Change Report

Group 1
Assessment 2

Emma Hogg
Isaac Rhodes
Ali Yildirim
Isioma Offokansi
Seungyoon Lee
Toby Meehan

a) Upon the release of the results for assessment 1, we requested the feedback for team 3. We worked to break down the feedback to clarify what parts of team 3's project needed improvements and what techniques our own group did not incorporate during assessment 1. The full spreadsheets can be found in Figure 1 under the change report section on our website.
The comments in the 'Bad' column is how we decided on what part of the deliverables needed changes. Another way we decided what changes were to be made to the deliverables was the requirements. The requirements mainly impacted the implementation of the project which then influenced many aspects of the change report deliverables. This can be seen in the actual content of the class diagrams (Figures XXX) in the changed version of the architecture deliverable.
The spreadsheet of feedback as well as all subsequent changes to the deliverables were made using Google products such as Google Docs and Google Sheets. We used these products to allow for simultaneous work on the deliverables from any device with minimal additional hardware and applications which helped to mitigate R2 in the new requirements report.

Having a spreadsheet for both our project and team 3's project reflected many similar issues that the projects had such as some diagrams being difficult to read due to size or quality. However, the spreadsheets also caused a few minor conflicts in regards to what changes needed to be changed because they provided multiple options in how to continue with the project.
An example of this would be in the methods and planning deliverable. In assessment 1, our team assigned specific roles to each member so people knew what they were responsible for, team 3 did not assign roles to their members. We discussed whether we wanted to keep the team roles and after some deliberation we decided that we worked more efficiently when people were assigned to admin roles on top of the work that needed to be done. Many differences from the spreadsheet and deliverables were settled like this.

Using the assessment PDF and our feedback spreadsheet, we divided the work that needed to be done for assessment 2 and used Trello to create individual tasks. Each member of the team chose tasks that they felt they were most suited for. Members were mainly responsible for the work needing to be done for their chosen tasks, including whatever changes needed to be made to deliverables.
We believed Trello to be the most suitable task manager because of its features. Trello allowed for flexible task delegation and allowed us to add comments, dates, and attachments to be added to individual tasks. The visualisation provided by Trello greatly helped understand the tasks and our progress with them.
To ensure everyone was happy with the direction and progress of the deliverable changes, weekly meetings were held through the runtime of the project where feedback was given to any changes made between meetings.

The following pages provide detailed documentation of what changes were made to the change report deliverables and why those changes were made.

Requirements

Old: Old Requirements
New: New Requirements

Below, please find a table describing any changes that were made by our team to the Requirements document and the justifications as to why we changed them.

ID	Change Made Explanation	Justification
1	Alterations to the Introduction of the document.	We noticed very little was said about meeting with the customer in the introduction which we carried out both for Assessment 1 and 2. We felt this was important to add to understand how these conversations influenced our requirements for the game. We also noticed the lack of distinction between user and system requirements therefore we understood the need to clarify this to ensure the document was not confusing to the reader. Finally, they included no references to previous research proving the reliability of their technique. We understood the importance of clarifying this to prove the method we took was reliable and effective for the project.
2	Added Endless Mode requirements to tables	The product brief mentions the need for a new gamemode endless mode as a vital requirement therefore we needed to include this in the deliverable.
3	Remove the user requirement UR_NUMBER_OF_CUSTOMERS and addition of FR_CUSTOMER_ARRIVAL	This requirement is a system requirement rather than a user requirement therefore we decided to move this into the functional requirements under the name FR_CUSTOMER_ARRIVAL. This ensured that the document was accurate so that everything in the game was implemented correctly. We also had to update the customer arrival functional requirement to cover the new requirements for Assessment 2.
4	Change some user requirements to fit Assessment 2	In Assessment 1 there was a maximum of 2 cooks and only two recipes. We had to change these requirements to support assessment 2 which requires 3 cooks and four different recipes. Also the number of cooking stations can now vary in the game as they can be bought back therefore we had to alter this requirement. This was important to add to ensure we were covering all that is needed for the new product brief.
5	Change the priority for UR_REPUTATION_POINTS	The reputation points are a requirement of the product brief to implement in Assessment 2 therefore this is important to change as this is a necessity.
6	Add a user requirement of UR_MONEY	A new requirement for assessment 2 is that the user can buy back cooking stations therefore they require a way to buy these back. To do this we need to have a form of money they can earn. This requirements specifies how the user can earn money that can be used to buy more stations
7	Add the user requirement UR_BUY_ITEMS	We added this user requirement as in the product brief there is a need to be able to buy back cooking stations while playing the game using the money the user has earned. This is therefore a necessity to implement and therefore is an important requirement to add.
8	Alter the UR_UX priority from May to Shall	One of the stakeholders for this project is potential future students. The game therefore needs to be attractive and catch the attention of the young adults. This is therefore important to implement as this is 50% of the stakeholders for this product.
9	Alter the description of UR_PLATFORM	The requirements of the game is for it to run on Windows and Linux therefore we changed picking two to be the specified two required.

