

# IE04 Integrated Exercise for Software II

Introduction

R. Yoshioka, P. Fayolle, M. Mozgovoy, U. K. Rage

# Course Description

- Integrated Exercise for Software II
- Duration : 10/4<sup>th</sup> ~ 1/31<sup>st</sup> (2<sup>nd</sup> semester)
- Objective:  
Obtain knowledge & skills to develop practical software
  - Plan and design software based on requirements
  - Perform collaborative software development
  - Make use of knowledge & skills of other courses

# Student Learning Goals

1. Understand **software requirements** and define corresponding **models**
2. Create a **software design** that satisfies **functional and quality requirements**
3. Setup a suitable **development environment** and effectively **manage development** activities
4. **Design sufficient tests** and **verify** software
5. **Validate** software by checking that it meets the needs of **stakeholders**

# So, what will we do?

- What will you develop
  - You will receive an RFP (Request For Proposals) of a software
  - You will propose and design software that satisfies the RFP
  - Your software will have a graphical user-interface and use a database
  - You may decide the programming language and frameworks
- How will you work
  - You will work in teams of 4-5 students each
  - You will develop the software incrementally in phases
  - You will validate and verify your software
  - You will present your software in the final class

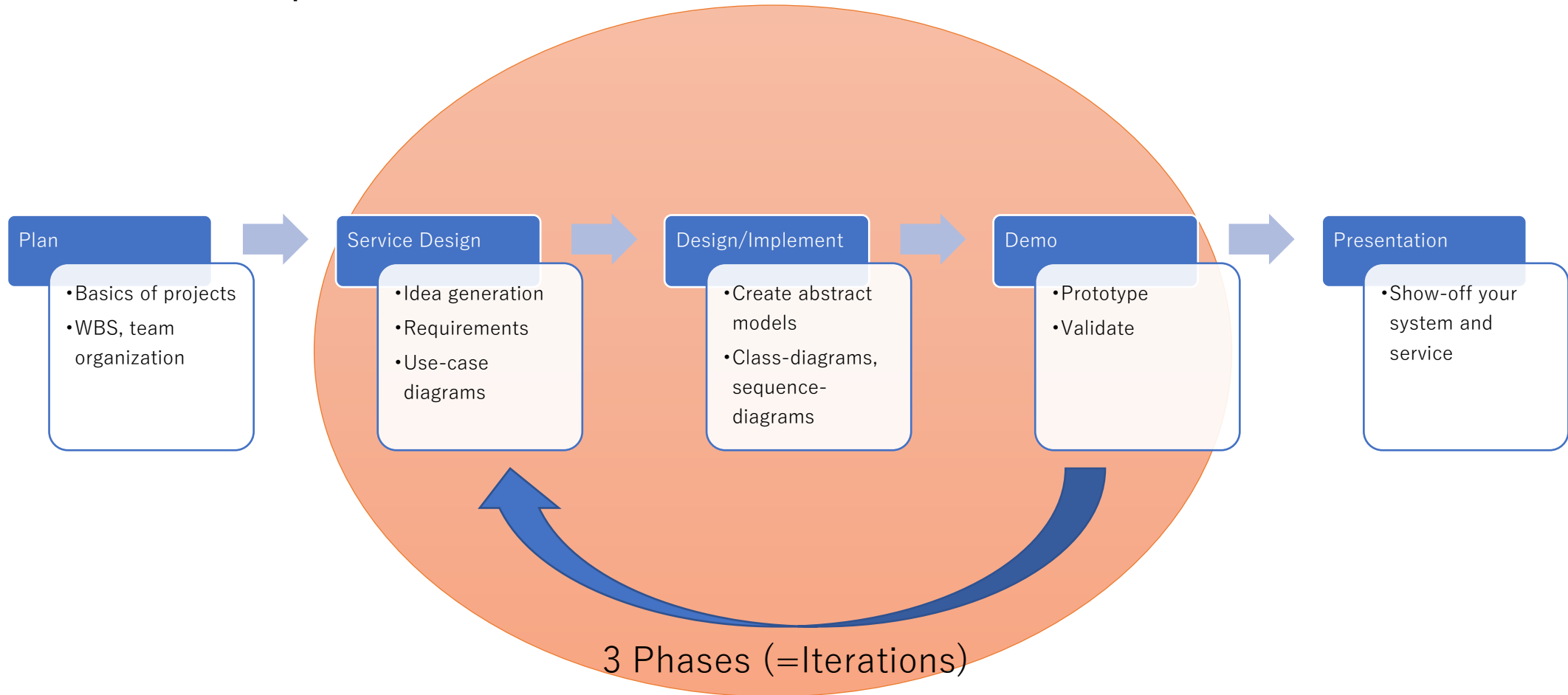
# Expected Output

- A software system that satisfies given requirements
  - Proposed service (user-stories)
- Description of service
  - Target users
  - Location of usage
  - User-stories (as use-cases)
- Output methods
  - Software system
  - Project documentation
  - Presentation

