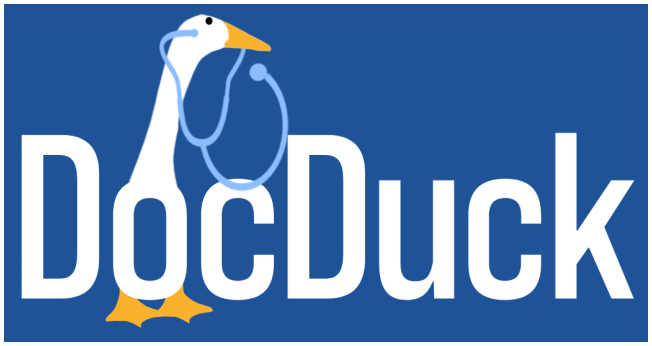

DocDuck Quality Assurance Manual

SWEng Group 1

May 7, 2024

Contents

1 Introduction **4**
1.1 Company Profile 4

2 Roles and Responsibilities **5**
2.1 Organisational Structure 5
2.2 Project Manager 6
2.2.1 Role Description 6
2.2.2 Risk Management 7
2.2.3 QA Metrics 8
2.3 QA and Documentation Manager 9
2.3.1 Role Description 9
2.3.2 Risk Management 10
2.3.3 QA Metrics 10
2.4 Design and Specification Manager 11
2.4.1 Role Description 11
2.4.2 Risk Management 12
2.4.3 QA Metrics 12
2.5 Testing and Integration Manager 13
2.5.1 Role Description 13
2.5.2 Risk Management 13
2.5.3 QA Metrics 14
2.6 Finance Manager 15
2.6.1 Role Description 15
2.6.2 Risk Management 15
2.6.3 QA Metrics 15
2.7 Software Manager 16
2.7.1 Role Description 16
2.7.2 Risk Management 16
2.7.3 QA Metrics 17

2.8 Marketing Manager 18
2.8.1 Role Description 18
2.8.2 Risk Management 18
2.8.3 QA Metrics 19
2.9 Communications Manager 20
2.9.1 Role Description 20
2.9.2 Risk Management 21
2.9.3 QA Metrics 21
2.10 Graphic Design & GUI Manager 22
2.10.1 Role Description 22
2.10.2 Risk Management 23
2.10.3 QA Metrics 24

3 Deliverables **25**

4 Project Management Methodology **26**
4.1 Requirements and Specifications Phase 26
4.2 Agile Design and Planning 27
4.2.1 Design Function 27
4.2.2 TDD (Test-Driven Design) 27
4.2.3 Small Releases and Demos 27
4.2.4 Implementation and Coordination 28
4.2.5 Testing and Integration Sprint 28
4.3 Quality Auditing Reviews 29

Quality Assurance Manual Version	**Release Date**	**Changes**	**Contributors**
0.1	14/11/23	Initial Version. Created Sections: 1, 2, 4, 5,	wab515 jrb617 nc1196 ttc515 zm926 htsm500 js2140 rw1834 lw2380
0.2	30/11/23	Section 3 Added. Table format standardised across roles. Clarification of 2.2.3. Unnecessary vision statement removed. Company profile adjusted to be in line with QA Manual standards. 4.2 modified to match an Agile method apposed to Waterfall. Cleaned up structure.	jrb617 htsm500 js2140 lw2380
1.0	5/12/23	Remade Section 4 to be completely Agile. Improved formatting and grammar. All QA Metrics quantified. Project Manager paragraph rewritten for clarity	jrb617 zm926 htsm500 js2140 lw2380
1.1	26/01/24	Updated Role Hierarchy fig 1, Tweaked formatting	jrb617
1.2	07/05/24	Role Hierarchy updated with reallocated roles, Grpahics Designer and GUI Developer renamed to Graphics Design and GUI Manager	jrb617

1 Introduction

1.1 Company Profile

The company was formed to provide assistance to companies with a maintenance engineer division. the major business focus for the company is to improve overall clarity for the maintenance engineering sector. Our company's emphasis is on affordability for the growing business as well as continued support with additional future features.

Our group is a team of experienced developers with backgrounds in many different avenues combining together to provide a high quality product with a variety of features and accessible to all skill levels.

2 Roles and Responsibilities

2.1 Organisational Structure

To ensure the product can be delivered at a high quality, on time, and on budget. The company structure has been divided into a series of roles for the different members. These roles break down into 2 areas, organisation and software. The organisational roles lead the running of the company and design of the product. The software roles lead the implementation and testing.
Each role has at least one deputy assigned to it. They are to be familiar with the responsibilities of the role and able to take over the duties should the main manager be unable to through illness etc. They also take up smaller tasks and assist with larger ones to distribute labour more evenly amongst the company.
Communication between team members is achieved through weekly team meetings and reviews. Additional progress reviews are also conducted prior to deliverable submission.
An illustration of the roles and the interactions between them is shown below:

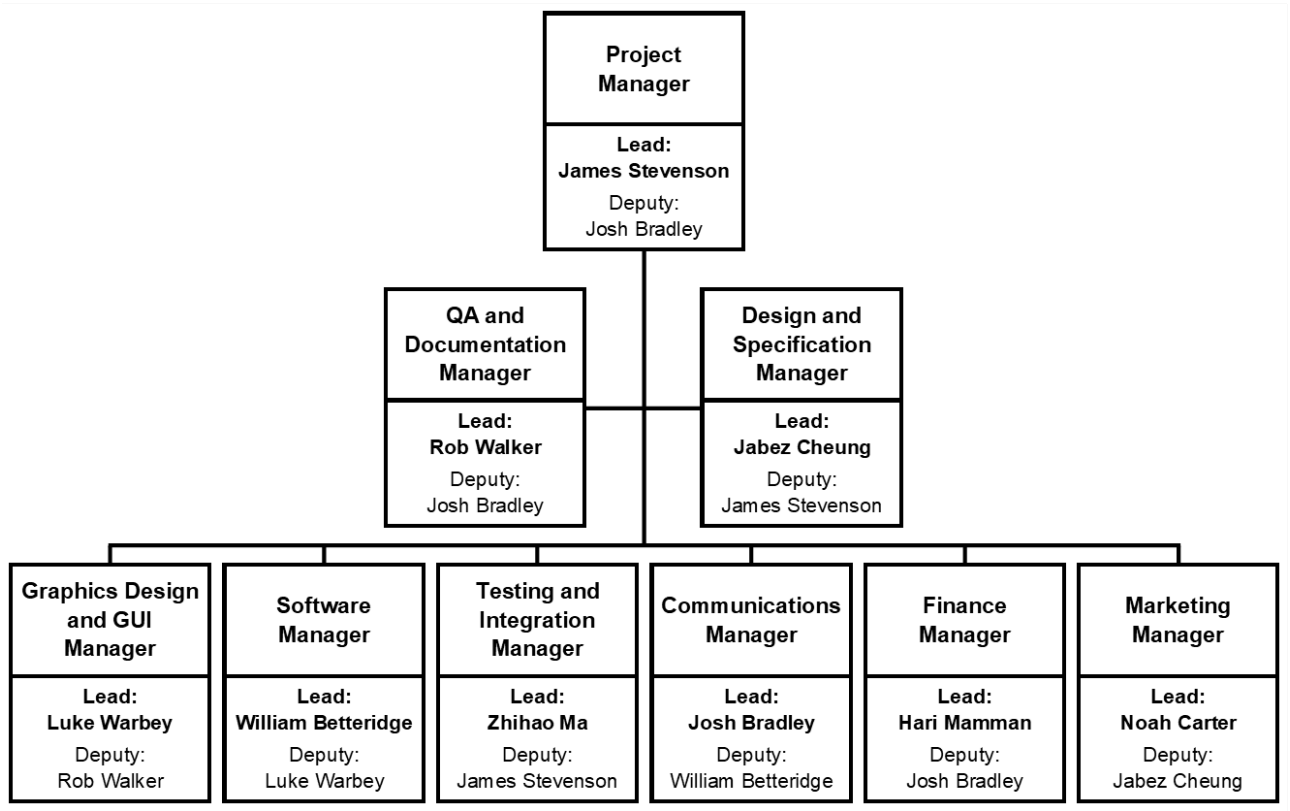

Figure 1: Role Hierarchy

2.2 Project Manager

Group member: James Stevenson

2.2.1 Role Description

The Project Manager oversees the entire project, ensuring it achieves its goals within the set time frames. They also guarantee that all team members have assigned roles and contribute effectively. The Project Manager maintains regular communication with each team member to address any challenges and facilitate a smooth progress. Additionally, they ensure the availability of necessary assets for the project. This involves various responsibilities such as:

- Arranging regular meetings as well as extra meetings if required.
- Making sure the agenda is known and set for the future meetings.
- Assigning roles that complement each individual members skill set.
- Attaching secondary roles for each member to ensure that the role can still be handled in-case of emergency and that the primary role member stays on track.
- Ensuring a healthy working environment where each member feels listened to.
- Making sure that each member feels they have made an appropriate contribution to the project as a whole.
- Ensuring contingencies are in-place in-case unexpected emergencies occur during development.
- Having appropriate backup solutions to problems.
- Organise a realistic project plan and time schedule.
- Liaise with the customer to ensure the product meets with their expectations.
- Perform the final check on deliverables before approval.
- Communicate with the finance manager to keep within budget for the project.

2.2.2 Risk Management

Risk Description	Likelihood (1-5)	Impact (1-5)	Priority (1-25)	Mitigation
Team Members clashing	2	4	8	Team members feel comfortable to voice any and all concerns at meetings and solution is found accomplished by private 1 on 1 meetings.
Missing internal deadlines	3	5	15	Have constant communication with members of each section of the project and find solutions to problems before deadlines are reached, such as swapping members of the team at fault or adding a new team member to increase productivity
Team Members prolonged absence	2	4	8	Each role has a backup member capable of picking up where the other team mate left off, if absence lasts even longer multiple members may be assigned to that role.
Failure to implement feature	3	4	12	Ensure a rollback state is created before each new features implementation, if the feature is mandatory check if there is a simpler alternative feature that could provide similar results.

