Quality Assurance Manual

Group 3

June 7, 2022

Contents

Revision History

Revision	Date	Author(s)	Description
0.1.0	15.02.22		Doc created
0.2.0	14.03.22	DM1306	Editorialise. Changed "Deliverables Overviewer" to "Group Chair". Added Manager/Co-ordinator where required. Removed Akif and Alan. Action AEW Comments
0.2.1	18.03.22	DM1306	Re-add Akif to roles
0.2.2	30.04.22	PG850	Expand some sections. Add Developer role.

1 Introduction

This document shall set-down the framework for managing our project, and for delivering a quality product.
Outlined here are the roles that the team will take to allow us to deliver a quality product, as defined by the Functional Specification, including an overall role description, the risks attached to the role and how they will be mitigated, and the role's QA Metrics.
The deliverables are also outlined here with due dates and the roles involved in producing them. Sections on the overall project risks, metrics and methodology are also included.
For an overview of terms used in this document, please see Appendix C

2 Roles

2.1 Overview

To help ensure on-time completion of the project, we shall give ownership of certain areas of the product to certain people with certain roles and responsibilities - though it will be noted that these roles do not have hard boundaries, and all members shall be expected to take some part in all aspects of the project.
Our role structure is *relatively* flat and should hopefully be somewhat *self organising* - the highest precedence in decision making is shared between the "Business Manager" and "Group Chair"; beneath these are "Documentation Co-ordinator", "UI Co-ordinator", and "Infrastructure Manager". The "QA And Testing Co-ordinator" role is effectively outside of this structure as it pertains to all parts, although is not able to make major changes or executive decisions without agreement from at least one of the two leading roles.
Most roles will be shared by multiple members of the project, due to the large amount of required roles compared with the number of team members available.

2.2 Group Chair

2.2.1 Role Description

This role should plan the structure of the software, setting clear standards for code, and overseeing the integration of components into the product. This role shall also work with the Client to ensure that the received specification is complete - being accountable for ensuring that the design and development process runs smoothly.
The Chair shall communicate directly with QA & Testing to help guide the metrics and to aid in Test design, and should communicate regularly with the Business Manager to ensure an accurate an overview of the business progress requirements, and to aid in discerning direction.
The Chair shall "chair" meetings when present, shall lead design review meetings, and shall have the final decision over major directional changes.

2.2.2 Role Risk Management

Risk	Severity	Likelihood	Mitigation
"Integration" failure	Low	Medium	Ensure coding standards are being followed.
Client requirements not satisfied	High	Medium	Review prioritization of components with Business Manager.
Run-time errors	Medium	Medium	Work with QA&T to review testing.

2.2.3 Role QA Metrics

Metric	Description
Product Completeness	How much of our specification is implemented?
Product Deadlines	Internal and External deadlines for the Product and Features are met or exceeded.
Build Errors/CICD Failures	Minimise - attempt to bring to nil.

2.3 Business Manager

2.3.1 Role Description

The Business Manager is responsible for the managing the regular "business" and financial aspects of the project. They shall work with the Group Chair to liaise with the Financial Backer to secure funds for the overall project, and with other companies to negotiate and secure contracts.
In Development, the Business Manager should work with the Group Chair to have an overview of the development and to aid in directing the working of the team toward the common goal.

In the Business-Management area, the Business Manager will take overall responsibility for organizing and delegating the finances and accountancy and shall work with the Chair to produce a marketing strategy for the product; the Business Manager shall also manage time recording, and overall progress reports fed-back Group Chair and the group. When the Group Chair is not present, the Business Manager shall "chair" any meeting.

2.3.2 Role Risk Management

Risk	Severity	Likelihood	Mitigation
Budgetary Overrun	High	Medium	Keep accurate budget sheets. Use chargeable resources sparingly
Failure to meet business requirements	High	Low	Have regular "business" reviews.
Failure to meet software requirements	High	Low	Business Manager be present at Retrospectives

2.3.3 Role QA Metrics

Metric	Description
User-story clarity	Agreement on issue must be made between Client and Team
Deadlines Met	Deliverables submitted as timetabled.
Budget	Actual budget used in at or below the target.

