Project Management Documentation

LEO-Based Assessment Tool

---

1. Introduction

This document describes the **project management approach** used during the development of the *LEO-Based Assessment Tool*. The project was carried out as part of the **Software Engineering Project (SENGPRJ)** course at FHTW.

The goal of the project management process was to ensure structured planning, transparent progress tracking, effective teamwork, and timely delivery of all required project artifacts.

---

2. Project Organization

2.1 Team Composition

The project was developed by **Group 6**, consisting of the following members:

- El Khadra Tareq
- Kovalko Markiian
- Yalcin Pinar
- Zhou Lukas

2.2 Roles & Responsibilities

Responsibilities were distributed across the team and included:

- Backend development (Spring Boot, REST API, database logic)
- Frontend development (Electron-based UI)
- API design and system integration
- Testing and debugging
- Deployment and configuration (Docker, AWS, Neon)
- Documentation and presentation

Each task was assigned to specific team members and tracked throughout the project lifecycle.

---

3. Project Management Methodology

The project followed an **agile software development approach** with iterative development cycles.

Key principles applied:

- Incremental development

- Continuous integration and testing
- Regular reviews and feedback
- Flexibility to adapt requirements

Development was organized into **iterations (sprints)**, each with clearly defined goals and deliverables.

---

4. Planning & Requirements Engineering

4.1 Requirements Definition

At the beginning of the project, functional and non-functional requirements were defined, including:

- Learning outcome–based assessment (LEO concept)
- Role-based access (Teacher / Student)
- Graph-based LEO structure with dependencies
- Cascade grading logic
- Recommendation logic for next LEOs
- Deployment and usability requirements

Requirements were formulated as **user stories** and continuously refined during the project.

4.2 Backlog Management

Project backlogs were maintained using the following tools:

- **Microsoft Azure DevOps Boards** – main product backlog and sprint backlogs
- **GitHub Projects** – implementation tracking and task visualization

Backlogs included:

- User stories
- Technical tasks
- Testing and documentation tasks

Each backlog item was prioritized and assigned to a sprint.

---

5. Sprint Planning & Execution

5.1 Sprint Planning

Before each sprint, the team:

- Selected backlog items for the sprint
- Estimated effort
- Assigned responsibilities
- Defined sprint goals

5.2 Sprint Execution

During each sprint:

- Tasks were implemented incrementally
- Code was committed regularly to GitHub
- Progress was tracked using task status updates (To Do / In Progress / Done)

---

6. Progress Tracking & Time Management

6.1 Task Tracking

Progress tracking was performed using:

- **GitHub Project Board** for implementation status
- **Azure DevOps Boards** for detailed task and backlog management

Each task was linked to concrete development activities, such as backend features, frontend views, testing, or deployment.

6.2 Time Tracking

Time spent on tasks was tracked within the project management tools. This allowed:

- Monitoring of individual and team workload
- Evaluation of effort distribution
- Reflection during sprint reviews

---

7. Reviews & Quality Assurance

7.1 Sprint Reviews

After each sprint, a review was conducted to:

- Evaluate completed features
- Demonstrate implemented functionality
- Identify issues and improvements
- Adjust the backlog for upcoming sprints

7.2 Continuous Quality Assurance

Quality was ensured through:

- Code reviews and refactoring
- Unit and integration testing
- Validation of requirements against implementation

---

8. Collaboration & Communication

Team collaboration was supported by:

- Regular team meetings
- Continuous communication during development
- Shared repositories and project boards

Clear communication and transparent task assignment contributed to effective teamwork.

---

9. Tooling

The following tools were used for project management and collaboration:

- **GitHub** – source code management and version control
- **GitHub Projects** – task tracking and implementation planning
- **Azure DevOps Boards** – backlog and sprint management
- **Docker & Docker Compose** – deployment consistency

---

10. Evidence for Project Management

Evidence for project management activities includes:

- GitHub Project Board (task assignment, progress, completion)
- Azure DevOps Boards (backlogs and sprints)
- Commit history and pull requests
- Sprint planning and review artifacts

Links:

- GitHub Project Board: https://github.com/users/piy678/projects/7
- Azure DevOps Boards: https://dev.azure.com/BWI-25WS-SEPR-Team06/LEOBasedAssessment/_sprints/backlog/LEOBasedAssessment%20Team/LEOBasedAssessment/Sprint%201

---

11. Reflection on Project Management

The agile project management approach enabled:

- Structured development despite project complexity
- Clear visibility of progress and responsibilities
- Early identification of risks and challenges

Using both GitHub Projects and Azure DevOps Boards allowed detailed planning as well as high-level progress tracking.

Overall, the project management process supported successful collaboration, timely delivery, and fulfillment of all course requirements.

Group 6 — SENGPRJ
Supervisor: *Thomas Mandl*