THE UNIVERSITY OF
MELBOURNE

SWEN90014
Masters Software Engineering Project

ASSESSMENTS & PRESENTATIONS

Coordinators: Philip Dart and Patanamon Thongtanunam

Overview

- Schedule
- Letter of Agreement
- Assessment Details
- Team presentations:
  - *Design Concepts + Architecture (Executable & 4+1 view model)*

Schedule Overview

Date	Milestone	Collaboration
Sun 4 Aug	*Resolved Software Requirements (proposed)*	*Team x (2 or 3)*
Sun 11 Aug	*Design Concept, incl. reuse plan*	*Team*
Sun 1 Sep	Sprint 1 (incl. Executable Architecture)	Team
Sun 22 Sep	Sprint 2	Team
Sub 23 Sep	**Sprint 3 started**	**Team**
Non-teaching period: Mon 30 Sep – Sun 6 Oct		
Sun 20 Oct	**Sprint 3**	**Team**
Mon 21–Fri 25 Oct	Acceptance	Team
Sun 27 Oct	Team self-assessment and lessons learned Reports	Team/individual
Swot-vac: Mon 28 Oct-Fri 1 Nov		

Schedule Detail

Date	Milestone	Collaboration
	Resolved Requirements (baseline)	*Team x (2 or 3)*
	Design Concept and Reuse Plan	*Team*
12 Aug – 1 Sep	*Sprint 1*	*Team*
2 Sep – 22 Sep	Sprint 2	Team
23 Sep (Week 1 of Sprint 3)	**Sprint 2 Review & Retrospective:** **- Demo to Client and record feedback** **- Review your Sprint 2 (What's good or need an improvement)** **Sprint 3 Planning:** **- After the client meeting, Adjust backlog and revise sprint 3 scope**	**Team and Client**

Letter of Agreement

- All student projects are covered by IP agreements
- A legal document is needed for the projects that have clients outside from UoM
- The Letter of Agreement concerns include:
  - University's obligations and responsibilities
  - Industry Partner's obligations and responsibilities
  - Ownership of intellectual property
  - Confidentiality

Letter of Agreement: Example

THE UNIVERSITY OF MELBOURNE

Industry Software Project Letter Agreement ('Letter Agreement')

Hellen *Client*

Client's organisation

Date: 26/09/2019

RE: ***Course/Subject: SWEN90014: Masters Software Engineering Project***

Dear Hellen,

Thank you for agreeing on behalf of *Client's organisation* (**Industry Partner**) to be a partner and provide a project brief for the subject "***SWEN90014: Masters Software Engineering Project***" (**Subject**) run by the Melbourne School of Engineering at the University of Melbourne (**University**). The opportunity for students to gain relevant professional training and skills through exposure to an industry setting relating to their area of study is invaluable.

The purpose of this Letter Agreement is to set out the obligations of both the University and the Industry Partner in relation to the industry-based project described in brief in Item 1 of the Schedule to this Letter Agreement

Letter of Agreement: Process

- *Team supervisor will* prepare LoA
- *Team supervisor will collect the signatures of* ***clients*** and ***students*** in the project
- Your (students) tasks:
  - Read the whole LoA document (11 pages)!
  - Write your name on the student deed poll
  - Sign the student deed poll
  - Return the signed documents to the supervisor
- If the clients are from UOA, you don't need to sign the documents

Student deed polls (Example)

Page 1 of a deed poll

STUDENT ASSIGNMENT AND CONFIDENTIAL INFORMATION DEED POLL

THIS DEED POLL is made on the day of 20

Name	Your details	**(You** or **Your)**
Address		
Telephone		

Page 3 of a deed poll

EXECUTED AS A DEED POLL

SIGNED SEALED AND DELIVERED BY)
)

Your signature

.....

Signature of (student)

Your teammate's or supervisor's signature

.....

Your teammate's or supervisor's name

.....

Print Name of Witness

Assessment: Overview

- Analysis and process-related documentation, 8K-10K words, (60%)
  - To be delivered to supervisor for feedback via the prescribed toolset as per the schedule
- Final release, 2K-3K LOC (30%)
  - To be maintained under version control and delivered to the Client at Acceptance.
- Individual report, not exceeding 2K words (10%)
  - Outlining individual contribution and lessons-learned

Assessment: Submission

- **Analysis and process-related documentation, 8K-10K words, (60%)**
  - All documentation in Confluence, JIRA, BitBucket will be assessed
  - Other supplementary materials *may* be submitted through LMS or via email if documentations are recorded outside of the standard tools (the provided Atlassian toolset)
  - No additional report needed!
- ***Deadline***: By Week 12 (28 October)

