Change Report

Group 5 :
Callum Watson
Jack Guan
Craig Slomski
Yaseen Khan
Chase Mo
Kamrul Islam

Process of Software Evolution:

Introduction:
Over the course of the making changes to the original game there are three different components to maintenance:
1) Repair software faults: Refactoring the code to correct any existing errors
2) Adapting software to a different operating environment: Ensuring the code is suitable for the change in testing environment & development environment.
3) Adding and modifying system functionality: Adding the remaining Assessment 1 functionality that were not initially implemented.

We will be using system re-engineering as opposed to methods such as scrapping the old system and replacing the old system with a new system. As re-engineering has the added benefit of reduced risk (linked to risk register R) and reduced costs (limited time deadline surrounding the project).

Conventions:
Naming conventions / referencing systems for Requirements, Risks, Architecture Diagrams have been adjusted to reflect any removed components, components from the previous team and components added by the current team. They are as follows:
ExistingReference_R: Removed Components
ExistingReference / _1: Initial Components (which have been unchanged) for Assessment 1
ExistingReference_2: Refactored/Added Components for Assessment 2
This method is used to improve traceability whilst not collapsing the existing referencing system used to link the whole project together.
In addition, to this we will use a Change Log table, to track when changes have been made covering: Change reference, responsible team member, change date, description of change and whether it was approved and who by. This convention was used in order to have a clear log of all changes made, so we can distinguish previous teams work with our own work, know who is responsible for changes and be able to revert changes if changes are not approved.

Tools Used:
Tools to keep track of changes include Google Drive where we will have existing backups of the original team's documentation for referencing throughout making changes.
In addition we will be using Google Docs to edit our current project in order to utilise version control to keep track of changes.
Alongside this, using Gantt Charts to date the changes and show which team members are working on which adjustment.
We have decided since this is a small scale project the above explained free software will suffice as it enables us to make changes, and clearly trace and track them with ease. If this was a bigger project with more variables, we would opt for more professional software.

Requirements (RE):

Change Reference	Responsible Team Member	Change date	Description of change	Approved by
CRE0	Kamrul	01/03/23	Updated user requirements to reflect Assessment 2 Brief and change in stakeholder requirements	Callum
CRE1	Kamrul	20/03/23	Simplified SSON to be precise and to-the-point. Updated UR, FR and NFR to further reflect Assessment 2 requirements. Amended and implemented requirements which were initially supposed to be included in assessment 1.	Callum
CRE2	Kamrul	26/04/23	Added extra requirements to fulfil feedback given to Assessment 1's original implementation. Extra requirements added which were planned from the beginning of Assessment 2.	Jack

Adding new requirements to meet change in customer's needs
- Implement five special power ups that chefs can obtain (e.g. speed increase, shorter cooking times)
- Implement support for different levels of difficulty in the game (e.g. easy, normal, hard)
- Implement facilities that allow players to save the state of the game at any point and resume a saved game later

Original and updated requirements can be found under the deliverables section on the website.

For assessment 2, the full product brief required implementing which contained new features which includes, but not limited to: 3 chefs, endless mode, reputation points etc.
These features were appended to the necessary tables within the requirements document, along with the priority of implementation of the said features.
It is necessary to maintain a clear distinction between assessment 1 and 2 features with assessment 2 taking higher priority in relation to implementation. Requirements suffixed with '_2' show that it is a requirement that must be fulfilled in this stage of the project. This includes, but is not limited to, features required for assessment 2 and features required but not implemented for assessment 1 - this means that features that were deemed to have a 'May' or 'Should' priority may not be as important at this stage of development.
In summary, there is not a significant change required in this deliverable, however, it must be noted that it is the base of the project and its features and should be referred to on a regular basis, as well as updated when necessary.

Changes & Justification:

The changes made were simple with mainly one overall goal to meet which was to set a base for the changes needed to be made to reflect assessment 2's objectives. This provides a set of goals, in relation to the product brief, to complete the implementation of the game. Once the 'Shall' requirements are met, there should be further work to implement the other priority requirements, which also may not have been implemented previously in assessment 1. There is a clear priority ranking system, which was followed when adding the new requirements, to aid the group in prioritising which features to add and which features may, or may not, be added, providing there are no restraints regarding time and resources.

