Software Quality Assurance Plan for

Blood Bank Management System

Prepared by:

Munir Abbasi 20K-0244

Raza Ali Abidi 20K-1061

Mustafa Zahid 20K-1045

Assignment 1

Table of Contents

1. Introduction

The software we for which the quality plan is being developed is the Blood Bank Management System that will be developed by our team. We hope to produce such an SQA plan that can ensure the maximum level of quality for our software. The idea for the software stemmed from the visit of Indus Hospital's blood donation drive in FAST-NUCES Karachi. We hope to produce such a system and make it highly usable and beneficial.

2. Group Members

The project is headed by Raza Ali Abidi. He put together this team with experts from the class. Raza will head the project by providing software quality engineering principles while heading the project. The backend record management will be carried out by Mustafa Zahid. He is a seasoned backend engineer whose interest in database engineering prompted his interest in the project. The software construction will be carried out by Munir Abbasi due to his experience in constructing similar software such as his dairy farm management system. [1]

3. Project Scope

We will help deliver a management system that can record the blood donations and then their depositions. We will not provide an integration with other software systems of the hospital but this will rather be a separate software. We will provide tracking services in case of any emergency such as blood poisoning. We will not provide corrective measures that is the task of the software user. [2]

4. Software Quality Goals

Our plan outlines how we will ensure that our software quality goals are met and also ensure which software quality goals will be fulfilled.

4.1. Code quality

1. The software construction team will make sure that the code quality is improved by reducing the complexity of the code. This will be achieved by:

- Making use of small methods. We will reuse code where it is possible and then create a smaller less complex method. This will reduce lines of code and also improve readability.
- Try to reduce the if/else conditional statements. Most of the time, we do not require an else statement in scenarios.

2. Code will be made reusable to improve the quality by:

- Reusing components of code.
- The developers must ensure that they introduce modularization and also maintain a high cohesion in the placement of the module. This also ensures that the code is understandable.
- The developer should use dynamic link libraries to improve reusability.

4.2. Requirement quality

1. The software quality engineering process must be such that the requirement is complete.

- This can be ensured by questioning the client about what they want and try copy it down as they say without addition. If things are still incomplete or require elaboration, the requirement engineer must question the client.

2. The requirements must also be made understandable.

- This can be achieved by dividing the SRS into sections so it is easy for the developer to construct the software as per the client's needs.

4.3. Design Quality

1. Design quality must be such that the appropriate design patterns must be used for the Blood bank management system.

- The Software Design Specification should include an appropriate design pattern which would be the Adapter method as interfaces will change for the users but the objects will be the same.

4.4. Quality control effectiveness

1. Quality control effectiveness can be achieved by making sure that the completion rate of the work is met.

- The expected and actual completion time must be close. This can be ensured by making accurate guesses for the total software completion.
- Domain experts can be used such as a colleague [3] who can provide accurate estimates.

2. The resource allocation for the workload must be kept balanced.

- As mentioned in the project proposal, we have divided the workload as per expertise. This can help increase QC effectiveness.

3. Testing effectiveness must also be ensured so that the errors can be found, and then the error can be corrected appropriately.

5. List of Software reviews

We had performed previews of the software was done of some blood banks like:

- Indus Hospital
- Liaquat National Hospital
- Hussaini Blood Bank
- AKU Hospital

We had and will also perform peer reviews, have meetings to discuss progress, and also have our instructors and teachers review our software for improvements regularly. This can ensure we have a smooth process for a smooth and well-designed end product.

We will also have formal technical reviews to:

1. Uncover errors.
2. Make the project manageable.
3. Ensure standards are met.

We will also have client reviews to get their acceptance and to ensure we build the right product.

SQA Audits

We will also have SQA audits where members will report their progress. Any problems faced will be discussed. Any changes that affect the project will be discussed.

6. Software Testing

We will test our software. Some basic testing has been performed on the software.

- Login test is done whether the wrong input does not allow opening of the software.