10	Remove user requirement UR_AESTHETICS	This is a user requirement that should not have been included as it should be a functional requirement. The user interface should have been kept at a higher level of abstraction here so we removed this requirement.
11	Change UR_ACCESSIBILITY description	This user requirement should be a functional requirement therefore we have changed the description to be specific to the user. They need to be able to navigate the game easily therefore made alterations to cover this. The functional requirement will cover the specifics.
12	New user requirement of UR_TIMER	This requirement should have been implemented in Assessment 1 and it was however this was not noted down in the table. Therefore, adding this to the table was crucial to ensure traceability for requirements and testing.
13	New user requirement of UR_MINIGAME	When the user is chopping, frying and baking they need to play a short minigame to ensure that they do not fail the preparation step. The user must be able to do this and hence we felt it was important to add to the user requirements
14	Addition of the user requirement UR_PREPARATION_FAIL	In Assessment 2 it is possible to fail both the cutting and the cooking steps therefore this must be implemented into the game hence we added this requirement which covers this.
15	Addition of the user requirement UR_POWER_UPS	The user should be able to activate power ups in the game to aid them during the game. This is a new essential requirement and hence was important to add to the requirements table to ensure we implement it.
26	Addition of user requirement UR_DIFFICULTY	For assessment 2, the user should have an option as to what difficulty scenario mode is in. Therefore, this is a requirement that we needed to add to ensure that it was implemented.
21	Addition of user requirement UR_SAVE	There is a new requirement for the user to be able to save the game in its current state for assessment 2. This was not applicable in assessment 1 and therefore needed to be added to the table of requirements.
22	The addition function requirements for making the new items	This is an important functional requirement to add was the capability of allowing the user to make jacket potatoes and pizzas for the customers as specified in the product brief
23	Additional FR_BAKING requirement	The user requires an additional station to bake potatoes, burger buns and the base of the pizzas therefore the system must support this additional station.
24	Changing the cooking station functions allows users to fail the step.	It is possible to fail the cutting step, baking step and frying step in assessment 2 and this was not previously considered in the original requirements therefore this must be added to the requirement already listed and must be linked to the new user requirement UR_PREPARATION_FAIL.
25	Addition of the functional requirement FR_TUTORIAL	After speaking to the customer we found that having a tutorial in a separate place to the actual game would be preferred therefore we needed to update this requirement in the functional requirements table to change this.
26	Removal of the requirement FR_PREPARE_STAGE	This requirement was not very specific therefore we have specified the steps that can be failed in the specific station requirements. This helps with clarity over the requirements.
27	Addition of functional requirements supporting the need for power ups.	The system is required to show power ups randomly throughout the game for assessment 2. Therefore, the system will support the fact that if a chef interacts with a power up they will then obtain the powerup and the user will be able to see a visual that shows it is activated. This is therefore described over multiple requirements to cover all needs.

