

IO6 - PRIMAE: Platform Framework
Platform Timeline and Methodology Document

Author: UCY



Participation and Recreation
through Inclusive Martial Arts
Education and E-Learning



Co-Funded by the
Erasmus+
Programme of the
European Union

Executive Summary

The purpose of this document is to describe the Software Engineering methodology to be followed for the development of the web platform for the PRIMAE project. Furthermore, the scheduled timeline for each step is proposed. This is the first version of the document, and it might be revised after feedback from the consortium partners.

Contents

Executive Summary	2
General Information	4
Platform Implementation Methodology	7
Time Plan	9
References	10

General Information

*Intellectual Output 6: “**Platform Framework**, producing an accessible platform for persons with and without a disability who want to be involved in inclusion in sport, coaching, coach education, assessing, IQA and teaching. Open to all Europeans and based on all other outputs, the platform for an inclusive eLearning Curriculum will be created.”*

- **Leading:** University of Cyprus
- **Technological support** by AREADNE, Futuro Digital
- **Pedagogical support** led by Ikkaido
- **IO6 Official Start Date:** 1st November 2019
- **IO6 Official End Date:** 27th February 2021



Figure 1 – E-learning development structure

Software Development Life Cycle

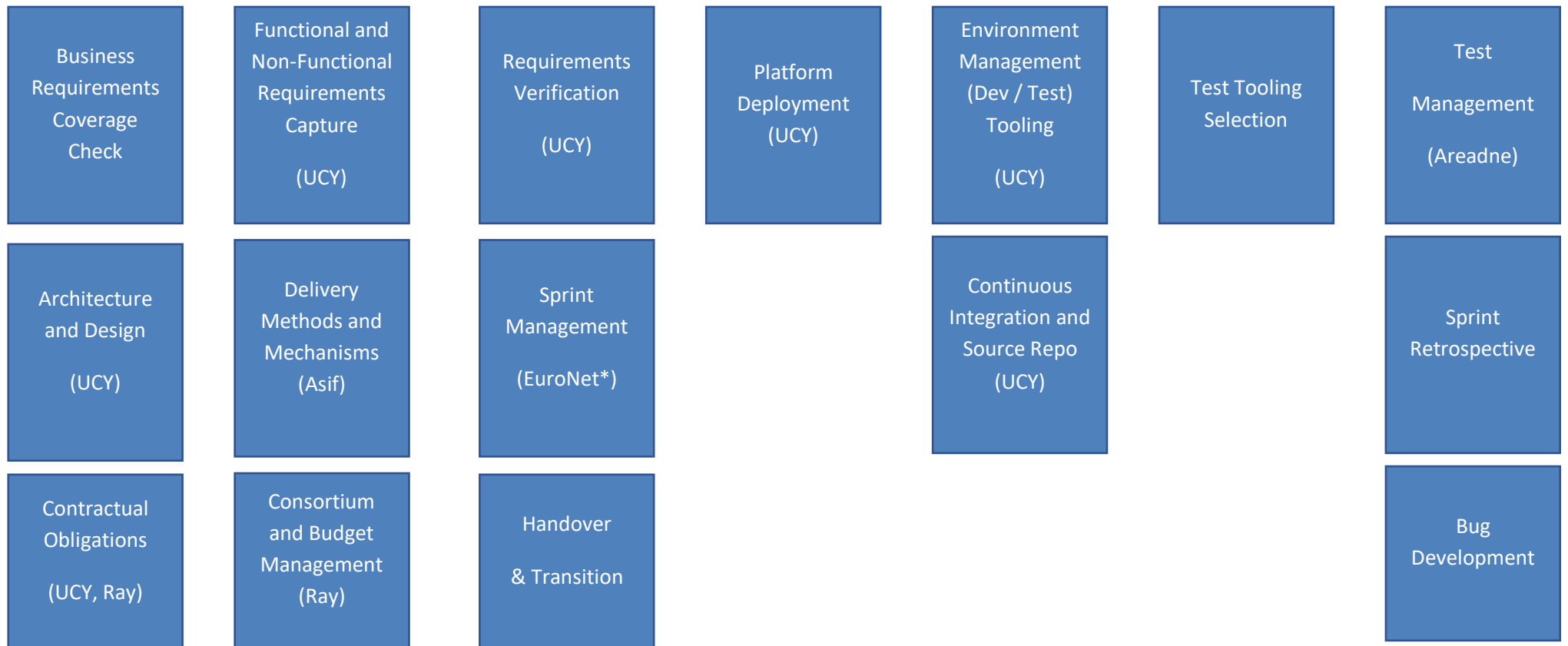


Figure 2 – Development areas

Platform Implementation Methodology

For the development of the platform, the Waterfall [1] Software Engineering Methodology will be followed. In this methodology, the activities of the development are broken down into sequential steps, and each step depends on the results of the previous one. Each step can be revised after receiving feedback from consortium partners; testing; or evaluation with users and should be finalised before proceeding to the next one.

