Agile Software Development Practices (SOFT2412/COMP9412)

School of Computer Science

Assignment 2 – Agile Software Development with Scrum and Agile Tools

Group assignment (3-5 students)
Weight: 25%
Due: Multiple deadlines (see submission requirements below)
Deliverables: Multiple technical reports and multiple project demonstrations (listed below)

1. Background

The goal of this assignment is to work as a team to develop a software product using the Scrum methodology and Agile development tools and practices. In this software development project, your team must apply Scrum practices and principles properly to build a software product in Java. Each team also must use the Agile software tools they set up during the first group project to follow CI/CD practices in their development. These tools must be set up and ready before writing any code for your application.

Each team will have a stakeholder – a client/customer (your tutor) for their project. Your team will have the opportunity to discuss and ask questions related to software requirements from the project client during the lab/tutorial time. Your tutor will act as a client whom you should satisfy in terms of delivering a working version of the software.

As a team, you will get hands-on experience in organising and conducting Agile software development using the Scrum methodology and Agile development tools. Specifically, you and your team will be asked to:

- Set up and use Agile development tools and CI/CD practices;
- Experience the importance of Agile values and principles;
- Understand and apply Scrum roles;
- Write and estimate User Stories;
- Plan and execute development Sprints;
- Conduct Scrum events;
- Continuously improve the development process; and
- Produce various Scrum artefacts.

2. Forming Scrum Teams

In this group project, you will continue to work with your team from your Agile Development Tools project (Assignment 1). Therefore, it is crucial that you reflect on your team dynamics and difficulties, and think as a team about how you can collaborate and work in a more productive environment. The team structure must follow the Scrum method which should help your team to be more self-organised. Your team need to decide on the following roles:

1. 1 Product Owner;
2. 1 Scrum Master; and
3. 2-3 Core Team members.

After team formation, each team will need to discuss and agree on the above Scrum roles. Both the Product Owner and Scrum Master must also contribute to the development of the project (writing code using CI/CD) in every Sprint, in that they are also required to develop parts of the submitted software product.

3. Virtual Scroll Access System

Prologue: In the enchanting realm of Edstemus, a mystical guild of computer science elves known as the 'Whiskers' sought to explore the ancient scrolls of wisdom stored within the Library of Agility. These scrolls, though not made of parchment and ink, were binary files, each containing a trove of digital knowledge. The revered wizard Ankit of the realm of Edstemus has tasked some of the best Whiskers with creating a Virtual Scroll Access System (VSAS) to facilitate the retrieval and sharing of these digital scrolls. Your group has been selected for this noble quest.

Objective: Your task is to design and develop the Virtual Scroll Access System (VSAS), a digital repository to grant Whiskers access to the library's knowledge, represented as downloadable binary files. As far as you are concerned, the contents of these files doesn't matter as long as the specifications are met (that is, you are not required to do anything with the data other than store it). This system will serve as a portal to the realm of digital wisdom, where library staff and members can efficiently access, download and upload these files. You may use a graphical or terminal-based user interface.

Specifications:

All of the following are just basic requirements from the teaching team. You are encouraged to extend these features and come up with more innovative and interesting ideas on your own.

3.1. User Management

Registration and Login: Users can create accounts with detailed profiles, including personal information such as their phone number, email address, full name and customisable ID keys. All accounts must be locked behind a username and password. No two accounts can have the same ID key.
Update User Profiles: Logged-in users can update and change their information on their profile, including their password.
Guest Users: Users may anonymously use the application without logging in, but are only able to view scrolls and may not upload or download scrolls.
Admin Users: Make a single admin profile that has access to special admin privileges. Admins should be able to:

- View a list of all users and their profiles
- Be able to add and delete other users
- View stats such as the number of downloads/uploads for each scroll ever passed through the application.
  User Type Display: The user's name and type should be displayed on the main UI.
  Password encryption: The password for login stored (either in a database or locally stored file) should be encrypted using any hashing algorithm.

3.2. Digital Scroll Management

Adding New Digital Scrolls: Users can add scrolls to the virtual library. Each scroll will have a unique name and ID for categorisation. When adding a scroll, the user will be asked to upload its binary file.
Edit and Update Digital Scrolls: Users should be able to make modifications to the scrolls they have uploaded. They should not be able to edit other Whiskers' scrolls.
Remove Digital Scrolls: Users should be able to remove any scrolls that they have uploaded to the platform.

