



IS212 (AY 2024-25, T1) – Scrum Project Instructions

Executive Summary

Overall Goal	<i>Deliver a quality software increment for a customer using scrum</i>
Weightage	35% of all course assessments
Deliverables & Due Dates	
Scrum Process Consultation (ungraded)	<p>During the Week 9 class:</p> <ul style="list-style-type: none"> Each team will have an informal (ungraded) conversation with their instructors about how they are conducting their sprints We recommend that teams start sprinting in Week 5 to get their 'hands dirty'. Figure out what's working and what's not, then use the Week 9 conversation to improve your process for the final sprint(s)
Software Release & Scrum Deliverables (graded)	<p>By 23:59 on Friday Week 12 (all sections):</p> <ul style="list-style-type: none"> You are to submit a working software release based on the customer briefing that (i) implements the seven core features and (ii) was built by your team by following a scrum process Submit all evidence of your scrum process and all code, including recordings of your final sprint meetings Submissions <u>must</u> be done via eLearn; late penalties apply
Interactive Q&A Sessions with Instructors (graded)	<p>During your Week 13 Class:</p> <ul style="list-style-type: none"> Your team will meet your instructors for a closed-door 20-minute interactive Q&A session (i.e. no formal presentation) Your instructors will ask you questions about your scrum process and software increment You are encouraged to show evidence from your deliverables when you answer your instructors' questions
Peer Review (graded)	<p>By Friday Week 14, 11:59pm:</p> <ul style="list-style-type: none"> Complete your intra-team peer evaluation Non-participation in the project / peer review will result in penalties

Project Goal

The overall goal of this project is to give you hands-on experience of running an agile software project. It is designed to give you a sense of how real-world software developers collaborate both within a team and with an external customer (*the good, the bad, and the ugly*).

Based on the customer briefing provided to you in Week 1, and the subsequent clarification sessions, you are to nurture a product backlog of user stories, design a software system, and ultimately build a first release that increments **seven core functionalities** (specified below).

You are to manage your coding activities using the **scrum process** and are to document evidence of your various scrum activities and meetings throughout the entire project. Unlike other SCIS projects, our focus is on ***the process you are following***, so please do not hack a system together in Week 12 and expect to score well. We want to see a **sustainable** scrum process that is calibrated to the velocity of your team's developers (*and we consider that a 5 pax team typically has a lower velocity than a 6 pax team!*). For example, it would be reasonable to conduct three 2-week sprints with around 6hrs/week coding work per team member. Exceeding 10hrs/week per person suggests something is going wrong.

Note that your final sprint **must** be scheduled/planned to end by Friday Week 12 **at the latest**, as your deliverables are all due that evening by 23:59.

Finally, we want to see evidence of the steps you have taken to ensure the quality of your code: steps such as test case design, continuous integration, and/or refactoring. It may be the case that your team starts sprinting before we teach these topics. If that's the case, it's fine to incorporate them only into your later sprint(s) when your sprinting process should be at its most refined.

First Release – Core Functionality

The customer briefing describes an entire system that is beyond the scope of a single-term project.

The customer has identified **seven core features and the expectations** that they would like to see addressed in your first release (i.e., your submission in Week 12). In particular:

Role	Function	Description
Human Resources and Senior Management	View overall and team schedule	Able to see the list and number of staff in the office and at home by department
Managers and Directors	View team schedule	Able to see who is in the office and at home in own department
Managers and Directors	Approve and reject arrangements	Approve and reject arrangements. To have a list of pending requests

Staff	View team schedule	Able to see who is in the office and at home in own department
Staff	View own schedule	Able to see own schedule
Staff	Apply for arrangement	Apply for arrangement
Staff	Withdraw arrangement	Withdraw an approved arrangement or cancel a pending request

If you can complete these core features ahead of the Week 12 release, you can incorporate further features from the customer briefing as part of your final sprint(s). But you should not torpedo the quality of your sprinting process simply to fit in more features.

Note that even though the Week 12 release only requires working code for these **seven** features, you may still want to consider features beyond the scope of the first release in your product backlog and C4 models.

Details on how to reach the customer for further clarifications have been conveyed separately to your section. DO NOT spam public Slack channels with such questions, as they will not be answered.

Note that each section has a different customer, so clarifications may differ slightly between sections. You should only consider the clarifications offered to your section.

Deliverables – Working Code & Evidence of Scrum

It is important that you start collecting evidence of your team's sprinting process from the first sprint (Week 5) onwards, as this forms a significant part of your Week 12 submission. The quality of your scrum process (and thus the evidence you collect) weighs heavily when we grade your projects.

We encourage you to decide as a team how to manage your scrum project. You could, for example, manage your product and sprint backlogs using spreadsheets. Alternatively, you could explore tools such as Jira. The specific process you put together will determine the type of evidence you need to collect, e.g., PDF exports of Google Sheets vs. exports/screenshots from Jira.

Note that it is fine for your initial sprint to be a bit 'rough', as long as you show evidence of reflecting on this and taking steps to improve your subsequent sprints. We will view your final sprint as a **"showcase sprint"** that demonstrates your process at its best.