28	Addition of functional requirement FR_MINIGAME	The system must support the new user requirement of being able to play a minigame. The system must provide the minigame to determine whether the cooking/chopping/baking stage has passed and hence is an important requirement to include.
29	Additional of functional requirement supporting the difficulty setting	There must be a menu where the user can choose the difficulty of the game. The game will then have to implement the difficulty chosen. This is a new requirement for assessment 2 therefore is vital for the system to support.
30	Addition of functional requirement FR_BUY_ITEMS	The game must also support the fact that cooking stations can be bought back in game. This is part of the product brief, hence this had to be added as a functional requirement to the requirements deliverable. We have not included chefs in this due to speaking to the customer who advised that this is not required depending on the number of stations you choose to have. In our case, the game would become cluttered going against the needs for the customer.
31	Addition of functional requirement FR_MONEY	The game must support the concept of the user having money as a big functionality of the game. This is the basis of buying stations therefore this is a requirement that must be implemented to fulfil other requirements.
32	Addition of functional requirement FR_SAVE	The save mechanism is something that the Assessment 2 product brief states is required. Therefore, this is something that the system must support. This is a functional requirement that must be added to the game.
33	Change the description for NFR_OPERABILITY	We noticed that the description for this non-functional requirement was not very specific (“by new player”). Therefore, we felt it was important to specify who the new player would be and the operating system on which they would play the game to ensure this requirement has enough detail to confirm whether this has been met or not. Previously, it would have been difficult to be certain that this requirement had been met or not.
34	Change the fit criteria for NFR_RESPONSIVENESS	Previously, the document stated 1/60 seconds as a value with no proof as to why this value was chosen. We therefore did some research as to what the latency should be on average in current video games at 30 fps. We discovered that this value is accurate but felt the need to add some proof to back up our choice in this requirement ensuring we showed reliability.
35	Remove the non-functional requirement of NFR_TIMER_PRECISION	This requirement is something that we feel is difficult for us to guarantee and is not very clear. The OS controls this rather than the game. We felt this was not a requirement that we needed to look into so we removed this.
36	Change the fit criteria for NFR_AVAILABILITY	The idea of playable state we felt was very vague and we needed to update this in order to be able to confirm whether we have met this requirement or not. Therefore, we changed this to update what a playable state means and therefore added some clarity to the requirements. This is vital for the success of the game.
37	Change the fit criteria for NFR_PLATFORM_AVAILABILITY	After reading an online source, we noticed 95% does not cover the platforms that the game should be able to run on. We therefore altered this to make the fit criteria cover the true percentage. This was vital to add as otherwise we have a requirement that would contradict with a previous requirement regarding the platforms it should run on.
38	Remove the UR_FRAMERATE and create the non-functional requirements NFR_FRAMERATE	The framerate is not a user requirement but instead a quality that the system should have to improve the experience for the user. We also changed the fit criteria to be 30 fps over 60 fps as according to research this is the average at which video games should run. This was important to add so that we are not trying to meet unrealistic expectations.

Architecture

Old: [Old Architecture]
New: [New Architecture]

ID	Change Made Explanation	Justification
1	Plant UML justification	The original owner of the project used and referenced PlantUML but gave no reason as to why they used it. We felt that PlantUML was a crucial part of our project because it allowed us to create clear diagrams
2	Class diagram is broken up into packages	The original class diagram was very difficult to read even after zooming into the image. Since everything was already in packages we decided to separate them out. This provided very clear images within the document for the most part and all the images were added to our website so even the diagrams that had a slight drop in quality or an orientation to fit on a page could still be viewed clearly.
3	Sequence diagram	The sequence diagrams were confusing and lacked proper responses but the information that they were trying to convey was good. We reformatted both diagrams to allow for a more accurate representation of what was occurring in those sequences.
4	Removal of state diagrams	We felt as though, with the other clarified diagrams, we had enough structural and behavioural diagrams to convey the information in the deliverable.
5	Changed CRC cards	The original CRC cards had valuable information but did not contain stereotypes and were therefore RDD was not used. We changed them to more accurately portray an actual CRC card.
6	Changed the contents of the user requirements table at the bottom	As we started Assessment 2 we had an interview with our customer which greatly changed the requirement we had to fulfil. We changed the contents of the content of that diagram to show how we wanted to initially implement some of these features