2.4 QA & Testing Co-ordinator

2.4.1 Role Description

The role will co-ordinate with the rest of the group to create a testing plan, implement Unit Tests, shall work specifically with the Infrastructure Manager to implement the CICD pipeline, and shall work with the Group Chair and Documentation Co-ordinator to help produce the QA Manual.
With a "test-driven" approach, test cases must be developed prior to complete implementation, ensuring that we take time to think about how our interfaces are specified; these tests will be added to throughout the project, and may require replacing as and when interface designs are changed, following agile principals.
The role will also include creating reports throughout the project in alignment with our QA metrics, ensuring that Documentation, Code, and Business Practices meet our specified standard of Quality, and are completed in a timely manner.

2.4.2 Role Risk Management

Risk	Severity	Likelihood	Mitigation
Group QA targets not hit	Medium	Medium	Work with members to ensure targets are hit.
Bugs missed in testing	Medium	High	Ensure test suite is correct.
Schedule overrun	High	Medium	Redesign the schedule, after the Retrospective, for the next time segment.
Missing work	Medium	Low	Ensure all Documents and Code are in the Git repository.
Code not standards compliant	Low	Medium	Ensure all members are informed of our coding standards.

2.4.3 Role QA Metrics

Metric	Description
Test Coverage	Maximise code-coverage of test suite, aiming for $\geq 70\%$.

2.5 Documentation Co-ordinator

2.5.1 Role Description

Work with other Group members to collect and manage the documentation for the product, ensuring that content and formatting is correct and adheres to a consistent style typeset using LaTeXwhere possible. The role shall also ensure that documentation is written using correct grammar, and is understandable by the target audience.

2.5.2 Role Risk Management

Risk	Severity	Likelihood	Mitigation
Documentation not consistent	Medium	Medium	Use standard style and typesetter - the standard LaTeXpreamble may be found in the Documentation Repository. Co-ordinate with others writing documentation to ensure all writing members' written styles are conformant with one-another.
Incorrect use of English	High	Medium	Remember Grammar. Use a spellchecker.
Loss of control of document versions	Medium	Low	Ensure that document revision tables are completed promptly and accurately.

2.5.3 Role QA Metrics

Metric	Description
Spelling Mistakes	Minimise.
Language Coherence and Consistency	Upon reading, documents should be Coherent and should be visually and stylistically consistent.
Formatting Coherence and Consistency	Documents should be coherent and consistent.
Documents produced on-time	

2.6 UI Co-ordinator

2.6.1 Role Description

Implement the User Interface by co-ordinating with the rest of the group and through consulting the Functional Specification.

2.6.2 Role Risk Management

Risk	Severity	Likelihood	Mitigation
UI not displaying all requisite features	High	Low	Ensure UI is designed to the Functional Specification.
User Experience hindered by incomplete features	High	Low	Work with the "owner" of that feature to implement a fix.

2.6.3 Role QA Metrics

Metric	Description
Usability	How easily *usable* is the UI? Minimise the number of clicks to navigate to the most used options. Minimise User surprises.
Client Satisfaction	Client must be satisfied by the design and implementation - measured via client feedback.

2.7 Infrastructure Manager

2.7.1 Role Description

The Infrastructure Manager manages the Git repository, CICD pipelines, and any other miscellaneous Infrastructure required for the success of the project.

2.7.2 Role Risk Management

Risk	Severity	Likelihood	Mitigation
Unresolvable Git Conflict	Medium	Medium	Resolve Conflict, with input from relevant git users.
Build/CICD breaks	High	Medium	Don't break build system.

2.7.3 Role QA Metrics

Metric	Description
Unresolved Merge Requests	Minimise the number of active, unresolved Merge Requests.
Unresolved Issues	Minimise the number of active, unresolved Issues.
Git Accessibility	Minimise the amount of time a member cannot access the Git.

2.8 Finance Co-ordinator

2.8.1 Role Description

The Finance Co-ordinator shall keep track of the income and outgoings of the group, co-ordinating with the Business Manager to help in management of the group's finances and accounts. They shall, alongside the Business Manager, liaise with the Financial Backer to secure funds for the overall project, and with other companies to negotiate and secure contracts.

