Requirements (Assessment 2)

Team name: Endeavour (Team 28)

Team Members: Sarah Berry, Finley Brown, Yupeng Di, William Griffiths, Kurtas Joksas, Fraser Masson

Requirements

When determining the requirements for this software engineering project, we determined 3 key areas to obtain the information required to express the requirements for the project, which involved the product brief, the stakeholders, and our own team decisions.

Our first step was to analyse and quantify the requirements from the product brief provided to us. This involved creating an initial list of all the requirements from the document and discussing the implications of these with the team. As a lot of these requirements were critical to the success of the project and their implementation was required, many have been classified as functional requirements of the system.

Once the requirements from the product brief were categorised and discussed, we then moved onto the second stage, which involved interviewing a stakeholder in order to obtain additional requirements and expectations of the system. This process was performed by arranging a meeting with the customer, and asking them a set of predetermined questions. These questions were discussed between members of our development team beforehand and were specifically written for achieve a few key objectives:

- Elaborate on requirements from the product brief in order to achieve a greater understanding of the specifics regarding them and any potential constraints.
- Obtain new functional requirements for the project. (additional critical functionality)
- Gather some expectations of the user for the project.
- Understand if and where there was flexibility for deciding our own requirements.

These questions covered a lot of topics regarding the project including the style of the game, a target audience, mechanics and other aspects that previously had not been covered in the product brief. While the interview was being conducted with the customer of the product, members of our team noted down key information / transcripts of the answers provided, in order to ensure we had the correct information provided by the stakeholder. This data was again aggregated, and discussions were made with our team around the answers provided.

Once we had gathered all the required data from the product brief alongside additional prompts from the stakeholders, another discussion was performed in order to determine the final initial requirements of the project, mostly prompted from the answers provided by the customer. These requirements were determined from specifics of the system that the stakeholder expressed to us were our decision to make, and were mostly non-functional requirements, such as how certain features of the game would be implemented.

Taking all of this data into consideration, we have decided to format the provided requirements into 3 tables - consisting of:

- The user’s requirements and expectations from the system
- The functional system requirements
- The non-functional system requirements

Additionally, this statement of requirements will include criteria for quantifying the success of the requirement and the level of priority we’ve set them.

	User Requirements		
	ID	Description	Priority
1.1	UR_NAME	Game must be named “Auber”	Shall
1.2	UR_UX	The user interface should be easy and intuitive to navigate	Should
1.3	UR_MAP	The game displays a map to the user which can be navigated and interacted with	Shall
1.4	UR_PLAYER	The player can navigate the map, heal and arrest hostiles	Shall
1.5	UR_HOSTILES	Hostiles act like other characters on the ship, but attempt to sabotage key systems and have special abilities	Shall
1.6	UR_LOGIC	Game is played in real time, with no turns.	Shall
1.7	UR_JAVA	Must be developed with the Java programming language	Shall
1.8	UR_ARREST	Auber must be able to arrest the infiltrators	Shall
1.9	UR_WIN	User must be able to win the game by arresting every single infiltrator before the systems are destroyed	Shall
1.10	UR_LOSE	If the infiltrators destroy every system the user loses the game	Shall
1.11	UR_INFILTRATORS	There must be infiltrators on the map based on the difficulty level chosen	Shall
1.12	UR_SOUND	No sound is needed	Should
1.13	UR_VIEW	Game must be from a top down view	Shall
1.14	UR_CONVENIENCE	Accessible to everyone and easy to learn	Should

	Functional System Requirements			
	ID	Description	User Requirements	Related Risks
2.1	FR_MENU	The system should provide a menu to allow the player to start and configure the game	UR_UX	-
2.2	FR_MAP_ROOMS	The system should have a fixed playable map area consisting of at least distinct 4 types of rooms	UR_MAP	-
2.3	FR_MAP	The map will be made of tiles on which the player and the AI can traverse, from tile to tile	UR_MAP	RISK 13
2.4	FR_TELEPORTER	The map should contain teleport pads in areas which will the player can use to teleport around the map	UR_MAP	-
2.5	FR_KEY_SYSTEMS	The map should contain at least 15 key systems which can be sabotaged by infiltrators, and defended by the player. These cannot be repaired.	UR_MAP	-
2.6	FR_ALIENS	There must be at least 2 types of alien characters on the map	UR_MAP	RISK 8 RISK 15
2.7	FR_SABOTAGE	The hostiles can sabotage key systems on the map	UR_HOSTILES	-
2.8	FR_SPECIAL_ABILITIES	The hostiles must have at least 3 special abilities	UR_INFILTRATORS	-
2.9	FR_PRISON	Holds hostiles once they have been arrested. Arrested hostiles cannot escape.	UR_MAP	RISK 8
2.10	FR_ATTACK_NOTIF	The player is notified of sabotage attempts	UR_UX	-
2.11	FR_HEAL	Player can heal in the infirmary	UR_UX	-
2.12	FR_ARREST	Player can arrest hostiles	UR_ARREST	-