Table 1: Project Manager potential risks and mitigations

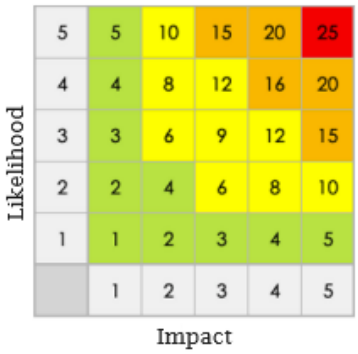

Figure 2: Priority Colour Chart

2.2.3 QA Metrics

Responsibility	Metric	Purpose
Customer communication:	Customer rating system	Consistent reviews and feedback from customer over product and documentation. Ask for customer to prove ratings out of 5 stars and customer reviews
Project tracking:	Smooth Project Pace	Set realistic internal deadlines and gantt charts, review and monitor how close we lineup to the goals set.
Healthy workspace	Developer working hours	Ensure that the group spreads their workload out even and effectively rather than completing all work at the last minute.
Workspread	Assigned workload difficulty	Ensure that one developer is not assigned with all the high risk high necessity tasks, that they are spread out evenly.

Table 2: Project Manager QA Metrics

2.3 QA and Documentation Manager

Group member: Rob Walker

2.3.1 Role Description

Quality Assurance is a set of procedures that ensures the software created by the company meets the requirements needed and to an acceptable quality. The QA and Documentation Manager is responsible for ensuring these processes are adhered to for the life cycle of the software to ensure quality, monitoring these processes, creating a document management system and reviewing documents to ensure they adhere to the requirements. Ensuring the processes created are adhered to requires regular communication and review with all members of the team.

Communication with members of the team will be done through group meetings, review meetings and documents created for team members to help them follow the requirements for all documents through the company. These documents will be designed to allow the QA and Document manager to communicate and check that requirements are followed. If change is needed in QA documents or processes, approval of the changes will be required approval by the QA and Documentation Manager and the Project Manager. An updated document/process will be created by the QA and Documentation Manager and will be sent out to all team members with the change clearly communicated.

The tasks for the QA and Documentation Manager are:

- Create QA processes and make clear to all members what should be followed, who should follow it, and the quality needed.
- Define the quality expected from documents in the company alongside the Project Manager.
- Create QA metrics for quality to be measured from throughout development.
- Review the use of the QA processes throughout development and carry out review meetings to ensure quality.
- Specify deadlines and how different types of documents should be handled in the company.
- Create document guidelines to be followed by all members of the team.
- Ensure the document guidelines are followed so the standard of documentation is up to high.
- Archive all documents throughout the project.
- Track all updates on documents and releases.
- Monitor deadlines for documents to make sure they are delivered.
- Ensure the latest versions of documents are being used.
- Ensure all meetings are appropriately held and documented.

2.3.2 Risk Management

Risk Description	Likelihood (1-5)	Impact (1-5)	Priority (1-25)	Mitigation
Failure to carry out a QA process	2	2	4	Have regular review meetings with team members and keep QA processes appropriate and up to date.
Missing or lost documents	1	5	5	Keep online and offline backups of all documents when they are created or updated
Module or Document clashing/incompatibility	2	4	8	Make sure the most up to date documents are used and all team members are aware of any changes
QA metric not met	2	2	4	Monitor and review all team members and their adherence to the QA metrics and processes.

Table 3: QA and Documentation potential Risks and Mitigations

2.3.3 QA Metrics

Responsibility	Metric	Purpose
Documentation:	Documents delivered	The number of documents delivered against the number of documents expected.
Deadlines:	Deadlines met	The number of submissions against the number of deadlines set.
Meeting QAs:	QA Metric Collection	The number of QA metrics met against the number of metrics expected to have been met.

Table 4: QA and Documentation QA Metrics

2.4 Design and Specification Manager

Group member: Jabez Cheung

2.4.1 Role Description

The Design & Specification Manager is responsible for providing a design of the product with specifications and features fulfilling the customer's needs. Seamless communication with managers is of paramount importance to keep track of the progress with all of the designed/requested features included with high-quality standards, meeting client's expectations.

- Read through the client's requirement statement and design a solution within the given budget and time.
- Produce the product specification by communicating with stakeholders including the client and the project team.
- Clearly define functionalities/performance criteria.
- Analyze the test results report and make sure all of the designed features meet the criteria and quality standard fulfilling client and end user's expectations.
- Communicate with other managers to keep track of the progress.
- Identify risks associated with the design or specification of the product and avoid over budget or time.
- Document the design and specification processes providing details of updates/changes made in each iteration.
- In association with the test manager, ensure the features requested by the client are in good working order and meet high-quality standards. Allocate group members to solve challenges and improve the quality of the product if needed.
- In association with the GUI design manager, ensure the product is easy to use and the services are easy to access, following Human Interface Guidelines [1].
- Seek feedback from the client for each version released identifying room for improvement.

2.4.2 Risk Management

Risk Description	Likelihood (1-5)	Impact (1-5)	Priority (1-25)	Mitigation
Delay in finalizing the product design	2	5	10	Identify the reason for the delay. Consider reordering by prioritizing the tasks, evaluate the time lost and report to the Project Manager.
Unfulfillment of quality standard & specification	2	4	8	Work with the Testing and Integration Manager to identify the reason for unfulfillment and the challenges involved, allocate members to improve the corresponding part of the product, evaluate the time needed and report to the Project Manager.

