Quality & Risk Management

Reese Gunardi - 12109627 ​

Pierce Kinghorn -12091381 ​

Myles Renaud - 12096375 ​

Jack Rolfe - 12017976​

Nick Kelly – 0194134

Table of Contents

[1 Quality assurance 2](#_Toc84169028)

[2 Monitoring Quality 2](#_Toc84169029)

[3 Controlling Quality 3](#_Toc84169030)

[4 Persona Based Quality Assurance 3](#_Toc84169031)

[5 Development & Quality Project board Process 4](#_Toc84169032)

[6 Risk Management Plan 5](#_Toc84169033)

[6.1 Risk Identification 5](#_Toc84169034)

[7 Risk Assessment 6](#_Toc84169035)

[7.1 Risk Mitigation 7](#_Toc84169036)

[7.2 Lessons Learned 8](#_Toc84169037)

# Quality assurance

Quality assurance is vital to the project’s success to ensure a high standard set and achieved for the final product, knowing this the development team used several key strategies to maintain a high baseline level of quality. Maintaining schedule is critical to project success to ensure that developmental paths align allowing fundamental architecture and features to be implemented in time to support asynchronous development of systems with prerequisites. This includes fully implemented features that are relied upon throughout development so that other features may be added. Code requires constant validation and refactoring to ensure optimal programming methods and code are the result. Individual team member updates allow everyone to understand the status of concurrent tasks and features being implemented, features that may affect other members' planned work.

To summarise, quality is controlled by.

* Maintain schedule
* Fully implemented features
* Code refactored and validated
* Individual updates

# Monitoring Quality

Monitoring quality starts by having well defined plans and procedures in place to support the organisation and monitoring of quality control. Sprint initiation meetings are scheduled weekly with predefined meeting goals and processes to determine user stories and other criteria the upcoming sprint must adhere to. Team acceptance testing helps monitor the quality assurance goals mentioned above by documenting the sessions. Code walk through are in conjunction with team acceptance testing as they can determine the same end results and require the same refactoring to meet the requirements. These are documented and sometimes recorded for reference and review as a team to ensure methods are suitable and in some cases adjusted going forward.  
KMS updates provide the mentioned documentation and procedures results to other team members outside of the current testing group, this also allows participants the ability to review issues when referencing in the future.

To summarise, quality is controlled by.

* Sprint initiation meeting
* Team acceptance testing
* Code walkthroughs
* KMS updates

# Controlling Quality

Controlling the quality of the project is accomplished through conducting sprint evaluations where the team collectively review the outcomes of the previous sprint in order to ascertain that the quality standards were met. In addition to this sprint evaluations also review what goals were achieved from the sprint and what was not completed throughout the sprint. Similarly, code quality reviews are performed with team members reviewing each other's code and checking if any possible adjustments or improvements can be made to greaten the overall code quality.

Testing data is performed by attempting all acceptance tests from each user story from the previous sprint. This will allow for sprint scheduling with knowledge of what acceptance test passed or failed. Refactoring future sprints take place last in terms of controlling the quality, wherein after all the evaluations, reviews, and testing are completed, the following sprint is planned around and adjusted to better suit the requirements are to ensure that all deliverables are completed in time at a high level of quality.

To summarise, quality is controlled by.

* Sprint evaluation outcomes
* Code quality reviews
* Testing data/acceptance testing
* Refactoring future sprints

In addition to these quality control methods there are several quality assurance strategies to assure high levels of product quality.

# Persona Based Quality Assurance

One of the primary quality assurance strategies implemented throughout the project was a personal based user acceptance and quality assurance testing and design process. Application requirements and features, use case and use processes, UI & UX and other general user focused needs were designed around several fictional personas modelled to represent the diverse range of students at CQU. This includes students of all age groups and education levels, students fresh out of high school, mature students looking for a career change or general education opportunities. Personas have different levels of technology literacy and comfortability as well as operating system and device preferences, this is designed to target the widest largest audience with the most varied experience and usage. Furthermore, cultural and linguistic considerations were taken into consideration to support the large number of international student's world view and comfortability with the application. Other considerations were of those who have special accessibility requirements or needs such as our persona Sarah who doesn’t allow her vision to impact her dreams.

# Development & Quality Project board Process

The project board process follows the path of initiating a user story to deploying on the server, many of the previous mentioned quality assurance tasks may be performed in this section, however it is important to note these are two different processes with different goals and end results.

