Project Plan

Business Marketplace for Edge Intelligence Sensors

08/03/2022 Eindhoven
Version: 2.1

CB-S3 Group 1

Members:
Aleksej Borisov: 2776286
Oleksandr Gurianov: 4178092
Mohammad Nazibul Khan: 4263308
Lars Kluijtmans: 4220269
Noelia Rodriguez Morales: 3635988
Esther Wolfs: 3329984

Tutor:
Nicole Zuurbier

Version history

Version	Date	Author(s)	Changes	State
0.1	08/03/2022	Esther Wolfs	Start project plan, chapter 1.1 - 2.2	Started
0.2	14/03/2022	Esther Wolfs	Continue start project plan, chapter 3.1 - 5.2	Started
0.3	17/03/2022	Esther Wolfs	Implement feedback	Finished
1.1	15/04/2022	Esther Wolfs	Add sprint 2 in planning	Started
2.1	12/05/2022	Esther Wolfs	Update for sprint c and sprint d	Started
2.2	14/05/2022	Esther Wolfs	Add testing strategy	Finished

Contents:

Version history **1**

1. Project Assignment **3**

1.1 Context 3

1.2 Goal of the project 3

1.3 Strategy 3

1.4 Research methods and methodology 3

1.5 Deliverables 4

2. Project Organisation **5**

2.1 Stakeholders and team members 5

2.2 Communication 6

3. Activities and time plan **7**

3.1 Phases of the project 7

3.2 Time plan and milestones 7

3.3 Sprint planning 7

4. Testing strategy and configuration management **9**

4.1 Testing strategy 9

4.2 Test environment and required resources 9

4.3 Configuration Management 9

5. Finances and risk **10**

5.1 Projectbudget 10

5.2 Risk and mitigation 10

1. Project Assignment

1.1 Context

Ivanti Edge Intelligence is a platform for IT administrators retrieving facts from within their environment. With their platform the administrator is able to discover, gather insights and take actions on happenings in their environment through software sensors. Characteristics about their platform are a simple user interface, fresh data as they retrieve their data directly from their endpoints, high performance, secure and scalable and they get better all the time with continuous delivery of new sensors, content and visualizations.

1.2 Goal of the project

The goal of this project is to create a digital marketplace for Ivanti that allows the company to provide their customers an user friendly environment from where they can download software products created by or related to Ivanti software solutions.

Ivanti currently does not have any specific tool to offer apps and packages to their customers or to allow the developers to upload their new applications or updates. With this project, we want to help Ivanti to extend their Edge Intelligence software so that they extend their catalog of services and offer a better experience to their customers.

The perfect outcome for Ivanti is a marketplace that works together with their already existing software, so it is easier to implement.

1.3 Strategy

For this project we have chosen to work with an agile approach. We will continuously keep delivering small features, instead of delivering the final product at once. This way we can work more flexibly and give a different level of priority to different features, and respond quickly to requests made by our client.

1.4 Research methods and methodology

The methodology that we will use is Agile scrum. (update later)

Research Target	Research method	Observations
Define relevant data to show in the description	Interview	which data is relevant for the user when is considering to download an app
How to display data in a graphical way	Literature study and available product analysis	display data about applications to users to help user select the best option

1.5 Deliverables

Project plan, software solution, architecture, backlog, test plan/documentation, research documents.

2. Project Organisation

2.1 Stakeholders and team members

Our team consists of 6 members, we also have a project tutor and a product owner.

Name	**Abbreviation**	**Role and functions**	**Availability**
Dennis Smits		*Product Owner*	
Nicole Zuurbier		*Project Tutor*	*Available on Monday between 9 am and 12 pm and Thursday between 1 pm and 4 pm*
Aleksej Borisov	*Alex*	*Developer*	*From Monday till Friday*
Oleksandr Gurianov	*OG*	*Developer*	*Monday: 09:00-12:00* *Thursday: 13:00-16:00*
Mohammad Nazibul Khan	*MNK*	*Developer*	*From Monday till Friday*
Lars Kluijtmans	*Lars*	*Scrum Master*	*Monday till Sunday*

Noelia Rodriguez Morales	*Noelia*	*Developer*	*Mondays 09:00-12:00 and Thursdays 13:00-16:00*
Esther Wolfs	*Esther*	*Developer*	*Monday till Friday*

2.2 Communication

The team members will meet each other in person at Fontys Eindhoven every week, at least on Monday morning and Thursday afternoon. For all other communication they will meet using MS Teams, either through voice calls or the normal chat. The goal is to discuss the progress of the project and to work on the project together.

To reach the project tutor Nicole Zuurbier, the team will be able to ask their questions in person at Fontys Eindhoven, on Monday morning and Thursday afternoon. For all communication they can use email or MS Teams.

The product owner Dennis Smits is available for questions and feedback via email or MS Teams.

