

Quality Assurance Manual

York Software Solutions

Software Engineering Project

**Department of Electronics
University of York**

Authored by Louis Cowell & Sam Merryweather

Additional contributions made by:

*Cameron Ellwood
Iyra Singh Gaura Bhandal
Ed Brooks
George Harris
Haruna Higa*

Document Change Log		
Change made by	Date	Notes
Sam Merryweather (SM)	28/01/19	Document Created
Louis Cowell (LC)	30/01/19	Added headings for overall structure
LC	01/02/19	Began process of collating documents into single QA Manual document
LC	02/02/19	Added Project Management Methodology & Software Development Methodology
SM	03/02/19	General Formatting Added Title Page, Change Log, Contents Page Added Software Manager and Source Controller QA Metrics Added Document Approval Added Software Test Report Template Added Disciplinary Procedure & Log Template Wrote Introductory paragraph Added Contents
SM and LC	04/02/19	Final Editing and Formatting Added UML Class Diagram

Document Approval Name: <i>Quality Assurance Manual</i> Document Serial Code: 100		
Reviewer(s)	Signature	Date
Louis Cowell <i>Project Manager</i>	LOUIS COWELL	04/02/19
Sam Merryweather <i>Quality Assurance Manager</i>	SAM MERRYWEATHER	04/02/19

1 - Introduction	4
1.1 - Company Overview and Vision Statement	4
1.2 – Personnel	5
1.3 - Organisational Structure	5
2 - Role Descriptions	7
2.1 - Project Manager	7
2.1.1 - Role Overview	7
2.1.2 - Risk Management	8
2.1.3 - Quality Assurance Metrics	8
2.2 - Quality Assurance Manager	9
2.2.1 - Role Overview	9
2.2.2 - Risk Management	10
2.2.3 - Quality Assurance Metrics	10
2.3 - Finance Manager	11
2.3.1 - Role Description	11
2.3.2 - Risk Management	11
2.3.3 - Quality Assurance Metrics	11
2.4 - XML/Server Developer	12
2.4.1 - Role Description	12
2.4.2 - Risk Management	12
2.4.3 - Quality Assurance Metrics	12
2.5 - Marketing Manager	12
2.5.1 - Role Overview	12
2.5.2 - Risk Management	13
2.5.3 - Quality Assurance Metrics	13
2.6 - Brand Manager	13
2.6.1 - Role Overview	13
2.6.2 - Risk Management	14
2.6.3 - Quality Assurance Metrics	14
2.7 - Communications Director	15
2.7.1 - Role Overview	15
2.7.2 - Risk Management	15
2.7.3 - Quality Assurance Metrics	16
2.8 - Testing Developer	17
2.8.1 - Role Overview	17

2.8.2 - Risk Management	17
2.8.3 - Quality Assurance Metrics	17
2.9 - GUI Manager	18
2.9.1 - Role Overview	18
2.9.2 - Risk Management	18
2.9.3 - Quality Assurance Metrics	18
2.10 - Marketing Assistant	19
2.10.1 - Role Overview	19
2.10.2 - Risk Management	19
2.10.3 - Quality Assurance Metrics	20
2.11 - Software Manager	20
2.11.1 - Role Overview	20
2.11.2 - Risk Management	21
2.11.3 - Quality Assurance Metrics	21
2.12 - Source Controller	21
2.12.1 - Role Overview	21
2.12.2 - Risk Management	22
2.12.3 - Quality Assurance Metrics	22
2.13 - Lead Developer	23
2.13.1 - Role Overview	23
2.13.2 - Risk Management	24
2.13.3 - Quality Assurance Metrics	24
2.14 - Integration Manager	24
2.14.1 - Role Overview	24
2.14.2 - Risk Management	25
2.14.3 - Quality Assurance Metrics	25
3 - Project Management Methodology	26
3.1 - Agile Overview	26
3.2 - Agile Life-Cycle	27
3.2.1 – Specification Phase	27
3.2.2 – Planning Phase	28
3.2.3 – Design Phase	28
3.2.4 – Implementation Phase	28
3.2.5 – Testing Phase	28
4 - Development Methodology	29

4.1 - Design Methodology	29
4.2 - Implementation Methodology	29
4.3 - Testing Methodology	29
5 - Deliverables	30
6 - Appendices	32
6.1 - Project Management Documentation	32
6.1.1 - Meeting Documentation Template	32
6.1.2 - Example Project GANTT Chart	33
6.2 - Disciplinary Procedure & Log	33
6.3 - Professional Conduct & Ethics Policy	34
6.4 - Company Documentation	37
6.4.1 – UML Use Case Diagram	37
6.4.2 – UML Class Diagram	37
6.4.3 – Software Test Report Template	38

1 - Introduction

This document outlines how York Software Solutions' Quality Assurance process will operate during product development. Included in this document are outlines of York Software Solutions' vision, members and hierarchical structure, comprehensive descriptions of each member's role within the company, as well as an overview of project management for all project phases and the relevant deliverables that must be submitted during the project lifecycle. Finally, there is an Appendix of document templates that will be used in York Software Solutions.

1.1 - Company Overview and Vision Statement

York Software Solutions is a software development start-up, founded in York, North Yorkshire in 2019. Our 7-person team consists broadly of a Development team, a Marketing team and a Project Management team as well as specialist roles such as Finance. Our company specialises in accessibility software, and we are passionate believers that technology is fundamentally a force for good in modern society and should wherever achievable be utilised to that end. Our first product, **urSign**, is to be an educational translation application for American Sign Language (ASL).

Our vision is as follows:

- 1. To uphold at all times the highest possible standards of software development, customer service and professional conduct.*
- 2. To produce software that aids, educates and inspires the users of our products, and to maximise the potential of our software through comprehensive market research and user feedback; listening to our users, and building software that delivers the best possible experience for them - what is good for our users is good for our clients.*
- 3. To gain a significant foothold in the accessibility software market over the coming years, and to eventually become genuine competitors at the very top in Europe and around the globe.*

1.2 – Personnel

Below is a list of all York Software Solutions employees and their roles within the organisational structure. Note that each member holds two roles so as to maximise the involvement of each team member and foster a culture of collaboration between departments within the company.

Louis Cowell - *Project Manager & Communications Director*

Sam Merryweather - *Quality Assurance Manager & Testing Developer*

Haruna Higa - *Finance Manager & XML/Server Developer*

Iyra Singh Gaura Bhandal - *Software Manager & Source Controller*

Cameron Ellwood - *Lead Developer & Integration Manager*

Ed Brooks - *GUI Manager & Marketing Assistant*

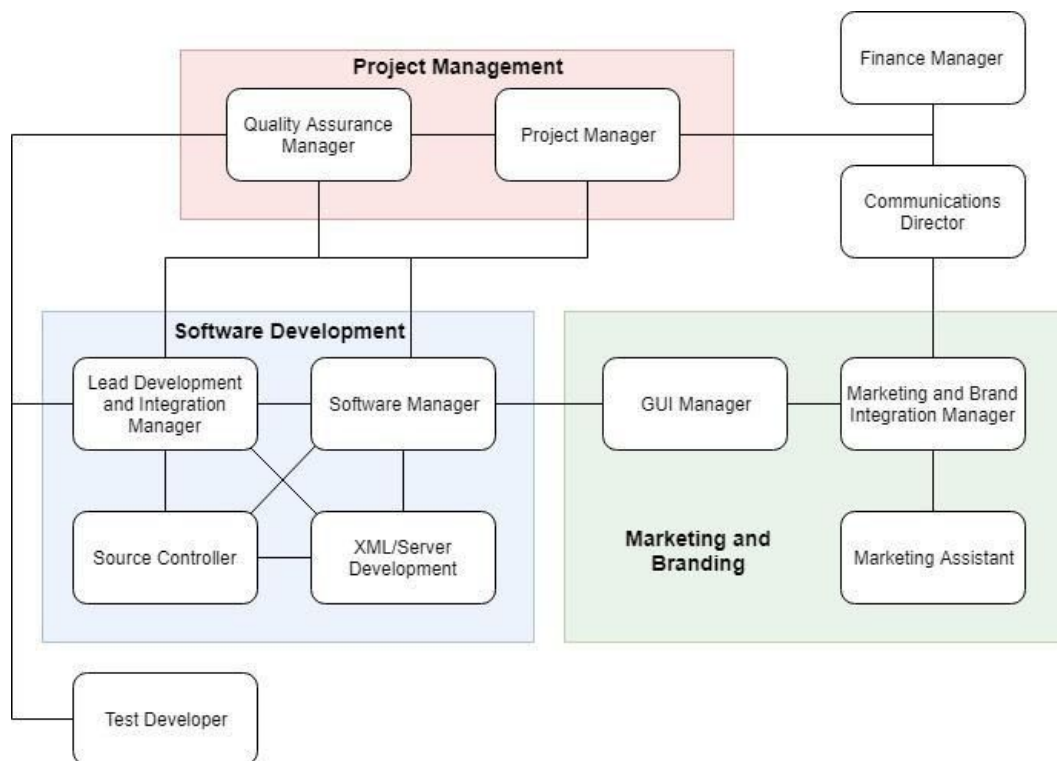
George Harris - *Marketing Manager & Brand Integration Manager*

1.3 - Organisational Structure

York Software Solutions operates with a reasonably flat hierarchical structure, fostering a culture of collaboration and allowing for maximum interaction between individuals and teams. Only when the situation absolutely necessitates it does any one member have jurisdiction over another.

