**Project Plan**

***SUE-Data-Science-Platform***

*Nathan Keyaerts*

<<

*This template can be used for all projects, especially software engineering projects. Chapters or parts that are not applicable can be removed.*

*Text in italic is background information and must be removed in the final version of your project plan.*

*Note that this is a template and can be changed for own purposes, e.g. you can adapt the layout to the layouts as used at the company of your internship.*

*For your project name, think of a name that highlights the most relevant aspect of your project, and specify whether it is about graduation internship or third year internship.*

>>

|  |
| --- |
| **Date** **:** **04.09.2024** |
| **Version** **:** **0.1** |
| **State** **:** **In progress** |
| **Author** **:** **Andrei Pieleanu, Mohammed Bouali, Björn Wildeman, Calvin Kwan** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| 0.1 | 04.09.2024 | Mohammed Bouali, Calvin Kwan, Andrei Pieleanu, Björn Wildeman | Initial Version | In progress |
|  |  |  |  |  |
|  |  |  |  |  |

**Distribution**

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

Contents

[1. Project assignment 3](#_Toc1351705883)

[1.1 Context 4](#_Toc444579039)

[1.2 Goal of the project 4](#_Toc1721418520)

[1.3 Scope and preconditions 4](#_Toc755872571)

[1.4 Strategy 4](#_Toc447662753)

[1.5 Research questions and methodology 4](#_Toc2005102979)

[1.6 End products 4](#_Toc1836506755)

[2. Project organisation 4](#_Toc454780821)

[2.1 Stakeholders and team members 5](#_Toc1600214434)

[2.2 Communication 5](#_Toc1042894997)

[3. Activities and time plan 5](#_Toc1370030314)

[3.1 Phases of the project 6](#_Toc1322402578)

[3.2 Time plan and milestones 6](#_Toc707767408)

[4. Testing strategy and configuration management 6](#_Toc55164467)

[4.1 Testing strategy 6](#_Toc789459952)

[4.2 Test environment and required resources 6](#_Toc524571918)

[4.3 Configuration management 6](#_Toc1300051465)

[5. Finances and risk 6](#_Toc500689687)

[5.1 Project budget 6](#_Toc1632997180)

[5.2 Risk and mitigation 6](#_Toc750843781)

# Project assignment

## Context

Sue is a leader in Cloud Native technologies and solutions. Our Cloud Native experts are pioneers and innovators in the field. They are passionate about learning and immersing themselves in the newest technologies. We provide cloud-native solutions for various customers throughout the Netherlands, including the Dutch Police, Booking.com and more. We provide a range of services and solutions, and pride ourselves on being a leader in the cloud-native space.

## Goal of the project

The goal is to build upon the existing CLI tool by adding more features utilizing API calls.

## Scope and preconditions

|  |  |
| --- | --- |
| **Inside scope:** | **Outside scope:** |
| 1. Creating the API application | 1. Implementing machine learning functionalities |
| 1. Expanding the application’s functionalities | 1. Modifying the CLI version of the application |
| 1. Documenting the application | 3 Building and accessing SUE infrastructure |
|  | 4 Long term support |

## Strategy

**Project Strategy: Agile (Scrum)**

**Justification:**

The choice of an Agile approach, specifically Scrum, is justified for several reasons:

1. Flexibility: Adapts to evolving project requirements.
2. Collaboration: Promotes cross-functional teamwork.
3. Quick Deliveries: Provides rapid, incremental releases.
4. User-Centric: Focuses on user feedback and satisfaction.
5. Risk Mitigation: Identifies and addresses challenges early.
6. Quality Assurance: Ensures high-quality development.
7. Transparency: Keeps stakeholders informed.
8. Continuous Improvement: Allows for ongoing enhancements.

Overall, Scrum aligns with the project's dynamic nature, enabling effective development and user-focused improvements.

## Research questions and methodology

Main research question

How do we implement the functionalities that are in the current CLI into an API that can be integrated into SUE’s infrastructure so that they can offer it as a service to their clients.

Research sub-questions

How will the API be deployed?

How can we ensure that our software solution is GDPR compliant?

## End products



# Project organisation

## Stakeholders and team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **Role and functions** | **Availability** |
| *Nathan Keyaerts*  [*nathan.keyaerts@sue.nl*](mailto:nathan.keyaerts@sue.nl) | *N.K.* | *Project owner* | *Mon 9:00-12:00 13:00-16:00*  *Wed 9:00-12:00 13:00-16:00*  *Thu 9:00-12:00 13:00-16:00* |
| *Björn Wildeman*  [*b.wildeman@student.fontys.nl*](mailto:b.wildeman@student.fontys.nl)  *+31 06-23255046* | *B.W.* | *Group member*  *Programmer* | Mon-Fri 9:00-12:00 13:00-16:00 |
| *Andrei Pieleanu*  [*a.pieleanu@student.fontys.nl*](mailto:a.pieleanu@student.fontys.nl)  *+40 786908232* | *A.P.* | *Group member*  *Programmer* | Mon-Fri 9:00-12:00 13:00-16:00 |
| *Calvin Kwan*  [*c.kwan@student.fontys.nl*](mailto:c.kwan@student.fontys.nl)  *+31 639731213* | *C.K* | *Group member*  *Programmer* | Mon-Thu 9:00-12:00 13:00-16:00  Fri 9:00-12:00 1 |
| *Mohammed Bouali*  [*m.bouali@student.fontys.nl*](mailto:m.bouali@student.fontys.nl) | *M.B.* | *Group member*  *Programmer* |  |
| *Frank Schürgers* | *F.S.* | *Coach* | Monday 9:00-12:00 13:00-16:00 |
| *Dennis Cools* | *D.C.* | *Technical teacher* | Wed 9:00-12:00 & Thu 9:00-12:00 |
| *Frank Coenen* | *F.C.* | *Technical teacher* | Thu 9:00-12:00 13:00-16:00 |