RFP: Project Theme

**“TA Report System”**

# Development Process



# Master Schedule

Week	Date	Phase	Activity	Deliverables (to submit)	Classroom
1	10/4	Plan	Team formation, understand RFP	Team profile	<b>Common</b>
2	10/11	Req. Def.	Requirements & use-case analysis	Product backlog	Separate
3	10/18	Phase 1	Requirements definition, use-case analysis	Phase backlog, Use-case diagram, Class-diagram (model)	Separate
4	10/25	Phase 1	Design, implementation	Use-case diagram (description), Class-diagram (model), UI definition	Separate
5	11/8	Phase 1	<b>【Phase demo】</b>	Class-diagram, sequence-diagram, program	<b>Common</b>
6	11/15	Phase 2	Revise & update requirements and use-cases	Phase backlog	Separate
7	11/22	Phase 2	Design, implementation	Use-case diagram (description), Class-diagram (model), UI definition	Separate
8	12/6	Phase 2	<b>【Phase demo】</b>	Class-diagram, sequence-diagram, program	<b>Common</b>
9	12/13	Phase 3	Revise & update requirements and use-cases	Phase backlog	Separate
10	12/20	Phase 3	Design, implementation	Use-case diagram (description), Class-diagram (model), UI definition	Separate
11	1/10	Phase 3	<b>【Phase demo】</b>	Class-diagram, sequence-diagram, program	<b>Common</b>
12	1/17	Test	Validate and verify	Test-plan (including test-cases)	Separate
13	1/24	Test	Final improvement & prepare for presentation	Test report, issue-list	Separate
14	1/31	Finish	<b>【Final presentation】</b>	Final report (presentation slides), program	<b>Common</b>



# Progress Report and Demos

- Objective
  - Teachers will provide feedback on the contents and guidance on project activities
- Progress report
  - Each team will give a 5 min. progress report in each class
  - The report will be a brief presentation using slides
  - The report should cover contents in the template
- Phase demo
  - Each team will give a short demo at the end of each phase
  - The demo must be based on working program (combination with UI-mockups acceptable)
  - The demo will be given together with the progress report

# Notes on Required Deliverables

- All deliverables assume formats used in [FU14 Introduction to Software Engineering](#) course
  - Reasonable adjustments/improvements are acceptable
  - All deliverables must be shared on designated repository on GitHub
- Use-case diagram(s) is required
- Class diagram of model-classes (as in the MVC model) is required
- Sequence diagram(s)
  - At least one sequence diagram is required: select an appropriate use-case
- Software:
  - All code produced for the project is required to be shared on GitHub
  - Working code must be [pushed to GitHub by Phase 1 Demo](#), and [incrementally improved continuously throughout the course](#)
- Product backlog
  - Describes the values your software provides to users
  - It is a list of “user stories”
    - Describe a function in terms of user’s values (benefits)
    - Who is it for? What is it for? What does it do?
- Phase backlog
  - A task-list for the phase
  - Select user-stories from the product backlog

# Evaluation

1. Presentation at each review (phase demo and final) 25%
  2. Software product (documents and software) 40%
  3. Final exam (individual interview) 25%
  4. Participation in class activities and attitude 10%
- Assessment of software product
    - Degree of fulfillment of requirements (number of functions and their quality), performance, creativity, and innovation.
  - Assessment of student contribution
    - Explanation provided at each phase demo
    - Development records (documents & repository).
  - Final examination
    - Students will be interviewed during their final presentation (QA session)

# Summary

- Experience the development of practical software
- Practice cooperative activities in teams
  - Team members should **inform, consult, and assert** each other
- Let's be creative!
  - Enjoy making people happy with your skills and abilities

# Course Web-site, etc.

- Web-site
  - <https://ie04-aizu-2023.github.io/>
- GitHub repository
  - <https://github.com/ie04-aizu-2023>
  - You must login to view private repositories

# Tasks for today

- Regroup with your team members (the list will be provided to you)
- Join GitHub classroom
- Input your team profile in the README.md at the top level of GitHub
- Read the RFP
- Prepare for next-week's first progress report

# Team Profile

- Team name
- List of members
  - Student ID, name
- The type of software you will develop:
  - Desktop application
  - Web application
  - Smartphone application
  - Others: please describe
- Programming language & framework