LAPR3 Technical Details
LEI - 2021/2022 – 1st Semester, 2nd Year
v4.0, January 4th

Table of Contents

1 LAPR3 Technical Details

1.1 Development Process

The software development process should be iterative and incremental while adopting good design practices (e.g., GRASP and SOLID principles), coding standards and using a Scrum approach.

1.2 Unit Testing

The implementation process must follow a TDD (Test Driven Development) approach. Unit tests should be developed to validate all domain classes.

Unit tests are based on application design (e.g., Sequence Diagrams (SDs) & Class Diagrams (CDs)). Each sprint evaluation of the project will include an analysis of the quality of testing and the use of a test-driven development approach. Code coverage and mutation coverage are based on unit testing and is performed using quality assurance tools.

1.3 Tasks/Issues & Planning

For this project, task creation, division and planning must be used with Jira issues. Each user story should be created, prioritized, estimated, and assigned to team members. For each user story, three tasks must be created. Each task should focus on:

- **Analysis:** This task should focus on the Use Case Diagram and System Sequence Diagram for each user story. If necessary, possible changes to the Domain Model.
- **Design:** This task should focus on the Class Diagram, Sequence Diagram and Relational Model (Normalised).
- **Implementation:** This task should focus on implementing Test Cases and Code.
- **Review:** This task should focus on reviewing the implementation.

For example, for the user story "User Registration", the following task/issues should be created:

- User Registration [**Analysis**].
- User Registration [**Design**].
- User Registration [**Implementation**].
- User Registration [**Review**].

1.4 Sprint Review

No User Interface (UI) is requested for the system administration. If you wish, you can provide one to better support user interaction or demonstrations. Nevertheless, all inputs and outputs should be performed using text files.

You should have test case scenarios to automatically demonstrate a logical sequence of the requirements by using Unit and/or Integration Tests. In both cases, requirements should be fully tested using Unit Tests and demonstration scenarios should be implemented.

2 Sprint Deliverables

2.1 Moodle

Your project should always be up to date on Bitbucket.

The version that will be assessed every Sprint is the one with the commit time closest to the deadline. Nevertheless, **all team members** should submit the work on Moodle.

At the end of each Sprint students must submit on LAPR's Moodle a **ZIP** file containing:

1. The project repository (all folders in the repository), in a single ZIP file named **LAPR-YYYY-GXXX-REPOSITORY[1].zip,** containing the following technical documentation:
    a. The Project Report that must be written in the project's Readme.md file.
    b. Requirements Engineering:
        i. System Sequence Diagram (SSD) for each user story.
    c. Engineering Analysis:
        i. Domain Model (DM) for the whole project.
            1. Excerpts for each user story.
    d. Engineering Design:
        i. Class Diagram (CD) for the whole project.
            1. Excerpts for each user story.
        ii. Sequence Diagram (SD) for each user story.
        iii. Relational Model (Normalised) for the whole project.
    e. Sprint Self and Peer-Assessment (*i.e.,* excel form).
2. <u>A compilation status report. Please check section 3.2 for more details.</u>

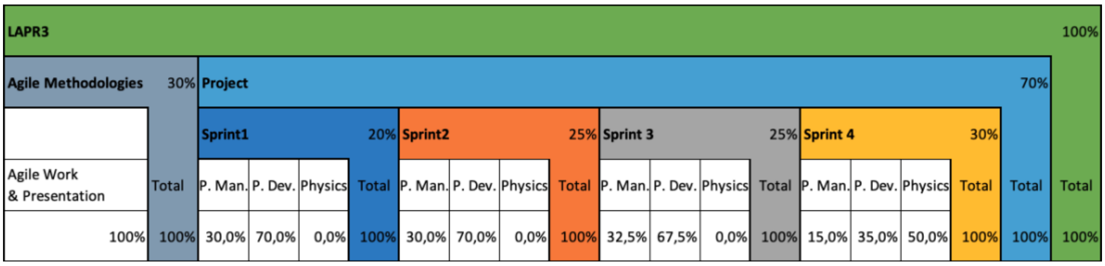
[1] Where YYYY stands for year, e.g., 2021 for 2021-2022 school year and XXX stands for team number e.g., 001.

3 Assessment

Assessment is performed everyday while students are in class and receive immediate feedback. Each Sprint there's a Sprint Review on your Laboratory classes where the project is assessed on a technical basis and a Sprint Review on T/P classes where your work management is assessed.

Student are assessed on the user stories development they participate in. If students are enrolled in other subjects, for LAPR3 the analysis, design, implementation and testing phases of each user story is assessed. If a student in only enrolled in LAPR3, the student may skip the analysis if the user story mainly focuses on other subjects.

For LAPR3, assessment is comprised by two moments, one regarding the Agile Methodologies work prepared on the first three weeks of the project (30%) and the Project (70%) as seen on Figure 1. For the Project, Sprint 1 corresponds to (20%), Sprint 2 (25%), Sprint 3 (25%) and Sprint 4 (30%). For each sprint the weight for Project Management (P. Man), Project Development (P. Dev.) and Physics varies according to the Sprint as shown in Figure 1.



