**Project Plan**

***Energy Grid North***

*Gertjan Schouten*

|  |
| --- |
| **Date : 2/23/2021** |
| **Version : 0.1** |
| **State : INCOMPLETE** |
| **Author : Group S63-1** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| 0.1 | 23/2/21 | Group 63-1 | Eerste opzet | Opzet |
|  |  |  |  |  |
|  |  |  |  |  |

**Distribution**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Receivers** |
|  |  |  |
|  |  |  |

# Table of Contents

[Version history](#_heading=h.k9b98fr5kf6u) 2

[**Table of Contents**](#_heading=h.sq6khqwadnim) **3**

[Project assignment](#_heading=h.1t3h5sf) **3**

[Context](#_heading=h.4d34og8) 4

[Goal of the project](#_heading=h.2s8eyo1) 4

[Scope and preconditions](#_heading=h.17dp8vu) 4

[Strategy](#_heading=h.ql7p9azcy99x) 5

[Research questions](#_heading=h.26in1rg) 5

[End products](#_heading=h.lnxbz9) 6

[**Project organisation**](#_heading=h.8tlg69ny27v7) **6**

[Stakeholders and team members](#_heading=h.1ksv4uv) 7

[Communication](#_heading=h.44sinio) 7

[Activities and time plan](#_heading=h.2jxsxqh) **8**

[Phases of the project](#_heading=h.z337ya) 8

[Time plan and milestones](#_heading=h.3j2qqm3) 8

[**Testing strategy and configuration management**](#_heading=h.hty04bc0bc11) **9**

[Testing strategy](#_heading=h.1ci93xb) 9

[Test environment and required resources](#_heading=h.3whwml4) 9

[Configuration management](#_heading=h.2bn6wsx) 9

[Finances and risk](#_heading=h.qsh70q) **9**

[Project budget](#_heading=h.3as4poj) 10

[Risk and mitigation](#_heading=h.49x2ik5) 10

# Project assignment

## Context

*<<Describe the company and context briefly.>>*

The Dutch energy sector has asked us to create a simulation tool for energy production and consumption. The energy sector consists of multiple stakeholders with the most important ones being TenneT (Balance Responsible Party), utility companies and consumers. With the rise of renewable energy it is important to balance the production of non-renewable energy to prevent the grid from blackouts or grid failures.

## Goal of the project

*<<Describe the goal of the project. Take into account:*

*The why, what is the reason for doing this project ?*

*What would the new preferred situation look like ?*

*What are the advantages of this project?*

*How does this project add value to the company/context?*

*Which possibilities are offered by the ICT product that the project will realize ?*

*>>*

The goal of the project is to make a loosely coupled system that is able to visualize energy production, transactions and simulations for multiple different stakeholders that all have their own needs and requirements.

The reason for doing this project is to learn about enterprise level software architecture and implementation of this concept. The advantage of this project is that the energy production and consumption can be predicted more accurately, which can lead to less energy waste.

## Scope and preconditions

*<<What activities, and which end products (to what extent or quality) belong to the project, and which don’t >>*

|  |  |
| --- | --- |
| **Inside scope:** | **Outside scope:** |
| 1. (Large-Scale) Consumers Front-end | 1. Maintenance |
| 1. Utility Front-end | 1. Production/Consumption in other regions |
| 3 BRB Front-end |  |
| 4 Producers Front-end |  |
| 5 Backend API (Gateway) |  |
| 6 Raw Energy Data Database |  |
| 7 Processed Energy Data Database |  |
| 8 Consumenten Database |  |
| 9 Sales Database |  |
| 10 Production/Consumption in north region |  |
| 11 Requirement specification |  |
| 12 Research report |  |
| 13 Application testing |  |

*<< Indicate any preconditions. E.g., think of technology choices that have already been made by the company. Note that you are also expected to retain a critical, but constructive, mindset for choices already made >>*

## 

## Strategy

The strategy that we will approach is SCRUM.

The reason for choosing SCRUM is because we as a group (S63-1) are part of a larger group (S63) that need to communicate with each other. Meanwhile we have to be a *self organizing* team. With the use of microservices we can deliver software products within the agile sprint cycle to be delivered on time all while being transparent with the product owner and other teams.