Table 5: Design & specification potential risks and solutions

2.4.3 QA Metrics

Responsibility	Metric	Purpose
Resource Management:	Cost of adding certain features to the product	Resources required to complete a certain task, including time, human power, tools, etc.
Specification Management:	Features implemented vs features requested	Is the design fulfilling the features and specifications requested by the client meeting high-quality standards without major defects.
Client Feedback:	Feedback from the client	Amount of elements being approved or to be improved.

Table 6: Design & specification QA metrics and measurement

2.5 Testing and Integration Manager

Group member: Zhihao Ma

2.5.1 Role Description

The Testing and Integration Manager leads the efforts to maintain high quality and reliability of software releases. This role encompasses strategic planning and execution of testing and integration activities. Responsibilities include:

- Strategising and executing a comprehensive test plan to ensure the robustness and performance of software before deployment.
- Managing the integration of new software with existing systems, ensuring minimal disruption and maintaining system integrity.
- Overseeing the creation, maintenance, and evaluation of automated test frameworks to facilitate continuous integration and deployment processes.
- Collaborating with cross-functional teams, including development, operations, and product management, to align testing strategies with business objectives.
- Ensuring that all testing and integration activities are conducted in accordance with regulatory standards and company policies.
- Conducting risk assessments for testing and integration processes, implementing mitigation plans to address identified risks.
- Providing leadership and guidance to the testing team, fostering a culture of quality and continuous improvement.
- Analysing test results, reporting to stakeholders on software quality, and making recommendations for improvements.
- Keeping abreast of new testing tools and strategies, integrating innovative technologies and practices to enhance the testing and integration life-cycle.

2.5.2 Risk Management

Risk Description	**Likelihood (1-5)**	**Impact (1-5)**	**Priority (1-25)**	**Mitigation**
Inadequate test coverage	2	4	8	Implement test coverage tools and regularly review test cases.
Flaky tests causing false positives/negatives	3	3	9	Regularly review and update tests, and remove non-deterministic factors.
Regression defects	2	4	8	Implement a robust regression test suite and run it at regular intervals.
Technical debt in test code	3	3	9	Allocate time for regular refactoring and technical improvements.
Integration delays	3	4	12	Promote continuous integration and set clear milestones.

Table 7: Testing and Integration Potential Risks and Mitigations

2.5.3 QA Metrics

Responsibility	Metric	Purpose
Requirement Test Coverage	Percentage of requirements for which test cases have been written. $\frac{\text{Number of requirements with test cases}}{\text{Total number of requirements}} 100$	Measures the extent to which the specified requirements are verified by tests, ensuring that all features are validated.
TDD Adoption Rate	Ratio of features developed using TDD to total features developed. $\frac{\text{Features developed with TDD}}{\text{Total features developed}} \times 100$	Reflects the adoption level of TDD practices, promoting early defect detection and design simplicity.
Test Case Pass Rate	Percentage of test cases that pass successfully in the first run. $\frac{\text{Test cases passed on first run}}{\text{Total test cases run}} \times 100$	Indicates the initial quality of the test cases, aiming for a high success rate which can reduce the need for repeated testing.
Phase-Specific Test Coverage	Ensuring test cases cover all functional scenarios for a specific phase, like the login page.	Assures that every functional scenario is covered, such as confirming all user roles can log in correctly, which leads to more reliable software usability.
TDD Cycle Compliance	Frequency of test-first approach where a failing test is written before writing new code. $\frac{\text{Number of TDD cycles followed}}{\text{Total development cycles}} \times 100$	Encourages rigorous application of TDD, enhancing code quality and reducing the likelihood of regressions.
Test Regression Rate	Frequency at which a change causes a previously passing test to fail. $\frac{\text{Number of regressions}}{\text{Total changes}} \times 100$	Helps identify unstable features and prioritize the need for corrective actions to maintain software reliability.
Automated Test Suite Run Frequency	How often the complete automated test suite is run.	Tracks the regularity of test suite execution, which is crucial for continuous integration environments and for catching defects promptly.

Table 8: Testing and Integration QA Metrics

2.6 Finance Manager

Group member: Hari Mamman

2.6.1 Role Description

The Finance Manager is responsible for handling all financial and accounting affairs. They should be in regular contact with the Project Manager to ensure that the group is kept on budget and expenses are being reduced where possible. Also, they should be in communication with the Marketing Manager in order to help produce sale demonstrations. The Finance Manager tasks include:

- Managing the company's financial accounting, monitoring and reporting systems
- Managing the company's budget
- Providing and interpreting financial information
- Work with the Marketing Manager to ensure there are appropriate contracts across groups
- Work with the Project Manager to make sure the company receives financial backing
- Formulate strategic and long-term business plans
- Produce accurate financial reports to specific deadlines

2.6.2 Risk Management

Risk Description	Likelihood (1-5)	Impact (1-5)	Priority (1-25)	Mitigation
Poor Budgeting	2	4	8	Monitor at regular intervals and adjust as needed.
Risks due to insufficient liquid assets	2	5	10	Monitor and stick to an accurate budget, ensuring that there are plans in place for sudden shocks.
Ineffective communication of financial information	1	2	2	Ensure clear and transparent communication and provide accurate and timely finance reports.

