

CS3203: Software Engineering Project

Introduction

LN - INTRO



School of
Computing

Teaching Team



Dr. Bimlesh Wadhwa
bimlesh@nus.edu.sg



Dr. Martin Henz
henz@comp.nus.edu.sg



Tzer Bin



Yahui

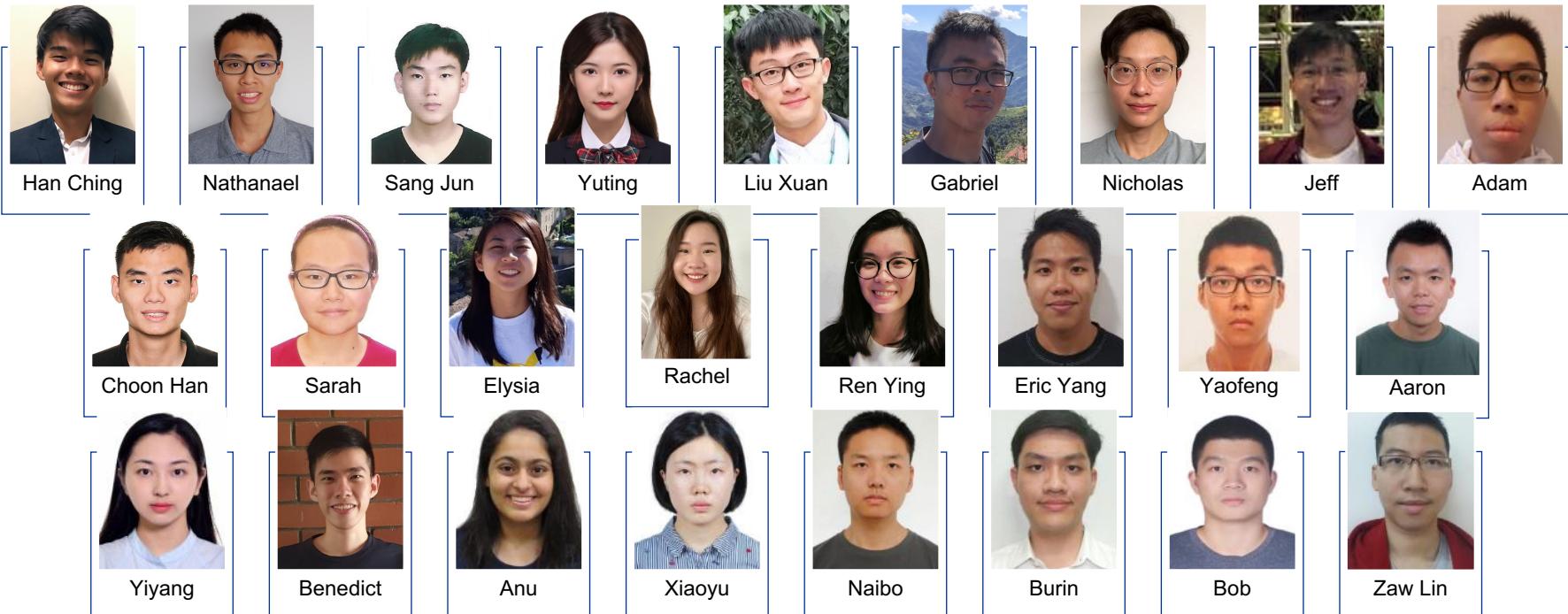


Bangjie



Eric Han

Teaching Team (Tutors)



Why software project courses?



Development in industry vs. course assignments



Bigger project size

Architecture, practice design strategies
Work in teams
Project planning: development phases, task relevance / priority



Functional and quality requirements

Functional application
Performance, maintainability



Quality documentation



Work with deadlines

CS3203 vs. other project-based courses

CS3203 – Software Engineering Project

- Specified requirements and tools, Application of SE practices,
- Build and test an application