2.8.2 Role Risk Management

Risk	Severity	Likelihood	Mitigation
Budget Overruns	High	Medium	Keep track of what money is going where and when it is going, notifying the other members of the company also.
Contracts not fulfilled financially	High	Low	Notify the buyers of the contract that payment is due and secure it.

2.8.3 Role QA Metrics

Metric	Description
Financial Viability	Does the company have the required capital to proceed with development.
Optimal Pricing	Ensure that an optimal price is calculated for contract work and for the product sales structure, factoring in Overhead Recovery.
Good Financial Management	Minimise expenditure.
Contracts Pricing	Optimise the contract provision and purchase price.

2.9 Developer

2.9.1 Role Description

The Developer role is to build the product outlined in the Functional Specification, whilst co-ordinating with all other roles, to meet the requirements of the project posed to the group.

2.9.2 Role Risk Management

Risk	Mitigation
Software not adhering to standards set out by Functional Specification	Work with the Group Chair and Businesses Manager ensure that all members are aware of the specification.
Product not passing QA tests	Work with QA to ensure the product meets the QA specification.

2.9.3 Role QA Metrics

Metric	Description
Development of Software On Time and Up To Specification	Software must be developed to the specification described in the Function Specification.
Fulfillment of Features	Product features must be implemented.
Fulfillment of Time Plan	The product must be delivered to the client in a timely manner.

3 Deliverables

Deliverable	Person(s) in Charge	Target Recipient	Due
QA Manual	QA Manager + Business Manager	All Team Members & Business Mentor	Spring Week 5, Monday
PWS	All Team Members	Other Development Teams	Spring Week 6, Thursday
Functional Specification	Documentation Manager + Deliverables Overview	Customers + Developers	Spring Week 4, Friday
Design Documentation	Deliverables Overview	All Developers	Throughout Project
Inter-Group Contracts	Business Manager	Other Development Teams	Spring Week 8, Tuesday
Test & Integration plans and reports	Testing Manager	Customers	Spring Week 10, Friday & Spring Week 10, Friday
Financial Report	Business Manager	Customers + AEW	Spring Week 7, Friday & Spring Week 9, Friday & Summer Week 3, Friday & Summer Week 6, Friday
First Iteration Source Code	Deliverables Overview + UI Developer	Customers	Spring Week 10, Friday
First Iteration User Manual	Deliverables Overview + Documentation Manager	User	Spring Week 10, Friday
Minutes of Meetings	Business Manager + Documentation Manager	All Team Members	Summer Week 8, Thursday
An example Multimedia Presentation	All Team Members	Customers	Summer Week 8, Monday
An HTML-tour of the product	Infrastructure Manager	Customers	Summer Week 8, Monday

4 General Project Risk Management and Mitigation

These are risks associated with the overall project, and not tied to any specific role. Role-specific Risks are declared in §2.

Risk	Severity	Likelihood	Mitigation
Overall quality is not of a deliverable standard	High	Medium	Communication within the team and re-read QA to implement changes to improve the quality of the product.
Lost work	Medium	Medium	Backup everything and keep up communication within the team so every member knows where all work is kept.
Sections of work overrun their internal deadline	Medium	Medium	Update and redesign the schedule for the current time scale as and when needed. Be strict with time to get all deliverables completed.
Illness impeding a members ability to complete work	High	Low	Ensure that all members have some working knowledge of all parts of the project.
Loss of one or more group members	High	High	Ensure that all members have some working knowledge of all parts of the project.
Errors within code being difficult to debug	Low	High	Employ several members of the team to help and debug as a group.
Imperfect User Experience due to missing parts	Medium	High	Implement missing features and ensure our core features work well.
Errors when merging code between coders	Medium	Low	Make sure to follow coding standards
Business requirements not met	High	Medium	Meet regularly to review requirements and subsequent outcomes.
Product not deliverable by the deadline	High	Medium	Regular team meetings, and coding meetings to ensure required work is being completed.

5 Project QA Metrics

These are QA Metrics associated with the overall project, and not tied to any specific role. Role-specific Metrics are declared in §2.

Metric	Description
Effectiveness of Product and Quality	What percentage of the Functional Specification is implemented in the shipped product and what percentage of the initial idea has been fulfilled?
Usability of the Product	In end-user testing, how easy is the product to navigate and control. Users should review on a 1-5 scale.
Efficiency of Code and System Resources	How much CPU time does the program take to run? How much memory does the program require to run?
Fulfilment of Current Task	Are the planned tasks completed before the deadlines we have set for them? Where are deadlines being overrun and what can be done to meet them in future

5.1 Version Control

5.1.1 Documentation

Each document shall contain a Version History table for quick-check version control, containing an overview of changes between versions. Version numbers shall be as follows: Increment the most significant number for a major Release document, increment the middle number for a major change, and increment the least significant number for minor changes.
Documentation shall also be kept in a git repository to provide full version control.

5.1.2 Source

Sources shall be kept in a git repository to enable easy version control of sources.

6 Project Methodology

6.1 Agile

Our project shall be completed using an *Agile Methodology* to aid us in meeting our Product Specification. Within our methodology we should especially remember the following principals and avoid much of the strict, named, Dogma that has become associated with Agile since 2001.

> "The best architectures, requirements, and designs emerge from self-organizing teams."
>
> "Continuous attention to technical excellence and good design enhances agility."
>
> "Working software is the primary measure of progress."

Iterative, Incremental, and Test Driven does not mean, however, that we should be *"debugging code into existence"* (See Expert C Programming Chapter 8 - Software Is Harder than Hardware!, by Peter van der Linden); incremental development requires building a working, simple, skeleton of the program first and then incrementally implementing the more complex parts - not "writing a fast slapdash first attempt, and then getting it working by successive refinements over a period of weeks by changing parts that don't work."

6.2 In Person, or Remote?

One of the Agile Manifesto's 12 principals is that "The most efficient and effective method of conveying information [...] team is face-to-face conversation." With this (and the potential for changing situations) in mind we should aim to have some face-to-face meetings, but with much of the work and smaller, procedural, meetings being carried out remotely where possible.

6.3 Reviews

Our Documentation and Product shall be subject to regular reviews through design review meetings, as well as automated testing through CICD pipelines managed by the Infrastructure Manager.
To merge into the Master branch, it is required that a Merge Request be opened and at least one other person must review and either approve or close the Merge Request ensuring that each addition has an overview-review before it may be included - it is hoped that this will catch many potential flaws or style violations.
More in-depth formal code reviews shall be completed at regular intervals or after large (potentially breaking) changes; these should hopefully catch defects that were missed by merge-reviews and testing.
The Review process shall be led by the Group Chair, but must include all group members to allow them to benefit from any flaws or new methods identified by the review process.

6.4 Help

If members of the group require help in certain language features, it may be that other members have an understanding of these. In this instance, help shall be available on request in whatever form is thought necessary; Pair-programming is not envisioned as a regular activity, although it may be undertaken where it is deemed beneficial to the group.

A Project Coding Style

We shall follow the Sun Microsystems Java Code Conventions, available from Oracle here: *https://www.oracle.com/technetwork/java/codeconventions-150003.pdf*, but with some additional points due to new Java Language features introduced since 1997.

- Do not write methods which *may* return null. Instead, for clarity and safety, an Optional must be returned in such a case.
- Use of automatic Type Inference (the "var" keyword) is permitted for local variable declarations.
- Use of Lambda Expressions and Functional Interfaces is encouraged where practical.

B Role Assignment

Roles	Members
Business Manager	Boris
Group Chair	David
QA & Testing Co-ordinators	Sam, Pat
Documentation Co-ordinators	Brandon, Akif
UI Co-ordinator	Peter
Infrastructure Manager	David
Finance Co-ordinator	Brandon
Development	All

C Terms

- Agile: A set of development principals valuing interactions and which allow rapid, agile response to changes.
- Retrospective: Reflecting on past success or failure to improve future process.
- CICD: Continuous Integration and Continuous Delivery. Automated test, build, and deployment.