The diagram below outlines the overall structure of the company. Each team member is represented on the diagram as being connected to others based on the amount of communication prescribed between them due to their role. The diagram above roughly illustrates the company's hierarchy; however, as above, it should be noted that due to its small size, the company operates more with a mesh-like structure than a top-down hierarchy, with constant communication between every individual and department beyond the connections shown on the diagram. All members will also hold an equal voice in discussions of company matters and product direction.

It should also be noted that while several different roles are formalised within the company hierarchy, in practice several of these roles may be held by the same person – see **1.2 – Personnel** and **2 – Role Descriptions** for a comprehensive list of personnel and descriptions of each role respectively.



The company is divided up into several main teams, each with a different purpose outlined below:

- **Project Management** - Concerned with overseeing a project's overall progression from start until delivery to the client. Consisting of the Project Manager and Quality Assurance Manager, Project Management must maintain good communication with all other teams to ensure every aspect of the project remains on course.
- **Software Development** - Carries out the majority of the development for the project maintaining consistent communication with management to ensure each software module is on track as laid out in the project progression plan.
- **Marketing and Branding** - Focuses on ensuring the product fits into the company's desired spot in the market of the product currently in development. Some members of the team, especially the GUI Manager, will be working closely with Software Development to make sure a strong brand comes across in all software written.

Other company members who don't fall into specific teams include:

- **Testing Developer** - Must frequently communicate with both Software Development, especially the Lead Development and Integration Manager, and the QA Manager to ensure product modules are completed to an appropriate standard.
- **Communications Director** - Focuses on liaising between the Client and Management to ensure the product is being developed to the correct specification. They may also communicate with the Marketing and Brand Integration Manager to make sure they know how the client wishes the product to look and fit into the target market.
- **Finance Manager** - Ensures the project is always within budget by liaising with the Project Manager and Client via the Communications Director.

2 - Role Descriptions

The role of the Project Manager is to act as the overarching organisational lead on the project – planning, scheduling and delegating work as it comes. The overall goal of the Project Manager is to ensure a smooth, timely and high-quality delivery of the product while meeting client specifications and budget requirements. The Project Manager seeks to ensure clear channels of communication between themselves and the constituent members of the team as well as communication between team members as without communication the project can quickly derail. Furthermore, the Project Manager is a point of contact between any one member of the team with any other and has the general responsibility of dealing with any problems arising with the project or with individual members.

2.1 - Project Manager

2.1.1 - Role Overview

The specific responsibilities of the Project Manager are as follows:

- **Company Strategy:**
 - Formalise company vision statement and ensure it is always upheld
 - Work with the Finance and Marketing managers to provide accurate and realistic costings and marketability models which are in the interests of both the company and the clients
 - Conduct contractual negotiations with staff and clients
 - Formally approve budget & marketing strategy
 - Keep abreast of industry trends and adjust company strategy accordingly
- **Team Organisation:** As above, the Project Manager is responsible for ensuring that the interaction between teams runs smoothly to achieve the project objectives. The Project Manager therefore must:
 - Arrange and minute team meetings, decide agenda and agree actions. Clarify actions with each team member and ensure that previously set goals have been met
 - Ensure that each member of the team understands the objectives and has a clear idea of what they need to be doing at any given stage in the project life cycle.
 - Delegate tasks to the relevant team member or department and regularly check in on progress
 - Use weekly meetings with the team to gauge each member's progress and workload – if necessary, re-delegate work to lighten the workload of an overloaded team member
 - Conduct disciplinary matters should any team member not produce the expected output – though first check if there is a valid reason why this may not be the case. Again, potentially re-delegate work as necessary
- **Project Planning:** Agree upon and hence provide a realistic project life cycle and timeline that is satisfactory to both the client and the team members, ensuring a quality product, timely delivery and a pleasant, studious work environment in which team members are obligated to pull their weight without becoming over-stressed and thus producing a lower-quality output. This includes:
 - Using GANTT charts and other workflow analysis tools to provide a clear and actionable project plan from initial design through dynamic client adjustment to completion and subsequent delivery
 - Ensuring regular and consistent re-evaluation of the product against client requirements
 - Scheduling core employee hours including both group work and meetings

- o Setting deadlines for internal deliverables *before* external delivery to the client to ensure sufficient time for quality control
- **Document Control:** Work with the Quality Assurance Manager to ensure documents including reports, meeting minutes and memos are stored and documented appropriately, with change logs implemented as standard. Additionally, all documents produced independently of the Quality Assurance Manager and the Project Manager must be checked by each before being signed off into official company documentation

2.1.2 - Risk Management

Risk	Mitigation
Client requirements are too vague	Consult with team and return to client with a comprehensive list of necessary clarifications.
Client requirements are unrealistic	Explain this to the client, ensuring that this is done in a way so as not to make the company seem incompetent. Explain in detail why the company will be unable to meet the requirements and provide examples of alternatives that may be agreeable.
Product is not on track to meet requirements	Stop all production and arrange meeting with team members. Demonstrate that product isn't on track and work together on an actionable plan to correct this. Do not resume work until all members are clear on what needs to be done.
A group member isn't pulling their weight in terms of work output, attendance or communication	Check in with team member to ascertain why they haven't been active. Assign another team member to their tasks in the interim. Conduct disciplinary matters if necessary.
Failing group dynamics – group members do not get on or communicate effectively	Make best possible effort to reconcile the team members in question – remind members that regardless of personal matters there is a job to be done, the success of which both members are stakeholders in, and that communication is key to that success.
Document control quality begins to slip	Hold meeting with QA manager and ask them to help rectify situation. Remind all team members the importance of document control and make more of a personal effort to keep on top of it.

2.1.3 - Quality Assurance Metrics

Metric	Measurement
Product is of high quality	Client requirements all met – seek confirmation from the client themselves.
Product is delivered on time	Product is delivered in its entirety before agreed deadlines.
Product is delivered under budget	Difference between allocated budget and project costs is a positive number.
Company employees are happy	Ensure that every member is happy with their workload and company dynamics during meetings, holding individual meetings if needed.
Company vision statement is upheld	Check every major decision made against company vision statement, ensuring that the company vision is enshrined in the decision-making process.

2.2 - Quality Assurance Manager

2.2.1 - Role Overview

The Quality Assurance (QA) Manager's role is to ensure the company's output is of an appropriate level, meeting the specification approved by the customer. They are responsible for overseeing the development of the product as a whole, being mindful of the customer's requirements and product specification at all times. QA Management also involves streamlining development by identifying problems in software as early as possible, whilst also establishing suitable solutions by collaborating with the Lead Developer and Software Manager.

The QA Manager must maintain good communication with both the Lead Developer and Software Manager to ensure the product progresses at the rate agreed upon by the company. They must also regularly update the Project Manager, who should be made aware of all appropriate information on the progression of the product. Much of this communication will be carried out during the company's weekly group meetings, but meetings with team members with more specific agendas may also occur.

The specific responsibilities of the QA Manager are as follows:

- **Specification:** Assess whether software meets the specification agreed upon with the customer
 - Determine what can be done to bring the product back on track to the specification the customer expects
 - Communicate with the Communications Director to ensure the customer is informed of project progression
 - In collaboration with the Project Manager, determine deadlines for different sections of the product

- **Planning:**
 - Establish a QA plan that outlines a set of standards and QA Metrics that can be used to assess the standard of a product
 - Ensure project progression is being appropriately monitored and that the appropriate documentation is completed in a timely manner
 - In collaboration with the Project Manager, determine deadlines for different aspects of the product
 - Ensure project deadlines are met in sufficient time without sacrificing product quality or functionality
- **Testing:**
 - Ensure tests are completed and documented to the company's standard
 - Ensure the Lead Developer is made aware of the results of tests and any actions that may need to be made as a result of the testing process.
 - Help the Software Development team to identify problems in the software and aid in providing realistic solutions
- **Document Control:** Along with the Project Manager, manage and maintain the documentation of the project. This includes changelogs, review documents, marketing & finance material, meeting minutes etc.

