous improvement and efficiency making it perfectly suited to the needs of this project and our small team.

Benefits of Scrum for our project:

Adaptability to evolving needs: Building a data platform involves some unknowns. User needs and functionality can adapt based on data gathered during development. Scrum's nature allows us to easily apply these changes into sprints, ensuring the site remains relevant.

Faster delivery and user feedback: Scrum prioritises the delivery of working features in short sprints. This allows us to quickly launch a basic version of the site with core functionality. Early user feedback from these sprints is invaluable in prioritising features and refining the overall user experience.

Efficient workflow for a small team: With a limited team size, clear communication and collaboration are critical. Scrum encourages this through daily stand-up meetings and sprint reviews. Team members stay on the same page, identify roadblocks early and address them together, resulting in a more efficient development process.

Reduced risk and early course correction: Scrum focuses on delivering working features in increments. This allows us to identify and address potential issues early during sprints, minimising rework and development costs associated with significant changes later in the project.

Continuous improvement based on data: The nature of Scrum allows for continuous improvement of the site. Data collected throughout the development process (user behaviour, feature usage) informs future sprints.

Quality Management

To ensure that our dashboard meets the expected performance, reliability, and usability, we apply the following quality management practices.

Quality Assurance: Every line of code is reviewed by a team member to ensure quality and to ensure we keep up to our coding standards.

Quality Control: Weekly meetings are held to review progress and discuss sprints. In addition, we will have three stand-up meetings per week to answer questions, discuss challenges, and reassign work if needed.

**Coding Readability:**

* Using indentations to mark the beginning and end of control structures to specify code blocks.
* Writing compact code and avoiding longer functions. Ideally, each function should have a single task.
* Using the DRY (Don’t Repeat Yourself) principle. Automating repetitive tasks whenever necessary.
* Avoiding deep nesting. Keep code straightforward and easier to follow.
* Keeping the length of lines manageable for better readability

**Standardizing Headers:**

* Having consistent headers in each module containing:
  + Module Name
  + Creation Date
  + Creators Name
  + (Modification History?)
  + Summary of Functionality
  + Functions and Variables

**Unique Identifiers:**

* Assigning meaningful, unique names to each variable.

**Comments and Documentation:**

* Writing comments to explain complex code. Making sure that the reader gets guided through the logic and algorithm.

**Exception Handling:**

* Use try-catch blocks to manage exceptions.

**Peer Review:**

* **Conducting peer reviews for all code changes**

**Testing and Verification:**

* Writing unit tests to verify logic of each module and function.
* Ensure, through testing, that logic works together as intended.

**Closing Notes:**

* Test all code on real devices, browsers and on live servers to ensure it is working as intended.

# **In Scope and out of Scope**

During this project we can only focus on a certain amount of features to show data within the analytics dashboard. There are endless amounts of features that could be implemented, therefore it’s important to put boundaries in place and decide on the most important features that we can implement. These are features that will give the most value to the stakeholders as well as making sure that that certain feature can be done within the given timeframe.

The goal of this project is to help the ecommerce company make additional profits through analysing their data. This is not a clearly defined objective, so our real objective will be to deliver a fully functional website with the necessary API connections from the marketplace & content management system, graphs showing insights in sales & the amount of units sold per product as well as a database that will save all the incoming data.

We will acquire customer reviews for the seller (ecommerce company) via the marketplace API. These are only the seller reviews, not the product reviews. We can only access the product reviews via web scraping, which is not within the scope of this project.

During week 6 of the project, we will make sure we have a working API connection, we’ve set up our database as well as having a web hosting service to host the domain name. We’ve gathered a lot of information about the company and what kind of data would be valuable, but we’re still thinking of ways to solve some of the problems they are having such as not knowing if they should keep more stock for B2C clients because the profit margins are way higher as well as other issues that could potentially be solved with the software application.

In week 7 we’ll implement our first feature which is to show a graph with the amount of sales for that day, in the deliverables section we’ll go into more detail on these features mentioned here in the timeline.

In week 8 we’ll give more insights on these sales, particularly the amount of revenue separated for the webshop & marketplace. The amount of profit can be calculated as well with information given from the company such as transport fees, cost of goods, commission, software costs etc.

This will be helpful to know how much discounts will do for profit, because software costs, warehousing costs & employees have a set monthly price, increasing sales with lower profit margins could be a way to increase profits with more information on this data.

During the 9th week we’ll track warehousing stock so we have more insights if the amount of stock is going up or down overall, each month new products will be published on the marketplace but the amount of stock for these products is never the same because of the pre-sale for B2B only clients. Tracking this will give more information on how much inventory to buy in.

# **Deliverables**

During this project we will make an analytics dashboard for an ecommerce company. This includes a working software application with API connections to the marketplace & content management system.project

The postgreSQL database includes historical data as well as data in cache for the incoming orders.

It will include all the code for the frontend of the website & the domain itself.

The first feature that will be included is a sales dashb

# **Prerequisite**

# **Success Criteria**

The following must be provided to call the project successful:

A satisfied stakeholder and a working dashboard. We want to reach these goals by taking care of the following: data accuracy and integrity, data visualization effectiveness and security and compliance.

Data accuracy and integrity: Being able to assess the accuracy and integrity of the data in the dashboard. Making sure to be able to reliably be able to detect if an error occurred to prevent data errors from occurring.

Data visualization effectiveness: Evaluating the effectiveness of data visualization techniques used in the dashboard. We will measure by clarity and the ability to gain insights effectively to users.

Security and compliance: Verify that the dashboard keeps up to a certain level of security and compliance standard.

# **Work Break Down Structure**