Method Selection and Planning

Old: Old Plan

New: New Plan

ID	Change Made Explanation	Justification
1	Additional paragraph to describe the analysis/design/ implementation of the project	Team 3 gave no description on how they formed the basis for their implementation so we added a concise outline on how our project developed. We believe that adding this gave more clarity to how our group
2	Provided an explanation to why GitHub was preferred over other code repositories	Both the owners of the original project and our team used GitHub to hostour projects but team 3 did not provide any information about their experiences with other code repositories. We felt that including this was important to include because it led to other important decisions, such as using GitHub Pages to host our website.
3	Specify which IDE were used and why they were used	The original deliverable gave an array of IDEs that were used in the project but gave no explanation on the impact using all of them had on the workflow. We refined the list of IDEs used and provided sufficient reasons, requirements, and impacts that using them had on the project.
4	Additional reasoning for our choice of document manager	Our use of Google products was the same as the owners of the original project and some of the reasons that they made for using them were the same. However, we felt as though we should further explain our choice and mention our experience with other products within the same field so it was clearer why we ended up using Google Products.
5	Specified who was given what roles within the SCRUM methodology and gave descriptions on what those roles responsibilities were	In the original document we noticed that team 3's agile development style matched our own but they gave no in depth details as to who was assigned crucial roles or what those roles were responsible for. We felt as though this was a very crucial piece of information as knowing who was responsible for what made the workflow in our group smoother. We made sure to add this information into our deliverable.
6	Provided SCRUM sprints information via a logbook	We added this to tie in with the rest of our experience with the SCRUM methodology and the progress of our project as a whole.
7	Changed details about how we managed sprint meetings	The original project mentioned how they handled meetings but it didn't reflect the flexibility we tried to incorporate into our plan. We believed that that flexibility greatly helped to reduce stress and improve communications within the team so it was important to mention.
8	Added a paragraph about our use of a task manager	In the original document we noticed that team 3 did not mention how exactly they arranged the tasks that needed to be done. We used Trello which helped us lay out our tasks visually and clearly assign them to people. This helped in team organisation and ensuring people were not working on the same tasks unnecessarily so it felt we had to mention
9	Provided more personally tailored reasons for our choice in communications	While the original project mentioning Discord as their form of online communications resonated with us, the reasons that they provided were different from our own. We believed it necessary to reflect that difference in our deliverable.
10	Removing Gantt charts from	The Gantt charts did not reflect our movement through the project, of

	the bottom of the deliverable	course, so we remade them. We believed that having our weekly Gantt charts on the deliverable wouldn't be useful because cramping them onto small pages ruined the clarity of the images, as seen by the original project's Gantt charts, so we added clear versions of our Gantt chart onto our website and added a link to the end of the document.
11	Changes to the plan	Our planning greatly differed from the original owners of the project, of course. The unfortunate absence of one of our team members greatly changed our initial plan so we had to adapt to a greater workload for each member. We felt that mentioning the changes to our plan was very important.

Risk Assessment and Mitigation

Old: Old Risk Assessment
New: New Risk Assessment

After reading the risk assessment, we felt that it was important to add some specific risks for Assessment 2. We were aware that further risks may appear when taking over another group's project and therefore felt we should add these to the current table of risks. If any of the new risks were to occur we would therefore be ready with the knowledge of what to do in order to reduce the impact of the problem. This ensured that this part of the project would run smoothly and effectively.

Below is a table describing any changes that were made by our team to the Risk Assessment and Mitigation document and the justifications as to why we changed them.