2.2.2 - Risk Management

Risk	Mitigation
Work is not completed to an appropriate level	Hold a meeting with the appropriate group member(s) and ascertain as to why the quality of their work has slipped. Workload redistribution may be required to lift the load on struggling group members.
Software produced does not meet the customer specification	Hold a group meeting with the Project Manager, Lead Developer, Software Manager and Communications Director and any other relevant group members working on the part of the program that is not meeting the specification and reassess how the group is working to make sure the specification is met.
Documentation is not completed to the company's standard	Meet with those responsible for filling in the missing documents and find out why it was not completed, focusing on making sure they are able to complete it in the future

2.2.3 - Quality Assurance Metrics

Metric	Measurement
Work is completed to the standard outlined by the QA plan	Work meets the specification outlined in the QA Plan
All documents completed by team members conform to the company standard	All documentation is checked by both the QA Manager and Project Manager to confirm it conforms to the company's standards.
Program meets the specification agreed upon with the customer	Development is carried out with customer specification and appropriate deliverables in mind at all times
Program is delivered in a timely manner	Program is delivered to the customer at the agreed specification before the deadline

2.3 - Finance Manager

2.3.1 - Role Description

The role of the Finance Manager is to oversee the finances of the entire organization. They are responsible for the financial health of the organization, and in order to do this must keep in mind the long-term financial goals of the organization, take responsibility for any risk, planning, and cash management. Communicating with the Project Manager, they must plan and hold discussions on managing the business and workers' activities.

The specific responsibilities of the Finance Manager are as follows:

- **Finance Management:** Communicate with and manage relations with bank and with the Management team to plan financial matters, ensuring that bankruptcy is avoided and that profits are maximised
- **Financial Review:** Identify when a review of the project finances is necessary, and conduct said financial review(s)
- **Financial Records:** Maintain appropriate records for all payments and invoices, including wages and incidental costs
- **Regulations:** Adhere to all financial rules, laws, and regulations

2.3.2 - Risk Management

Risk	Mitigation
Bankruptcy	Weekly cash flow is managed and forecast is used to plan accordingly. Regular maintenance and reports should be made, and a review called for if deemed necessary.
Unforeseen costs	Prudence is practiced, and a weekly cash flow forecast is used to minimise the impact of such costs.
Financial Business Plan is rejected	Plan and consider costs with attention to detail. If needed, consult with Financial Advisor regarding any questions.
Required deliverables incomplete or not submitted on time.	Internal deadlines are set and adhered to.

2.3.3 - Quality Assurance Metrics

Metric	Measurement
Company is in good financial health	Payments are made on time and the balance of the account is never negative.

Good relations with the bank and any associated companies are maintained	All payments are made in a timely manner, documented, and communication between parties is clear.
--	---

2.4 - XML/Server Developer

2.4.1 - Role Description

Responsible for the design and implementation of the server structure and XML files containing the data. All content must be stored and transported as an XML file, which should be readable by the software to deliver the content as intended. The XML/Server Manager should work closely with the Software Manager and Lead Developer to ensure the functionality of the server meets the requirements of the software, and that the XML files are written clearly and optimally for the software.

The specific responsibilities of the XML/Server Manager include:

- **Working Server and Data Storage:** As per the Agile development methodology, ensure that the server is always functional at the end of every server iteration, and that the XML data is readable.
- **Data Management:** Ensure all data is stored properly in an XML file, and that XML files are written in a manner consistent with the rest.
- **Testing:** Communicate and work closely with the Software Manager and Testing Developer to develop suitable testing methods of the server.
- **Records:** Maintain all relevant documentation and previous versions. These should be clear and understandable for any member of the team to have an accurate understanding of the XML data and server functionality.

2.4.2 - Risk Management

Risk	Mitigation
Deadlines for server development are not met	Review reason for delay, and consult with the lead developer to adapt strategy.
Server does not meet needs or is inefficient	Follow the project timeline and functional specifications. Make use of testing to optimize server speed and capacity
Failure of data storage and retrieval	Manage the server to ensure all data is provided correctly.

2.4.3 - Quality Assurance Metrics

Metric	Measurement
Data is read successfully from the XML files	Communication and cooperation with the Lead Developer to ensure compatibility of the server and software
Data can be successfully stored and retrieved from the server	The required tests are passed, with no errors or failures.

2.5 - Marketing Manager

2.5.1 - Role Overview

The Marketing Manager is responsible for both the day-to-day marketing operations within the company as well as developing, maintaining and implementing a long-term strategy, in line with the company objectives. They will also monitor and measure the results of marketing campaigns, reporting the effectiveness to the Project Manager. Data collated should be analysed to inform future marketing strategy development and publication of future marketing materials. The budget for marketing should also be properly allocated to campaigns, with the Financial Director affirming fund distribution.

The specific responsibilities of the Marketing Manager are as follows:

- **Production and Deployment of Marketing Material:** Working with the marketing team to produce materials consistent with the chosen marketing strategy and the brand, then publicising these materials in an effective way.
- **Marketing Analysis:** Once a marketing campaign is underway, it's success will need to be properly quantified, measured and analysed in order to inform future campaigns. The overall strategy may need to be re-evaluated upon analysis of marketing data, if the current strategy is deemed to be ineffective, or simply not the most effective.
- **Marketing Budget:** Working closely with the Finance Manager, the Marketing Manager should manage funds related to marketing campaigns, allocating funds for marketing materials, brand integrations, social media campaigns and other expenses.

2.5.2 - Risk Management

Risk	Mitigation
Marketing is ineffective	Prompt and thorough analysis of marketing campaign data as well as current and future market trend data should allow for marketing effectiveness to be continuously evaluated, with strategies being altered if needed.
Budget not properly allocated – Marketing runs over budget or too much of budget is allocated to Marketing	Maintain a strong line of communication with the Financial Director to ensure funds are well managed.
Target audience is unclear after research	Proper understanding of the product, the current market and competitive products should inform the profile for the audience of products made by the company.

2.5.3 - Quality Assurance Metrics

Metric	Measurement
Marketing outreach has been successful	For a marketing campaign, the percentage of the target market reached is high
Large proportion of target market reached	There is a large number of positive product user reviews from users in the target demographic
Target market correctly identified	The profile of potential clients is known, specific and informs marketing strategy

2.6 - Brand Manager

2.6.1 - Role Overview

The purpose of the Brand Manager is to ensure that all products and services marketed under the brand are recognisable, cohesive and resonate positively with the current and potential client base. They must develop, implement and execute marketing strategies for the brand in order to spread and maintain a unique and positive image. If the Brand Manager has done their job right, the public will be able to quickly and easily recognise the brand and associate it with quality and professionalism. They will work closely with the GUI manager to produce intuitive interfaces which are unmistakably consistent with the brand.

The specific responsibilities of the Brand Manager are as follows:

- **Brand Recognition:** Working alongside UI/UX designers and Marketing team to ensure consistency in branding through common colour schemes, logos and other elements. Users should easily and automatically associate all products under the brand.
- **Market Growth:** Monitoring market trends, competitive products and meeting with clients to inform marketing strategy development, implementing new and revised techniques when necessary. Strategies chosen/developed should be based on the current and future market, to the furthest extent the future market can be accurately predicted.
- **Product Consistency:** While a product is developed in stages by a team of many different people, the Brand Manager should work to make sure each part of the product ‘feels’ to the end user that it belongs amongst the others. That is, sections should flow into each other without the user ever knowing that part A and part B were developed by different people, they should just ‘feel’ that A and B belong together.
-

2.6.2 - Risk Management

Risk	Mitigation
Products ‘feel’ separate or disjointed and are not cohesive in style	Maintain strong communicative relationships with UI/UX team members, ensuring they work together to use a common design scheme and smooth transitions
Product is confused with competitive products	Thorough research into competitive products will inform design choices, making decisions which are conducive to the product’s market while remaining unique
Brand underperforms on the market	Meet regularly with the Marketing team to pool research and promptly analyse data to develop innovative strategies to market the brand

2.6.3 - Quality Assurance Metrics

Metric	Measurement
--------	-------------

Products are easily recognised and associated	When shown a group of products, clients easily and quickly distinguish our brand from others
Potential user base sees regular growth	Regular re-evaluation of the current and potential client base yields larger and more diverse groups

2.7 - Communications Director

2.7.1 - Role Overview

The Communications Director is tasked with maintaining the image of the company. Their main responsibility is conducting client and user-facing communication on a day-to-day basis, acting as a key point of contact with the client in order to ensure important matters such as costings, company vision, marketability and so forth are communicated with clarity and purpose. While the Communications Director role is formalised as a separate entity to the Project Manager within the company's hierarchical structure, the two roles are nevertheless currently held by the same person.