## Communication

There will be weekly meetings with the stakeholders, every Thursday morning, where we will brief them on the progress and get their feedback, with teachers we will meet Mondays, Wednesdays and Thursdays on site at TQ.

Between the group members we will be communicating at school when we meet according to the Fontys schedule and via discord when we are not together, we will have standups in the mornings of when we meet at school, the goal of the standups is to keep each other up to date and to know if everyone has any issues or bugs that need to be fixed.

# Activities and time plan

## Phases of the project

The project consists of 6 sprints, the first is reserved for problem analysis and understanding the work that was previously done,

## Time plan and milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Phasing** | **Effort** | **Start date** | **Finish date** |
| Sprint 0 | 1 | 02-09-24 | 22-09-24 |
| Sprint 1 | 5 | 25-09-24 | 13-10-24 |
| Sprint 2 | 8 | 23-10-24 | 10-11-24 |
| Sprint 3 | 13 | 13-11-24 | 01-12-24 |
| Sprint 4 | 21 | 04-12-24 | 22-12-24 |
| Sprint 5 | 13 | 08-01-25 | 19-01-25 |

# Testing strategy and configuration management

## Testing strategy

We will be utilizing a Test-Driven development for our backend. That means we will be writing unit tests for our methods to ensure less bugs and guarantee quality. As the project gets more complex, we will add integration tests to ensure our layers communicate without issues. For our frontend, we will add performance tests and/or E2E tests. At the current moment, we don’t know which one suits the best, so we have to conduct research.

Regarding code coverage for our unit tests, we are looking for a percentage greater than or equal to 80%. We will set up a CI/CD pipeline to test our changes and for every pushed branch, we will perform merge request sessions.

## Test environment and required resources

We are looking in using SonarQube to measure our tests to ensure we have the above-mentioned coverage (80%), as well as removing code smells.

When pushing features into production or development the CI/CD environment will run a pipeline to automatically run all tests to see if the newly added code does not break anything else. If all tests are passing, then the CI/CD environment will do a static code analysis and generate a report on any code smells and/or bugs.

## Configuration management

Version control plays a crucial role in software development. For this project, we will be utilizing the GitLab environment provided by Fontys, which will host our codebase and facilitate the use of an integrated CI/CD system. All development work will be carried out in the develop branch, and from there, feature, issue, or bug fix branches will be created as needed. Once the work on these branches is complete, the changes will be merged back into the develop branch. This will be done through pull/merge requests, which must be reviewed by another team member. Merges to the main branch will occur only from the develop branch and only at the end of each sprint.

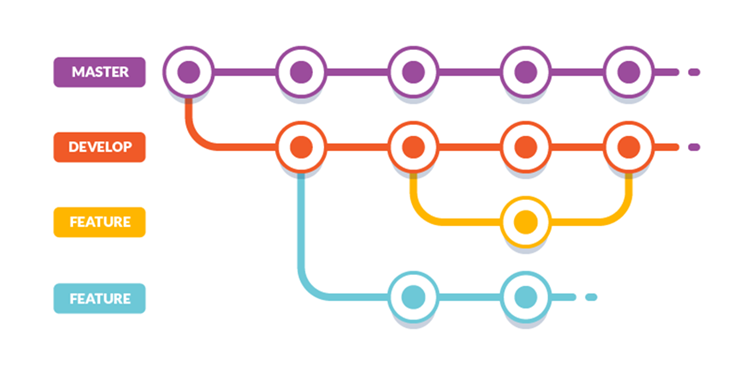


Figure 1: Visual representation of the branching strategy.

# Finances and risk

## Project budget

Currently, no costs have been anticipated for the development of the project. Should this change in the future, it will be documented accordingly, and the stakeholders informed about it.

On 12.09.2024 we have received an AWS account where we can work on our cloud. However, the costs for maintaining AWS will be handled by the people at SUE.

## Risk and mitigation

|  |  |  |
| --- | --- | --- |
| **Risk** | **Prevention activities** | **Mitigation activities** |
| 1. Technology complexity | Perform thorough research and learning for the technology | Communicate with other individual that has experiences or expertise in the technology |
| 1. Deprecated technology | Choose a well-maintained technology that is regularly updated | Search for another similar technology that is well-maintained |
| 1. Miscommunication | Perform daily communication with other team members and stakeholders | Contact the other team members to clarify the misunderstanding |
| 4 Hardware failure | Be mindful with activities that utilize personal hardware | Request a new laptop from ISSD and continue work from there until the problem has been solved. |