Architecture (AR): (Export all UML diagrams and any that don't fit can go on the website**)**

Change Reference	Responsible Team Member	Change date	Description of change	Approved by
CAR0	Jack & Callum	08/03/23	Updated Sequence Diagrams to reflect new requirements	Callum
CAR1	Chase	08/03/23	Updated Class Diagrams to reflect new requirements	Callum
CAR2	Callum	02/05/23	Updated UML Diagrams for after development	Kamrul

Changes & Justification:

CAR0
To avoid clogging up the document, we extended the methods used to generate sequence diagrams similar to the previous team, using PlantUML. Hosting them on our website with links to the images and which requirements they are based on.

CAR1
We extended the methods used to generate class diagrams similar to the previous team, using PlantUML. Also hosting them on the website.

Sequence Diagrams:

Provided by Team 5: Runtime Terrors

Diagram	Requirement
Customer_Patience.png	UR_FAIL_STATES_2
Reputation.png	UR_FAIL_STATES_2
Powerup.png	UR_DIFFICULTY
SaveAndLoad.png	UR_CONTINUE_2
Screen_Navigation.png	UR_FUNCTIONAL
Unlocking_Chefs.png	UR_CONTROL
Unlocking_Stations.png	UR_CONTROL

Architecture Diagrams

Architecture Diagrams added for Assessment 2 have been listed below and the user requirement relating to those diagrams.

Class Diagrams:

Provided by Team 5: Runtime Terrors

Diagram	Requirement
GameMaster_Scenario_class.png	UR_LAYOUT, UR SOUND
Machine_And_Utility_Class_Diagram.png	UR_ACTION_TIME
Person_class.png	UR_CONTROL
Save_Classes.png	UR_CONTINUE_2
Screen_Classes.png	UR_FUNCTIONAL
Entire_Architecture_Diagram.png	For reference

CAR2
Similar to the previous team, we generated UML for the post implementation in order to review the changes made across the development, this can be found on the website.

Method Selection and Planning (MP) Change Log:

Change Reference	Responsible Team Member	Change date	Description of change	Approved by
CMP0	Callum	01/03/23	Adjustments to team composition to reflect the new team and addition of weekly meetings, an essential part of team communication.	Kamrul
CMP1	Callum	01/03/23	Adjustments to tools used to reflect tools that we used in order to be successful in Assessment 1 which we will use. In addition to tools the previous Team used which we have adapted.	Kamrul
CMP2	Callum	01/03/23	Replaced Systematic Plan with updated Gantt charts and plan for Assessment 2	Kamrul
CMP3	Callum	08/03/23	Elaborated further detail on the Scrum Agile method to show a change in the planning of this project and reformatted document to meet requirements	Kamrul

Changes & Justification:

CMP0

It was essential to change the team composition section as it was very specific to the previous team's structure and was built around having a team of 7 whereas we are composed of a smaller team composition of 6 and it is important to play to different strengths.

Team Composition:

Keeping with the principles of the agile methodology to assign roles based on the core strengths of each of our team members discovered through Assessment 1. With half of the team (Chase, Jack, Yaseen) really familiar with coding and implementation and the remaining 3 members (Kamrul, Callum, Craig) of the team being able to support them through testing and continuous integration. In addition, having 2 members familiar with the documentation (Callum and Kamrul) who are able to lead the planning of the overall second stage of the project.

In order to set manageable tasks for each week, weekly online/in person meetings will take place where each of us will set ourselves goals for the week and will be replicated in a Gantt Chart (made in plantUML) with deadlines and team allocation.

To summarise, we have broken down the assessment into its constituent parts, divided our team according to proficiency in writing, testing, continuous development, implementation and project management, and distributed the workload according to these divisions. All of which have a supporting team member in the event a team member falls behind in their workload.
In addition to weekly meetings to track progress and visualising that progress and setting deadlines through the use of Gantt Charts.

CMP1

Despite most tools staying the same, we can simply adapt the tools the previous team have used to suit our needs such as Tiled. However, tools we have become familiar with in Assessment 1 (Such as plantUML) that also suit our needs, we have included to make the most out of available resources.