The specific responsibilities of the Communications Director are as follows:

- **Marketing Strategy:** Work closely with the Marketing Manager to ensure the marketing strategy communicates clearly the core company vision and the key benefits of the product. This may include paid advertising and social media strategy.
- **Accessibility:** Given that the business is focussed around accessibility, the Communications Director must ensure that matters of product accessibility are handled sensitively and accurately, with market research focussed around maximising accessibility and involving the disabled users that the products are designed to help at every stage of the design process.
- **Client Coordination:**
 - Perform detailed analyses of client requirements and convert them to actionable objectives
 - Effectively communicate objectives to the team
 - Report progress to the client and use feedback to ensure that product development is always in line with client requirements
 - Act as a key point of contact with client to discuss important matters

2.7.2 - Risk Management

Risk	Mitigation
Marketing material is unclear	Hold regular meetings with the Marketing Manager to formulate consistent marketing strategy.
Company image is in some way tarnished	Write clear company communications & ethics policy which all team members must strictly adhere to. Ensure legal, ethical business and financial practices are followed at all times. In the event of company falling into disrepute, conduct damage limitation via social media and media outlets.

2.7.3 - Quality Assurance Metrics

Metric	Measurement
Communications with client go smoothly	Company and client come to mutually beneficial arrangement, client always appears happy with the result of meetings
Potential user base sees benefit of product	Users express early interest in product, sales of first release are promising
Marketing material clearly communicates vision	Focus groups or similar do not ask for clarification on matters which they would be expected to understand in the first instance

2.8 - Testing Developer

2.8.1 - Role Overview

The Testing Developer's main responsibility is to establish the test plan used to carry out complete testing on sections of a product, under the supervision of the QA (Quality Assurance) Manager. This will consist mainly of a set of automatic test scripts that will test the functionality, stability, and usability of a section of code. These tests verify that sections of the program meet the specification produced by the customer and are collectively referred to as the test plan. Each test will outline how the software should react, what should be seen, and any potential issues that could be raised.

The Testing Developer is also responsible for producing detailed reports of each test carried out that cover the exact circumstances under which the test was completed as well as any changes that need to be made. These reports will then be fed back to the QA Manager and Lead Developer to assess how the results impact the product and any actions that need to be taken.

The specific responsibilities of the Testing Developer are as follows:

- **Planning:** Under the supervision of the QA Manager, establish a test plan and ensure it is well understood by all developers
- **Testing:**
 - Following the test plan, write and carry out extensive tests on code segments using automatic testing methods
 - During testing, complete and formalise the appropriate documentation outlining the results of the test
 - Inform the QA Manager of the results of all tests completed and any actions that need to be completed

2.8.2 - Risk Management

Risk	Mitigation
Program modules fail a test	Fill out the appropriate documentation on the test failure and feed this back to the QA Manager who will inform the lead developer of the issue.
Program module is not ready for testing	Meet with the Lead Developer and explain any issues which may make the module unsuitable to be tested at that time.

2.8.3 - Quality Assurance Metrics

Metric	Measurement
Program Modules function as expected at an appropriate standard	Test reports are completed to the standard outlined in the QA Plan, and the QA Manager is made aware of any issues that may have occurred during the testing process

2.9 - GUI Manager

2.9.1 - Role Overview

The GUI Manager is responsible for the design and implementation of the Graphical User Interface (GUI). This concerns the layout and implementation of buttons, menus and windows, and considers the user interaction with these interface elements.

Since this part of the program concerns direct interaction between the user and the software, market research should be used to determine the preferred user experience of the typical user. Therefore, the GUI Manager must liaise with members of the team concerned with market research so that they have a clear view going into the design of what needs to be accomplished. More market research should also be undertaken to find out on what operating system the initial release of the program should be designed for as subsequent versions for support on other operating systems may require additional GUI development.

The GUI Manager should also be in close contact with the Software Manager, as all the GUI research conducted by the Marketing department will need to be developed and implemented, and members of the Development branch may be needed for consultation or as another pair of hands for coding, particularly if development is behind schedule.

As well as implementing the interactive components for the purposes of user interaction (main menus, user profile screens, content pages, buttons etc.), another big aspect of GUI development is designing a visually appealing program, giving a more professional feel, improving the user experience of the application and ultimately selling more copies, increasing profit and allowing further potential development.

Being a management role, this also entails directing team members developing this part of the program as well as reporting to the Project Manager with regular updates and updating any files to the shared file space. As such it is the responsibility of the person with this role to make sure the development of the GUI is kept on the schedule defined by the Project Manager.

2.9.2 - Risk Management

Risk	Mitigation
Program isn't fully interactive	Make use of testing to make sure that all buttons and sliders work for all inputs
UI layout is inefficient	Design basic layout before starting and use market research to find the most intuitive layout
Visuals aren't aesthetically pleasing	Use market research to find what would suit our audience best

2.9.3 - Quality Assurance Metrics

Metric	Measurement
GUI is delivered on time	Follow project timeline closely, reporting any unexpected delays to the rest of the team

GUI has a high degree of usability	Ensure that every GUI design decision is informed by relevant market research
GUI is consistent across all views and platforms	Consult regularly with the Brand Integration Manager to ensure that the GUI design is in-line with the company branding policy and that every element is 'on-brand'

2.10 - Marketing Assistant

2.10.1 - Role Overview

The purpose of this role is to assist the Marketing Manager in researching competing products to find the best way to design our program, as well as defining a target user base to market to. In order to maximise sales, the program must be well received by users and reviewers alike, which is achieved by creating an application that is carefully designed to meet the needs of the particular user demographic, as well as offering a competitive product compared to others available on the market.

The specific responsibilities of the Marketing Assistant are as follows:

- **Business Model:** Research which business models are industry standard for functionally similar applications and work with the Finance Manager and Marketing Manager to determine how the company should approach the monetisation of the product
- **Target Demographic:** Research the target demographic, finding out the target audience's preferences from the beginning to allow for a user-centric design which should improve the user experience and thus give higher customer satisfaction and result in more sales than if no audience research was conducted.
- **GUI Design:** Conduct research to inform the program design with respect to colours, text styles and user interface, so that the product will appeal to the primary audience demographic. For example, if it were aimed primarily as a teaching tool for school children, the design might include bolder text, bright colours etc. to help engage the user more. If the app were to be designed more with the elderly in mind, then perhaps a high-contrast display would be beneficial for those with poor eyesight.
- **Future Works:** After the initial release, conduct continuing market research to inform the design of future product releases in terms of extra functionality and changes that might be made in future versions. Keeping the market research and client interaction active even after product launch will help ensure that the application sells well and *keeps* selling well.

2.10.2 - Risk Management

Risk	Mitigation
Unclear target audience	Proper understanding of the product, the current market and competitive products should inform the profile for the audience of products made by the company.

Incorrect business plan	Look at multiple competitors to see what business plan has had most success.
Research yields range of different results	See if results can be matched to a certain audience demographic which the team can then decide on.

2.10.3 - Quality Assurance Metrics

Metric	Measurement
Marketing outreach has been successful	Marketing efforts produce a measurable increase in sales
Large proportion of target market reached	There is a large number of positive product user reviews from users in the target demographic
Target market correctly identified	The profile of potential clients is known, specific and informs marketing strategy

2.11 - Software Manager

2.11.1 - Role Overview

The role of the Software Manager is to ensure the fast delivery of high-quality software according to the functional specification of the product, and guide the software development in a managerial capacity, including testing and UI/UX. This includes ensuring that the correct resources are procured and put to use in a way which is compliant with the usage restrictions of the product.

The specific responsibilities of the Software Manager are as follows:

- **Functional Specification:** Define a consistent, comprehensive functional description of the software and provide a reasonable release date for each version.
- **User Stories:** Define which stories make it into the current feature and ensuring there is always someone available to work on them.
- **Planning & Review:** Ensuring the completeness of the backlog in the development process and its review every iteration.
- **Dependency Management:** Identify the dependencies between activities at a high level and ensure a possible sequential completion is followed through in order to prevent dependency loops.
- **Implementation:**
 - Plan the implementation of the functional specification prior to the beginning of each development cycle
 - Delegate development tasks among Development team
 - Decide upon formats and coding standards, ensuring that these are consistent
 - Decide upon the choice of programming language(s), hierarchy of the code and the interactions between objects if applicable via UML diagrams, state transition diagrams etc.