3. Activities and time plan

3.1 Phases of the project

For this project we work with an agile approach. We work in sprints of 3 weeks, this means that every phase consists of 3 weeks. The first phase of our project is dedicated to research. We will start by writing the necessary documentation, like the project plan and the backlog. We will also write user stories and try to make a plan for the next sprint. At the end of this phase we will deliver our documentation and implement the feedback we will get.

3.2 Time plan and milestones

Every sprint consists of 3 weeks, we will update our plan at the end of every sprint, to include the next sprint.

Phasing	**Effort**	**Start date**	**Finish date**
1. Documentation Draft (Project Plan, Backlog, Wireframe, Activity Diagram, Project Presentation), ready Online Agile Environment, first Project Prototype	~18 hours	07/03/2022	25/03/2022
2. Documentation (Project Plan, Backlog, Project Presentation, Test Plan, Test Report), working Project Prototype	~24 hours	28/03/2022	15/04/2022
3. Work on user stories, make working demo, make unit tests, connect to mongodb	~40 hours	18/04/2022	13/05/2022
4. Finish making most user stories, make the unit tests, remake the api, update css, connect to remote mongodb, fix errors, make research document	~40 hours	16/05/2022	03/06/2022
5.		06/06/2022	24/06/2022

3.3 Sprint planning

Sprint	Member	Task
A	Lars	Make user stories, C4 Diagram
	Noelia	Make user stories, C4 Diagram

	Mohammad	Make user stories, C4 Diagram
	Aleksej	Make wireframe, C4 Diagram
	Oleksandr	Make wireframe, C4 Diagram
	Esther	Make project plan, C4 Diagram
B	Lars	Implement user story (update my content/create my content), TICT, Activity diagram
	Noelia	Make structure for react, TICT, Activity diagram, Login/logout
	Mohammad	Implement user story (delete my content/view statistic), TICT, Activity diagram
	Aleksej	TICT, Activity diagram
	Oleksandr	Make wireframe, Activity diagram
	Esther	Implement user story (read all/my content), TICT, Activity diagram
C	Lars	Fixed bugs, made footer and header same as wireframe, fix merge errors, make unit tests, work on database diagram, made user stories
	Noelia	Made demo, fixed merge errors, learned how to use mongodb, worked on database diagram
	Mohammad	Made demo, finished user stories for this sprint
	Aleksej	worked on user story 6
	Oleksandr	Made a presentation and a search bar
	Esther	Finished user stories, worked on demo, worked on database diagram, updated documentation, made sprint D planning.

4. Testing strategy and configuration management

4.1 Testing strategy

To make sure that all our functionalities are working as expected, we will make a test plan. In this document, we will write the test cases and our expected result compared to the actual result. We will also write unit tests and make sure we have coverage for every single method. The acceptance criteria for our tests are described in the user stories.

To make sure the front end of our application is user friendly we will write a UX feedback report, after asking different users for feedback. The functionalities of the front end are also covered in the test plan.

4.2 Test environment and required resources

Testing our code is necessary to maintain the quality. We have set up a gitlab CI/CD environment, with a runner. Currently we have two pipelines, one for the main build and one for the tests. Everytime someone pushes their work the pipelines are triggered and we can see whether they pass or fail and during which stage. This way we know what part of our code gives us errors, so we can look at that and fix it.

4.3 Configuration Management

For version control we have set up a git repository. Everything that is necessary for the project is in the git repository, the code, the documentation and the presentations.

When working on a user story, the person working on that story makes a new branch to work in. When they finish their story and are sure it is working, a merge request is made so that it can be merged into the main branch. Another group member who has not worked on the story is assigned to review the code, to make sure it is understandable and correct.

To try to avoid merge conflicts, we make sure no one is working in the same branch at the same time and we try not to make any changes directly in the main branch. Always pulling the latest version and not merging a lot of code at the same time is another way we use to try to avoid the merge conflicts. However, sometimes we still get merge errors. To deal with this, the person who tried to merge and the person who made the original code that is giving the conflict work together, to see what changes need to remain.

5. Finances and risk

5.1 Projectbudget

For this project there is no budget.

5.2 Risk and mitigation

Risk	Prevention activities	Mitigation activities
1 The product owner becomes unavailable	There should be at least two product owners, in case one of them becomes unavailable.	
2 A team member gets covid	https://www.wikihow.com/Prevent-Coronavirus	If the team member is not feeling too ill they can work from home
3 Laptop crashes and all data is lost	Always make multiple backups on different hard drives/in the cloud	
4 Code crashed after implementing new feature or changing a feature	Frequently push new features to git	Revert to the previous stable version
5 A team member doesn't finish their part of the project	Always have at least two people working on a feature and don't let an important thing depend on just one person	Communicate with all members on the status of the project and the progress that has been made