

Change Report

Cohort 1, Team5:

Lewis Mcshane

Adam Howard

Morgan Davis

Road Gibbons

Zijun Zou

Muhidin Muhidin

Formal Approaches to Change Management

Change management refers to documentation of the changes that have been identified from the requirements, planning and the implementation of the specified changes to achieve the desired result (product).

There are various strategies and approaches when it comes to implementing change management. Our team has adapted an approach that is very close to Kotter's change management theory. This is because we thought Kotter's theory has a step by step model which is straightforward to follow and to incorporate into our project. In addition, we thought this approach would give us the flexibility to accept changes and prepare for it as we see fit [1].

Change Management Workflow

According to aligni's approach [2], ECM workflow starts once a change requirement is identified. Then the person who is responsible for the change requirement creates a change request (CR). After further discussions about the (CR) then a change order (C) is created using the change request (CR).

The following table gives details about participating members and each of the assessment components that have been assigned to them according to the Stage 2 of Kotter's theory on change management.

Assessment 1 deliverables to be changed	Participating members
Requirements	Zijun
Architecture	Morgan
Plans	Lewis
Risk Assessment	Muhidin
Assessment 2 deliverables	Participating members
Website	Muhidin
Change Report	Everyone
Implementation	Adam & Morgan
Testing	Roan
Continuous Integration	Morgan

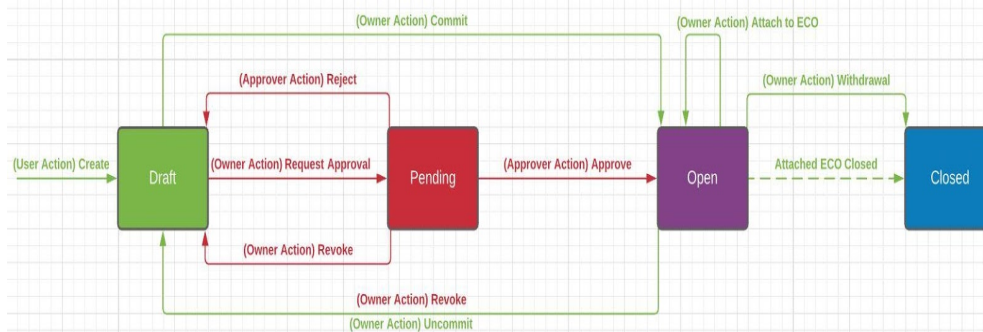
The whole process from when the change is request to the incorporation of the change can be summarised into the following steps:

- 1) Create an ECR
- 2) Add a through description, indicate affected parts and identify relevant participants (team members).

- Using steps(1 and 2), each participating member created a change request(CR) on the relevant assessment component that was assigned to them according to the tables above.
- 3) Discuss the change request(CR), agree and approve the change request.
 - Group meeting on Mondays and Thursdays and the facebook group chat were used discussions
 - 4) If approved (the workflow is enabled)
 - 5) Create the change order(CO) from change request(CR)
 - 6) Create necessary part revisions to include changes
 - 7) Discus further, agree and approve part revisions to include changes
 - 8) If approved (workflow is enabled)
 - 9) Release effected part revisions:
 - To avoid confusion a numerical naming convention was used.
 - For example, part revisions have number 2 extensions. Requirement 2 represents a part revision including the changes from Requirement 1 and Plan2 represents a part revision including the changes from Plan1 and so on.
 - 10)The change order (CO) is closed
 - 11) And finally the change request is closed too.

The described processes and how the workflow is applied to both the change request(CR) and the change order(CO) are precisely captured in FIGURE1 AND FIGURE2 respectively. Further details on the changes made on the deliverables from assessment1 from Eng1 Team4 can be found in next section under the name Proposed Changes To Assessment1.

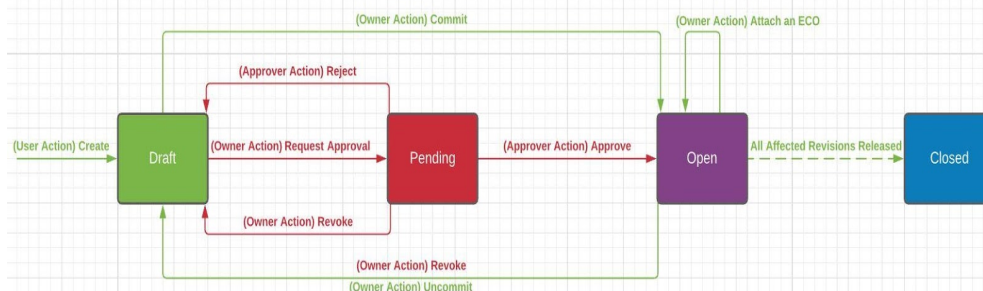
ECR Workflow



Approval Workflow

Simple Workflow

ECO Workflow



Approval Workflow

Simple Workflow

Proposed Changes to Assessment1

In this section a brief explanation and justification of any changes made to Assessment deliverables from the other team is given.