- **Quality Assurance:** Communicate with the QA Manager to ensure that the design meets the quality standards of the software and can deliver appropriately to the users (i.e. that it accurately represents the functional specification and actively achieves the company vision).

2.11.2 - Risk Management

Risk	Mitigation
Lack of well-defined development strategy	Arrange meetings to validate the backlog and arrange completion of the tasks sequentially
Implementation as specified is unrealistic or even impossible	Discuss alternative specifications with the Project Manager to meet the client requirements with minimal disturbance to the existing code
Incompatibility with legacy hardware or software	Explore documentation to find if there is another way of accomplishing the task, and whether it might be better to sacrifice compatibility with one particular hardware or software platform for more accurate working on the other platforms. Go back to the client with concerns and modify the functional specification accordingly.

2.11.3 - Quality Assurance Metrics

Metric	Measurement
Amount of extra software work that needs completion	Count the items in the backlog, the items remaining on the current iteration, and the number of errors and bugs in the current code (reported by the Lead Developer).
Software completeness	Run through the tests for each module to make sure they cover every method, and ensure that everyone's software role on the current iteration has been undertaken. Notify Integration Manager of any set-backs.

2.12 - Source Controller

2.12.1 - Role Overview

The Source Controller's main responsibility is coordinating the use of the Source Code Management (SCM) system at the company. This requires a good understanding of how to use the source control software and furthermore what to do if it goes wrong by accident, and how to recover from such a situation. They should have a strong line of communication with the Software Manager and the Lead Developer, as the Software Manager has the final say on what code in particular makes it into the finished product, and all code must pass through the source control system.

The specific responsibilities of the Source Controller are as follows:

- **Source Control:**
 - Choosing an appropriate source control system
 - Ensuring that all employees have adequate access, particularly within the Development team
 - Reviewing every 'breaking change' before committing, using the source control software's *diff* function
- **Security:** Ensuring that the codebase is kept secure and hidden from the public view, but with read-permissions at a minimum to anyone inside the company.
- **Working Software:** Ensure that the master branch is always clean and working, never with code that does not work at all, in line with the Agile development methodology. Any code that does not compile should be in a development branch or at the very least commented out.
- **Team Coordination:** Making sure that everyone with write access follows source control best practices (commit often, good commit messages)
- **Dependency Management:** Ensuring that external dependencies are managed (if applicable) and that a clone of the code base results in a full working copy of the software.

2.12.2 - Risk Management

Risk	Mitigation
There is a conflict between the file versions two members of the team are working on	Assist in the source control software automatic merging process to ensure that all code is accounted for with no feature loss, and best practices (commenting and indentation) are preserved.
Change log is not up to date with the last modified file	Speak to the lead developer about the feature implemented and issue a correction to the changelog into the system, updating the versioning if applicable.

2.12.3 - Quality Assurance Metrics

Metric	Measurement
Progress towards completion of code	Look through the recently added source to see if it hits any more points on the iteration to-do list. Speak with Lead Developer to make sure there is no new code to be added to the repository. Update the version numbers and merge experimental branches with consent of the Software Manager.

Code complexity	Look for code that has been duplicated in more than one branch or even in more than one file in the project to make sure there are no circular dependencies or multiple implementations. Track down the relevant commit and talk to the Lead Developer about rolling back.
-----------------	--

2.13 - Lead Developer

2.13.1 - Role Overview

The Lead Developer is responsible for coordinating the development of software modules and ensuring that all developers follow the software quality standards outlined by the QA and Software Manager. They need a clear understanding of the development processes of each software module in order to achieve the desired software specification. Furthermore, they should be in constant communication with the Software Manager to ensure that the products specifications are being met, as well as all other developers to aid with development and check module timelines are being met.

The specific responsibilities of the Lead Developer are as follows:

- **Software Specification:** Have a clear understanding of the software specification, software quality standards, development processes and development timelines outlined by the Software Manager and communicated these to the Development team
- **Delegation:** Assign the development of software modules to different Development team members based on their relative experience and strengths
- **Consistency:**
 - Ensure all developers have a clear understanding of their tasks, the chosen software quality standards and development processes to be used
 - Set an example for the Development team by producing high quality software following the design specifications and quality standards
 - Monitor the development of other the software modules ensuring consistency between each module and that each team is working efficiently towards their respective deadlines

2.13.2 - Risk Management

Risk	Mitigation
Code inconsistencies between software modules	Have clear programming standards including software format, variable naming style, commenting procedure etc. and provide both written and diagrammatic (UML) references for how the modules will interact with each other.
Code or module failure	Meet with Development team regularly to discuss any failures and their importance in the program to assign timeline for solving them. Ensure that a change log for each software modules is well maintained and can be passed along to Testing and Integration teams to aid with test cases
Software module development behind schedule	Consider how far behind schedule the module is and assign additional manpower also considering the development of other modules

2.13.3 - Quality Assurance Metrics

Metric	Measurement
Software modules are error-free	Record and report the number of compilation errors encountered and if completed module has been tested and is error free. Include all updates in the change log
Module development is completed and refactored on time	Record creation and completion dates and times for each module. Work closely with the Software Manager to ensure module development doesn't overrun

2.14 - Integration Manager

2.14.1 - Role Overview

The responsibility of the Integration Manager is to manage the integration of multiple software modules and the procedures associated with this phase of development. He/she should hold regular meetings with the Software team and the Testing Developer to make plan for integrating and testing each software module, following the Software Manager development timeline closely. Records should be created for each module stating the testing procedure taken on the module, which other modules it interacts with and how it's planned to be integrated into the overall program. These records should be reviewed and monitored by the Integration Manager to ensure that integration timeline is being met, and then delivered to the Software and Group Managers for further review.

The specific responsibilities of the Integration Manager are:

- **Integration Management:**
 - Design, plan and oversee the integration stage of the software development
 - Work closely with the Testing Manager to ensure the process is handled correctly
 - Monitor the progress of the integration stage and ensure works are on schedule
- **Records:** Collate all testing and integration records from the Development team for review and completion of deadlines
- **Reporting:** Communicate the reviewed records and results of the integration stage as a report to the Software and Group managers

2.14.2 - Risk Management

Risk	Mitigation
Integration deadline not met for specified module	Review reason for delay and reschedule integration plan for specified module, and any other module that relies on the delayed module
Errors when integrating one or more modules	Communicate with Development team and Testing Manager to devise further tests for each module individually to identify the problem area
Records and module change logs not up-to-date	Set regular reviews of all available documents and modules to ensure most up-to-date versions are available

2.14.3 - Quality Assurance Metrics

Metric	Measurement
Integration deadlines met	Manage and record all events that have been completed and the date of completion
Number of modules integrated successfully	Communicate with Testing Developer to collect testing and integration reports and assess their success and completion

3 - Project Management Methodology

3.1 - Agile Overview

The following is a comprehensive description of the project management methodologies employed by York Software Solutions to ensure fast delivery of reliable, high-quality software. York Software Solutions uses Scrum, a lightweight variant of the Agile project management methodology, combined with a Test-Driven Development (TDD) paradigm to produce simple, robust software by developers who value customer collaboration. The Manifesto for Agile Software Development, authored by seventeen prominent software developers, is as follows¹:

"We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more."

The Agile manifesto is centred around twelve core software development principles²:

1. *Customer satisfaction by early and continuous delivery of valuable software.*
2. *Welcome changing requirements, even in late development.*
3. *Deliver working software frequently (weeks rather than months)*
4. *Close, daily cooperation between business people and developers*
5. *Projects are built around motivated individuals, who should be trusted*
6. *Face-to-face conversation is the best form of communication (co-location)*
7. *Working software is the primary measure of progress*
8. *Sustainable development, able to maintain a constant pace*
9. *Continuous attention to technical excellence and good design*
10. *Simplicity—the art of maximizing the amount of work not done—is essential*
11. *Best architectures, requirements, and designs emerge from self-organizing teams*
12. *Regularly, the team reflects on how to become more effective, and adjusts accordingly*

The Scrum methodology of Agile Software Development is characterised by the Development team holding regular meetings to review progress of a series of development phases. Scrum is lightweight, meaning that the management overhead is kept to a minimum in order to allow for maximal development and testing time. Within the Scrum framework are a series of single iterations, called Sprints. Sprints are short, defined by a set period of time which often referred to as a 'timebox'. Each timebox is typically 1-4 weeks and at the end of each Sprint, working software is demonstrated to the

¹ Kent Beck; James Grenning; Robert C. Martin; Mike Beedle; Jim Highsmith; Steve Mellor; Arie van Bennekum; Andrew Hunt; Ken Schwaber; Alistair Cockburn; Ron Jeffries; Jeff Sutherland; Ward Cunningham; Jon Kern; Dave Thomas; Martin Fowler; Brian Marick (2001). "Manifesto for Agile Software Development". Agile Alliance. Retrieved 14 June 2010.