CS3216/CS3217 – product development

- Exposure to software development specific to a platform

CS3281/CS3282 – Thematic Project Course

- Enhancements of an in-use brownfield project

ATAP, SIP, iLead and NOC

- Industry and out-of-classroom experience

UROP and FYP

- Research experience, Individual project

CS3203 vs. earlier courses you took

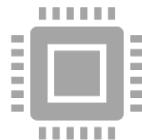
| Data Structures CS2030, CS2040 | SE course CS2103 | SE course CS3219 | SE project CS3203 |
|--|--|--|---|
| Learn about tree, graph, queue, hash table, ... | - | - | Work with many interrelated trees, graphs, etc. |
| Mini scale | Small scale | Medium Scale | Medium scale |
| - | Class-level decomposition | Component-level decomposition | Component-level decomposition |
| Interface operations and algorithms for tree, graph, .. | Class interfaces & information hiding Public vs. private | Architecture design; Component interfaces & information-hiding | Component interfaces & information-hiding |

Module focus



Teamwork

Planning, communication, and cooperation



SE practices

Development Process
Decomposition and component interfaces (API)
Design and Algorithmic complexity
Comprehensive testing



High quality Technical documentation

Module Schedule

- 7 Lectures: Week 1-5 & 7-8
Learn about project requirements, software design and software engineering practices
- 10 Consultations: Weeks 3-12
Meet with tutors to update progress and get comments/suggestions
- 2 Tests: Weeks 5 and 11
Assess requirement analysis and design
- 3 Project iterations: Recess Week, Week 10 and Week 13
Go through iterative development of the project (+ final project presentation in Reading Week)

Module Tools

LumiNUS

Announcements,
schedule, course
materials,
quizzes, submissions,
team formation, tests ...

GITHUB Wiki

<https://github.com/nus-cs3203/project-wiki/wiki>
Project requirements ...

Zoom

Lectures, consultations,
presentations ...

Microsoft Teams

Communication, discussion,
group formation ...

Send queries to Microsoft
Teams / Teaching team

Module Grading

| | | |
|-----|-----|--|
| 35% | 7% | Lecture(Weekly Quizzes + participation) |
| | 20% | Consultations(participation + oral presentations) |
| | 8% | Test1~3% Test2~5% |
| 65% | 15% | Iteration 1 – BasicSPA(code) + Report-draft |
| | 5% | Iteration 2 – Report + SPAProgress/Demo |
| | 45% | Iteration 3 – AdvSPA + Extension + Report + Presentation |

Individual penalty/bonus based on peer review and tutor observations.

Module Project

- SPA - Static Program Analyser
 - BasicSPA
 - AdvSPA
 - Extensions
- Your project should compile with standard C++ 11 compiler (VS 2019) with library (STL)
- Recommended tools:
 - Visual Studio 2019
 - » IDE / testing tool / debugger / automated build tool