Tools:

During the handover to Assessment 2, Google Drive had further uses. It was used to store deliverables and documentation as it has a built-in Word processor as well as version control to allow team members to see contributions to the document which was immensely useful during this phase. During this phase we updated requirements from our customer and used that to update the system and user requirements and constraints. This then informed our updated design of the architecture and the UML diagrams we made such as class diagrams and sequence diagrams. To assist this process plantUML was used again as it provided tools to design UML diagrams, furthermore it allowed for the creation of gantt charts to help log our progress and illustrate the overall timeline of the project. We chose this over technology such as Draw.io, plantUML would ensure consistency as multiple team members would contribute to the architecture diagrams.

For the Implementation Phase GitHub was the main collaborative tool used. It allowed for easier project synchronisation with its features such as pull, push and branching. It also has conflict resolution support which Google Drive lacks to help support the development of the project. In addition, GitHub is useful in the Testing Phase as it allows testing to take place in dedicated testing branches and allows for automatic tests to take place in order During the actual implementation of the project we decided on using LibGdx, a game engine using Java. This was chosen over another engine in consideration Litiengine due to its more active community and more extensive documentation available.
Tiled was used for the tileset and collision of the map in the previous team's assessment and we have adapted that in order to expand the map to import into LibGDX

CMP2

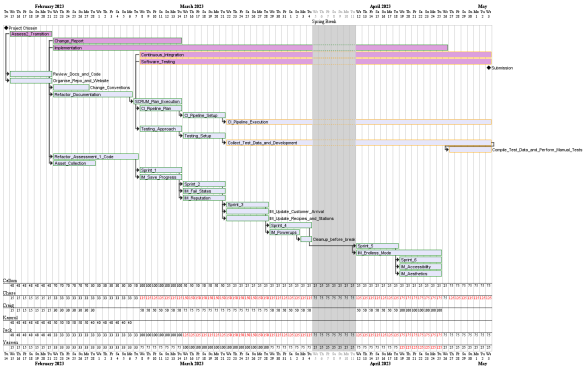
A rough plan was introduced on 01/03/2023 that was incrementally developed with sprint reviews.

Planning Testing and Continuous Integration ; Sprint 1: IM_Saving_Progress ; Starting: (8/3/2023)

Sprint 1 addressed the features that posed the highest risk to the project according to our project backlog which was being able to save progress and return later.
Meanwhile the team members not currently implementing as this big change took place focused on writing up the planning behind Continuous Integration and Testing, allowing us to trail and set up these in the following week. Ended: (14/3/2023)

Sprint Review:

The previous three points in the product backlog have been addressed previously before key sprints were brought into this. However, in this sprint we have achieved the one goal we set ourselves. It was useful having one goal as this feature requires the most refactoring and would have been difficult implementing features in parallel to this one since it affects a significant portion of the game. This was successfully implemented over the week giving us an opportunity to focus on planning consistent testing and continuous integration for the rest of the project.
Gantt charts have been updated to reflect the whole project plan.



CMP3

See for more detail in the attached Plan2.pdf document on our website. The SCRUM method used by the previous team was justified but not fully detailed. We have since elaborated on that and have focused on the changes in the sprint planning [1] and included a product backlog [2] (a sample shown in the screenshot)

Feature and User Requirement:	Note:
As a player, I can see a leaderboard of local high scores from Scenario mode UR_LEADERBOARD_2	Implemented prior the SCRUM change, Assessment 1
As a player, I can hear background music whilst playing the game UR_SOUND	Implemented prior the SCRUM change, Assessment 1
As a player, I can unlock up to three chefs UR_CONTROL	Implemented prior the SCRUM change
As a player, I can save my progress and return later on an open day UR_CONTINUE_2	Deemed most high risk so prioritised first. Most potential future refactoring
As a player, my mistakes cost me reputation points UR_FAIL_STATE_2	
As a player, I can see fail states that cause me to lose reputation points UR_FAIL_STATE_2	Depends on having reputation points to lose

Risk Assessment and Mitigation (RA) Change Log:

Change Reference	Responsible Team Member	Change date	Description of change	Approved by
CRA0	Callum	01/03/23	Updated existing risks and removed redundant ones	Kamrul
CRA1	Callum	01/03/23	Changed Team Member Allocations	Kamrul
CRA2	Callum	08/04/23	Added Risk Log	Kamrul

Changes & Justification:

No major changes were made to the Risk Management Plan as it was very well explained and very adaptable to our portion of the project.