² Kent Beck; James Grenning; Robert C. Martin; Mike Beedle; Jim Highsmith; Steve Mellor; Arie van Bennekum; Andrew Hunt; Ken Schwaber; Alistair Cockburn; Ron Jeffries; Jeff Sutherland; Ward Cunningham; Jon Kern; Dave Thomas; Martin Fowler; Brian Marick (2001). "Principles behind the Agile Manifesto". Agile Alliance. Archived from the original on 14 June 2010. Retrieved 2nd February 2019.

client. This management paradigm places emphasis on working software over slavish devotion to documentation and is intended to maximise the value of team working hours to the client.

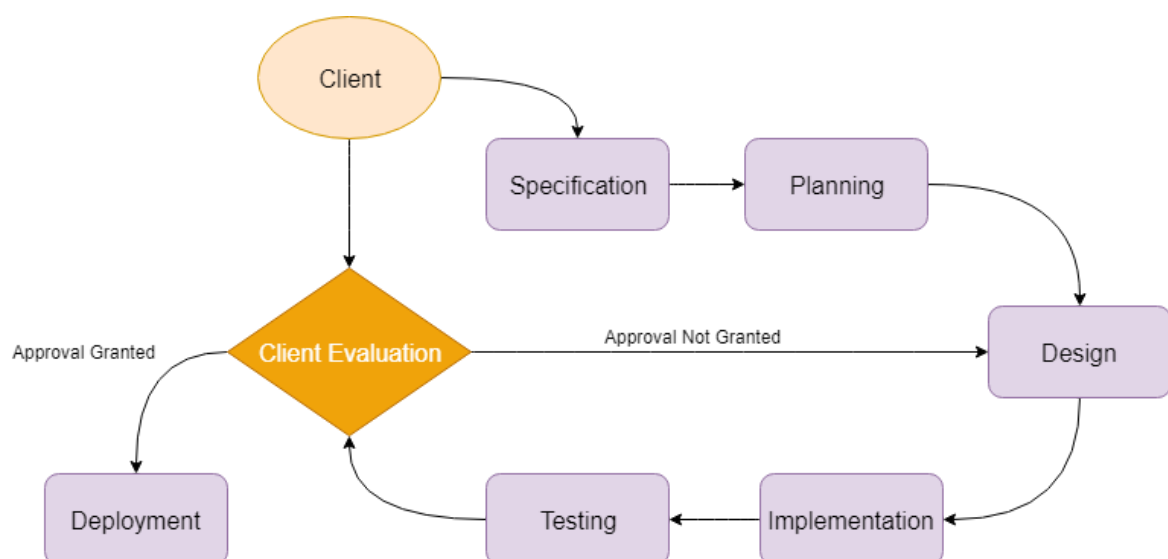
Over the course of a Sprint, one or more *user stories* are typically completed. A user story is a short functional description of some software feature from the perspective of the user. For example, “The user should be able to change the preferences in their user account” is a user story as it describes in as simple terms as possible a single functional requirement of the user. If a user story is too complex, it may take more than a single Sprint to complete – therefore wherever possible it is best to break complex stories up into smaller, more manageable stories in terms of development time.

User stories may have requirements that go beyond the current understanding of the Development team and require further investigation before they can be completed. This is the function of a *spike*. A spike is a functional investigation used to inform the completion of a user story. For example, the Development team may investigate the amount of work it would take to implement video recording capability within an application using the peripheral webcam and microphone of the host device.

In the TDD cycle, traditional programming paradigms are eschewed in favour of writing tests before code, with the intention of then writing code to pass the test. This ensures that the functional specification of the product is at the absolute forefront of the development process as all code is written with the express purpose of fulfilling a functional requirement.

3.2 - Agile Life-Cycle

Our approach to the Agile life-cycle can be expressed in the following diagram:



3.2.1 – Specification Phase

During this phase, an initial consultation is conducted with the client in order to ascertain their specific requirements in terms of project management metrics such as budget and delivery schedule. This is informed by the functional requirements and specification of the product, following a group discussion about what is realistically achievable in the given time frame with the specified budget. Once a specification, budget and schedule has been agreed with the client, the Planning Phase begins.

3.2.2 – Planning Phase

The Planning Phase is where a company consultation takes place in order to devise an internal schedule, broadly delegate design work and come up with a bespoke development strategy for the specific product outlined by the client. The Planning Phase necessitates no functional design or actual software development beyond basic descriptions of each software module within the overall product design. Note from the above diagram that once the Planning Phase is complete, the iterative development cycle begins in earnest with the Design Phase.

3.2.3 – Design Phase

The Design Phase is the starting point in our development lifecycle, whereby the product is designed in full or redesigned in part to meet client specifications and/or evaluation, depending on whether this stage has been reached via initial specification or client evaluation following delivery. The Software Manager, with assistance from the rest of the Development team, devises an implementation of the functional requirements from the top down, and thus produces a functional specification alongside a test plan written by the QA Manager and Testing Developer. Similarly, the Marketing team comes up with a Marketing, Social Media & Branding strategy, alongside a comprehensive program of market research, and the Finance Manager begins monitoring of the financial aspects of the project.

Throughout this process, the Management team and Communications Director maintain constant lines of communication with the Client to ensure the project stays on track with the Client's requirements and expectations. This is also true of the Implementation Phase.

3.2.4 – Implementation Phase

This phase is made up of a series of development iterations, or sprints, each of which lasts around a week. Although in terms of timescale this phase is perhaps the longest, this division of the phase into a series of increments, with constant delivery and re-evaluation of the software, makes it more like a development lifecycle in its own right. This constant, regular delivery of working software is central to the Scrum variant of the Agile methodology, which in turn makes it central to that of our own. Each team member is expected to contribute to the software development during either this phase or the Testing Phase, alongside their additional duties.

3.2.5 – Testing Phase

The Testing Phase concerns the various tests undertaken by the Development team, with assistance from others within the company, to ensure a quality delivery of software to the Client. The testing procedure is overseen by the Software Manager, Lead Developer and Testing Developer - however as with the Implementation Phase it is expected that other members of the company assist with the implementation of the test procedure. Upon completion of the Testing Phase, the finished product is delivered to the Client for evaluation. Note that this evaluation is separate from the software deliveries at the end of each sprint cycle; this evaluation is of the product as a whole, alongside the relevant accompanying marketing and financial material.

If the outcome of the Client evaluation is that of approval, the product is deployed according to a Deployment Plan drawn up by the company. In the case of Client requirements not being met, and thus the product is not approved, the process begins again at the Design Phase, although the product will in all likelihood not be redesigned from scratch but rather aspects of it will be changed to better fit the requirements of the client.

4 - Development Methodology

The Functional Specification document is produced based upon requirements given by the client concerning the function of the product upon deployment. The Quality Assurance (QA) Manager, in conjunction with the Software Manager, will go through the Client's requirements via the Communications Director in order to extract the desired functionality and identify the constraints given by the potentially less technical description from the Client. If more clarification is needed, the Client may be asked to provide a more specific functional description. Sometimes requirements may be contradictory or inconsistent, by which point several possible interpretations should be found, and further enquiries should be made of the Client. Since the contract has not yet been agreed by this point, there may be several iterations of the Client requirements, with the Client being consulted on the exact definition throughout the process. After the document is finally agreed on by the Client, the Project Manager meets with the departmental managers of the company, delegating the fulfilment of the more precise specification to each department.

4.1 - Design Methodology

The design is implemented by the Software team, and as such that department will be the primary group tasked with the architecture design. The design of the product is informed by the Functional Specification document, which is a list of specific features to be implemented. The process of creating this document is the design process; specifically, the Specification ought to match the software being used in the project. The company utilises the **Java** programming language, which allows easier cross-platform support, a great boon for reducing the amount of work required to create multiple functional specifications - if the functions are roughly the same on all platforms, less work needs to be done to differentiate them. The Functional Specification does not delve into the specifics of implementation (such as the naming or hierarchy of modules in a Java project), it only provides a list of features which can be ticked off as they are completed.