The methodology is displayed in Figure 1.

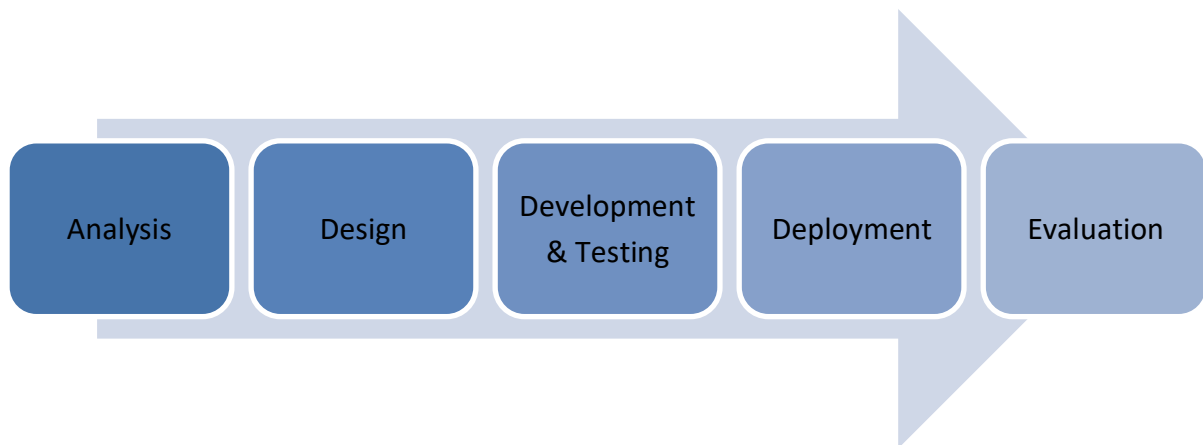


Figure 3 - Platform Development Methodology

The steps of the methodology to be followed, in more detail, are:

- *Step 1 - Requirements Analysis.*

This step aims at analysing and capturing all the requirements of the platform and documenting them in a specification document. Initially, a first version of the requirements reporting will be created based on the project application; the discussions during the weekly meetings; the partners shared documents; and the experience of the development team.

The first draft specification report will be presented to the partners for review. Based on the feedback, a final specification report will be again presented to the partners to review and confirm the finalisation of the first step.

- *Step 2 – Design.*

In this step, the design of the platform will be created based on the result of step 1, in order to plan the application of the specifications. A design for functionality, including database design, mock-ups and workflow scenarios are some potential design elements that might be created by the team, as needed. A non-functional prototype – the first version – might be designed, demonstrating the functionality to be developed in a simple but easy way for the

users to conceptualize it.

The design elements will be presented to the partners and after processing their feedback, the development team will need to receive the consent of the consortium to proceed with the actual implementation.

- *Step 3 – Development and Testing*

In this step, the development team will proceed with the actual implementation of the platform, based on the agreed specification and design from the previous steps.

During the implementation, both the development team, as well as partners, will need to test the platform in several stages as needed.

Testing is incorporated into the Implementation step as the two procedures are dependent on each other and cannot be executed in sequential order. They rather need to be running in parallel. Testing will be happening on different levels. Unit testing will be needed to verify the correct functionality of a single component within the platform. Integration testing will be needed to verify that several functionalities interact, communicate, and work well with each other. System testing will eventually verify that the platform as a whole is functioning correctly. When no bugs are arising anymore, the test will be completed. Figure 2 displays this procedure.

Feedback from testing will help to fix possible bugs and improve the platform’s final version. The final version will need to also incorporate the final elaborated teaching modules.

