# DIICSU Agile Software Engineering

# Assignment Specification

Agile Software Engineering is an iterative and flexible approach to software development that prioritises collaboration, customer feedback, and incremental delivery of functional software. Unlike traditional methodologies like Waterfall (which follows a rigid, linear sequence of phases), Agile focuses on adaptability, continuous improvement, and delivering value to users early and often.   
  
Agile is guided by the **Agile Manifesto** which emphasises:

1. **Individuals and interactions** over processes and tools.
2. **Working software** over comprehensive documentation.
3. **Customer collaboration** over contract negotiation.
4. **Responding to change** over following a plan.

You are going to undertake an agile project using mainly the Scrum methodology across 4 weeks.

* Sprint 1 – week beginning 3rd March
* Sprint 2 – week beginning 10th March
* Sprint 3 – week beginning 17th March
* Sprint 4 – week beginning 24th March

Sprints 1 and 3 are formative, meaning you will only be given feedback to help improve your team. Sprints 2 and 4 will be summative.

Each week will take the following format:

* Monday at 2pm – planning for the next sprint
* Monday at 4pm – Daily Scrum 1
* Tuesday at 7pm – Daily Scrum 2
* Wednesday at 7pm – Daily Scrum 3
* Thursday at 4pm – Daily Scrum 4
* Friday at 10am – Daily Scrum 5
* Friday at 2pm – Sprint Review

Note, that these meetings are all during class time, so the lecturer will be present and grading based on these meetings. This means if you fail to attend any of these meetings you will be down graded.

### Planning meetings

The planning meeting in week 1 of the project will have the following format:

1st 45 minutes – creating a product backlog

2nd 45 minutes – sprint planning for 1st sprint, deciding on what is in the sprint backlog

From week 2 onwards, the planning meeting will take the following format:   
1st 45 minutes – retrospective on the previous week

2nd 45 minutes – sprint planning for next sprint, deciding on what is in the sprint backlog.

During each of these 45 minute sessions, the lecturer will spend 5 minutes with each group monitoring progress and coaching where necessary.

### Daily Scrum

Each team will have a 10-minute daily scrum meeting each workday. These will be during the timetabled session. The lecturer will attend to monitor how these are being performed and to provide coaching where necessary.

### Sprint Review

At the end of each sprint,, a customer will come to discuss what has been achieved so far. Each team will have 10 minutes to demonstrate their software and elicit feedback.

### Roles in the Team

There are 3 roles in the team:

1 x Product Owner - Takes the inputs of what the product should be and translates them into a product vision and a Product Backlog. Makes sure the user is kept as the focus of the project.

1 x Scrum Master - Does whatever it takes to make the Scrum Team successful, such as removing organisational impediments, facilitating meetings, protecting the team

The rest of the team are the development team.

The roles held by each team member will rotate at the end of each sprint. No team member should be the Product Owner or Scrum Master more than once. Also, everyone must do at least one of these jobs at some point in the project.

### Artefacts

The following artefacts should be maintained at all times. The lecturer may look at their current state at any time, and use this as part of the continuous grading:   
**Product Backlog** – This must be in the group's Microsoft Teams folder and accessible and editable by all team members.

**Sprint Backlog** – This must be in the group's Microsoft Teams folder and accessible and editable by all team members.

**Code Repository** – This should be hosted on GitHub. This should contain: All source code for your project and a link to a live version of your project (if web-based). Remember to add the lecturer to your GitHub repository (KarenPetrie1980)

## Communication

All communication in this project must be in English. Groups will be downgraded whenever that is not the case, as English is the medium of instruction for this degree programme, and this project is being assessed.

All written communication between the team must be in the Microsoft Teams channel set-up for this purpose.

## Required to be Submitted

As this project is mainly graded continuously based on all the meetings and the artefacts in both Microsoft Teams and Github, there is only 1 small submission required.

This is an individual contribution document that outlines what you feel you contributed to sprints 2 and 4. A template for this is provided. This includes the AI declaration.

