

Change Report

Team 24

Joe Cambridge

Joss Davis

Emily Dennison

Louise Evenden

Amber Hemsley

Josh McWilliam

Lianyu Zhao

a) Summary of Processes, Tools and Conventions

We used Google Docs to plan our project approach. We created “minutes” Google documents during every meeting where we planned what we had done since the previous meeting and what we would do for the next meeting. We also used our Gantt chart to plan out our project. This helped us track our weekly performance and adapt to changes based on team members’ availability. We used PlantUML in order to create the Gantt chart, as we used this for assessment one and we were familiar with it.

We met every week or biweekly, depending on the required work. We took a short holiday break to let our team members rest.

We had folders on Google Drive containing all the files needed for Assessment 2. Having folders made it much easier to keep track of these original requirements formation. For example, we had a folder called “Team 26 old docs”, which contained all of the documents we inherited from DevCharles. Then we had another folder called “New Docs” containing our updated versions of the documents from DevCharles. This made it easier for us to track all the documents in one place. Towards the end of the project, we created a new folder called “Final Documents” containing the final versions of the documents to be submitted and added to the website.

We created the UML diagrams for the Architecture document using diagrams.net on google drive. We also used IntelliJ Ultimate to create UML diagrams efficiently and effectively, allowing us to update the UML after individual team members made distributed changes.

We used the Agile methodology as our software development framework. We used this as our methodology because it worked so that we could work simultaneously on different parts of the project instead of waiting for individual team members to complete them before we could start the next stage of development. This worked well in our team as none of us were working on the project full-time, and we completed tasks at different speeds.

We created a table to track any changes made to the code and the reasons for the change. This let us track our changes and made submitting this assessment easier. It also meant there was less chance of people accidentally working on the same parts of code, helping manage time. We created this table using the table function in Google Docs. We also tracked changes to the website so that if anyone else needed to work on it they would know what Emily had already done.

We used Git as our version control system, with the repository hosted on GitHub. This is the same stack as our first assessment project, so we were all familiar with how to use it. The team whose code we inherited also used GitHub for their development, so we could simply fork their repository.

b) Changes made to deliverables

The last group's unaltered deliverables can be found at:

jmm889901.github.io/ENG1/docs/original-docs/og-architecture.pdf
jmm889901.github.io/ENG1/docs/original-docs/og-implementation.pdf
jmm889901.github.io/ENG1/docs/original-docs/og-method.pdf
jmm889901.github.io/ENG1/docs/original-docs/og-requirements.pdf
jmm889901.github.io/ENG1/docs/original-docs/og-risk.pdf
jmm889901.github.io/ENG1/document.html

Our deliverables, and the updated versions of the last group's can be found at:

jmm889901.github.io/ENG1/docs/Change2.pdf
jmm889901.github.io/ENG1/docs/Impl2.pdf
jmm889901.github.io/ENG1/docs/Test2.pdf
jmm889901.github.io/ENG1/docs/CI2.pdf
jmm889901.github.io/ENG1/document.html
<https://jmm889901.github.io/ENG1/docs/requirements-phase2.pdf>
<https://jmm889901.github.io/ENG1/docs/architecture-phase2.pdf>
<https://jmm889901.github.io/ENG1/docs/plan-phase2.pdf>
<https://jmm889901.github.io/ENG1/docs/risk-phase2.pdf>
<https://jmm889901.github.io/ENG1/javadoc/index.html>
<https://jmm889901.github.io/ENG1/piazzaPanic.jar>
<https://jmm889901.github.io/ENG1/licence.html>
<https://jmm889901.github.io/ENG1/testing.html>

Requirements changes

We updated the requirements to ensure they covered any new requirements required for Assessment 2. We did not change the first part of the other team's document as they had very clear and concise information about the requirements elicitation process. We updated the requirements table for the assessment two requirements. We also added a few requirements to improve the other team's original requirements table format.

We added several requirements given to us as part of assessment 2:

- Added requirements for game ending upon reaching 0 reputation points (added FR_REPUTATION, FR_GAME_OVER).
- Changed requirements to allow customers to arrive in groups of 2 or 3 (changed FR_CUSTOMER_FLOW).
- Added endless mode requirements (added FR_GAME_MODES, FR_ENDLESS_MODE).
- Added adjustability for scenario mode requirements (added FR_SCENARIO_MODE, changed FR_COMPLETION).
- Added requirements for failing preparation steps (added FR_PREP_FAIL).
- Added requirements for investment system for cooks and stations (added FR_INVESTMENT).
- Added requirements for save files (added FR_SAVE_FILES).
- Added requirements for difficulty level (added FR_DIFFICULTY).
- Added requirements for power-ups (added FR_POWERUPS).
- Added requirements for losing reputation points (FR_COMPLETION_TIME_LIMIT).