LAPR3																			100%
Agile Methodologies	30%	Project																70%	
		Sprint1			20%	Sprint2			25%	Sprint 3			25%	Sprint 4			30%		
Agile Work & Presentation	Total	P. Man.	P. Dev.	Physics	Total	P. Man.	P. Dev.	Physics	Total	P. Man.	P. Dev.	Physics	Total	P. Man.	P. Dev.	Physics	Total	Total	Total
100%	100%	30,0%	70,0%	0,0%	100%	30,0%	70,0%	0,0%	100%	32,5%	67,5%	0,0%	100%	15,0%	35,0%	50,0%	100%	100%	100%

Figure 1 - LAPR3 Assessment

3.1 ITP Metrics

Every sprint, until each next Monday, students should fill in a self/peer-assessment enquiry using ITP Metrics website. More information will be made available on Moodle. This works as a control point for your team and interaction between members and it's not gradable. Yet, for each sprint that students do not reply to this survey, incur in a penalty of 1 value from 0-20 on your final grade.

3.2 Project Management

This assessment is performed according to the following rubrics and focused on the used agile methodology. More information will be available on Moodle:

- Create the Sprint Backlog. (10%)
- Task splitting from User Stories. (10%)
- User Stories prioritization. (5%)
- User Stories estimation. (5%)

- Daily Meeting occurrences and registered on Jira. (10%)
- Daily Progress registered. (5%)
- Retrospective registered. (10%)
- Burndown Chart. (10%)
- Sprint Review Demonstration. (10%)
- Percentage of Sprint conclusion. (10%)
- Overall Sprint management. (15%)

At the end of each sprint, a status report document should be submitted. This document must contain the essential information about the sprint review and sprint retrospective and a brief analysis of project indicators and team performance. A template will be provided in Moodle.

3.3 Project Development

3.3.1 Code Quality Assessment (35%)

Code quality assessment is continually performed on your project. It takes into consideration the following measures and is performed automatically:

- Code Duplication. (Weight of -13)
- Technical Debt. (Weight of -13)
- JUnit Code Coverage. (Weight of 34)
- PIT Mutation Testing Coverage. (Weight of 21)

The following development requirements must be met to achieve the highest possible grading:

- JUnit Code Coverage should be above 90%.
- PIT Mutation Testing Coverage should be above 85%.
- A maximum of 5% Code Duplication is allowed.
- A maximum of 5-day Technical Debt is allowed.

The following development requirements must be met for the project to be graded:

- JUnit Code Coverage should be above 80%.
- PIT Mutation Testing Coverage should be above 65% (Sprint 2), 70% (Sprint 3) and 75% (Sprint 4).

From Sprint 2 onwards, software products that do not compile on Jenkins, have an automatic grading of 0 (zero).

From Sprint 2 onwards, software products that show up as "Failed" on SonarQube, have an automatic grading of 0 (zero).

On each day of Sprints ~~2,~~ 3 and 4, the last commit pushed to the repository must leave the project in a state that enables it to be built with "Success" on Jenkins and have the "Passed" state on SonarQube. Failure to comply with this rule, induces a team penalty of 0,20 points per day on a scale of 0 to 20 on your project final grade, up to a maximum of 3 points.

3.3.2 Commit Messages (7,5%)

Bitbucket will be connected to Jira's issue management system. Therefore, Git commit messages should include a description, a keyword, and the task/issue number according to Jira Smart Commits[2]. An example commit message is presented next, where <ISSUE_KEY> should be replaced by the issue ID on Jira: **"<ISSUE_KEY> #close User Registration [Analysis]"**

Commits must be made by valid Bitbucket users in the team. **Please be careful when committing messages, so that they are not made by a local PC user or someone else.**

GIT commit messages are rated using a [0-5] grading system:

- 5 – All commits have a well-formed commit message
- 4 – At least 90% of commits have a well-formed commit message
- 3 – About 50% of commits have a well-formed commit message
- 2 – Not Applicable
- 1 – About 25% of commits have a well-formed commit message
- 0 – Less than 25% of commits have a well-formed commit message

3.3.3 Project Report and Documentation (10%)

Project report with sections for Analysis, Design, Implementation and Review.

3.3.4 Requirements (35%)

The percentage of functional requirements done.

3.3.5 Sprint Review Demonstration (12,5%)

Sprint Review demonstration, where requirements should be presented to the client.

3.4 Physics

This corresponds to some User Stories that students must develop regarding applied physics on Sprint 4.

This assessment is performed according to the following rubrics focused on applied physics:

- Search the different types of vessels. (7,5%)

[2] https://confluence.atlassian.com/fisheye/using-smart-commits-960155400.html

- Identify the differentiating characteristics. (7,5%)
- A sketch of the vessel's in geometric figure. (7,5%)
- Choosing a reference for the center of mass. (10,0%)
- Center of mass determination. (17,5%)
- Determination of the center of mass - in a container. (10,0%)
- Considers the distribution of cargo on the vessel. (10,0%)
- Calculation of total mass and pressure exerted on water. (15,0%)
- Determination of vessel height variation above water level. (15,0%)

3.5 Plagiarism

Any attempt of copy or plagiarism (using third-party code) not explicitly mentioned in the report, will be heavily penalized and may lead to project annulment. Failure to comply with these policies and procedures will result in disciplinary action.

4 Relevant Hyperlinks

- Bitbucket
  - https://bitbucket.org/
- Jenkins
  - https://jenkins.dei.isep.ipp.pt/
    - Note: login with student number (e.g., 1010101)
- SonarQube
  - https://sonarqube.dei.isep.ipp.pt
    - Note: login with student number (e.g., 1010101)

5 Revision History

V1.0	Initial Release.
V2.0	Updates to section 2.1 and 3.2.
V3.0	Update to section 3.3.1.
V4.0	Update Figure 1 on chapter 3. Update section 3.4.

Notes: new changes are underlined while removed content are strikethrough.