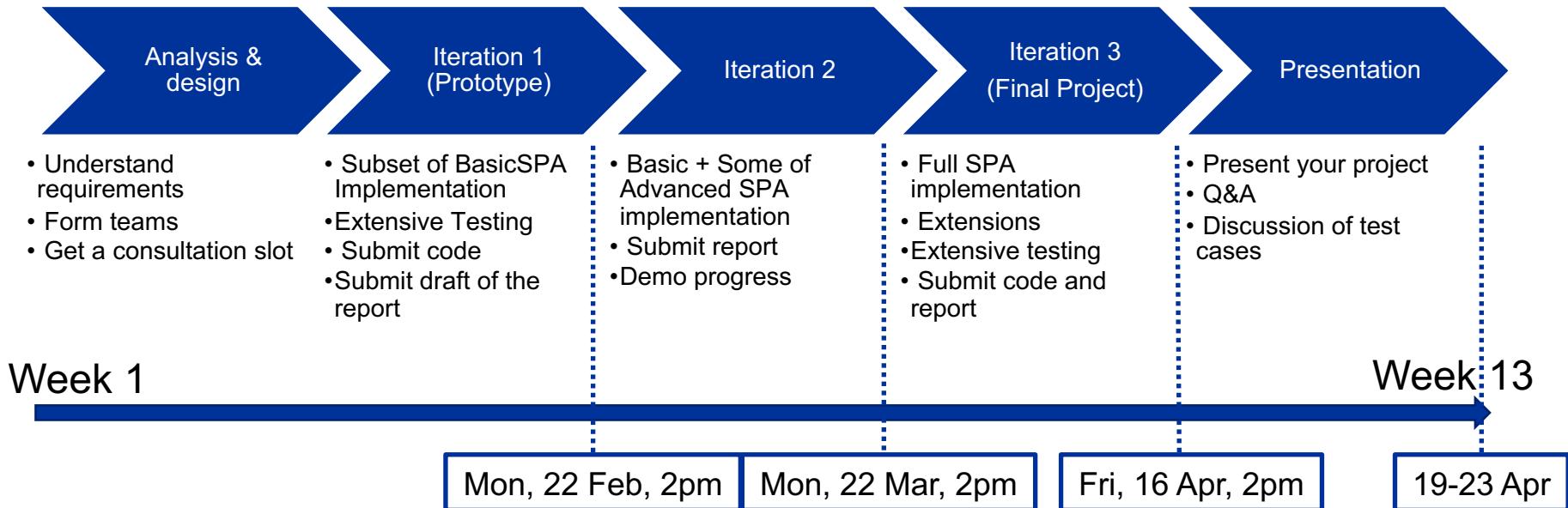
Module Project

- 2 possible ways to get started
 - **Windows Startup Solution:** For Windows-only teams (including Windows installed via bootcamp / VM); easy to get started
 - **Cross Platform Startup Solution:** For teams with at least one member not using Windows at all; harder to get started
 - The whole team must use the same startup solution.
 - More details will be released by the end of Week 1 and there will be a tools lab in Week 2.

Project Team

- Six students per team
- Step 1: choose some team members
 - Form a group of 2-3 people
 - Register your group on LumiNUS under Class & Groups -> Class Group -> Project Team Formation by Mon, 18 Jan, 2pm
- Step 2: randomly get some team members
 - Groups (and anyone without a group) will be randomly paired to form teams of six
 - Teams will be announced by Mon, 18 Jan, 6pm
 - Let us know if you have any issue in working with students in your team by the end of Tue, 19 Jan

Project Iterations



Project Consultations

- Meet with tutors to update progress and get comments/ suggestions
 - 10 Sessions: Weeks 3 to 12
 - Time slots:
Tue 10am-12pm, 12-2pm, 2-4pm, 4-6pm
Wed 10am-12pm, 12-2pm, 2-4pm, 4-6pm
-
- * Consultation with tutor - 1hr
 - * Team meeting – 1 hr

Oral Presentations

Three individual presentations (~ 10 mins each)

1. Problem domain and Development Process

BasicSPA requirements, SIMPLE, PQL, SPA architecture,
Development Process, Development Environment

2. Design of the (own)component developed

Data structure/algorithm, testing, its interaction with other SPA
components

3. Design of a component developed by another team member

Data structure/algorithm, testing, its interaction with other SPA
components

Lectures

- Week 1-5, Week 7-8
 - Project requirements and SE practices
- **Blended Learning** mode from Week 2 onwards
 - Pre-lecture videos
 - In-class activities (2 hours): **Tue 10am-12nn OR 2pm-4pm**
 - » Group work and presentations
 - » Allocation of time slots for Blended Learning lectures to be announced by **Mon, 18 Jan, 6pm**
- **Post lecture quizzes** due every Mon 2pm after a lecture
 - **Week 2-6, 8-9 Mon 2pm**

Why Blended Learning?

- More interaction, more guidance
- Get exposure to more examples in software engineering
- Gain experience in solving interesting practical problems
- Develop team skills
- Develop presentation skills

Tests

- Test 1
 - Project requirements
 - Take-home, due in Week 5 (**Mon, 8 Feb, 2pm**)
 - Open-book
 - Work and submit individually or as a group of two

Tests

- Test 2
 - Project requirements and software design
 - Online test, Week 11 during lecture hours (**Tue 30 Mar, 10-12 and/or 2-4pm**)
 - Open-book
 - Partly individual, and partly group-test