Note: a full peer review is not required for this project as the lecturer will be able to monitor contributions at the daily meetings.

## Deadline for Submission​

The deadline for the individual contribution document is:   
Sprint 2 – Saturday 15th March at 20:00

Sprint 4 – Saturday 29th March at 20:00

## Hand in method​

The individual contributions will be submitted through My Dundee. The Git hub repository, product backlog and sprint backlog are expected to be kept updated throughout the project.

## Feedback method and date ​

Oral feedback from the lecturer will be received regularly in the daily meetings. This will be particularly prevalent in weeks 1 and 3, which are formative only, so the lecturer can provide assistance without prejudice.

Formal written feedback will be received within 3 weeks of the deadlines for submission. This will be on My Dundee.

## Penalty for late submission​

One grade point per day late (meaning if a submission is one day late and marked as a C2, it will receive a C3 grade). A day is defined as each 24-hour period following the submission deadline, including weekends and holidays. Assignments submitted more than 5 days after the agreed deadline will receive a zero mark (AB).

## Percentage of the module​

Sprint 2 is worth 40% of this module

Sprint 4 is worth 40% of this module

## AI statement

A diagram of a diagram

AI-generated content may be incorrect.

AI may be used to complete any element of this assessment. Noting that most of the grading is based on how the group collaborates to complete the task. Not on the final code produced to complete the project.

You must complete the AI declaration as part of the individual contribution submission to indicate your use of AI.

## Marking scheme

### Sprint 2

Adopting an Agile approach, you are to plan and perform your second sprint. This will be assessed on the following aspects:

* Creating and maintaining a product backlog consisting of suitable user stories
* Create and update a sprint backlog
* Develop components of the software
* Use agile methods and apply appropriate quality standards consisting of:
  + Refactoring of code
  + Perform Code Review
  + Daily Scrums
  + Use of Pair Programming
  + Regular and Informative use of GitHub
* Your meeting with project stakeholders

The emphasis of this assignment is the creation and maintaining of a product backlog, followed by the completion of a sprint to develop code for the required software.

The software language and development stack that you use is up to you. Hosting of the project is also under your control.

*It is your responsibility to ensure that you abide by any third-party service agreements surrounding security of keys and usage of data. Please remember that data limits are in place when using some services. Questions should be directed to the module leader.*

See full rubric at the end of the document.

### Sprint 4

Adopting an agile approach, you are to plan and perform the fourth sprint to continue development of the application that was described earlier. In this sprint you will be assessed on the following aspects:

* Reflect on the outcomes of Sprint 3 through a Sprint Retrospective
* Develop the next set of user stories agreed with the project stakeholders
* Apply the actions agreed by your team at your sprint retrospective
* Use agile methods and apply appropriate quality standards consisting of:
  + Refactoring of code
  + Perform Code Review
  + Daily Scrums
  + Use of Pair Programming
  + Regular and Informative use of GitHub

The emphasis of this assessment is the finishing of the product.

See full rubric at the end of the document.

