Software Engineering II

Course Information

The Software Engineering (Part II) builds on the material introduced in SENG201 (Introduction to Software Engineering) and is intended as a companion course to SENG302 (Software Engineering Group Project).

In this paper, you will learn about software processes, including Agile Software Development (e.g., Scrum, Kanban), effective source code and documentation management, resilience engineering, acceptance testing, software metrics and software design principles.

At the end of each lecture it is expected to finish the prep slides before the next lecture

Lecture Material

Resources

Assignment Submissions

Course Outline   Lectures outline

Term 1 - Agile Software Development

- 23/02 Recap on software engineering methods
- 26/02 Scrum 101
- 02/03 (Agile) Requirement analysis
- 05/03 Scrum team management
- 09/03 Continuous integration
- 12/03 Continuous delivery
- 16/03 Testing and mocking
- 19/03 Resilience & reliability
- 23/03 Software quality metrics
- 26/03 Software architecture 101
- 30/03 User Experience 101

Term 2 - Object-oriented design principles

- 27-30/04 Generics & Collections
- 04-07/05 Object-oriented design
- 11-14/05 Design pattern I
- 18-21/05 Design pattern II
- 25-28/05 Design by contract
- 01-04/06 TBD

Lab/tutorial outline

Term 1

- 01-03/03 Tutorial - Scrum (1)
- 08-10/03 Tutorial - Scrum (2) [Assignment 1]
- 15-17/03 Conceptual modelling and `JPA`
- 22-24/03 Acceptance testing with Cucumber
- 29-31/03 `Facking` and mocking with `Mockito`

Term 2 (Provisional, please check forum for latest updates)

- 03-05/05 Generics & Collections
- 10-12/05 Design
- 17-19/05 Design Patterns I
- 24-26/05 Design Patterns II
- 31-02/06 Design by Contract

Assignments deadlines

- 08-10/03 Assignment 1 - Scrum tutorial 2 (during lab time), self-enrol via group forming page
- 01/04 Assignment 2 - Reflection report
- 04/06 Assignment 3 - Acceptance testing and design patterns

Grades

- Assignment 1: Scrum tutorial 15%
- Assignment 2: Reflection report 5%
- Assignment 3: Coding assignment 20%
- Final Exam: 60%

Labs

Lectures

Lecture One

SCRUM

Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest value. Scrum is a lightweight framework that helps people and teams to generate value through adaptive solutions for complex problems, scrum was created by `Ken Schwaber` and `Jeff Sutherland`.

Scrum is a method of generating agile development within a workplace/team, it does this by using adaptive planning and works to achieve early delivery of the product in order to assess and improve the work flow within a development team.

Here are the lead roles in a `SCRUM` environment workplace.

1. A Product Owner orders the work for a complex problem into a Product Backlog.
2. The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
3. The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.
4. Repeat

Here is an Overview of Scrum from Scrum.org.

Lecture Two

The Five Values Of Scrum

- Openness
- Focus
- Respect
- Courage
- Commitment

Initial startup for scrum

Some bootstrap effort is needed before starting

- start from a vision of the product
- refine its objectives
- discuss with the stakeholders, including the Product Owner, create an initial backlog with user stories
- agree on working mode and standards

`Scrum` and `Kanban` are quite different here

- `scrum` larger initial phase is needed before planning the first sprint
- `kanban` task-oriented, so faster start-up

Part I - what are we going to do?

- the PO presents highest priority stories or epics
- estimate the complexity of each story
- use previous velocity (i.e. number of points delivered previously)
- first draft of sprint backlog
- a sprint should be as coherent as possible

Part II - how are we going to do it?

- selected stories are broken down into tasks by the team
- tasks are described thouroughly (i.e. anyone can pick it up)
- estimations of tasks are set collaboratively, i.e. reaching a consensus
- the team knows the steps to implement each story
- the team knows what each task needs to produce

Lecture Three

Scrum Roles

Product Owner - translates user demands into stories - maintains and prioritises the list of things to do - negotiates content of releases and timing with the team

Scrum master - acts as a coach - facilitates communication inside and outside the team - represents managements, but protects team members

Team aka developers - everyone is a developer - everyone works on everything

Epic: - large piece of work that may be implemented over many scrums

Story: - a well defined piece of work - confined within a sprint (unless too much is allocated)

Product backlog: - contains tasks (stories) - everything that still needs to be done - maintained by PO - some estimations may be missing

Sprint backlog: - items that will be handled within our current sprint - allocated from product backlog and estimated

Lecture Four

Dealing with Unplanned - Distinguish between development issues and bugs * issue: a problem identified at the latest before a review * bug: a problem discovered at the earliest during later regression tests

Dealing with impediments - Long meetings: stick to essential stakeholders and keep them time-boxed - Illness: unavoidable, but refrain from ignoring it - broken builds: implicit top priority for the whole team to fix it - tools: always frustrating, but you need to fix it first - third parties: even more frustrating, consider alternatives - scope creep: pay special attention to review the stories and broken down tasks - unreliable PO: learn how to deal with them - team problems: retrospectives and seek for assistance - external incentives: remember Scrum is about teamwork