



4.2P GR7

101131147 | MONIQUE KUHN  
101111372 | JAKE SCOTT  
102259710 | TIEN PHU NGO  
101100655 | LACHLAN BURNS  
102095118 | JAYDEN MCQUEEN  
102079989 | DAVID STARE

Naurin Afrin | Friday 12.30

SWE30010 Development Project 2: Design,  
planning and Management

### **Project Proposal: PHP-SRePS**

In this project quality is defined by the software's Functionality, Reliability, Usability, Efficiency, Maintainability and Portability.

In regard to a Scrum approach, measuring quality would involve a high level review of the sprints followed by a low level review of the code which would address errors in the code, readability of the code, integrated testing and performance testing.

From the results of low-level testing we would classify each point and feature of the software to.

- No action
- Refer for repair
- Reconsider overall design

### **Definition of 'Done'**

The product will be considered 'done' when the software is in a state where it can be deployed while being considered 'high quality'. The quality will be determined through test cases where the program will be given specific initial conditions and compute outputs accordingly. These outputs from the program will be compared to known results of the initial conditions to check the program is functioning correctly. Ideally the program should pass all tests to ensure complete functionality. However, a more practical approach would be to ensure that the quantity of failed cases is less than or equal to 5% of the total test cases created.

### **Quality Management**

The quality of the software is determined through the product quality model: ISO/IEC 25010. This model outlines 8 key characteristics of a product that need consideration during the development process which include: functionality suitability (fs), performance efficiency (pe), compatibility (c), usability (u), reliability (r), security (s), maintainability (m) and portability (p). By developing software with these characteristics and accommodating the needs of the client (outlined in the scope in 2.2P), the software can be deemed 'high quality'.

Below are some points of a quality control checklist that relate to the client's needs and the above characteristics:

No.	MSCW*	Characteristics of the ISO/ICE 25010 Model	Requirements of the Software	Methods to ensure the software's functions are reliable
[1]	<b>M</b>	<b>Fs, u</b>	Generate monthly sales reports which can be in CSV format	Test cases which can verify the conversion is in the right format
[2]	<b>M</b>	<b>Fs, u</b>	User-friendly graphical interface	Present the interface to people outside of the development team to ensure it is easy to follow. Adoption should not exceed 15 minutes.
[3]	<b>M</b>	<b>Fs, m</b>	Record, edit, and view sales data for pharmaceutical products	Develop tests with edge cases that have known results, to ensure the same result is obtained through the program
[4]	<b>M</b>	<b>Fs, u</b>	Generate alerts for low stock	Tests where stock is set to be intentionally low in order to determine if an appropriate alert is shown
[5]	<b>M</b>	<b>Fs, u, m</b>	The program can perform predefined mathematical operation on existing data to predict monthly sales for items based on previous sales	Develop tests with normal and edge cases with a previously known result to ensure the program predicts accurately
[6]	<b>M</b>	<b>Fs, pe, u, r, s, m, p</b>	Fully workable application and database system	Test the program on different machines (of the same operating system) to a 90% success rate using unit/integration/system tests
[7]	<b>M</b>	<b>Fs, u, m, r</b>	Display Items that are in demand	Develop test cases where items are set intentionally high, in order to determine if the appropriate items are displayed
[8]	<b>M</b>	<b>Fs, u, m, r</b>	Display stock numbers of all items in store	Test cases where the display command is executed, and all appropriate items are displayed
[9]	<b>W</b>	<b>Pe, m, r, s</b>	Online/Cloud database	Test cases to ensure the data is stored without corruption and effectively
[10]	<b>C</b>	<b>Fs, r, s</b>	Login system with different permissions	Test cases where users with varying permissions attempt to access unauthorized data and are rejected
[11]	<b>W</b>	<b>u, m, r, c, p</b>	Mobile access	Test cases where the program is run on mobile devices to ensure it has the same effectiveness and user-friendliness as its Windows version
[12]	<b>S</b>	<b>Pe, m, r, s</b>	Backups/redundant storage	Test cases to ensure the backup storage contains the same data as the primary storage (without corruption)
[13]	<b>C</b>	<b>Fs, s</b>	Data encryption	Data cannot be accessed by undesired personnel
[14]	<b>M</b>	<b>Fs</b>	Local storage	The program can take input and store it in local storage
[15]	<b>S</b>	<b>Pe</b>	Low resource demand	RAM used while running: <100MB & Executable file is less than 50MB in size
[16]	<b>S</b>	<b>r</b>	Exception Handling	Display appropriate error messages when required
[17]	<b>S</b>	<b>p</b>	Portable system	The program contains 1 executable file and is supposed to run on its own without any third-party library installed