It should be noted that we have taken over the project from Eng1 Team4 - Sample Text.

The precise location where the deliverables are held is under the menu tap Assessment1 Deliverables at : <https://eng1team5.github.io/assessment1.htm>

I. Requirements

1. The requirements do not fit the requirement.

Because the user interface should be easy and intuitive to navigate. The previous **UR_RULES** was limited for user requirements, so it was changed to **UR_UX**.

- **UR_UX** (The system should offer a pleasant user experience)

2. The requirements missing.

In order to appropriate intended users (**UR_AUDIENCE**) and the users could be navigated easily and intuitively (**UR_UX**), there are some functional requirements missing in the previous functional table. Therefore, two functional requirements should be added.

- **FR_SETTING** (The user should be able to adjust settings such as sensitivity and controls)
- **FR_DEMO** (There should be a demo mode where the game plays itself to show the user how the game works.) should be added.

3. The functional requirements and non-functional requirements are confusing.

The scalability was a non-functional requirement, so I removed **UR_SCALABLE_UI** from user requirements and **FR_UI_SCLABLE** from the functional requirements table. Then **NFR_SCALABLE** was added and quantified in the non-functional requirement table.

- **NFR_SCALABLE** (The games UI must be scalable (set to different resolutions)) and the fit criteria is the game window scale down to a screen of 1x1 pixels.

4. The fit criterion could be more precise.

The fit criterion of **NFR_PLAYABLITY_INSTRUCTIONS** is that < 1minute spent on giving player instructions, the fit criterion could be more precise.

- **NFR_PLAYABLITY_INSTRUCTIONS**

(A new player, with minimum prior knowledge, must understand the objectives and controls of the game quickly (be able to attempt to play)) and the fit criterion is 95% of players will be able to teleport, heal their health and arrest operatives with 1 minute of instructions.

II. Abstract and Concrete Architecture

1. Both the abstract and concrete architecture were too complicated.

We decided that it would be best to remove some of the less important connections between the entities on the diagram, namely the uses. Then we reorganised the structure of the diagrams so that there is a clear divide between the screens and the entities involved with the game. Finally, we reorganised the concrete architecture so that it's split into 3 separate diagrams with links between them. This was because the original was already split into the HUD, the AI and the main section. So it would reduce the complexity and make it easier to understand with them split into 3 simpler diagrams, rather than one with a more complicated structure.

2. It's visually quite bland since there's no colour.

We decided that since the architecture was already so complex, adding lots of different colours could make it needlessly complicated and harder to read at a glance. Therefore, we added a single colour, yellow, to denote when the different concrete diagrams are connecting to one another. This way it makes it easier to comprehend all 3 diagrams as 1 complete architecture because it's more obvious how they are connected.

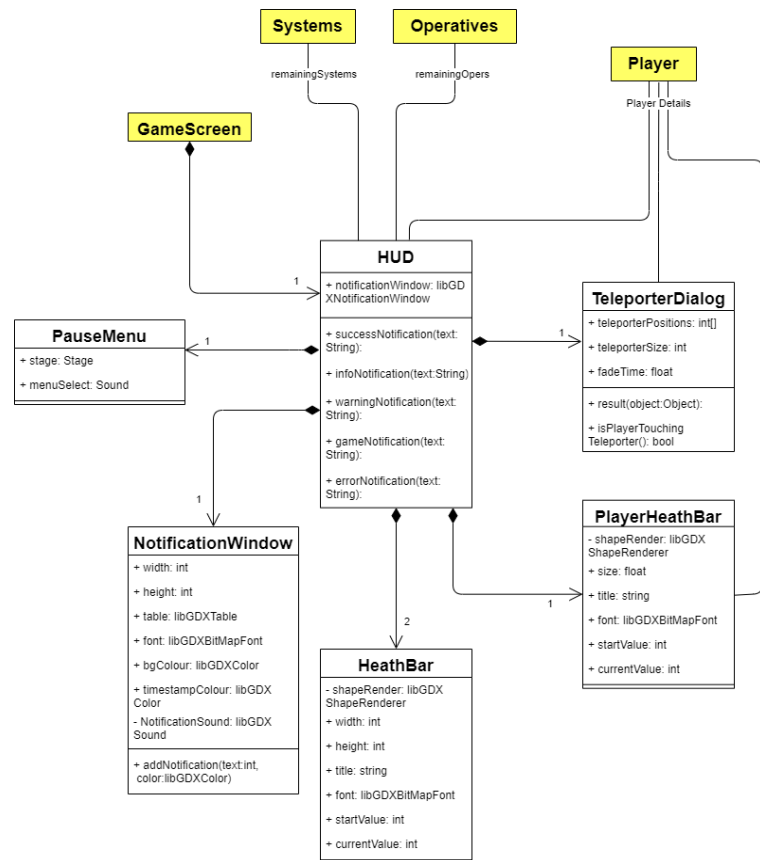
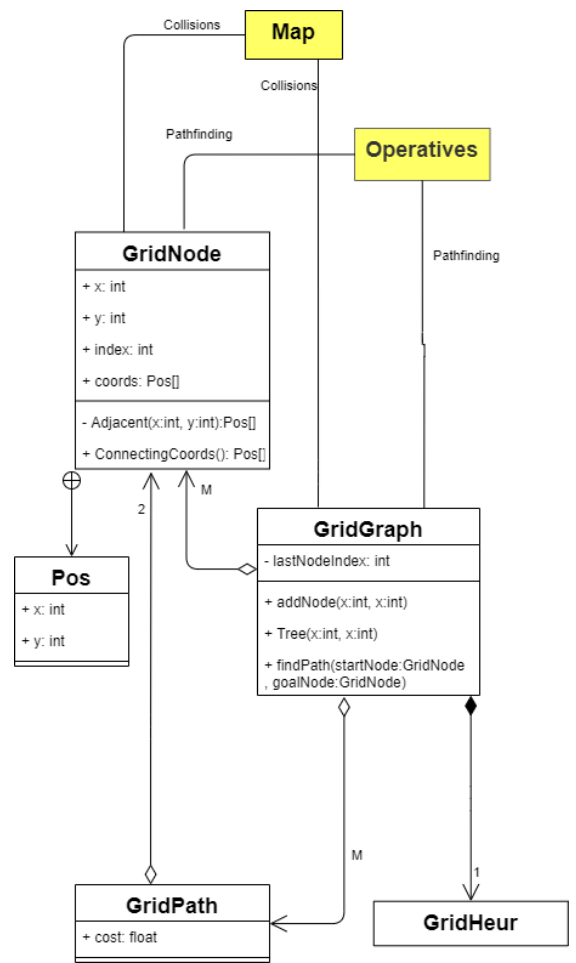
3. It's quite hard to read at a glance.

We tried to make the structure of the architecture more readable by making a few changes:

- Increasing the font size.
- Rearranging the nodes in a more visually comprehensive manner.
- Reducing unnecessary clutter within the nodes.

4. There were some requirements that were missing from the implementation and weren't in the architecture.

So, we added those new parts into the architecture seen as the Pause Menu and the update() function of the Operatives, since they contain the new operative abilities (FR_OPER_ABILITY & UR_OPERATIVE_ABILITIES). We added them to both the abstract and concrete architecture.



III. Methods and Plans

1. Some aspects of team organisation are not presented

We decided that there were some important details about team organisation that had not been covered. Therefore, we added details, including team roles and how they were assigned. We also included an explanation of why a democratic structure was chosen for the organisation of the team.

2. Some of the tools that we used after we took over the project had not been covered in the Tools Used section of the document.

We added details about the tools our team used for assessment 2 that the team we had taken over decided not to use in assessment 1 or had not written about. This included the IDE IntelliJ, as well as the communication applications Zoom and Messenger.

IV. Risk Assessment and Mitigation

Changes to Risk Format & Registrar

1. Risk ownership was introduced into the tabular risk representation .
2. Also new sections on the process followed to identify relevant risk and risk analysis were introduced which were both lacking in the previous teams risk assessment.

Risk Descriptors

1. Changes were made on the risk prescription to clarify the meaning and where possible details were added to put emphasis on the risk so it can be understood better.
2. Where appropriate irrelevant risks were deleted because the risks were no longer applicable due change management and project integration. For example, the previous owners of the project have used Trello to keep track of project progress and Discord as a communication platform. For simplicity we kept meeting logs and used facebook which was already available to most of us as a communication platform.
3. Similar risks were put together into relevant categories to increase clarity and to avoid repetition which could lead to confusion.
For example , R3 , R4, R5 and R6 which all represented risks that could result from disruptions on group communication were put together under the ID R2:- Disruptions on group communications in Risk2.pdf. Also R8 and R9 and R10 which all represented risks that could result from team members becoming unavailable due to various reasons, were put together under the ID R4:- unplanned absences in Risk2.pdf.

Duplicate Risks

1. Where appropriate relevant risks were transferred with little or no change to avoid duplication. However, New ID's were assigned to the transferred risk. For example R13 on the former groups risk assessment which represent requirement change due to new ideas and R3 which also represented requirement change due to the customer were combined under a new ID R6 under the description of requirement change.

Likelihood and Severity

1. Necessary adjustments were made on the likelihood and severity of the risk so the risks are not being exaggerated and mimic real case scenarios that are applicable to the project.

Reference list:

- [1] Anastasia Belyh, (2019, Sep. 20). *Major Approaches & Models of Change Management*. CLEVERISM [Online]. Available: <https://www.cleverism.com/major-approaches-models-of-change-management/> [Accessed: Feb. 01, 2021].
- [2] Aligni Incorporated, (2021). *Engineering Change Management*. ALIGNI [Online]. Available: <https://www.aligni.com/doc/engineering-change-management/> [Accessed: Feb. 01, 2021].