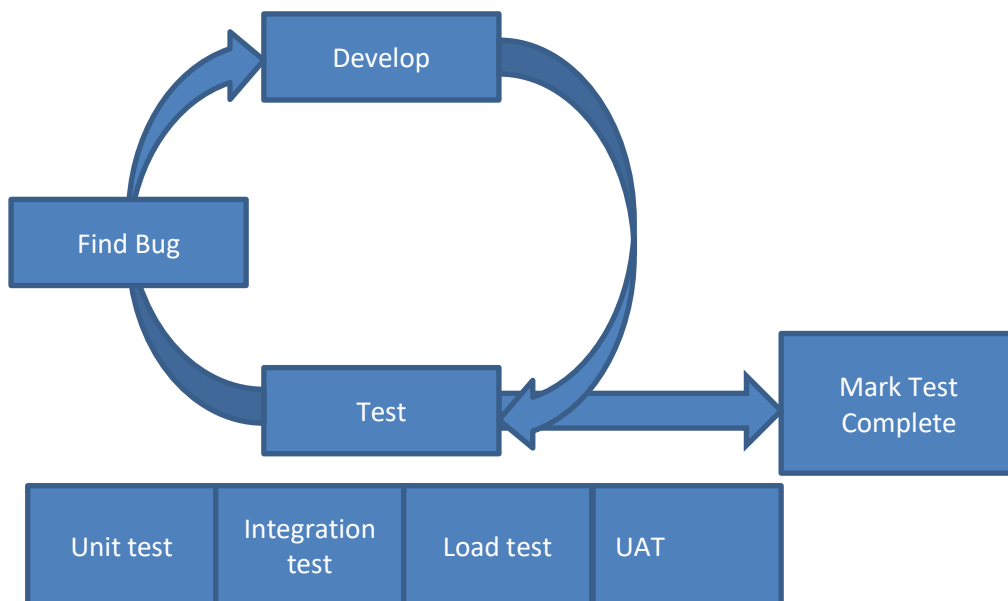


Figure 4- Development and Testing Phase

- *Step 4 – Deployment*

In this step, the final result of the Implementation step, i.e. the developed web platform, will be made accessible to all users, online. From that point on it will provide continuous, full, and open-access service.

- *Step 5 – Evaluation*

This step aims at evaluating the platform’s usability, ease of use, usefulness in functionality and other user-related areas. Feedback should be collected by real users.

Requirements Lifecycle

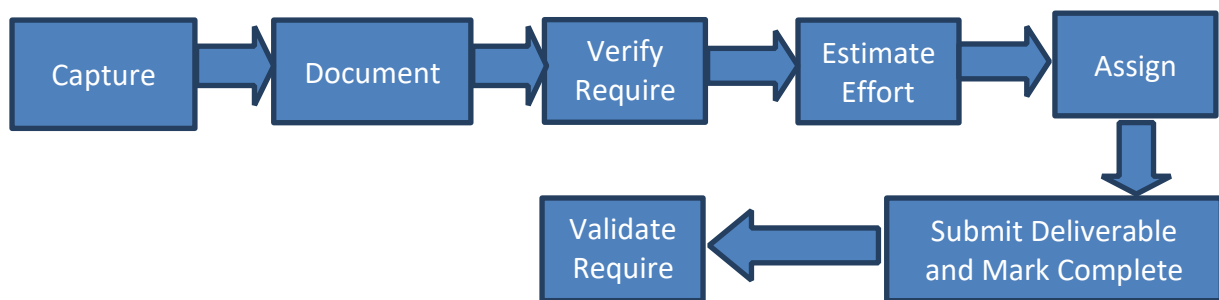


Figure 5 - Requirements Cycle

Figure 4, presents in more depth the lifecycle of a requirement, throughout the engineering methodology. Each requirement will be captured, documented and verified in the Analysis step. Then, in accordance with the Design phase, the effort will be calculated for each requirement and its implementation will be assigned to some members of the development team. During the development and testing phase, the responsible team members will need to implement, test and finalise the respective functionality. The requirement will be initially validated as met during testing in the previously mentioned phase, and then after deployment, it will also be evaluated by the users.

Time Plan

Step 1 - Requirements Analysis: by end of July 2020 (31/07/20)

Step 2 – Design: by the end of October 2020 (30/09/20)

Step 3 – Development & Testing: by end of January 2021 (31/01/21)

Step 4 – Deployment: by end of February 2021 (28/02/21)

Step 5 – Evaluation: based on training and events scheduled.

References

[1] Royce, Winston W. "Managing the development of large software systems: concepts and techniques." *Proceedings of the 9th international conference on Software Engineering*. 1987.

Legal Disclaimer

The project was funded by the European Union through the ERASMUS + programme. The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.