ID	Change Made Explanation	Justification
1	Additional paragraph included to the introduction describing the risk monitoring	We noticed that the previous deliverable included no information as to how they carried out risk monitoring. This is an important part of the document therefore we felt it was important to include some information as to how we did this in Assessment 2.
2	Additional column to the risks table named "Happened?"	The extra column to this table was necessary in order to note down our risk monitoring techniques. This column was used to show when a risk happened, what we did to mitigate the impact and how this impacted our project. We felt this was vital to include as a part of our risk monitoring process.
3	Add a paragraph to explain what the column "Happened?" means and what it will include	The "Happened?" column could be very confusing to someone reading the document without any explanation. We, therefore, included a description as to what information you could find in this column of the table to ensure that the deliverable is clear and easy to read and understand.
4	Update the "Happened?" column for the risks that occurred over the project.	We lost a member of our team in the early stages of the project. It was important to update that this had happened in that table so all of the team understood what we needed to do and so we progressed as efficiently as possible with minimal impact. Not only that but we had problems with bugs in the code and tests failing during development. We therefore included that this had happened in the column to ensure it was known as to how we solved this problem with minimal impact on the project and so the whole team were aware of the occurrence.
5	Update the mitigation technique for risk R11. Change it to researching prior to starting the project so that we are prepared if a library is discontinued.	We noticed that this risk mitigation technique would not really prepare us in a situation where this risk occurs. We therefore felt this was important to update to ensure that if the risk does occur we are prepared. The new mitigation method would result in the fact that we would be prepared if a library is discontinued as we will know alternatives to this and be able to switch fairly easily minimising the impact of this.
6	Update the mitigation technique for risk R14. Change to look at research other options to ensure we are prepared to switch software to reduce the impact of this if it occurs.	After reading this risk assessment we felt that risk R14 did not reduce the likelihood nor the impact of the risk occurring. We therefore felt that if the software did not have good support for technical issues we would be hugely impacted and the risk would not be mitigated. Therefore, we felt we had to come up with a better solution to mitigate the impact than the one given.

7	Addition of the new risk R17 linking to there being a bug in the game.	This project had not previously been tested to ensure that the game had no bugs in it therefore we knew there would be a risk that we could find one while testing. There is also the risk that we could produce bugs in the game while we continue to develop it for Assessment 2. We therefore needed to be prepared as to how to deal with this if it were to occur. We therefore discussed the likelihood and impact and added a mitigation technique that would help us to minimise the impact.
8	Addition of the new risk R18 linking to the fact that some of the tests may fail and therefore code needs to be fixed	This project had not previously been tested resulting in there being a risk that some parts of the code need to be fixed before we can progress with the implementation. We therefore do not want this impact to be large, holding back progress. We discussed the impact and how we will react if this occurs within the Mitigation column. Due to these reasons we thought this was important to add to the table.
9	Addition of the new risk R19 linking to the fact the game could become boring and not therefore meet the needs of one of the stakeholders	When implementing the game it is very easy to focus on functionality and forget about one of the main aims of the project: for it to be attractive to potential future students. We therefore felt it was important to include a risk that covers this possibility. It will ensure that we stick to the task in hand and if we go off course we can return to make meaningful progress towards the enjoyability factor of the game quickly. We would be neglecting one of the stakeholders if this risk were to occur so it is important to include.
10	Addition of the new risk R20 linking to the controls of the game becoming too complex	As the game becomes more detailed and complex there is a risk that the game could become too complex for the player to play. We therefore need to consider what to do if this occurs. We felt this was important to add as we would be neglecting the needs of one of the stakeholders (the student) if we do not consider this. Our mitigation technique ensures that the customer is involved if this were to occur.
11	Addition of the new risk R21 linking to there being problems during the presentations	When presenting there is always a risk that something can go wrong. We therefore need to be prepared to recover if this were to occur. If we are not prepared there is a big risk that this could go badly wrong which would have a big impact on the success of the game. Therefore, understanding how to prevent this impact is very important to include in the risk assessment.