4.2 - Implementation Methodology

Using the Functional Specification, the product is implemented by the Software team alongside the GUI Manager and Brand Integration specialist within the company, whom advise on the look and feel of the product, at least in the stages in which there is a graphical interface to show. Any changes or additions or features at this point must be integrated into the Functional Specification. Using the Agile development process, the backlog takes the form of a list of features that are incomplete or partially complete, and every iteration this backlog is reviewed first and given priority over new features. The rate of implementing new features may fall behind if the backlog accrues too many items. The implementation uses the *Git* Source Control Management (SCM) system to track the status of files and folders, keep a history in case the software must be reverted to a previous state and provide a log of who authored each piece of code, when and for what reason. As such, it can be easy to quantify individual contributions, and transfer the project between different working computers. The GitHub platform provides a way to keep a secondary copy of the Functional Specification - through tick-boxes in a 'to-do list' style.

4.3 - Testing Methodology

Once a module is deemed functional by the Lead Developer will forward it into the testing phase of development. It will be tested using automated test scripts written by the Testing Developer in conjunction with the QA Manager. These scripts should self-document the results of the test into the console. Once the test has been completed in full, the Testing Developer should fill in one of the company's official software test reports which will then be passed to the QA Manager. They will then assess whether or not the module passes the test. If the module passes, the next sprint can begin and work can be started on the design of the next module. If not, the module will go back into the design phase for redevelopment.

5 - Deliverables

Deliverable	Producer	Recipient	Due
Quality Assurance Plan	QA Manager	All company members	When company formed
Project Progression Plan (Including GANTT Chart)	Project Manager	All company members	Upon Project start
Functional Specification	Lead Developer	All company members	Upon Project start
UML Diagrams	Lead Developer	All company members	Upon Project start
Module Development Logs	Lead Developer	Project Manager, QA Manager, Software Manager	Throughout Development
Software Implementation Plan	Integration Manager	Project Manager, QA Manager, Software Manager	Upon Project start
Software Integration Records	Integration Manager	QA Manager, Software Manager and Lead Developer	Throughout Development
Software Testing Plan	QA Manager, Testing Developer	Testing Developer, Lead Developer,	Upon Project start
Software Test Reports	Testing Developer	QA Manager, Lead Developer	During development when sections of the program are ready for testing
Meeting Logs (including attendance, agenda, minutes, and actions)	Project Manager/Chair of meeting between group members	All company members	At the end of every meeting
Financial Business Plan	Finance Manager	Tony Ward	February 8 Week 5 Friday
Financial Report I, II, III	Finance Manager	Tony Ward	Through Spring and Summer terms
Financial Financial Summary Report	Finance Manager	Tony Ward	Summer Week 6 Friday
Profit and Loss Statement	Finance Manager	Tony Ward	Summer Week 6 Friday
Weekly Timesheet	All group members	Finance Manager	By the start of weekly Monday team meetings
Payslip	Finance Manager	All company members	Weekly

Marketing Plan	Marketing and Brand Integration Manager	Project Manager, Software Manager	Before first iteration
Market Research	Marketing and Brand Integration Manager, Marketing Assistant	Project Manager, QA Manager, Lead Developer, GUI Controller	Throughout Development
Social Media Strategy	Marketing and Brand Integration Manager	Project Manager, QA Manager, Lead Developer, GUI Controller	Throughout Development
Product demonstration	All group members	Client	Final project deadline
Completed Product	All group members	Client	Final project deadline

6 - Appendices

6.1 - Project Management Documentation

6.1.1 - Meeting Documentation Template

Meeting [NUMBER]

Serial Code	Meeting Date	Attending	Absent

Round The Table Updates:

To be conducted in no particular order.

Team Member	Update	Remarks
Haruna	Financial	
Haruna	XML/Server Development	
George & Ed	Marketing	
George	Brand Integration	
Ed	GUI	
Louis	Project	
Louis	Communications	
Sam	QA	
Sam	Testing	
Iyra & Cameron	Software Design & Development	

Agenda:

Item	Added By	Remarks

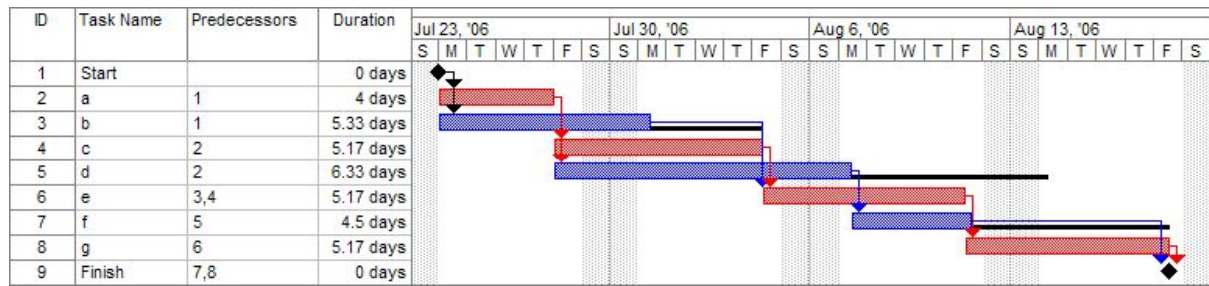
Actions:

Actions from previous meeting completed? Y/N

Item	Remarks

Any Other Business:

6.1.2 - Example Project GANTT Chart



3

6.2 - Disciplinary Procedure & Log

The company operates on a 'three strikes' system, whereby three disciplinary meetings called on the behalf of an employee can result in automatic dismissal from the company. This will not necessarily always be the result, and it is not the only route to dismissal; for example, a violation of the Professional Conduct & Ethics Policy is sometimes grounds for dismissal, if the offence is sufficiently serious.

Occasionally, a disciplinary matter will necessitate a report or memo being drawn up by the Project Manager, Quality Assurance Manager or relevant departmental manager. The format and wording of such a report is at the discretion of the author and, as with any report, subject to the company standard Quality Assurance process. That is to say that it must be signed off by the Quality Assurance Manager and Project Manager before being logged in the company archive.

Each disciplinary meeting entered into the log will be accompanied by a memo, sent by email to the employee in question, giving the formal reasoning for the disciplinary action taken and requesting that it not happen again. A copy of the report, if any, should be attached to the email. Below is a template of how such a disciplinary memo should be written:

Dear [EMPLOYEE],

York Software Solutions expects at all times that our employees display a basic standard of performance, that our standards of Professional Conduct & Ethics are respected, and that you exhibit exemplary behaviour at all times. It is our great regret that on this occasion you have failed to uphold these standards, and following the disciplinary meeting of [DATE], I must issue you with a disciplinary memo and warning. The reasons for this warning are as follows [REASON]. Please do not let it happen again.

I attach a copy of the Disciplinary Report.

Yours sincerely,

[MANAGER]

Disciplinary Meeting Log:

Below is a log of all disciplinary meetings - including the date of the meeting, the employee in question, the reason for the meeting and the action, if any, to be taken.

³ Source: https://en.wikipedia.org/wiki/File:Pert_example_gantt_chart.gif, Reproduced with full permissions. DOA: 04/02/19

Date	Employee	Reason	Action Taken

Signed:

_____ Project Manager

_____ Quality Assurance Manager

_____ Employee

Note that in the case where the Project Manager or Quality Assurance Manager are the subject(s) of a disciplinary meeting, the relevant signatory shall be replaced by an appropriate departmental manager.

6.3 - Professional Conduct & Ethics Policy

Below is a copy of the Professional Conduct & Ethics Policy that each employee must sign before they are allowed on the payroll at York Software Solutions. It details York Software Solutions' broad policies with regards to professional standards, ethics and the law.

The following outlines the standards of conduct and ethical principles that all company employees are expected to uphold. York Software Solutions aims to succeed in earning our client's trust via our commitment to the highest possible standards of professional integrity and ethics at all times. By maintaining these core values at the heart of what we do as a company, we can be sure to sustain a constantly growing reputation with clients and users, and healthy working relationships with our colleagues.

If you believe a violation of these standards has occurred, it is of paramount importance that you get in touch with the Quality Assurance Manager, the Project Manager or a trusted colleague as soon as possible in order to preserve the integrity of the company and to prevent any indication of your own complicity in said violation. The Company operates a zero-tolerance policy on retaliation against those who report breaches of professional conduct.

If these professional, ethical and legal standards are not met, the Company will take the appropriate action to rectify the matter and disciplinary action may be considered. Each employee agrees to:

1. Abide personally by the *Professional Conduct & Ethics Policy* at all times.
2. Report any and all suspected violations to the Management team.
3. Act immediately to prevent or mitigate breaches professional or ethical standards.

Each key tenet of the Policy is outlined below.

Diversity & Respect:

As an accessibility software development firm, diversity is at the centre of everything we do. York Software Solutions is committed to creating a safe working environment that works for everyone regardless of race, gender, sexuality, physical or mental disability, age or religion. Harassment on the basis of these factors shall be considered a grave violation of the Policy and may result in immediate dismissal from the Company.