## Sprint 2

Grading Rubric

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A1 | A2 | A3 | A4 | A5 | B1 | B2 | B3 | C1 | C2 | C3 | D1 | D2 | D3 | M1 | M2 | M3 | CF | BF | QF |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 5 | 2 | - |
| **A** | | | | | **B** | | | **C** | | | **D** | | | **MF** | | | **F** | | | |
| User Stories and Product Backlog  20% | * User stories and product backlog have been completed to a professional level, with consistency used throughout. Consistency and effort have been added to all stories with careful thought on the levels assigned * Evidence is present of product backlog evolving throughout the sprint, with additional stories being added as project evolves | | | | | * User stories and product backlog are completed to a high-quality standard, but some details are lacking. * There is some evidence of the product backlog evolving throughout the sprint with new user stories being added, but this is limited | | | * User stories and product backlog have been constructed but there are issued relating to the factors included. * The product backlog shows little evidence of evolution through the sprint. | | | * User stories have been constructed but essential information is missing that has made it difficult to properly order these within a product backlog. The backlog is badly formed and does not follow convention. * Very little evidence of backlog evolution throughout the sprint. | | | * The user stories and product subsequent product backlog are badly formed. Essential information is missing, and the result is a list of features that would not fit the traditional view of a user story. * The product backlog was viewed as a static artefact throughout the sprint | | | * As MF but with substantial failings | | | |
| Sprint Backlog[[1]](#footnote-1)  20% | * The sprint backlog is completed to a professional level and associated charts explaining progress are included. * It is evident from the backlog who individual tasks have been assigned to and their progress throughout the sprint. * There is evidence of the backlog being updated after every scrum meeting within the code repository | | | | | * The sprint backlog is completed to a high standard and there is some evidence of charts being used to demonstrate progress, however these are unclear in places. * Some parts of the backlog may be confusing to understand due to missing information * There is evidence of the backlog being updated after every scrum meeting within the code repository | | | * The sprint backlog has been completed but it is viewed as more of a static document that was completed at the end of the project. * There are errors in the backlog that make it difficult to understand aspects related to different stories and their level of completion. | | | * Sprint backlog has been completed but there are clear faults in how it has been created. * It is difficult to understand information regarding the user stories that have been completed, who was responsible, and their progress throughout the project | | | * The sprint backlog is completed to a very low standard. It is not clear what tasks are being carried out or how these have been measured * No suitable visualisations have been used to see the sprint burndown | | | * As MF but with substantial failings | | | |
| Software Artefact Development[[2]](#footnote-2)  10% | * The quality of the developed software artefact is at a professional level and polish has been applied in order to create an artefact that is usable in an industrial setting * Care has been taken to consider front and back end elements and how they work together to create an excellent user experience | | | | | * The overall quality of the created software artefact is at a high level, however there are elements that require additional work in order to be deemed complete * Whilst care has been taken to focus on overall application experience, there are areas that need additional work. | | | * The software has been created to a good standard but there are elements that need additional work * Some care has been taken to focus on the overall software experience * Care has been taken to focus on the brief provided by the client. | | | * Functional software has been created, but there are clear areas that are not ready for demonstration. * Little care has been taken to focus on the brief provided by the client | | | * The overall quality of the created software is poor, or the level of complexity of the software is at a very low standard. * It is apparent that little time or effort was spent on developing the software and the overall experience suffers due to this | | | * As MF but with substantial failings. | | | |
| GitHub Usage  10% | * GitHub and code versioning have been used successfully throughout. * Commits are logical, atomic and well commented. * Git flow was considered and used effectively when creating branches. Commits are reviewed by other team members and not the code creator. | | | | | * GitHub and code versioning have been used throughout the project. * There are areas where use of GitHub features could have been improved relating to the use of branching, issues, and project management | | | * GitHub and code versioning were used within the project but with major flaws in commit history or usage of branching | | | * GitHub was used within the project but with limited commit history or commenting. | | | * Minimal usage of GitHub throughout the project with an extremely small number of commits being made. | | | * As with MF but with substantial failings | | | |
| Agile Method Usage[[3]](#footnote-3)  30% | A large number of agile (and related) processes being used well that extends what has been covered in class. This may include:   * Code reviews * Pair programming   The daily scrum meetings are excellent, with all team members actively participating and showing shared ownership of the project. | | | | | Agile processes being used that were covered in class. This will include the items mentioned within (A) grade. Processes not introduced in class may also have been used  The daily scrum meetings are good, but some team members could contribute more. | | | Some agile (and related) processes that were introduced in class have been evidenced. Evidence of between 2-3 processes in total have been used.  The daily scrum meeting is just every student giving an update, no real discussion or collaboration | | | Very little agile processes have been used within the project. No processes that extend topics shown within the class have been used.  Some students do not participate in the daily scrum meeting. | | | There is no real evidence of agile processes being used within the first sprint. Any processes that have been attempted and evidenced have serious flaws.  Daily scrum does not take place | | | * As MF but with substantial failings | | | |
| Sprint Review[[4]](#footnote-4)  10% | It is expected that the sprint review will be carried out to a professional level by all groups. The following criteria will be used, regardless of grade, to judge the success of the sprint review:   * Welcoming of customers and setting scene for the sprint review * Overview of what will/will not be demonstrated in the sprint review * Demonstration of new functionality * Discussion of key events or problems that occurred during the sprint * Present upcoming product backlog items * Conclusion of the meeting | | | | | | | | | | | | | | | | | | | | |
| Individual Effort  Adjusted based on performance from group grade | Your individual grade for this assignment will be adjusted based on the effort that you have reported within your CW submission document | | | | | | | | | | | | | | | | | | | | |