Table 9: Finance Potential Risks and Mitigations

2.6.3 QA Metrics

Responsibility	Metric	Purpose
Reporting Accuracy:	Percentage of error-free financial reports	Have the deputy Finance Manager and Project Manager proofread financial information to minimize errors within reports when being presented to financial backers.
Budgeting:	Percentage variance between budgeted and actual expenses	Monitor and control budgetary performance, identifying places where costs can be saved.

Table 10: Finance Manager QA Metrics

2.7 Software Manager

Group Member: William Betteridge

2.7.1 Role Description

The Software Manager is responsible for planning, designing and coordinating the implementation of the software components of the project. They should have a full understanding of the specifications and the requirements of the project in order to develop the desired product. Also, they should be in regular communication with the QA and Documentation Manager to ensure the product meets the specification and the quality of the software design is to the requirements. They should also be in communication with the Testing and Integration Manager to ensure that the product is thoroughly tested and is suitably integrated into any design flows and project timelines that are being worked on. The Software Manager tasks should include:

- Develop a clear overall software design based on the specifications and the customer requirements.
- Analyse the functional specification to determine necessary specific features to be programmed in Java, and split it up into multiple sections to be distributed to other group members.
  - Identify and decide on the interfaces, libraries, objects, data encapsulation and hierarchy that will be used.
- Work with the Design and Specification Manager to decide on the methods and techniques to be used in the design phase.
- Monitor the process of the coding and its implementation.
- Ensure that the design standards are followed in order to remain consistent between different sections of code written amongst the project group.

2.7.2 Risk Management

Risk Description	Likelihood (1-5)	Impact (1-5)	Priority (1-25)	Mitigation
Code failing to work	5	3	15	Implement test driven development to reduce unsolvable errors.
Code from group members not compatible	3	4	12	Make sure there is good quality version control and all group members agree on standards when writing code
Delay when finishing modules of code to be integrated	3	4	12	Keep track of all tasks and agree on temporary deadlines prior to final deadlines to allow for time to finish sections
Dependencies and Libraries being used become end of life (EOL)	3	3	9	Make sure all dependencies being used are actively being supported and updated. Look for similar alternative dependencies if a current one becomes EOL, to be able to transition to using instead.

Table 11: Software Potential Risks and Mitigations

2.7.3 QA Metrics

Responsibility	Metric	Purpose
Time Management	Time spent coding vs time planned	Ensure delivery of software is within plan by keeping track of total time spent on work.
Good coding practice	Code is clearly written and is understandable with minimal effort required to understand, all team members are able to quickly pick up where each other left off.	Consider lines of comments against lines of code. Use of standardised coding practices which are well documented and can be clearly understood.
Errors	Compilation errors and uncaught exceptions	Produce a compilation report to indicate the number of errors during compilation and whether code is error-free and tested.
Bug Fixes	Average time spent to test and fix a bug	Time taken identifying and fixing bugs and errors is suitable in comparison with the total number of hours spent testing.
Dependencies and Libraries	All libraries being used are up to date and supported	Utilising libraries and dependencies which are up to date and currently supported rather than old deprecated software is good practice and ensures longevity and security in the furture.

Table 12: Software QA Metrics

2.8 Marketing Manager

Group Member: Noah Carter

2.8.1 Role Description

The role of the Marketing Manager in engineering application development is imperative to the success of the business. This can be accomplished in many ways, but the main idea is to stay ahead of the curve and fill any gaps in the market by having a dynamic marketing plan. They should be in regular contact with all managers in the group to ensure all marketing materials and campaigns can be kept up to date with regards to the application development.

Some of the key roles of the marketing manager include:

- Conducting market research to identify any new potential consumers and businesses by understanding their needs and preferences. The Marketing Manager should stay informed on current industry trends and conduct market studies to identify any gaps in the market. Market Research also includes accessibility by making sure thorough research is completed in this department and ensuring our software is made as accessible as possible.
- Clearly defining the target audience for the product
- Collaborating closely with the rest of the team to oversee and be aware of any emerging developments.
- Overseeing and creating strategic launch campaigns for the product to maximise market impact.
- Ensuring the implementation of effective branded marketing communications across various platforms including the company website, any print communications, and advertising.
- Overseeing the management of media and marketing personnel as well as navigate relationships with any external PR agencies.
- Evaluating the effectiveness of any marketing strategies executed.
- Working with the Finance Manager to explore appropriate business models included product pricing.

2.8.2 Risk Management

Risk Description	Likelihood (1-5)	Impact (1-5)	Priority (1-25)	Mitigation
Market Access	4	4	16	Gain authorisation from certification bodies as required. Check whether end users are those with purchasing authority, model selling strategy accordingly.
Pricing issues related to core products	3	2	6	Find trusted sources with appropriate pricing.
Competition	5	4	20	Ensure quality of product. Produce appropriate advertising campaign. Provide excellent after-sales service.

Table 13: Marketing Potential Risks and Mitigations

2.8.3 QA Metrics

Responsibility	Metric	Purpose
User Satisfaction:	Collect user feedback and satisfaction surveys.	Evaluate the overall satisfaction of end-users with the visual design and user interface. Feedback can provide insights into the effectiveness of the design in meeting user expectations.
Brand Awareness:	Market survey alongside customer feedback.	Evaluate results from market survey, as well as asking consumers how they discovered our product to test campaign effectiveness.
Share of the market:	Site traffic.	Research the current market and evaluate the products standing in comparison to the competitive and ever-changing landscape of product development.