This also allows us to quickly improve other services over time as we learn more about them and from other teams as we have less waste and work more efficiently.

## Research questions

*<<*

*Describe the research questions that are most relevant to your project. For each research question, describe the approach and/or methodology. Use the Dot Framework to specify strategies and methods - see* [*http://www.ictresearchmethods.nl*](http://www.ictresearchmethods.nl) *for details.*

*Note that research is not only part of the initial phases (like analysis) of the project, but runs throughout the whole project. E.g., in the realization phases, you will probably do research in the Workshop and Lab context.*

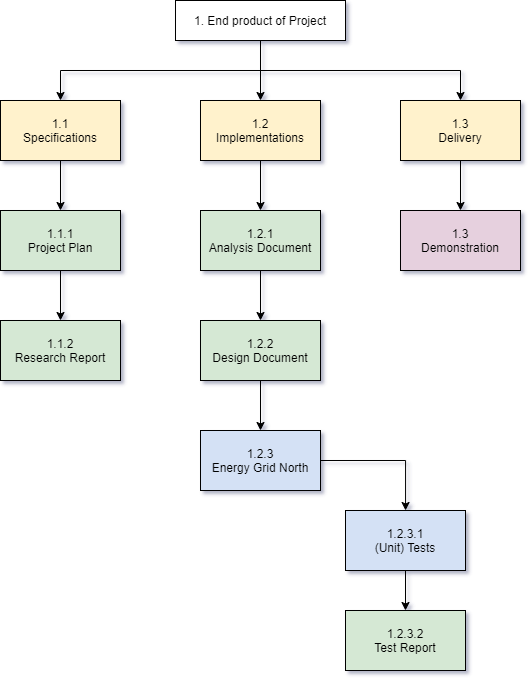
*Also realize that during the project your research questions may change, and that new ones will come up. That is normal for any project ☺, and is not a problem as long as you involve the right stakeholders, and keep your deliverables updated and in sync.*

*>>*

The initial research questions are subject to change but currently are as follows:

|  |  |
| --- | --- |
| **Research Question** | **Methodology** |
| What is the amount of energy that energy parks such as wind and solar energy produce in the northern region? | Library - Literature Study |
| What do the different stakeholders wish to see and use in the application? | Field - Stakeholder Analysis  Library - Expert Interview |
| Do we need Reactive Extensions in our Microservices architecture to improve the efficiency? If so, which approach will suit our case? | Field - Stakeholder Analysis  Library - Expert Interview, Literature Study |
| Which backend and frontend frameworks are best suited to implement a microservice architecture with which energy production, consumption and transactions can be monitored? | Library - Available product analysis, Best good and bad practices, Expert interview  Field - Document analysis  Lab - A/B testing |

## End products



*<< A Product Breakdown Structure (PBS) lists the end products that you realize, including a description of each product. In software engineering, the end products are more than just the project plan and the application itself. E.g., requirements documents, architecture documents, research reports and test reports are all end products. These are all important products that are required for effective handover. They are also necessary for further maintenance and follow up-projects. The PBS can change during the course of the project..>>*

# Project organisation

## Stakeholders and team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **Role and functions** | **Availability** |
| *Gertjan Schouten* | *GS* | *Product Owner* | *Tuesday morning: 30 minutes*  *Thursday morning: 30 minutes* |
| *Nico Kuijpers* | *NK* | *Software Architect* | *Thursday afternoon: 30 minutes* |
| *Frank Coenen* | *FC* | *Enterprise Software Engineer* | *Monday morning: 30 minutes*  *Thursday afternoon: 30 minutes* |
|  |  |  |  |
| *S63 - 2 (East)* | *East* | *Energy grid partner* | *Monday, Tuesday, Thursday* |
| *S63 - 3 (South)* | *South* | *Energy grid partner* | *Monday, Tuesday, Thursday* |
| *S63 - 4 (West)* | *West* | *Energy grid partner* | *Monday, Tuesday, Thursday* |
| *S64 - 3 (Belgium)* | *Belgium* | *Energy grid partner (Foreign)* | *Monday, Tuesday, Thursday* |
|  |  |  |  |
| *Consumer/Prosumer* | *Cons/Pros* | *Stakeholder* | *n.a.* |
| *Large-scale consumers* | *LSC* | *Stakeholder* | *n.a.* |
| *Utility companies* | *UC* | *Stakeholder* | *n.a.* |
| *Balance Responsible Party (TenneT)* | *BRP* | *Stakeholder* | *n.a.* |
| *Producers* | *Prod* | *Stakeholder* | *n.a.* |
|  |  |  |  |
| *Jursley Gonzalez* | *JG* | *Software Engineer* | *Monday, Tuesday, Thursday* |
| *Nick Krijgsman* | *NK* | *Software Engineer* | *Monday, Tuesday, Thursday* |
| *Maarten Blömer* | *MB* | *Software Engineer* | *Monday, Tuesday, Thursday* |
| *Faruk Aydin* | *FA* | *Software Engineer* | *Monday, Tuesday, Thursday* |
| *Vincent Andersen* | *VA* | *Software Engineer* | *Monday, Tuesday, Thursday* |

## Communication

S63-1 group meetings will primarily be held online in the Discord S63-1 channel. The group members will be available on their assigned days in the Discord voice channel unless specified otherwise.

Official meetings with teachers will be held online in the S63-1 Teams channel within the formally assigned time slots. Unplanned meetings with teachers may also occur in this same channel but with other times.

# Activities and time plan

## Timeplan of the project

The project will contain sprints that have a length of 2-3 weeks where each sprint will have daily standups on the assigned days as a group.

The sprints will be set up at the beginning of every new sprint where relevant issues will fill the backlog.

At the end of the sprint a group evaluation and reflection will take place in order to improve group dynamics and software struggles. These problems will be addressed and tackled in the following sprints after they were found.

Near the end of the project there will be a lot of focus on testing and coupling the different microservices as they are being prepared to be handed in with the best possible quality.

## Time plan and sprint goals

|  |  |  |  |
| --- | --- | --- | --- |
| **Phasing** | **Effort** | **Start date** | **Finish date** |
| 0 Introduction | 1 | Week 1 | Week 3 |
| 1. Non functionals | 3 | Week 4 | Week 6 |
| 1. Messaging & Security | 6 | Week 7 | Week 9 |
| 3 Continuous deployment | 7 | Week 10 | Week 12 |
| 4 Observability & Distributed Data | 7 | Week 13 | Week 15 |
| 5 Emerging trends | 5 | Week 16 | Week 17 |

\*Op schaal van 1-10 tijdsbeslag

# 

# 

# Testing strategy and configuration management

## Testing strategy

*<<Which testing strategy do you envision? E.g., on which levels will testing take place? Consider that you could choose unit, component, integration, system, or acceptance testing.*

*Justify your strategy, and also set goals where relevant. E.g., percentage code coverage for the relevant unit tests. For each of the planned tests, indicate what will be automated and what not.*

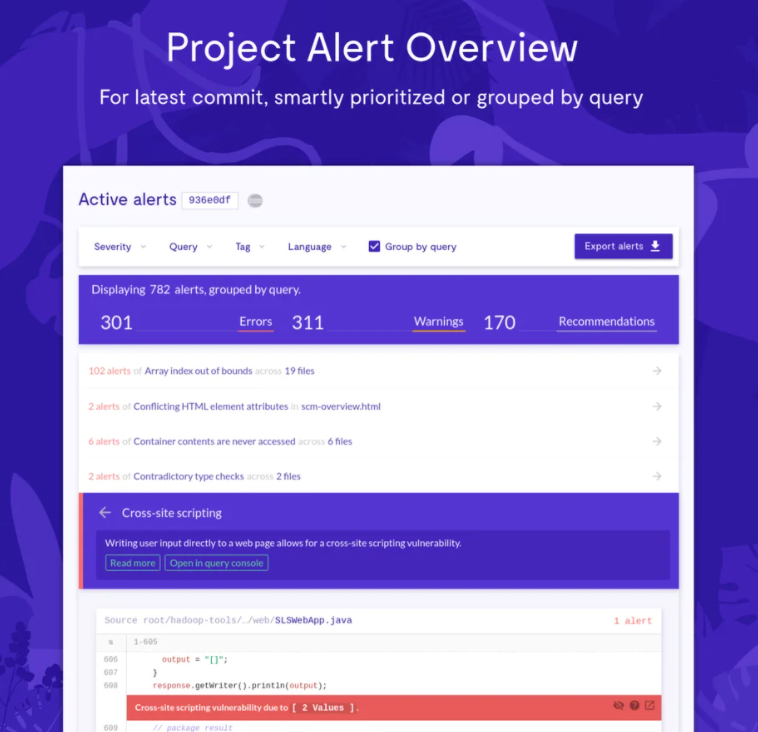
*Also think of quality testing setups like, e.g., Sonarqube.*