2.13	FR_MAP_SIZE	Map must have 10 rooms	UR_MAP	RISK 13
2.14	FR_SYSTEM_HEALTH	Key systems should have set, non-regenerating health	UR_LOGIC	-
2.15	FR_SYSTEM_DESTROY	Systems should be destroyed by attackers in 10 seconds	UR_LOGIC	RISK 13
2.16	FR_PLAYER_SPEED	Player's movement speed should be faster than infiltrators	UR_PLAYER	RISK 15
2.17	FR_HOSTILES_RUN	Hostiles should run from the player's teleportation gun	UR_HOSTILES	-
2.18	FR_ARREST	Player can't arrest hostiles if they aren't attacking anything	UR_PLAYER	-
2.19	FR_RESPAWN	Player should respawn at cloning bay if killed	UR_PLAYER	-
2.20	FR_HOSTILES_ATTACK	Hostiles' attacks should do 10% of the player's health	UR_HOSTILES	-
2.21	FR_HOSTILES_ABILITIES	Hostiles should have following abilities: Blinding player, player slowdown, player confusion	UR_HOSTILES	RISK 14
2.22	FR_ROOM_SIZES	Rooms should be different shapes and sizes	UR_MAP	RISK 8 RISK 15
2.23	FR_HOSTILES_SPAWN	All hostiles spawn on the map from the beginning of the game	UR_HOSTILES	-
2.24	FR_SYSTEMS_ATTACKED	A maximum of three systems can be attacked at once	UR_LOGIC	RISK 15
2.25	FR_ALIENS_COUNT	There should be at least 5 non hostile aliens on the map	UR_MAP	RISK 8 RISK 15
2.26	FR_HOSTILES_SPECIAL	There should be four hostiles with special abilities	UR_HOSTILES	RISK 14
2.27	FR_PLAYER_TILES	Player should have free movement between tiles; should not snap to tiles	UR_PLAYER	RISK 8 RISK 13 RISK 14

2.28	FR_TELEPADS	There should be three telepads on the station	UR_MAP	-
2.29	FR_BRIG	The brig should appear as a room on the map	UR_MAP	-
2.30	FR_MINIMAP	The game should display a minimap that shows what room the player is in	UR_MAP	-
2.31	FR_PLAYER_VIEW	The game should be zoomed in on the player	UR_MAP	RISK 15
2.32	FR_TELEPAD_DESTINATION	The player can teleport from any telepad to any other telepad	UR_PLAYER	RISK 13
2.33	FR_PLAYER_SPAWN	Player's spawn location on the map is fixed	UR_PLAYER	-
2.34	FR_ALIEN_SPAWN	Spawn locations for aliens should be random	UR_MAP	-
2.35	FR_ANIMATED_DOOR	Animate each door in the map when opened/closed	UR_MAP	RISK 15
2.36	FR_AUBER_POWERS	Auber has 5 new distinct abilities that he can pick up during his journey	UR_PLAYER	RISK 7
2.37	FR_LEVEL	User can pick the level of difficulty of the game ranging including easy, medium and hard	UR_CONVENIENCE	RISK 7 RISK 14
2.38	FR_SAVE	User can save the game and return to that position when they want to	UR_CONVENIENCE	RISK 7 RISK 14

Our group found that a few of the risks from the risk assessment table applied to almost all of the functional requirements. Risk 2, "Tools becoming unavailable", while unlikely, could potentially apply to all of these functional requirements if e.g. if a backup of the work was lost, or if one of the group members had a computer issue leading to the loss of their work. Risk 17, "Poor code quality leading to bugs or readability issues" is more likely to pose an issue than risk 2, however this can simply be mitigated by discussing the new code with the other group members.

Risk 18, "Loss of code or technical documents", is again very unlikely to happen considering our group uses Github as our online collaborative workspace, and if one group member loses their work they can download an earlier version of their work from github. However, if it were to happen.

Also, seeing as almost all of these functional requirements are specific features of the game e.g. the number of rooms the map has, the number of distinct abilities Auber has, any

‘Project’ or ‘People’ type rules can be applied to them, as deadline issues or specification changes could pose a risk to these requirements

	Non-Functional System Requirements				
	ID	Description	User Requirements	Criteria	Related Risks
3.1	NFR_DURATION	Game should last a satisfying amount of time	UR_UX	5 minutes minimum 10 minutes maximum	RISK 10
3.2	NFR_DIFFICULTY	Game should be enjoyable and not overly difficult	UR_UX	-	RISK 12 RISK 15
3.3	NFR_LANGUAGE	All in game messages should be in easy to understand	UR_CONVENIENCE	All in English language	-
3.4	NFR_SIZEABLE	Screen can be minimised and maximised	UR_UX	1920x1080 pixels max	-
3.5	NFR_NETWORK	Available to be played offline	UR_CONVENIENCE	No internet	