Table 14: Marketing Potential Risks and Mitigations

2.9 Communications Manager

Group Member: Josh Bradley

2.9.1 Role Description

Effective communication amongst a group is paramount to success. The Communications Manager is responsible for handling the overall communication amongst and outside the group. They coordinate communications amongst the other managers to enable the fluid transfer of information. They liaise with other groups to enable effective cooperation in Project Wide Standards and inter-group contracts.

The key tasks of the Communications Manager include:

- Overseeing internal communications within the group to ensure effective communication strategies.
- Ensuring all members of the group are up to date in all required areas: deliverables, expectations, due dates, etc.
- Overseeing all inter-group communications. Working with the Software Manager and Project Manager to develop project wide standards and inter-group contracts.
- Booking rooms for team meetings and notifying group members of their time and location.
- Disseminating meeting notes to members not at that meeting
- Ensuring all required information is readily available to group members who need it.

2.9.2 Risk Management

Risk Description	Likelihood (1-5)	Impact (1-5)	Priority (1-25)	Mitigation
Ineffective internal communications	3	2	6	Ask for clarification if unclear. Ensure that all members are communicating their progress via meetings and reviews.
Members unsure of their required work	3	4	12	Work with QA and Documentation Manager to create standardised and readily available documentation such as Gantt charts and meeting minutes
Groups unable to agree on project wide standard	2	5	10	Mediate with Project Manager, Software Manager, and their external counterparts to reach a solution which will work for all parties
Groups not fulfilling their contracts	2	5	10	Mediate with Project Manager their external counterpart to identify the issue and how to solve it and adjust time scheduling appropriately
Group Members unable to make meeting	4	2	8	Ensure Deputy is able to fill in if required, move meeting format to hybrid if appropriate

Table 15: Communications Potential Risks and Mitigations

2.9.3 QA Metrics

Responsibility	Metric	Purpose
Group Communication	Group members attending at least 80% of required meetings	Evaluate internal communication, whether meeting and work information is being passed on appropriately
Dissemination of responsibilities	Up to date Gantt chart (Monday.com) and completed meeting minutes	Ensure that all group members know what work is expected for them to complete
Inter-Group communication	At least once per week meetings with other groups to produce all required documents	Ensuring that appropriate information is passed between groups to enable effective co-operation. And to make sure contracts and sharing of code is maintained to a good standard.

Table 16: Communications QA Metrics

2.10 Graphic Design & GUI Manager

Group Member: Luke Warbey

2.10.1 Role Description

The Graphic Design & GUI Manager in the project plays a pivotal role in shaping the visual identity and user interface (UI) of the software. This role requires a unique blend of artistic creativity and technical proficiency. Key responsibilities include:

- User-Centric Design: Develop a deep understanding of user personas and incorporate user-centered design principles to create interfaces that are not only aesthetically pleasing but also highly intuitive and user-friendly.
- Wireframing and Mockups: Collaborate closely with the team to translate conceptual ideas into tangible wireframes and design mockups. These artifacts serve as the foundation for the development of the software's graphical elements.
- UI Prototyping: Utilize prototyping tools to create interactive prototypes, allowing stakeholders and end-users to visualise the flow and functionality of the UI. Iterate on prototypes based on feedback received during design reviews.
- Visual Asset Creation: Craft and optimise graphical assets, including icons, images, and other visual elements, ensuring they align with the overall design vision and enhance the user experience.
- Design Consistency: Maintain consistency in design elements throughout the software to create a cohesive and unified look and feel. This includes adherence to established design patterns and style guides.
- Responsive Design: Implement responsive design principles to ensure the software's UI is accessible and visually appealing across various devices and screen sizes.
- Collaboration with Development: Work closely with the development team to ensure the feasibility of design implementations. Provide necessary assets and guidance to facilitate a smooth integration of graphical elements into the software.
- Accessibility & Coding Standard Considerations: Integrate accessibility features into the design process, ensuring that the UI is inclusive and complies with accessibility standards and is formatted professionally [2].
- Stay Updated on Design Trends: Continuously stay informed about industry trends and emerging design technologies to bring innovative and modern design solutions to the project.
- Documentation: Maintain comprehensive documentation of design decisions, rationale, and guidelines to facilitate knowledge transfer and collaboration within the team.