- Data is entered correctly we used to try and catch blocks so if the exception occurs our system generates a message for the user. This will also be used to make sure that data integrity is preserved.

7. Validation and Verification activities

Verification would be performed throughout the development process to ensure that everything is being implemented according to the rules and regulations set up by the organization. After each iteration, it will be verified that the code written has exception handling and followed the standards.

After each iteration, we will make sure that the components/ features built are according to the requirements of the stakeholders/ end users. A validation process will take place after each iteration and after the full integration of all the features in a single software to ensure that the end product will be built following the requirement of the stakeholders. If some changes are required after the integration, the software will go back into the development process. For validation, we used Requirements Traceability Matrix while making the SRS of our software.

8. Software Testing Strategies

1. One testing strategy that will be used is the use of case studies. We will look at other blood bank management software or inventory management systems to elicit the shortfalls and eliminate the problem in our software. This improvement will set our software apart from other existing ones.
2. Another strategy we will use is that we will use a domain expert who we have mentioned [4] who can help us understand the working dynamics of the software users and develop code keeping in mind their needs to improve usability.

9. Software Testing Tools

The software testing tool that will be used by our team is going to be Squish and for regression testing, we plan to use the RCP testing tool.

10. Software Testing Techniques

We will follow the two major software testing techniques.

Static Testing

1. We will make use of static testing at the time of the code development. Here we will find defects in the code before execution.

- This will be achieved by utilizing peer reviews. This can make static testing easy as an accomplice with the developer can find errors quickly as compared to an individual.

2. The software testing technique of inspection can also be performed. This also helps achieve static testing.

Dynamic Testing

1. We will also make use of dynamic testing that can help achieve the code output we need.

- We will make use of black-box testing so that we can achieve functional testing. We can feed inputs and then compare the software result only based on the output. If the actual output matches the expected output, we will achieve this testing successfully.
- We can utilize white box testing during our unit testing to make sure that the code is working as expected. Then we can integrate that part of the code with the rest of the software.

11. Software Testing Methodologies

1. We will utilize multiple software testing methods. These methods will be applied in different environments, and serve different functions.
2. We will test our software during the whole development process rather than only focusing on the end of the software lifecycle. This can help elicit errors in the early stages to prevent a fault or failure later.
3. The SRS will include testable software requirements. This will make sure we can achieve what is needed and quantify it. We can make use of test-driven development to ensure specific program sections.
4. We will also evaluate the code for non-functional features such as security, reliability, and maintainability. This can be achieved by the developer easily.

12 Problem Reporting and Corrective Action

While developing this software we faced many problems in it and as we were using the Iterative model. So as soon as an error occurred in our project, we fixed it.

Problems that occurred during the development phase are:

» Not getting localhost code easily to connect SQL developer application with papyrus for the initial stage of development then we used another application to find the localhost code of our SQL developer

» to stop the system from receiving inaccurate data. An alphanumeric address field that permits spaces to be input before any digits or characters in the address is an example of this. As a result, it's possible that the intended address won't turn up when searches or sorting are done on the address field.

» Our code had many errors which were the main problem while developing this system but we debug the code and managed to remove the errors.

» It was difficult to design the user interface for this software as this software should be easily understood and must be friendly in use so we used design rules and Norman's Principles.

To solve the problems, we followed a reporting mechanism. Here as soon as a problem occurred, it was tried to be resolved using platforms such as Stackoverflow or Reddit. If the problem persisted, the team member informed the others. Then the corrective action was taken together. Since we had already distributed the responsibilities of work [5], it made it easy to resolve problems.

13. Document References

We have used 4 references from our project proposal.

1, and 2 are already discussed there in full detail and have been used.

3, 4 is the domain expert we have from the ICICI group who has an affiliation with Munir Abbasi.

5 is already mentioned as well. Roles were defined in the proposal.

Munir Abbasi=> Software construction

Raza Abidi=> Software quality assurance

Mustafa Zahid=> Database Management

The template used was from here and as outlined by course instructor.