3.3. Scroll Seeker

View Scrolls: Users should be able to view all available scrolls. You may implement this feature in any way you see fit (e.g. list of title names, icons, etc.).
Download Scrolls: Users can pick a scroll to download anytime during their browsing. They should not have to leave the View Scrolls screen to do it.
Search Filters: Implement filters for refining searches based on uploader ID, scroll ID, name and upload date.
Preview Scrolls: All users can preview scrolls on the platform prior to downloading them.

3.4. The Evolution of Literature

As the Library of Agility is never static, it is only natural that the requirements set for you by your client will follow suit. Throughout the development of your program, your client (and by extension, the supreme wizard) may ask for you to make small adjustments to your program to suit their needs. These will be discussed in more detail by your client (your tutor).

Epilogue: In the mystical realm of Edstemus, the Whiskers embark on their quest to harness the wisdom within the Library of Agility. May your journey through the Virtual Scroll Access System be swift and successful, as you unlock the digital secrets that await. Best of luck, brave Whiskers!

4. Scrum – The Project Sprints

You have a total of 4 weeks consisting of 3 main sprints to complete your project. The actual development of the software must run over 3 Sprints (iterations) and one preparation Sprint (Sprint 0). Each sprint will last for one week. Your team needs to follow Scrum methodologies for every sprint. Sprint 0 will be due in your tutorial in Week 9, with each subsequent sprint ending exactly one week later. Each team must follow the Sprint schedule as described below.

- Sprint 0: From the date of assignment release. Your team can use this period to decide on team roles, get familiar with Scrum methodologies and make progress on your initial development setup (Agile development tools). Ensure that you prepare initial user stories (including an initial product backlog) based on the requirements described above.
- Sprint 1: Starts in Week 9 and ends in Week 10 (both due the day after your tutorial)
- Sprint 2: Starts in Week 10 and ends in Week 11 (both on the day after your tutorial)
- Sprint 3: Starts in Week 11 and ends in Week 12 (both on the day after your tutorial)
- Individual oral quiz: Week 13 (on the day of your tutorial)

During your tutorial in Week 9, your team will have an opportunity to clarify and elaborate on the program and system requirements for the project. During each Sprint, your team must:

- Carry out development following Scrum practices and principles, as well as follow CI/CD practices using Agile development tools.
- Document all the Scrum events that happen, including user stories, relevant team members' interactions and the resultantly updated artefacts. This also includes the tools used to implement Scrum.
- Document the key events that occurred during the development of the Virtual Scroll Access system. This includes the use of Agile development tools and practices (including CI/CD) by all team members.

We recommend having a minimum of 5 stand-ups a week, even if they only last 15 minutes each. They can be conducted via Zoom or in-person.

At the end of each Sprint, your team must:

- Conduct the Sprint review during the tutorial and demonstrate the current version (increment) of the Virtual Scroll Access system to your client (tutor). Your tutor will provide feedback about the product and make some observations about the demo from the client's point of view (client role).
- The tutor will also note down their observations on each demo. and how's the work conducted using Agile tools and Scrum (your tutor's role). These will be used to assess the project demos. across the 3 Sprints.
- Submit a group Sprint report that documents all the work done in that Sprint as per the project requirements.
- Submit the version of the source code demonstrated to the client.

5. Building the Virtual Scroll Access System using Agile Tools

Each team must develop the Virtual Scroll Access System application in Java. All team members must collaboratively develop the application's requirements using Agile development tools and practices. Each team must produce a version of the Virtual Scroll Access software at the end of each Sprint and ensure that:

- The program must always produce correct output based on the requirements (based on this assignment description) and additional requirements from your client/tutor.
- The program must be built in a modular way since every team member is required to contribute to writing code for the software.
- Each team must carry on development using Scrum methodologies and the tools and practices (CI/CD) that were used in the first group project. Each team will be required to demonstrate their use of these tools. Automated builds and unit tests must be triggerable with appropriate test/code coverage and reporting. You must make sure your unit tests have good code coverage (>75% code coverage). For this project, it is acceptable to use Jenkins' polling feature to trigger automated builds and tests.
- Like the first group project, the team's development work must be available under the SOFT2412 GitHub organisation account. Your tutor must have access to and be able to inspect the source code at any point in time. Your tutor should also be able to inspect the CI/CD tools you've used during development at any given point in time.