Your deliverables should be **submitted in a single .zip file**. The contents of your zip file should be **organised into numbered sub-folders** as per the following:

#1	Deliverable	Type
1	Your final product backlog , containing all your completed and pending user stories (if you use a tool like Jira, you need to export your backlog/stories for the submission)	Document
2	Your system design , conveyed using the C4 model; this should include some C4-Code models for <i>significant design decisions</i>	Document(s)
3	Test cases covering the seven core features	Document(s)
4	<p>Documents from your sprint meetings (<u>all sprints</u>), for example:</p> <ul style="list-style-type: none"> • Sprint backlogs/goals/tasks at the start of each sprint • Records of estimation activities • Scrum task boards at the end of each sprint • Burndown charts at the end of each sprint • Any other relevant meeting documents, e.g., sprint retrospective boards <p>IMPORTANT: for your <u>final sprint only</u>, you must additionally include <u>recordings</u> of your sprint planning, review, retro, and (one) daily standup. As these will be too large for eLearn, please upload them to YouTube as unlisted videos, and include links to the videos as part of your eLearn submission. (If you upload all meetings in a single video, please make sure to <u>time-index it</u>.)</p>	Documents & Videos
5	<p>A self-contained slide deck summarising how your team addressed the rubrics of this course project. (Note that this will not be presented.)</p> <p>IMPORTANT: this <u>must</u> follow the slide template we will release in eLearn by Week 10.</p>	PDF Slide deck

6	<p>Your software release that increments the seven core features</p> <ul style="list-style-type: none"> • Include all your working code • Include a README file with clear instructions on how to run your software release • You can use any programming language¹ (but no commercial / low code software – <i>we need to see tests!</i>) • Please include any test classes containing unit/integration tests for your core features • Include any CI pipeline script that you utilised 	Code
7	<p>A deployment of your software release to a cloud-provider of your choice.</p> <ul style="list-style-type: none"> • Include a README file with a link to your deployment, and any instructions for accessing it (e.g. in case you have authentication). • Your deployment must be available to demonstrate during the Week 13 instructor Q&A meeting. 	README
8	<p>A README file that contains a link to your Git repository; you <u>must also ensure</u> that your instructors have been granted access to the repository. Please invite:</p> <ul style="list-style-type: none"> • G1-2: cposkitt@smu.edu.sg; ivantan@smu.edu.sg • G3-4: ctreude@smu.edu.sg; josephsung@smu.edu.sg • G5-6: ymlau@smu.edu.sg; ivantan@smu.edu.sg • G7-8: rajesh@smu.edu.sg; yzong@smu.edu.sg • G9: nicholastch@smu.edu.sg; josephsung@smu.edu.sg 	README & GitHub access

¹ If you are NOT using Python/Flask (as in our labs), please include full installation instructions so that we can recreate your software environment.

The following will take place **during** your Week 13 class. The specific timing will be conveyed to you by your instructors.

Deliverable	Type
<ul style="list-style-type: none"> Your team will meet your instructors for a closed-door 20-minute interactive Q&A session (i.e. no formal presentation) Your instructors will ask you questions about your scrum process and software increment You are encouraged to show evidence from your deliverables when you answer your instructors' questions 	Interactive Q&A

eLearn Submission Requirements & Late Penalties

- All deliverables must be submitted as a **single .zip file** and should be organised in numbered sub-folders (1–8) according to the table above.
- Please submit to your section's project 'assignment' on eLearn by **23:59 on Friday Week 12**.
- An intra-team peer review will be conducted in Week 14. There will be a penalty for those who are deemed to have not contributed after investigation by the teaching team. Having said that, please inform the teaching team **early** when issues arise within your team so that we can try to intervene before it is too late.
- Late submissions are subject to heavy penalties, as below. Please submit early. We will grade only your latest submission.

within 1 hour (one second late is late)	10% deduction from the marks you deserved
each subsequent hour	Penalty will increase: 20%, 40%, 80%, and then finally 100%

Project Rubrics

Projects will be graded holistically according to the rubrics on the next page:

Unsatisfactory	Fair	Good	Excellent
<ul style="list-style-type: none"> Team codes in a largely ad hoc manner; no discernible process or documentation Team did not show evidence of having used the estimation techniques covered in class for their user stories User stories somewhat follow the prescribed format but they were unclear and did not consider desired attributes of user stories System is documented with detailed specifications that might be invalidated by changes in-between sprints System is poorly designed Minimal evidence of ensuring software quality (e.g., test cases, unit tests) 	<ul style="list-style-type: none"> Evidence of occasionally following a scrum process with limited documentation Estimation techniques are used in an arbitrary manner without justification on why a particular technique(s) is/are used User stories somewhat follow prescribed format but need further conversations to reflect user needs and desired attributes of user stories System is designed beyond necessary significant design decisions with specifications that introduce inflexibility during sprint development System exhibits some elements of good design and some parts work on a public cloud provider Some unit testing is performed 	<ul style="list-style-type: none"> Team largely follows a scrum process supported by adequate documentation Estimation techniques are used in a consistent manner with adequate justification on why particular technique(s) is/are used User stories follow prescribed format and exhibit desired attributes but need further conversations to confirm user needs System is well modelled showing significant design decisions System is well-designed and solution is demonstrated on a public cloud provider System is well-covered by unit tests A basic CI pipeline is utilised when integrating code 	<ul style="list-style-type: none"> Team consistently follows an effective scrum process supported by comprehensive documentation Estimation techniques are used in a consistent manner with adequate justification on why particular technique(s) is/are used to produce valid and reasonable estimates User stories follow prescribed format and exhibit desired attributes with clear demonstration of user needs and acceptance criteria System is well-modelled showing significant design decisions that exhibit good practices of pattern reuse and design principles System is well-designed, code is modular, and solution is demonstrated on a public cloud provider System is comprehensively covered by unit tests and/or integration tests A comprehensive CI/CD pipeline is used during code integration