## Sprint 4

Grading Rubric

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A1 | A2 | A3 | A4 | A5 | B1 | B2 | B3 | C1 | C2 | C3 | D1 | D2 | D3 | M1 | M2 | M3 | CF | BF | QF |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 5 | 2 | - |
| **A** | | | | | **B** | | | **C** | | | **D** | | | **MF** | | | **F** | | | |
| Sprint Retrospective and Follow Up Actions[[5]](#footnote-5)  20% | * The sprint retrospective takes a focused approach and details what worked well, what could be improved, and what will be accomplished in the next sprint. This is supplemented by information given by each team member so that actionable items can be created * Actionable items are revisited at the end of the second sprint and reflected on. | | | | | * A sprint retrospective takes place, covering most of the points required (shown in [A]). Some team members give supplementary information so that actionable items can be created * A number of actionable items are reflected on but this may be slightly superficial | | | * A sprint retrospective takes place covering some of the required points (shown in [A]). Very few team members make an active contribution to this. * Actionable items are created but not discussed in great detail at the end of sprint 2 | | | * A sprint retrospective takes place, but this does not follow any sort of convention. * Little information is given regarding how actionable items came into realisation | | | * The sprint retrospective is carried out, but no real actionable items are created. * A reflection takes place but this contains no useful information | | | * As MF but with substantial failings | | | |
| Sprint Backlog[[6]](#footnote-6)  20% | * The sprint backlog is completed to a professional level and associated charts explaining progress are included. * It is evident from the backlog who individual tasks have been assigned to and their progress throughout the sprint. * There is evidence of the backlog being updated after every scrum meeting within the code repository | | | | | * The sprint backlog is completed to a high standard and there is some evidence of charts being used to demonstrate progress, however these are unclear in places. * Some parts of the backlog may be confusing to understand due to missing information * There is evidence of the backlog being updated after every scrum meeting within the code repository | | | * The sprint backlog has been completed but it is viewed as more of a static document that was completed at the end of the project. * There are errors in the backlog that make it difficult to understand aspects related to different stories and their level of completion. | | | * Sprint backlog has been completed but there are clear faults in how it has been created. * It is difficult to understand information regarding the user stories that have been completed, who was responsible, and their progress throughout the project | | | * The sprint backlog is completed to a very low standard. It is not clear what tasks are being carried out or how these have been measured * No suitable visualisations have been used to see the sprint burndown | | | * As MF but with substantial failings | | | |
| Software Artefact Development[[7]](#footnote-7)  10% | * The quality of the developed software artefact is at a professional level and polish has been applied in order to create components that are usable in an industrial setting * Care has been taken to consider front and back end elements and how they work together to create an excellent user experience | | | | | * The overall quality of the created software artefact is at a high level, however there are elements that require additional work in order to be deemed complete * Whilst care has been taken to focus on overall application experience, there are areas that need additional work. | | | * The software has been created to a good standard but there are elements that need additional work * Some care has been taken to focus on the overall software experience * Care has been taken to focus on the brief provided by the client. | | | * Functional software has been created, but there are clear areas that are not ready for demonstration. * Little care has been taken to focus on the brief provided by the client | | | * The overall quality of the created software is poor, or the level of complexity of the software is at a very low standard. * It is apparent that little time or effort was spent on developing the software and the overall experience suffers due to this | | | * As MF but with substantial failings. | | | |
| GitHub Usage  10% | * GitHub and code versioning have been used successfully throughout. * Commits are logical, atomic and well commented. * Git flow was considered and used effectively when creating branches. Commits are reviewed by other team members and not the code creator. * Advanced GitHub features are used to great effect. | | | | | * GitHub and code versioning have been used throughout the project. * There are areas where use of GitHub features could have been improved relating to the use of branching, issues, and project management | | | * GitHub and code versioning were used within the project but with major flaws in commit history or usage of branching | | | * GitHub was used within the project but with limited commit history or commenting. | | | * Minimal usage of GitHub throughout the project with an extremely small number of commits being made. | | | * As with MF but with substantial failings | | | |
| Agile Method Usage[[8]](#footnote-8)  30% | A large number of agile (and related) processes being used well that extends what has been covered in class. This may include:   * Code reviews * Pair programming * The daily scrum meetings are excellent, with all team members actively participating and showing shared ownership of the project. | | | | | Agile processes being used that were covered in class. This will include the items mentioned within (A) grade. Processes not introduced in class may also have been used  The daily scrum meetings are good, but some team members could contribute more. | | | Some agile (and related) processes that were introduced in class have been evidenced. Evidence of between 2-3 processes in total have been used.  The daily scrum meeting is just every student giving an update, no real discussion or collaboration | | | Very little agile processes have been used within the project. No processes that extend topics shown within the class have been used.  Some students do not participate in the daily scrum meeting. | | | There is no real evidence of agile processes being used within the first sprint. Any processes that have been attempted and evidenced have serious flaws.  Daily scrum does not take place | | | * As MF but with substantial failings | | | |
| Sprint Review[[9]](#footnote-9)  10% | It is expected that the sprint review will be carried out to a professional level by all groups. The following criteria will be used, regardless of grade, to judge the success of the sprint review:   * Welcoming of customers and setting scene for the sprint review * Overview of what will/will not be demonstrated in the sprint review * Demonstration of new functionality * Discussion of key events or problems that occurred during the sprint * Present upcoming product backlog items * Conclusion of the meeting | | | | | | | | | | | | | | | | | | | | |
| Individual Effort  Adjusted based on performance from group grade | Your individual grade for this assignment will be adjusted based on the effort that you have reported within your CW submission document | | | | | | | | | | | | | | | | | | | | |