2.10.2 Risk Management

Risk Description	Likelihood (1-5)	Impact (1-5)	Priority (1-25)	Mitigation
Compatibility Issues: The graphical elements created may not integrate seamlessly with the GUI, leading to compatibility issues.	2	4	8	Collaborate with Software Manager and conduct regular integration testing to identify and address compatibility issues early in the development cycle.
Performances Bottlenecks: Complex graphical elements can contribute to performance bottlenecks in GUI.	2	2	4	Optimise graphical assets for performance. Implement caching strategies and conduct performance testing to identify and resolve bottlenecks. Keep a balance between visual richness and system performance.
Maintainability Challenges: Poorly structured code for GUI development can result in difficulties in maintaining and updating the software.	3	4	12	Enforce coding standards and best practices for GUI development. Document the codebase thoroughly to facilitate easier maintenance. Conduct code reviews to ensure adherence to established guidelines.
Security Vulnerabilities: Inadequate handling of graphical elements in the GUI code may introduce security vulnerabilities.	3	5	15	Enforce coding standards and best practices for GUI development. Document the codebase thoroughly to facilitate easier maintenance. Conduct code reviews to ensure adherence to established guidelines.
Misalignment with Brand Identity: There is a risk of the graphic designer creating visual elements that do not align with the established brand identity and become inconsistent with GUI design. leading to inconsistency.	2	4	8	Conduct regular brand guideline reviews, ensuring the Graphic/GUI designer is familiar with and adheres to the brand's visual standards. Schedule periodic meetings with the team to clarify any ambiguities.

Table 17: Graphic Design & GUI Manager Potential Risks and Mitigations

2.10.3 QA Metrics

Responsibility	Metric	Purpose
User Satisfaction	Collect user feedback and satisfaction surveys.	Evaluate the overall satisfaction of end-users with the visual design and user interface. Feedback can provide insights into the effectiveness of the design in meeting user expectations.
Usability Testing Results	Analyse results from usability testing sessions.	Ensure that the graphic design aligns with established design guidelines and branding standards. Identify and rectify any deviations.
GUI Related Bug Reports	Monitor and analyse design-related bug reports.	Track the number and severity of bugs related to graphical elements. Address and resolve issues promptly to enhance the overall quality of the user interface.
Timeline Adherence	Track the adherence of design tasks to the project timeline.	Ensure timely delivery of graphical assets, preventing delays in the development and testing phases.
Prototype Effectiveness	Analyse feedback on interactive prototypes via numbered star rating and review description.	Evaluate the effectiveness of interactive prototypes in communicating the intended flow and behavior of the user interface. Use feedback for continuous improvement.
Documentation Completeness	Review the comprehensiveness of design documentation.	Ensure that design decisions, rationale, and guidelines are well-documented for knowledge transfer and collaboration within the team.

Table 18: Graphic Design & GUI Developer QA Metrics

3 Deliverables

Deliverable	Producer*	Recipient*	Due
Technical Requirements Document	Client	Project Manager, Design and Specification Manager, QA and Documentation Manager	Start of Project
Functional Specification	All Managers	Client	End of Design Phase
Quality Assurance Manual	All Managers	Client	End of Design Phase
Project Wide Standard (PWS)	Project Manager, Software Manager, Communications Manager	Client	End of Design Phase
Design Documentation	QA and Documentation Manager	Client	End of Design Phase
User Manual	All Managers	Client	With initial release and updated with subsequent releases
Inter-Group Contracts	Project Manager, Software Manager, Communications Manager	Other group Project Managers, Client	End of Design Phase
Test and Integration Plans	Testing and Integration Manager	Testing Team	Before Implementation Phase
Test and Integration Reports	Testing Team	Testing and Implementation Manager	Throughout the Project
Financial Business Plan	Financial Manager	Client	End of Design Phase
Source Code	Software Manager	Client	With Each Release
Code	All Software Developers	Software Manager	Specific Milestones
Meeting Minutes	QA and Documentation Manager	Project Manager	After Each Meeting
Example Multimedia Presentation	All Managers	Client	End of Design Phase
HTML Product Tour	All Managers	Client	End of Design Phase
Weekly Contribution Reports	All Members	Project Manager	Throughout the Project
Weekly Time Sheets	All Members	Finance Manager	Throughout the Project
Budget Analysis Reports (Costs, expenses and resources)	Finance Manager	Project Manager	Throughout the Project

Table 19: Project Deliverables

*Where a specific Manager is responsible, their deputy is likewise responsible

4 Project Management Methodology

4.1 Requirements and Specifications Phase

The project starts with a statement by the customer laying out what is required from the product. The group then analyses the requirements and generates a series of ideas for the software to produce. This goes through a series of development meetings to refine the concepts and narrow down the most effective and practicable. These are then discussed with the customer to ensure that they meet their requirements and are feasible to produce. The individual members whose ideas these are then finally develop them to present to the group for deliberation. The group then votes on which idea will be moved forward with as the final product and elect the Project Manager who reports this to the customer. This is then further developed by creating the final feature list and separating it into individual releases. To summarise:

- The customer provides the requirements for what the project must do
- The group generates ideas to meet this requirements
- The group meets with the customer to discuss and narrow down the ideas
- The group votes on the final idea and Project Manager.
- The Project Manager reports to the customer and discusses any changes required.
- The Project Manager, Design and Specification Manager and QA and Documentation Manager, then generate the final feature and release list.

4.2 Agile Design and Planning

In the Agile approach adopted for a software project, the Software Development Life Cycle (SDLC) emphasises an iterative progress over fixed phases; the design, implementation, and testing/integration. Each of these activities are integrated into each function development cycle, showing adaptability and responsiveness to changing requirements.