CRA0
It is important essential to keep the risk register updated to handle the current scenario, hence newer versions of the risks were made such as R9_2 and R10_2 which focus on testing and refactoring which were not handled in adequate detail (and were lower risk) which are now more relevant to Assessment 2. In addition, the following risks were removed so we have easy access to the key risks relevant to this section of the project. In addition to adding colour coding. Further reasoning:
R3 and R4: Were two risks that could be covered by one risk instead, the updated R1_2
R7 and R13: Were two risks that could be covered by one risk, the updated R5_2

R9_2	Technology	Big changes to the implementation side of the project lead to major refactoring required later.	M	H	Handle big changes first, in order to reduce the amount of refactoring needed	Implementation Team	Elaboration on key risk for part 2
R10_2	Technology	Updating the project may lead to certain requirements such as non-functional/functional requirements no longer being met.	H	M	Ensure to use a test-driven approach so each update has to pass a series of tests which are designed in regards to requirements. Including automatic unit tests and a short list of manual tests where needed	Testing Team	Elaboration on key risk for part 2
R1_2	Project	Lack of clear quick communication with the group, not utilising SCRUM agile methods	H	M	Discord Server to allow communication outside of Meeting Times which are: 4 hr/week during term time set aside for meetings 1 hr/week outside of term time set aside for SCRUM meetings	Project Leader	More clear, concise risk with a nod to agile methods
R5_2	Project	Code Files/Documentation stored on an independent device, not accessible to other team members	M	M	Push latest version any code (UML, Java, Markdown) onto Git with descriptions of changes. Always use Google Docs to edit the documentation files	Team (lead by report editor and github organiser)	More clear, concise risk

CRA1
The previous group had specific names of members from their team assigned to each risk, so therefore we had to change it to reflect our team structure. This included putting the responsible member's job title as the responsible party.

CRA2
Added in order to show whether risks had been taken or not, particularly focusing on the bigger risks that required alterations in the mitigation strategy.

Risk Log:

These are risks that were encountered that were considered high priority or were handled differently to how the mitigation strategy suggested based on the context.

Date: 27/02/23
Risk: R5_2
Code Files/Documentation stored on an independent device, not accessible to other team members
Context: Team 6 website which was hosted on GitHub pages went down which was being used to reference the website changes that we were making since it was a new format of website creation using markdown to which we were initially unfamiliar.
Solution: We had cloned Team 6's website repository and whilst figuring the specifics for hosting on GitHub pages, we found a free tool online to preview the website pages
https://markdown-it.github.io/

Date: 03/03/23
Risk: R1_2
Lack of clear quick communication with the group, not utilising SCRUM agile methods
Context: A couple of weeks in, we had not gotten an overall plan together for SCRUM implementation and what should be prioritised, which is a huge risk as it threatens large amounts of refactoring implementation later and no sense of direction or deadlines for the project.
Solution: Following the risk mitigation strategy, we hosted 2 hour bi weekly meetings during term time and 1 hour weekly meetings outside of term time. In addition, we filled out the systematic plan and laid out the Gantt Charts for future weeks

Date: 08/03/23
Risk: R9_2
Big changes to the implementation side of the project lead to major refactoring required later.
Context: With the change in requirements to add save states into the game, after taking the time to plan how this would be implemented, we came to the conclusion that this would be a difficult feature to implement and would require changing lots of prior code.
Solution: Following the risk mitigation strategy, we decided to prioritise this feature to be implemented first, in order to reduce the amount of refactoring the code would require later.

Implementation Change Log:

Change Reference	Responsible Team Member	Change date	Description of change	Approved by
CRI1	Jack	26/04/23	Licensing was updated and an update on the non-implemented features stated in requirements.	Kamrul

Changes & Justification:

CRI1
To obey and credit licensing rights, sources used in assessment 2, in addition to the original sources, were added to the licensing table along with links to the respective rights.

References:

[1] *The 2020 scrum GUIDETM* (2020) *Scrum Guide* | *Scrum Guides*. Available at: https://scrumguides.org/scrum-guide.html#product-backlog (Accessed: March 8, 2023).

[2] Lucid Content Team (2019) *How to develop a stellar scrum product backlog*, *How to Develop a Scrum Product Backlog* | *Lucidchart Blog*. Available at: https://www.lucidchart.com/blog/how-to-develop-a-product-backlog-in-agile (Accessed: March 8, 2023).