1. Quality Metrics for Sprint Backlog will be based on <https://www.agilealliance.org/glossary/sprint-backlog/> [↑](#footnote-ref-1)
2. Software Component Development marking will be based on the features shown to the client within the sprint review [↑](#footnote-ref-2)
3. Additional agile methods can be found within the Agile Alliance Subway Map <https://www.agilealliance.org/agile101/subway-map-to-agile-practices/> [↑](#footnote-ref-3)
4. Quality metrics for the Sprint Review is based on <https://www.scrum.org/resources/what-is-a-sprint-review> [↑](#footnote-ref-4)
5. Retrospective activities, along with a retrospective planner can be found at <https://www.funretrospectives.com/> [↑](#footnote-ref-5)
6. Quality Metrics for Sprint Backlog will be based on <https://www.agilealliance.org/glossary/sprint-backlog/> [↑](#footnote-ref-6)
7. Software Component Development marking will be based on the features shown to the client within the sprint review [↑](#footnote-ref-7)
8. Additional agile methods can be found within the Agile Alliance Subway Map <https://www.agilealliance.org/agile101/subway-map-to-agile-practices/> [↑](#footnote-ref-8)
9. Quality metrics for the Sprint Review is based on <https://www.scrum.org/resources/what-is-a-sprint-review> [↑](#footnote-ref-9)