* Backlog Initiation
* User Story – start
* Sub Task QA
* Internal Acceptance Testing
* Code Review
* Persona Acceptance & Unit Testing
* Pending Documentation – Team Approval
* Ready for Deployment - finish

The quality assurance process follows a strict plan in place to ensure that code and features implemented are put through rigorous testing but internally and in coordination with our personas. This is designed to ensure the projects work has faced the scrutiny of other members before being tested under user acceptance testing criteria. The first step is providing a user story to the backlog, that is reviewed by the team before being deemed ready to start, this includes detailed story, goals, assumptions, sub tasks, acceptance tests, business value and estimated effort and value.

User stories must be completed in full and meet the acceptance requirements of the team before being elidable for allocation. During sprints the user story is backed by the sub tasks to ensure the features being implemented are full and to the expected requirements.

Once the sub tasks and user story are complete internal acceptance testing and code review take place, these may be interchangeable as acceptance testing and code reviews both lead to refactoring of code and discovery of new issues or better implementation practices.

Once the above process is met with satisfaction the user stories are then put through the main user acceptance testing based on the user personas and their individualistic experiences and use requirements.

This is where the user story is set to pending documentation and team approval, all documentation related to this user story must be completed and released before being eligible for team approval. Once the team has signed off on the user story it is then declared ready for deployment from the development branch to the main to be pushed to the server via CI/CD.

# Risk Management Plan

## Risk Identification

There were several risks immediately identified during the initial planning phase of the development project. These risks being overall inexperience in project management, possible poor team member communication, poor teamwork, and poor team productivity. These were identified due to each member of the development team having little understanding about each member. This was due to it being each team member's first time working with one another. This initially led to a lack of understanding of each team member's skills and identifying where each team member could be most efficiently utilised.

After the initial team meeting and project analysis, the following risks to be identified were scope creep, flawed scheduling, lack of testing, and lack of formal reviews. These risks extend not only from the previously discussed risks but as well as the project development goal itself. Not having been able to meet and discuss the functional requirements of the project with the client, made determining the project scope and creating schedules difficult. With initial project demo constructions being done with a vague understanding of what the client wanted; these risks began to emerge.

To summarise the identified risks.

* Inexperience with project management.
* Poor team communication.
* Poor team member productivity.
* Poor teamwork.
* Lack of formal reviews.
* Lack of testing.
* Using new tools.
* Scope creep.
* Flawed scheduling.

# Risk Assessment

In terms of risk assessment, the risk that presented the greatest threat for possible project failure was poor team communication. This risk was determined by estimating the probability of the risk occurring along with the impact of the risk if it were to occur. Poor team communication was the greatest risk, leading to several problems if it were to occur such as poor productivity, and teamwork. Continuing this, a lack of proper communication would lead to a failure in determining project scope which in turn leads to scope creep and generally causing a myriad of additional problems.

In summary the greatest risk is.

* Poor team communication.

## Risk Mitigation

To avoid the impact of these risks, the development team has planned around and created possible solutions for each risk. The risks that were caused due to a lack of clear and consistent communication can be avoided by ensuring communication is prevalent through all stages of the project, this can be achieved by having members consistently communicate with each other using communication tools such as Discord.

Additionally, several other quality management strategies to be used to help mitigate risks are initiation requirements, utilising the Scrum framework, performing peer reviews, and sprint evaluations. The initiation requirements are a process in which the team reviews the project constraints and determines the overall scope of the project along with determining the technical requirements of the project. Scrum addresses the potential for poor project management with the Scrum framework allowing ample opportunity to clarify any uncertainties and helps mitigate the effects of scheduling flaws and inefficient work allocation.

Peer reviews allow the team to perform a type of formal testing and promote teamwork whilst providing a general higher level of quality work with each member reviewing each other’s work. Sprint evaluations are to be used to address any potential scope creep and eliminate any potential uncertainty of the current project status and sprint goals.

To summarise the mitigation methods.

* Communication tools.
* Initiation requirements.
* Peer reviews
* Sprint evaluations

## Lessons Learned

After completing the development project and navigating through all the identified risks, there were several lessons learned from the experience that each team member can take into future development projects. There were numerous benefits and lessons learned over the course of the project e.g., learning how to use new tools, working in a group, managing time. Regarding risk management itself, the major lessons learned as a team over the course of the development project would be that too many risks were taken on. The project requiring many new frameworks and the use of new development tools led to a slow start and a harsh learning curve. In conjunction with the project being each team member's first experience working with each other, the risk of possible project failure was high.

Another difficulty face during the project would be the lack of client input during the initial planning phase of the project. This led to the development not quite understanding the full scope of the project which caused scheduling and allocating work appropriately to be challenging. In terms of risk mitigation, however, this project exemplified how important communication is. Using communication tools with each member being in almost constant contact, many of the identified risks were mitigated and avoided.

To summarise the experience.

* Too much risk was taken on.
* Establish the scope of a project before development.
* Communication is the most vital key to success.