Software Engineering Exercise and Distance Learning Process

Goals

The goal of the exercise is to implement a limited software project and to make use of the software engineering tools (technical and organizational).

In a first step, the team assigns a role to each group participant. The role ownership means that he/she is responsible for the tasks assigned, but *not* that the task must be solved by the owner alone.

As part of the exercise, a reduced form of Scrum needs to be implemented. Scrum is intended to be used to plan the exercise and the distance learning units. The following contents are to be planned:

- User Stories
- Sprint Planning
- Sprint Review

During the exercises, there is always a stand-up at the beginning and a plan for the unit from the Scrum Master. To show the progress, a burn-down chart must be drawn.

All results should be committed into git in granular steps. This means that individual work steps can be traced and work in a team can be made independently.

Non-Functional Requirements

- Java
- git
- IntelliJ
- Scrum
- Compliance with Coding Conventions
- Java Doc

Version Control

Git

Deliverables

- When: 1. December 2023
- What:
  - Program Code
  - Configuration Management Manual
  - Design Document
  - Scrum Documents
  - Presentation Project incl. Lessons Learned (15 Minutes)
    - Including key figures for the project

References

- Java → https://openjdk.java.net/
- Scrum → http://www.scrumguides.org/

Support during the exercise

- Questions can be sent by e-mail to wolfgang.radinger-peer@edu.fh-campuswien.ac.at
- After a sprint, the Scrum Master sends a status and the sprint backlog for the next sprint

Additional non-functional requirements

Number	***Description***
NF100	The system must be developed with the help of the Scrum process model.
NF110	All files needed for development must be managed via git.
NF120	The system must be built according to the principle of 3-layer architecture.
NF130	The system must use at least two external Java libraries.
NF140	The system must use design patterns.
NF150	The team needs to set coding standards and make them verifiable.
sNF160	The Java program must be documented with the help of JavaDoc.