Important Dates / Deadlines

| Week/Day | Date/Time | Quizzes | Tests | Project | Presentations |
|-----------------|-------------------|-------------------------|--------------------|-----------------------------------|----------------------------|
| Week 2 Mon | 18 Jan 2pm | | | Project Group / Tools Lab Sign-up | |
| Week 3 Mon | 25 Jan 2pm | Week 1 & 2 Lecture Quiz | | | |
| Week 4 Mon | 1 Feb 2pm | Week 3 Lecture Quiz | | | |
| Week 3/4 | - | | | | Oral Presentation 1 |
| Week 5 Mon | 8 Feb 2pm | Week 4 Lecture Quiz | Test 1 (Take-home) | | |
| Week 6 Mon | 15 Feb 2pm | Week 5 Lecture Quiz | | | |
| Recess Mon | 22 Feb 2pm | | | Iteration 1 (Code) | |
| Week 7 Tue/Wed | Consultation Slot | | | Iteration 1 (Report Draft) | |
| Week 8 Mon | 8 Mar 2pm | Week 7 Lecture Quiz | | | |
| Week 7/8 | - | | | | Oral Presentation 2 |
| Week 9 Mon | 15 Mar 2pm | Week 8 Lecture Quiz | | | |
| Week 10 Mon | 22 Mar 2pm | | | Iteration 2 (Report) | |
| Week 10 Tue/Wed | 23/24 Mar | | | Iteration 2 (Demo) | |
| Week 11 Tue | 30 Mar 10am/2pm | | Test 2 (Online) | | |
| Week 10/11 | - | | | | Oral Presentation 3 |
| Week 13 Fri | 16 Apr 2pm | | | Iteration 3 (Code + Report) | |
| Reading Week | - | | | | Final Project Presentation |

Average Weekly Workload

| | Listed (Actual) |
|------------------------------------|-----------------|
| No. of hours for lectures | 4 (2) |
| No. of hours for consultations | 2 (1) |
| No. of hours for project and tests | 10 (14) |
| No. of hours for preparatory work | 6 (3) |
| Total | 20 |

Past Year Handbook

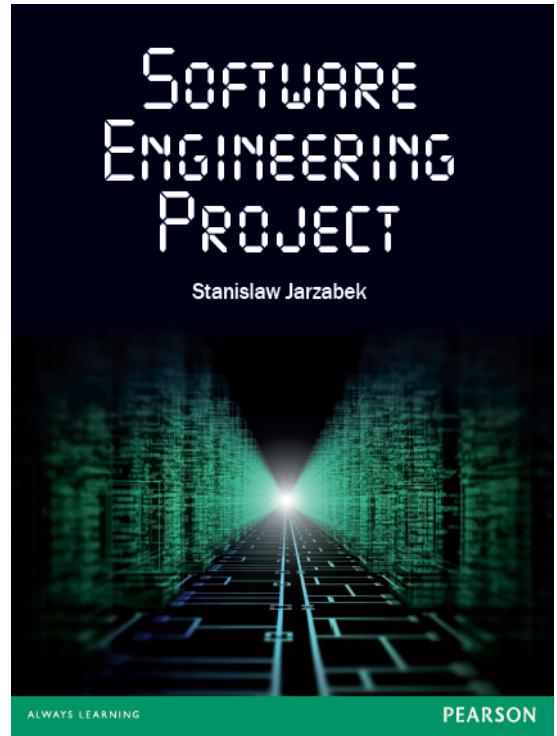
Software Engineering Project

Author: Stan Jarzabek

Publisher: Pearson

ISBN: 9789810692353

- Available at NUS Co-op @ Forum*
- Not mandatory and not recommended to purchase
- Refer to GITHUB Wiki instead



Other References

- [Software Engineering – A Practitioner’s Approach](#) by R. Pressman
8th edition, McGraw Hill, ISBN 978-0078022128
- [C++ How to Program](#) by Harvey & Paul Deitel,
9th edition, Pearson, ISBN 0133378713
- [The Mythical Man-Month](#) by Frederick Brooks,
2nd edition, Addison-Wesley, ISBN 0-201-83595-9

Past Year Module Comments

- Really forces you to work in a large team and complete a large project from scratch, very good experience.
- ...Also brings about many different aspect of computer science together such as programming languages, database implementation, testing and obviously software engineering.
- Learning project management and teamwork in a SDLC.
- I think the project aspect where we work as a team to come up with a working software in the end.

Past Year Module Comments

- Test 2 – section 1 has no purpose.
- There should be no reason to test it since we have been through the project and it does not add value to any CS understanding in the future
- I hope Test2 can have more group components.
- With a team of 6, I only got 4.5 teammates working with me. One of them is a ghost in this team, and he is so cheeky that he don't even see himself that way. While people are working so hard, he all the way stood aside.
- Everything is graded as a group, so group members that aren't working hard affect the group members that are working hard.
- workload is very heavy :(

Tips

- Make effort in achieving good teamwork!
- Spend sufficient time every week working on the project.
- Do not leave things to the last minute.
- Approach the teaching team whenever you encounter any difficulties.
- Focus on the whole development process, not just the implementation.
- Start testing early and test extensively.

The module could be more rewarding than you think.