*>>*

The code will be tested for desired and undesired results. This will help us to address known errors. If new problems arise during testing, we will write a new test case and solve the problem.

We will treat each service as a software module. This means that every service has their own (unit) tests. We will also test across different setups. This will give us more insights about the results. And the most important part of our tests will be the ones that are testing the cooperation between the services.

To check our code quality we will make use of LGTM.



## Test environment and required resources

*<< Describe the test environment. E.g., do you envision a DTAP (Development, Testing, Acceptance, Production) environment. Can you make use of a CI/CD environment or will you develop your own?*

*It often helps to use a picture to visualize the test environment.*

*If you already know, describe which resources are required for realization and testing. Think of hardware, cloud environments and specific tooling required for development and testing.*

*>>*

Every member of our group has their own setup. The tests can be started manually. But we will also create a CI/CD environment where the tests run automatically. We will make use of TravisCi and Heroku. TravisCi will handle the automatic tests, if the tests succeed the tested branch will be deployed on Heroku.

## Configuration management

*<< Describe the project approach with respect to version management. This might include things like tooling, branching strategy, promotion-, release- and baseline strategy.*

*Also, when relevant, think of a mechanism to deal with change requests and problem reports.>>*

We will work on a GitHub repository where we split versions within branches. When a version is tested and approved, the branch of this version will be merged to the main branch. Our CI/CD configuration will notify us when there are some conflicts.

# Finances and risk

## Project budget

*<< If specific budget is required for your project, indicate it here, and also what needs to be done to get budget approval. Think of hardware, applications, libraries, development environments, etc.*

*Regular costs that have already been covered, like an internship compensation, do not need to mentioned.*

*>>*

Next to the received compensation for deployment with Microsoft Azure, no budget is needed for the duration of the project.

## Risk and mitigation

*<< Investigate and define all risks affecting the project. For each risk indicate what has been done, or will be done during the project, to prevent the risk from being actualized, and define the mitigation actions, such as what you plan to do if the risk actually eventuates.*

*In a more elaborate version, you can also label the risks with their chance of occurence and impact. The advice is to focus on risks that have both a real chance of eventuating and some considerable impact. Direct risks, like what to do if your company supervisor is not available anymore, should always be described, as they have happened in the past quiet regularly.*

*>>*

|  |  |  |
| --- | --- | --- |
| **Risk** | **Prevention activities** | **Mitigation activities** |
| 1. The loss of a team member | Team members must try to indicate in advance whether there is something going on that can cause this. | Distribute designated duties of dropped team members among remaining team members. |
| 1. In case of illness of a team member |  | Discuss the severity of illness within the team and which tasks the team member can still perform. |
| 1. Damaged files | Clear agreements about the storage / storage of products made. | Backups of all products and current state of them. |
| 4 Time scrambling to complete a task | Team member must indicate in time when they run into something. Only work on tasks set up for a sprint and do not pick up new tasks until old tasks are completed. | Team members who have extra time can help with the unfinished task. |
| 5 Non-functional workstation |  | Borrow a laptop from the ISSD |
| 6 Poor quality code | Code reviews when a task is completed. | Testing of all code |
| 7 Estimation and scheduling | Sprint plannings, apply feedback from previous projects regarding scheduling. | Prioritize leftover tasks during schedule adjustments |