6. Submission Requirements

6.1 Weekly Demo

Each team must demonstrate to their client (tutor) a version of their Virtual Scroll Access System at the end of each Sprint in their scheduled tutorial. Each team member must demo their individual contributions (user stories / code tasks) they complete in each Sprint to their client. In the GitHub account, your team must label the source code (git tag) that will be demonstrated to the client with the appropriate version number so your tutor can inspect the appropriate source code. In the last Sprint, each team must demonstrate the complete Virtual Scroll Access System. Each Sprint demo will be conducted as follows:

- Sprint 1 Demo: Week 10's tutorials (all group members must participate)
- Sprint 2 Demo: Week 11's tutorials (all group members must participate)
- Sprint 3 Demo: Week 12's tutorials (all group members must participate)
- Individual Oral Quiz: Week 13 tutorials (all group members must participate)

All group members must attend all the Sprint Demos and participate in the demonstration. These weekly demos will be assessed by your tutor from both a client's point of view as well as how Scrum and Agile development tools were used in each Sprint. Your tutor will also observe the individual contributions based on the presented user stories, Scrum development and use of Agile tools.

6.2 Weekly Report and Source Code

Each team must submit a report at the end of each Sprint starting from the end of Sprint 1 (Week 11) through the link provided on Canvas. Only one submission per group is needed.

The report should include concise documentation of the team's development of their Virtual Scroll Access software using Scrum and the Agile development tools and practices as detailed in Sections 4 and 5. Note your team should provide details on all Scrum events and Scrum artefacts and include evidence to support the description and justification of the development work presented in their report. The source code version (commit) of each sprint must be also labelled on your GitHub repository using git tag. Furthermore, each group must submit an executable of their final source code at the end of Sprint 3 (Week 13) along with instructions on how the client can run the program for offline testing. The sprint reports and source code are due as follows at 11:59pm on the day after your tutorial each week:

- Sprint 1 Report & Source Code: Week 10
- Sprint 2 Report & Source Code: Week 11
- Sprint 3 Report & Source Code: Week 12

All group members must sign the assignment coversheet and attach it as the first page of each report. Contribution issues must be reported early (for any sprint) to your tutor so they can investigate it.

7. Group Member Contribution

If members of your group do not contribute sufficiently (in any of the sprints) you must alert your tutor within 2 days of the start of the next sprint. Each team member must be assigned user stories/features to implement during the sprint, and everyone's individual progress must be tracked through the project tracking tools implemented into your project (e.g. GitHub, Trello/Jira, etc.). Your tutor has the discretion to scale the group's mark for each member based on their contributions as follows:

Level of Contribution	**Proportion of Final Grade Received**
No contribution	0%
Poor contribution	50%
Insufficient contribution	75%
Sufficient contribution	100%

8. Marking Guide

The marking of each iteration will be based on the requirements detailed above. Below is a summary of the marking guide which will be used along with the detailed requirements listed above. For each assessable component, evidence must be provided in your project documentation and demonstrations to support any and all claims that you make.

8.1. Quality of Scrum Development (50%)

- Project Team (5%): Quality of team organisation and roles in accordance with Scrum methodologies and as detailed in the above assignment requirements.
- Sprint Goal (5%): Quality of sprint goals (sensibility, relevance and clarity).
- Tasks Board (Product and Sprint) (10%): Quality of the way the team created, managed and maintained the product and sprint backlogs in accordance with Scrum methodologies as detailed in the assignment requirements listed above (correctness, completeness, relevance, sensible and clear).

- Scrum Events artefacts (30%): Quality of the way the team carried out, managed and maintained Scrum events and artefacts in accordance with Scrum (in terms of correctness, completeness, relevance, sensibility and clarity). You should report all of the Scrum events and artefacts as well as the challenges/issues faced by the team during each sprint, including documentation on issues and conflict resolutions.

8.2. Agile Development Tools and Practices (15%)

- Quality of the application in terms of usability, maintainability, alignment with requirements and implementation of features as requested by the client
- Use of CI/CD tools and practices (GitHub, Gradle, JUnit and Jenkins) each week
- Your group's working source code must be on your Git repository with the appropriate labels/tags/releases of each end-of-sprint release
- Your final submission of your source code must also include a README.md file that describes how other developers and users can install, use and test your program

8.3. Quality of Application Development (20%)

- A clear and sensible explanation of the work carried out by the team in implementing, managing and using issue-tracking software. This should be further demonstrated through the project report and group demo including:
    - A description of how the group collaborated to complete the application using the above tools
    - Descriptions of individual and group contributions, group communication (you can reference other sections/parts if already presented previously in your report)
    - An overview of the CI tools and structure management used by your group (e.g. class diagrams, sequence diagrams, etc.)
    - How well the application produces the correct output/behaviour with various inputs and adheres to the functional requirements during your demo (recorded with a live Q&A)
- The above must be supported by appropriate evidence where applicable (e.g., screenshots, outputs, logs)

8.4. Quality of Demonstration (15%)

- Quality of the demonstrated version of the software from the client's point of view, including producing the correct output and behaviour based on the initial and additional requirements as elaborated by the client during each sprint and Q&A session (including running and testing the submitted code where required)

- A demonstration of how the demoed version/product meets the client's needs and delivers a high-quality output (well-tested and working without any errors, exceptions, mistakes or deviation from the core requirements)
- The team can respond to client questions and requests, as well as explain and justify the work they've done in the project and the results produced during the demo and Q&As

8.5. Quality of Individual Contribution

- Each team member made a fair and quantifiable contribution to the development of the software application using Scrum and Agile tools (as per the assignment requirements). All members collaborated and worked as a team as clearly evident by the GitHub repository and tools (e.g. Trello, Jira) and have provided evidence in each of the development activities above.
- Each team member should clearly indicate their contributions in the report in a table of technical and non-technical aspects of the contribution.
- The weekly demos will be assessed by your tutor from both the client's point of view as well as how well Scrum and Agile development tools were used in each sprint. Your tutor will also observe the individual contributions based on the presented user stories and demonstrations.
- All team members are required to present during each end-of-sprint demonstration. Team members who are unwilling to participate will be given a 0 for that sprint. Absences must be explained by a successful application for Special Consideration.

Note: *All assignment updates will be announced on Ed by the Unit Coordinator or a Teaching Assistant.*

Academic Integrity

While the University is aware that the vast majority of students and staff act ethically and honestly, it is opposed to and will not tolerate academic integrity breaches and will treat all allegations seriously.

Further information on academic integrity, and the resources available to all students can be found on the academic integrity pages on the current students website: https://sydney.edu.au/students/academic-integrity.html.

You may only use generative artificial intelligence (AI) and automated writing tools in assessment tasks if you are permitted to by your unit coordinator. If you do use these tools, you must acknowledge this in your work, either in a footnote or an acknowledgement section. For information on acknowledging AI please refer to the guidance in the AI in

Education Canvas site.

We use Turnitin, which includes AI detection, to detect potential instances of plagiarism or other forms of academic integrity breach. If such matches indicate evidence of plagiarism or other forms of academic integrity breaches, your teacher is required to report your work for further investigation.

Further information for on research integrity and ethics for postgraduate research students and students undertaking research-focussed coursework such as Honours and capstone research projects can be also be found on the current students website: https://sydney.edu.au/students/research-integrity-ethics.html.

Compliance Statement

In submitting this work, I acknowledge I have understood the following:

- I have read and understood the University of Sydney's Academic Integrity Policy.
- The work is substantially my own and, where any parts of this work are not my own, I have indicated this by acknowledging the source of those parts of the work and clearly indicated any quoted text by quotation marks or indentation according to accepted style standards.
- I have acknowledged any assistance provided in preparing the work including the use of copy-editing, proof-reading, and automated writing and drawing tools (including artificial intelligence (AI), reference generators, translation software, grammar checkers, but not spell checkers).
- The work has not previously been submitted in part or in full for assessment in another unit unless I have been given permission by my unit of study coordinator to do so.
- The work will be submitted to similarity detection software (Turnitin) and a copy of the work will be retained in Turnitin's paper repository for future similarity checking.
- Engaging in plagiarism or academic dishonesty in coursework will, if detected, lead to the University commencing proceedings under the Academic Integrity Policy. and the Academic Integrity Procedures.
- Engaging in plagiarism or academic dishonesty in research-focussed work will lead to the University commencing proceedings under the Research Code of Conduct and the Academic Integrity Procedures.
- Engaging another person to complete part or all of the submitted work will, if detected, lead to the University commencing proceedings against me for potential student misconduct under the University of Sydney (Student Discipline) Rule.

You are advised to keep copies of your assignment submission, drafts and any other research materials as evidence of your research and writing process. If you have used AI in the completion of your assignment, you should keep copies of the AI outputs.

Rubric

2023 Group project Assignment 2				
Criteria	**Ratings**			**Pts**
Quality of Scrum Development	**50 pts** **Full Marks**	**50 to >0 pts** **Partial Marks**	**0 pts** **No Marks**	/ 50 pts
Agile Development Tools and Practices	**15 pts** **Full Marks**	**15 to >0 pts** **Partial Marks**	**0 pts** **No Marks**	/ 15 pts
Quality of Application Development	**20 pts** **Full Marks**	**20 to >0 pts** **Partial Marks**	**0 pts** **No Marks**	/ 20 pts
Quality of Demonstration	**15 pts** **Full Marks**	**15 to >0 pts** **Partial Marks**	**0 pts** **No Marks**	/ 15 pts
				Total Points: 0