Assessment: Submission

- **Final release, 2K-3K LOC (30%)**
  - Source code recorded in BitBucket
  - The development history will be also checked
  - Inform your supervisor if source code and the development history are not stored in BitBucket provided by the university
  - *Deadline*: By Week 12 (28 October)
- **Individual report, not exceeding 2K words (10%)**
  - Details and submission are in LMS
  - *Deadline*: By Week 12 (28 October)

Assessment: Our expectations

- Engineering Ability
  - e.g., Problem Analysis and Solution, Design Synthesis, Analysis, and Specification, Implementation, Technical Creative Input, Evidence Gathering and Data
- Management
  - e.g., Processes, Task decomposition and planning, risk, professionalism, Quality Assurance
- Teamwork
  - e.g., Communication, Maturity, Attending to tasks
- Product
  - E.g., well documented, well designed, validated & verified, easy to deploy, etc.

Assessment: Our expectations

- Engineering Ability
  - e.g., Problem Analysis and Solution, Design Synthesis, Analysis, and Specification, Implementation, Technical Creative Input, Evidence Gathering and Data

All the details are provided in the LMS
Go to Subject Information > Assessment Guide

professionalism, Quality Assurance

- Teamwork
  - e.g., Communication, Maturity, Attending to tasks
- Product
  - E.g., well documented, well designed, validated & verified, easy to deploy, etc.

Assessment: Presentations

- **Architecture Review Session**
  - Week 11 (14 - 18 October): In the workshop or meeting hours (as per supervisor)
  - Present the architecture of your system and the source code to the supervisor and 2nd examiner
- **Demo & Acceptance Session**
  - Week 12 (21 – 25 October): To be scheduled by team, supervisor, and clients
  - Present your runnable system along with the acceptance narrative to the supervisor, 2nd examiner, and the clients.

Architecture Review Sessions

- Every team must have a presentation
- Closed sessions (i.e. presenting team only)
- Approximated time is 25 minutes
- Attendees:
  - All team members
  - Team supervisor
  - 2nd Examiner
  - Client attendance is *not* *expected*
- Main Focus: **All technical aspects**

Architecture Review Sessions

- Team supervisor will run the session
- The session content includes:
  1. Overview of architecture
     - incl. identifying incomplete elements
  2. Outline how Client can deploy your system
     - What are you delivering (form, content)?
     - What action needs to be taken?
  3. Questions regarding proposed changes
     - Illustrate response using architecture
     - Show relevant parts of code

Demo & Acceptance Sessions

- Every team must have a presentation
- Closed sessions (i.e. presenting team only)
- Approximated time is 50 minutes
- Attendees:
  - All team members
  - Team supervisor
  - 2nd Examiner
  - Client or rep. attendance *is* *strongly preferred*
- Main Focus: **Use cases (mostly non-technical aspects)**

Demo & Acceptance Sessions

- You (students) run the session
- The session content should include:
  1. Introduction
  2. Final Scope of your project
     - Delivered vs Not Delivered
     - Committed to vs Not Committed to
  3. Quality Assurance
     - Briefly describe why Client can be confident in your system that it is fit-for-purpose

Demo & Acceptance Sessions

- (Continue) The session content should include:
  4. Testing Report
     - Summary statistics: Categories and outcomes
     - Outstanding issues
  5. Demonstration
     - Team representative performs a *live* demonstration (**playing a video record is not acceptable**)
     - Team must record a video the live demo during the session for a submission

Demo & Acceptance Sessions

- (Continue) The session content should include:
  6. Delivery
     - Revisit outstanding issues (including from demo)
     - Handover (if the client accepted) or agree on handover arrangements
  7. Client provide feedback
  8. All outstanding issues and client feedback should be recorded in Confluence

Demo Guidance

- Demo capability (what users can do on the system), not functionality (what can the system do)
- Use realistic examples (based on user stories)
- Talk through what is happening during the demo
- Ensure you can set/reset easily for demos
- Pre-create elements to avoid delays
- Be prepared
  - Avoid issues if at all possible
  - Pre-announce if they can't be avoided
- **Don't argue** with the Client: just record

Demo Guidance

- Demo capability (what users can do on the system), not functionality (what can the system do)
- Use realistic examples (based on user stories)

All the details are provided in the LMS
Go to Resource > Preparing for presentation

- Pre-create elements to avoid delays
- Be prepared
  - Avoid issues if at all possible
  - Pre-announce if they can't be avoided
- **Don't argue** with the Client: just record

Lecture Identification

Coordinators: Philip Dart and Patanamon Thongtanunam

Semester: S2 2019