



**JOHANNES KEPLER
UNIVERSITY LINZ**

Institut fuer Wirtschaftsinformatik
Software Engineering

Praktikum Software Engineering

Antonio Garmendia

Unit 0 - Introduction & Preliminary Discussion

- **Introduction**
- **Grouping**
- **Evaluation**
- **Tools**
- **Task assignment for Workshop on 18.03**

- **Development of an application in a team**
 - Specify, plan and design a software product
 - Object-oriented programming and Testing (Unit tests & Code quality)
 - Work in a team
 - Application of SE tools
 - Version management (Repositories, GitHub)
 - Project management (GitHub Projects, Zenhub)
 - Build / Continuous Delivery (Maven + CircleCi)
 - Planning of the Sprints and Release Versions
 - Create of System Documentation (Architecture, Code, Test cases)

Development of a new Scientific Analyser Tool using Java and compatible libraries to develop either a desktop or a web application (e.g., Google Scholar)

A team of three developers should implement this project in several sprints over a period of 4 months creating all the necessary artifacts, such as: Software, Tests, Documentation, etc.

- Create, Read, Update and Delete (CRUD operations)
- Database Storage Solution
- Filter, Sort and Analyse
- Reports

- **High-Level Requirements**
- **Programming Language: Java**
- **Technology**
 - Backend: Java
 - Frontend: Swing, JavaFX. It is also possible the development of a web-based application. This is recommended if team members are familiar with web technologies.

- Working in teams of 3 students
- Tasks should be equally distributed considering the amount of effort
- Effort: 6 ECTS (~ 150 working hours) internship and group appointments included
- LVA-leader is your Client and Advisor
- Recommendation: Completion of the Software Engineering courses (Soft1, Soft2)



Each team member must participate in the implementation of the application – Equally distributed implementation tasks!

- **The Software Product is being developed in three releases**
 - *Release 1: April 08. 2020 (12.00 o'clock)*
 - *Release 2: May 12. 2020 (12.00 o'clock)*
 - *Release 3: June 24. 2020 (12.00 o'clock)*
 - *Final Product Delivery: July 12. 2020*
- **Submission per Release: Branch in Git with all the Documentation + Code**
- **Final Submission should be uploaded no later than 12. July 2020**

■ 3 Sprint Planning Meetings

- Mandatory attendance of the entire team
- 10 minutes presentation (Slide-Template)
- Each member should participate in the presentation
- Discussion, Status, Next Steps...

■ Three individual appointments (25.03, 29.04 and 3.06) per Team

- Feedback & Questions (30 Minutes)

	March					April					May					June					July
Datum	11/03	18/03	22/03	25/03	31/03	08/04	12/04	15/04	22/04	29/04	06/05	10/05	13/05	20/05	27/05	03/06	10/06	17/06	21/06	24/06	12/07
ToDo:	Instructions	Req. Workshop Sprint Planning 1	Sprint Planning Completed in Redmine	Project Meeting	Eastern Holidays		R1	Sprint Planning 2		Project Meeting		R2	Sprint Planning 3			Project Meeting (Code Review)			R3	Final Sprint Planning, Final Presentation	Final Release + Final Documenttion

- **Iterative development (Sprints)**
 - 1 Week to max. 1 Month
- **Prioritize a set of requirements, the Team decides which ones must be implemented in each sprint**
- **Result of a Sprint = New version of the product**
- **No dedicated roles in the team**
 - Between 5 and 9 developers per Team
- **High level of self-organization**

- **Goal: UI Prototype and OO Design**
- **Deliverables:**
 - First concept for building the application (with Features, Components,..)
 - UML Class Diagram with the most important classes (Class names, Hierarchies, Methodology, Patterns...) with a UML Tool!
 - UI Prototype
 - Continuous Integration in CircleCI
 - Presentation of the Project Status 1 (for Sprint Planning Meeting)

- **Goal: Prototype Implementation and Unit Tests**
- **Deliverables:**
 - Extended/updated UML Diagrams
 - Prototype Implementation:
 - First version of the User Interface
 - Some implemented functionality
 - Unit Tests for individual (important) classes
 - Use Case Description (see Use Case Template)
 - Presentation of the Project Status 2 (for Sprint Planning Meeting)

- **Goal: Documentation & Code Quality**
- **Deliverables:**
 - Extended/updated UML Diagrams
 - Extended Unit Tests
 - Implementation:
 - User Interfaces
 - Implemented most of the functionalities (all Features availables)
 - Code Quality Analysis with PMD, Findbugs, etc.
 - First version of the project documentation
 - Presentation of the Project Status 3 (for Sprint Planning Meeting)
 - Live Demo/Screencast of the Application

- **Deliverables:**
 - Final Project documentation
 - Executable, final version of the application
 - Github Documentation (Readme with Installation Instructions, etc.)
 - Javadoc for important classes, Interfaces and Methods

- **The criteria for assessment as follows:**
 - Functionality of the product
 - External Quality of the Product (Stability, Efficiency, User Interface)
 - Internal Quality of the Product (Quality of the design, Programming Quality, API-Documentation)
 - Widespread Unit Tests and Quality of the Unit Tests
 - Quality of the Documentation (Design, Test cases, Experience Report)
 - Presentations

- **Github Projects, ZenHub**
- **Git (GitHub)**
- **Maven**
- **CircleCI**
- **UML Editor / UI Prototyping Tool**
- **Code Quality: Static Code Analyzer**

- **Implementation details (detailed specification) in Github Projects**
 - For each release: Requirements, Tasks, Bugs, etc.
 - Assign to each task a responsible and a cost in time! – The responsible must implement the source code (Code + Unit Tests)
- **Create a Release Planning (Roadmap) in Github projects**
 - At the end of each release, the respective tasks, requirements, bugs, etc must be completed and closed.

- **GitHub to manage Code and Documentation**
 - Code must be committed in Github at least 1 per Week
 - Always enter the respective id for each commit (#TaskNr). – Each team member must write some code and make commits!
- **Quality feedback – The source code must be kept clean**
- **Document the problems that are not be fixed accordingly**

The submission for each release must be committed in a separate Github branch

- **Documentation, Tutorials, Links....**

<https://github.com/jku-win-se/teaching.ss21.prse.prwiki.en>

- **Now:**

- Build teams of 3 Students - 1 “Team Leader” Email to antonio.garmendia@jku.at [Subject: PR_SE2021 Team] (Name, Matr.Nr, email, GitHub user)
- Distribution of topics for the Workshop

- **For Next week (18.3.2020):**

- Get familiar with the requirements and prepare questions for the Workshop
- Plan the first version of the product and define the initial responsibilities for each member
- Get familiar with GIT, Maven, Github Projects...

- **By 25.03.2020: Complete planning for Release 1 in Github Projects**

- **Topic-1: Git**
 - Git Functions and Markdown
 - Git in Eclipse
 - Tutorial: <https://rogerdudler.github.io/git-guide/index.de.html>
- **Topic-2: Github Projects, Zenhub**
- **Topic-3: Maven + CircleCI**
- **Topic-4: UML Tools / Editors**
- **Topic-5: UI Prototyping + Tools**