We changed these requirements to accommodate the new requirements for assessment 2:

- Changed requirements for recipes to 4 (added pizza and jacket potato) (changed FR_RECIPES).
- Changed requirements for number of cooks to 3 (changed FR_COOKS).

We improved these original requirements format:

- Overhauled the user requirements table to better adhere to the definition of user requirements (and update for assessment 2).
- Overhauled the requirements to be more “objective” (i.e. testable).
- Re-introduce “UR_TIMING” as this can help keep track of how long the user takes to complete an order.

Architecture changes

We updated the diagrams in order to show our own UML diagrams as opposed to the UML diagrams for the old team's original project.

We didn't need to make many changes to this document as it mostly applied to our new architecture as well, except for changing the names of some Requirements where ours may have differed to theirs for example:

- FR_COOK became FR_COOKS.
- FR_CUSTOMER_FLOWS became FR_CUSTOMER_FLOW.
- FR_INGREDIENT_STATION became FR_INGREDIENT_STATIONS.
- FR_COOKING_STATION became FR_COOKING_STATIONS.

We did not have to change anything from the other team's architecture because we kept the same names and components. This ensured it was not confusing for anyone and was easier when implementing new features.

Method Selection and Planning Changes

We extensively changed the Method Selection and Planning deliverables to reflect our different work practices compared to the original team..

Team 26 used a modified version of the waterfall software development framework, adjusted so that a new stage could be started before the last stage was completed. We adjusted the deliverable to reflect our use of Agile. Agile allowed different stages to be completed simultaneously and reduced the amount of time we spent on management and paperwork versus actually developing software.

We updated the section about IDEs to reflect that we did not use the Live Share feature of Visual Studio Code, and did not consider using eclipse. Additionally, we added our use of IntelliJ to generate UML diagrams. In their assessment document, IDEs, Team 26 decided to use Visual Studio Code and IntelliJ. We did not change this. However, we did not use the Live Share feature of VS Code and did not consider using Eclipse so removed these lines. As a map creator our team and team 26 both decided to use Tiled, so we did not change this. We updated the section on Git to show our risks instead of their risks.

We changed the deliverable as we used a different method for making Gantt charts than Team 26. They used Asana, then Monday.com, then plantuml. They also used GitHub's Kanban board management system. Our team used plantuml the entire time. To communicate both of our teams decided to use Discord, and we removed Team 26's considerations of other chat platforms as we did not decide to use other platforms. To store and edit all the documents both teams used Google Docs, but we did not consider Word so removed this from our version of the document.

We updated their details about meeting contents as we ran our meetings differently from Group 26 and did not tend to work together during the meetings. We updated the paragraph about team organisation as team 26 didn't have a “leader” for the first few meetings but then decided that Andrey would become the “leader” to give out tasks and manage the team to work better. We decided to designate Emily as manager at the first meeting as we felt we needed someone to keep us on track.

We updated the paragraph about group splitting as our group setup was different from Group 26's. Both teams decided to split the groups into documentation and coding teams. This was done in both teams due to people having their strengths, and this meant each team member could work with their strengths. This also meant that more work was being done as it was split up. We updated our justification for this in our documentation to show our slightly different approach towards it.

We changed the majority of the Systematic Plan section of the document. This was because our approach to planning was totally different to the other group's.

Team 26 only created a Gantt chart for the implementation of the assessment while we created a Gantt chart for the whole of assessment 2. We split our Gantt chart into different sections based on what we needed to submit. We included this new Gantt chart in our new document, replacing the Gantt charts from the other group.

We also updated the details about holidays in the document. We did not work much over the holidays to let members rest, unlike Team 26, so updated this in the documents. We then had a small group meeting in discord with some members of the group, during this meeting tasks were given out and members were notified that tasks were shown in the minutes of that meeting. We removed their references to GitHub Tasks as we did not use these during our development, finding them frustrating.

Risk Assessment and Mitigation Changes

We added new risks to our new version of the other team's document. We also changed the formatting and colour scheme. We did not feel the need to make many changes to the content of the other team's risk assessment other than actually updating their risks table to make way for any new ones for assessment 2.

Improvements:

Compared with Assessment 1, we have made a lot of new improvements. In this document, we have displayed the risk categories more clearly and used different colours to display the risks and severity. We have also updated any new risks associated with assessment 2.

Newly added risks:

- 1: In-game graphics may not be easily distinguishable on smaller screens.
- 2: The project may be run on screens of different sizes, which may cause black borders or fail to run and affect the user experience.
- 3: The team members are absent from the team meeting due to their own reasons, so they do not know the current progress of the project or what has changed.
- 4: The codebase we have inherited has critical issues which delay development (e.g. no documentation, serious errors, etc).