4.2.1 Design Function

The design function in the Agile methodology is a continuous and collaborative process throughout each section. Key aspects of the design function include:

- **Requirements Refinement:** Collaboratively refine and prioritise requirements with stakeholders, converting them into actionable design tasks.
- **Iterative Design:** Continuously refine the design in response to feedback, whilst allowing for adjustments based on any changes in priorities.
- **User Feedback Integration:** Actively involve users in design decisions, making sure to incorporate their feedback in order to enhance user experience.
- **Adaptability:** Embrace any changes in functional and non-functional requirements, but still making sure that the design remains responsive to evolving project needs.

4.2.2 TDD (Test-Driven Design)

Test-Driven Design (TDD) is an integral part of the Agile design process, as it ensures code quality and functionality. The TDD cycle involves:

- **Write Tests First:** Developers create tests that define the expected behavior of the code before implementation; the first test will always fail.
- **Code Implementation:** Write code to meet the requirements outlined in the tests.
- **Automated Testing:** Execute automated tests to make sure that the code meets the criteria specified.
- **Refactoring:** Refactor code where needed in order to enhance design and maintainability yet making sure that tests continue to pass.

4.2.3 Small Releases and Demos

In Agile, the implementation phase is broken down into smaller, manageable releases; this reduces the risk of projects failing, as there is an earlier identification of any issues, and thus an earlier correction. Each release includes:

- **Incremental Development:** Implement features in increments, allowing for continuous delivery of valuable functionality.
- **Demo Sessions:** Conduct regular demo sessions to showcase completed features to stakeholders and collect feedback from them to further the refinement.
- **Flexibility:** Flexibly adjust priorities based on feedback, and prioritise these tasks over the next releases.

4.2.4 Implementation and Coordination

The implementation phase involves breaking down the design into smaller tasks, assigning them to the development team, and coordinating efforts:

- **Task Assignment:** Assign specific tasks to team members based on their expertise and workload.
- **Continuous Communication:** Maintain frequent communication between team members, Design and Specification Manager and Project Manager.
- **Iterative Progress Tracking:** Regularly track progress, ensuring design specifications are being met, and addressing any issues as soon as they emerge.
- **Budget Monitoring:** The Finance Manager monitors the budget, making sure the team is aware of the costs and are staying within said costs.

4.2.5 Testing and Integration Sprint

The Testing and Integration phase is an integral part of each sprint, ensuring that each function is thoroughly tested and seamlessly integrated into the evolving software product.

Component Testing and Integration

Within each sprint, testing and integration are ongoing processes:

- **Continuous Testing:** Rigorously test individual components within the software and involve all team members to achieve sufficient testing of all aspects to the application, which may be missed by just one or two team members carrying out testing.
- **Incremental Integration:** Follow a bottom-up integration strategy, starting with smaller units and progressively integrating into larger systems within each sprint.
- **Environment Simulation:** Initially conduct software simulation within the development environment, ensuring compatibility and functionality.
- **Version Control:** Maintain consistent version control to manage changes and ensure stable integration.

Error Management and Reporting

Effective error management is crucial for maintaining software quality. This includes:

- **Defect Logging:** Systematically log and track any defects, addressing them promptly in order to maintain stability in the software.
- **Regression Testing:** Make sure existing functionalities are still working as intended after any changes or refactoring.
- **Comprehensive Reporting:** The Testing and Integration Manager writes in-depth reports on the outcomes of tests, allowing for the team to make informed decisions.

Collaboration and Iterative Improvements

Collaboration and iterative improvements are key principles in Agile testing and integration:

- **Adherence to Schedule:** Strictly adhere to the testing and integration schedule within each sprint, ensuring timely completion of tasks.
- **Immediate Issue Reporting:** Team members promptly report any issues encountered during testing, allowing for a quick resolution.
- **Continuous Feedback:** Continuous feedback loops facilitate iterative improvements, ensuring the software's ongoing development meets the project goals.

4.3 Quality Auditing Reviews

Producing a high-quality product necessitates an equally high-quality level of production process; the methods utilised during the design and implementation phases are specifically crafted to reach these standards. This is also achieved through careful supervision of the QA metric collections. It is imperative that a review session takes place when wanting to progress onto the next phase of the project, as passing a review session is the only way to progress. These are often conducted by peers, stakeholders, or experts who have the experience and knowledge to provide feedback, suggestions and constructive criticism, however these members are a must:

- **Supplier:** The individual whose product is to be examined.
- **QA and Documentation Manager:** Establishes and oversees the overall review process, documents the results of the review session.
- **Inspectors:** Typically 3 members: one from the previous phase, one from the same team as the supplier, and one from the next phase, to ensure the product meets the necessary requirements for upcoming tasks.
- **Customer:** Not directly necessary, but the customer is often present especially when the deliverable is directly related to the final product.

Within the review session, the team will make use of multiple resources, including the QA metrics linked to the product, in order to look for any errors or inconsistencies. The QA and Documentation Manager will then create the Quality Auditing Report and submit it to the Project Manager. If the review session is failed, the product is unable to progress to the next phase until the errors are fixed.

References

[1] "Apple human interface guidelines." [Online]. Available: https://developer.apple.com/design/human-interface-guidelines/

[2] A. Korunović and S. Vlajić, "An example of integration of java gui desktop technologies using the abstract factory pattern for education purposes," *ETF Journal of Electrical Engineering*, vol. 29 No. 1, 2023. [Online]. Available: https://jee.ucg.ac.me/index.php/files/article/view/241