Harassment may include (but is not limited to):

- Verbal remarks (comments, inappropriate jokes etc.)
- Written communications (offensive emails, letters, memos etc.)
- Offensive pictures or videos
- Physical behaviour (pushing, unwelcome touching etc.)

A note on *indirect harassment* – one does not necessarily have to be offensive to a person's face to harass them. For example, constant remarks or emails concerning a particular person, sent to one or many other people can in many cases constitute indirect harassment.

Legality:

The law must be followed by our employees at all times; this necessitates full understanding of the laws relevant to our business in the markets in which we operate. This is particularly important in the context of our planned expansion into the pan-European and eventually global market. Aside from the personal legal consequences that may arise from breaking the law, the Company shall consider any knowing violation of the law during the course of one's work to also have been a violation of the Policy.

Communications:

York Software Solutions is a modern company, and as such we are aware of the power of communications technologies as tools for furthering our business goals. Social media in particular can be an extremely useful aid to stakeholder engagement. It is important to note however that communications on social media are a permanent record of our online communications, and that in using social media for business purposes we must maintain the image and reputation of the Company at all times. Our Policy with regards to social media is as follows:

- Social media strategy is strictly handled by the Marketing Manager. Any concerns or questions related to the use and/or misuse of social media should be raised with them. Employees must confirm all social media posts with the Marketing Manager beforehand.
- Employees must not use company social media accounts in a personal capacity.
- Social media accounts must be kept secure at all times. Only the Marketing team may keep copies of login information and social media accounts must never be left logged in on any machine.
- Social media posts must uphold the basic standards of integrity and ethics detailed elsewhere in the Policy.

Additionally, other online tools such as Slack, GitHub and email services must be used responsibly at all times. When using digital communications technologies, employees must adhere to the tenets concerning Diversity, Legality and Integrity detailed elsewhere in the Policy

Integrity:

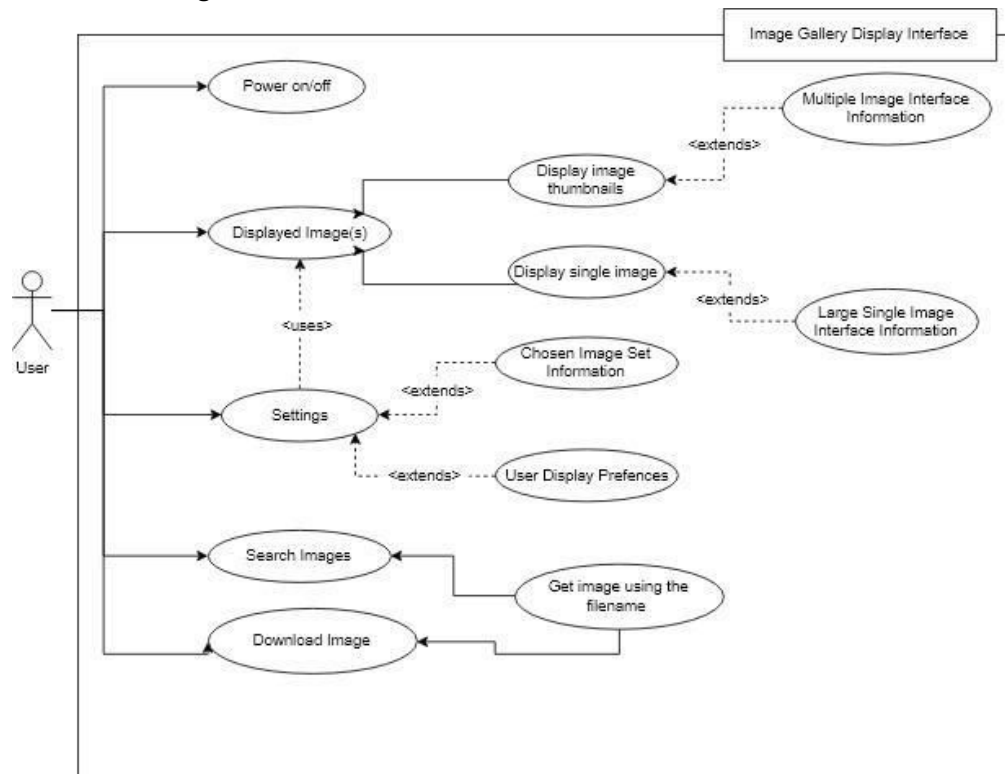
Integrity is vital to the continued success of our Company. We rely upon our relationships with users and clients to provide excellent service at all times and these relationships can be damaged irreparably by breaches of integrity. Collectively, we must strive at all times to:

- Be honest in our relationships with clients and users – let nothing get in the way of total accuracy in all of our spoken and written communication.
- Obtain business in a fair and ethical manner – this means complete transparency in how we obtain work and honesty concerning our capabilities and costs as well as those of our competitors.
- Handle competition ethically. Competitive advantages must only be obtained through fair means such as fast delivery of high-quality software at a low cost. Practices including, but not limited to, bribery, planned obsolescence and reverse-engineering of competitor's software are henceforth absolutely prohibited by this Policy.
- Obtain information fairly. Occasionally obtaining a competitive advantage necessitates seeking information about our clients or competitors. We must ensure that only publicly available information or otherwise ethically and legally acquired information is used to these ends.

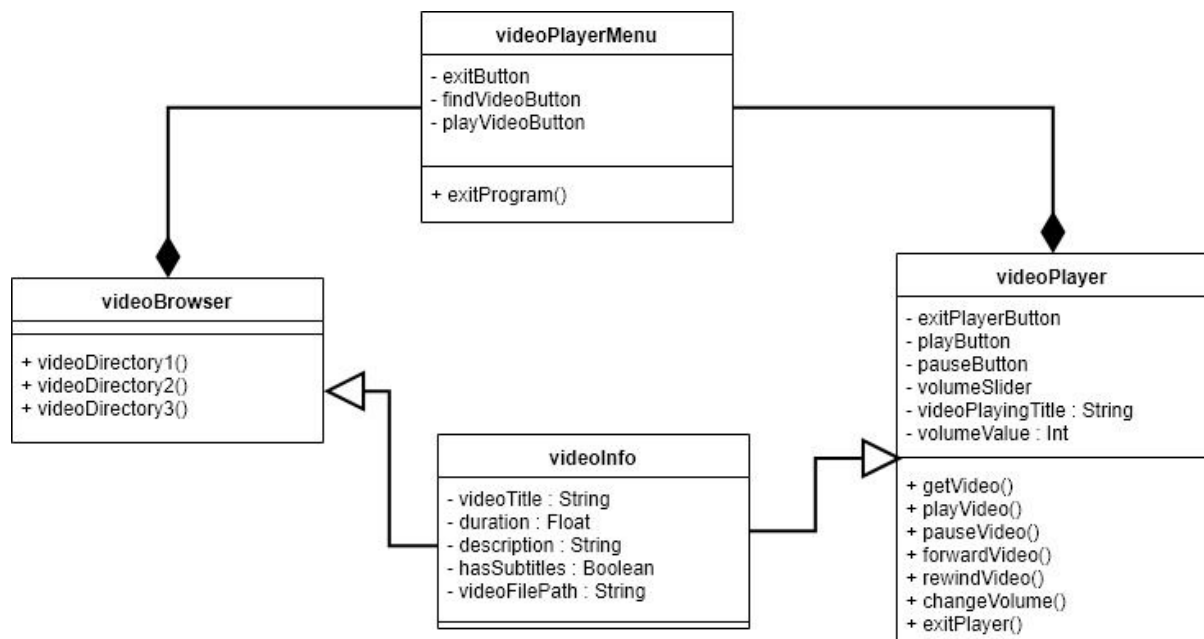
This concludes the *Professional Conduct & Ethics Policy*. Each employee must use their best judgement to apply the tenets found herein to their working lives and promise to uphold the Policy at all times. If a specific ethical situation is not covered in the Policy, employees must understand that this does not mean that ethical breaches related to the aforementioned situation are exempt from constituting violation of the Policy. The Company reserves the right to conduct disciplinary matters regarding any perceived violation of the core values expressed by the Policy, regardless of whether they appear specifically in the Policy or not. York Software Solutions' reputation in the industry are fully reliant on diligent compliance, and we thank our employees for always meeting their obligations under this Policy.

6.4 - Company Documentation

6.4.1 – UML Use Case Diagram



6.4.2 – UML Class Diagram



6.4.3 – Software Test Report Template

Software Test Report:

Project Name:

Module Name	Serial Code	Test Name	Tester

Test Description	Test Result	Errors	Warnings	Remarks

Tester Signature:

Date:

Approval:

QA Manager Signature:

Date:
