1.0 Change

1.1 Requirements Changes

The updated statement of requirements can be viewed here [1]. All changes to requirements as of assessment two are in bold and the table cell is coloured blue like so: **requirement change**. *Changes made to rectify any issues that arose from assessment one's feedback are highlighted in green like so:* change made *Requirements that have been removed due to assessment two have been coloured red to demonstrate that they are no longer valid, like so:* removed requirement.

The following document will explain and justify any changes that have been made to the statement of requirements following the customer's new specification with a brief description of the changes resulting from feedback. As our overall goal is to produce a playable and enjoyable game that the customer can market and sell, it is crucial that we listen to their needs and adjust requirements accordingly. The original statement of requirements can be found on the SEP-R-ated website [2].

Following the assessment feedback we revised our definitions for functional, performance and nonfunctional requirements. This change in definitions required a new set of performance and nonfunctional while the previous set of functional requirements were revised. We define our requirements as follows:

- *Functional requirements* are actions/functionality that a system must incorporate such that essential user functionality is provided.
- *Performance requirements* are constraints regarding the execution of system capabilities e.g. timings or space complexities.
- *Nonfunctional requirements* present constraints on the quality of a functional requirement.

1.1.1 Functional Requirements Changes

The game needs to be multiplayer and support two or more players (F1), this is a key concept that is essential to gameplay and a change to this could result in an inferior game. It is still our own decision whether or not to include networked play although there is a risk that too much time could be spent implementing this and core functionality could suffer [3]. But the as of phase two there is no need to implement an AI player.

F3 previously stated that there would be three resource types spread across tiles (Food, Ore and Energy) that the user can produce and collect. The revised requirements however state that the game only needs to feature the production of Energy and Ore; hence we have removed the requirement to produce food [1]. F2 remains unchanged because the game still incorporates plots of land or tiles with different characteristics; only these characteristics have now changed to include two resource types instead of the three that were originally required.

In line with F3 being changed to require only that the player can produce only two types of resources, the customer has also specified that Roboticons need to be customised to support only the production of these two resources. Therefore F8 has been changed to meet these new requirements and there is no longer the need for the Roboticon to be customised to produce Food [1].

The updated requirements for the market means that the player can buy and sell only to and from the market. Therefore there is no longer the requirement for trade between players (F6) has been changed [1]. Trade with the market and dynamic prices (F7) will remain key functional requirements for the game but trade between players is not required.

The customer only requires us to produce a limited GUI, as a result there is no need to implement requirement F13 and it has been removed from our statement of requirements [1]. Although it would be more visually appealing and easier to market with a fully-featured map that includes identifiable landmarks of the university, there is a risk of spending too much time on this unnecessarily [3].

The impacts of not fully implementing these changes in line with the client's needs have been assessed in the risk assessment, found on our website [3].

1.2 Methods, Plans Update
1.2.1 Method Changes
The update method documentation can be viewed [here](#) [4]. All changes can be identified by their formatting, any additions will be black text on a green background like so: changed text.

The first change to the method documentation is the adaption of our method to include test driven development styled implementation (*see section 1.1*). This change in methodology occurred early in the implementation stage of the project. We found that the team would naturally debug code as it was written before continuing onto the next features, therefore we thought it best to incorporate this behaviour into our methodology.

Tool selection and the justification for these choices also underwent changes (*see section 1.2*). We have included information concerning all tools that were under consideration and why these alternatives are not used. These changes were made following the assessment feedback as the documentation regarding tools was lacking in details. Through implementing these changes we hope to explain why we have chosen our preferred tools. This also serves as a reminder to the team to make use of such functionality, while providing both clients and management an insight into our daily operations.

The team expressed concerns regarding task assignment, monitoring and program testing during a scrum early in phase 2. To alleviate these concerns we integrated two new tools into our development method; ZenHub and Travis (*see section 1.3*). Job allocation and progress can be viewed by any member of the team, providing motivation to all members of the team to produce timely work of high quality. We believe that these additions to our tools is justified by the increased productivity and the convenience provided to all managerial roles.

The last change is in our organizational structure as individual responsibilities and development behaviour have changed (*see section 2.0*). The major change being the appointment of two lead developers, the appointment of these roles were necessary to obtain a clear hierarchy. This ensures that development proceeds swiftly with minimal disagreements whilst following a concrete plan of action. The change in development behaviour (i.e. the use of XP pair programming method, called lead-dev pairs instead of the 'packs') was implemented as we felt that having individuals working on their own sections was ill-advised. Through working in pairs it is less likely that progress will slow due to the complexity or amount of work, it also provides a way to keep developers motivated.

1.2.2 Plan Changes
Whilst the overall project plan presented in assessment one has not changed, we have provided a detailed plan for assessment three containing a Gantt chart showing task relations and a backlog containing task details (this can be accessed [here](#)) [5]. This was accomplished to ensure that the next phase will progress in a timely and predictably manner. Through hosting these documents on the website the whole team may access them at any time so they can both monitor progress and prepare for the oncoming tasks, it also allows for management to monitor our predicted progress. The plan for assessment four has not been implemented further, as we feel that planning for later phases should incorporate anything learned from our experience with past phases and as such we only begin planning one phase in advanced.

1.3 Risk Changes

The update risk documentation can be viewed [here](#) [6]. All changes can be identified by their formatting, any additions will be black text on a green background like so: changed text.

The first and most significant change we made was abandoning the individual top 10 risks system. We have set out with the intention of each team member keeping track of their own based on how they perceived the impact and probability. Based on these individual lists, the plan was then to update the overall team list on a weekly basis to reflect the changes the team identified. However we quickly found that this task became too time consuming and repetitive as there was a lot of overlap between each of the individual lists week by week. We tracked over 70 risks on weekly basis which occupied each team member's time and made updating the overall team list a complex task in itself. We found out that team members had a similar perception of the risks as the same risks kept emerging at the top of the individual risks lists. Due to this we have now adapted our approach to keep track of only the overall team risks which are discussed in weekly in a meeting. The majority of tasks are completed by multiple team members as opposed to individual work and as such keeping track of team risks has proven more efficient.

We have made changes to the way we track and present risks and mitigation over time. We are still using a spreadsheet to present the data because it allows for a simple tabular representation. The addition of risk categories (which we have previously identified) from Laurie Williams' Risk Management format [7] and colour coding from the rank table will make the visual representation easier to understand and should therefore make tracking risks a more streamlined task.

A skilled individual, the risk owner has been assigned to monitor each risk. The team has followed the Risk Ownership section of The Risk Management Guide [8] in order to appoint an appropriate individual. In accordance with this method of risk management we have appointed various individuals with elements of the risk responsibilities. The Risk Manager is responsible for the creation and enforcement of risk policy, as well ensuring that all risk owners have prepared sufficient risk mitigation strategies. Whilst all members of the team are responsible for identifying and reporting any arising issues.

We have now adapted the use of a pivot table. We now track 20 unique risks of which the probability of occurring or their impact could change on a weekly basis. This system, though efficient, generates vast amounts of data. Pivot table allows us to filter this data by a particular field, removing any information that is not needed at the time and as such makes tracking or updating the risks a simpler task.

Bibliography

[1] SEPR, "Updated Statement of Requirements," [Online]. Available:
https://seprated.github.io/Assessment2/Req2.pdf. [Accessed 21 January 2017].

[2] SEPR, "Requirements," [Online]. Available:
https://seprated.github.io/Assessment1/Req1.pdf. [Accessed 19 January 2017].

[3] SEPR, "Risk Assessment and Mitigation," [Online]. Available:
https://seprated.github.io/Assessment1/Risk1.pdf. [Accessed 19 January 2017].

[4] SEPR, "Method Deliverable," [Online]. Available:
https://seprated.github.io/Assessment2/Plan2.pdf. [Accessed 19 January 2017].

[5] SEPR, "Plan," [Online]. Available:
https://seprated.github.io/assessment3.html. [Accessed 19 January 2017].

[6] SEPR, "Updated Risk Assessment and Mitigation," [Online]. Available:
https://seprated.github.io/Assessment2/Risk2.pdf. [Accessed 21 January 2017].

[7] L. Williams, "Risk Management", 2004

[8] RuleWorks, "The Risk Management Guide - Ownership, " [Online]. Available:
http://www.ruleworks.co.uk/riskguide/risk-ownership.htm. [Accessed 21 January 2017].