No.	MSCW*	Characteristics of the ISO/ICE 25010 Model	Requirements of the Software	Methods to ensure the software's functions are reliable
[18]	<b>M</b>	<b>Fs</b>	Report History	Previous records can be viewed in manageable quantities, and as far back as the history goes.
[19]	<b>C</b>	<b>u</b>	Data Visualisation	Graphs generated are clear and easy to understand
[20]	<b>S</b>	<b>m, r</b>	Full refactoring to ensure optimized code.	Upon completion of refactoring, the team will check over and approve the optimisations.
[21]	<b>C</b>	<b>m</b>	Documentation	A full, detailed operations manual must be developed and appropriate commenting standards should be used
[22]	<b>S</b>	<b>Fs, u</b>	Startup	Product must be able to run on a desktop computer, with an executable file to launch the program.
[23]	<b>C</b>	<b>u</b>	Training session	Staff should be able to work the application at the conclusion of the training session

\*MoSCoW represents MUST, SHOULD, COULD, and WON'T

## **Reflection**

### **101111372 | JAKE SCOTT**

After completing our individual 'Definitions of Done', we compiled them into a more professional looking document which now includes: an introduction, definitions, and the factors in a more readable table format. The group came up with a couple of extra items such as the program generating alerts for low stocks, and sorted the items by how necessary/achievable they are.

### **102259710 | TIEN PHU NGO**

All team members met in the team meeting, went through all of the definition of done team members have come up with, divide them into 4 different categories and sorted them by necessity. In the end, I am happy with the team's list of definitions of done.

### **101100655 | LACHLAN BURNS**

During the tasks meet-up time we were successfully able to go through everyone's list of their definition of what is involved to complete the project, how this could be measured, and then merged them all into one document. From this we were able to use the Moscow approach to define each point to fall under being a must, should, could or won't do. Following this, we then further defined the points through the ISO/IEC 25010 model to their corresponding characteristics; functionality suitability (fs), performance efficiency (pe), compatibility (c), usability (u), reliability (r), security (s), maintainability (m) and portability (p). Overall, I was happy with how the team came together to complete this task.

### **102095118 | JAYDEN MCQUEEN**

After completing my individual "Definitions of Done" task, I was unable to attend this week's meeting. This meeting is used every week to facilitate the collaboration of our individual work into a group document. Upon looking at the complete document, I am very happy with the end result. It looks very professional, and I agree on the definitions of done that have been defined by our group

### **101131147 | MONIQUE KUHN**

The intermediate meeting between tutorials have been extremely helpful for the team. It has substantially increased our workflow quality and efficiency. This week, the team took the time to work through and discuss all the generated qualities, finally discussing the appropriate prioritization. Although I suspect the team may be taking on some hefty goals, our collaborative team nature should assist in ensuring we achieve as much as possible.

### **102079989 | DAVID STARE**

The team had individually created their own definitions of 'done' by listing objectives needed to be completed and practical goals for the number of unit test cases passed. The team then compiled these lists into one ensuring nothing important wasn't left out. The team agreed that at least 95% of the unit test cases would be passed before the program would be ready to be released. The team was happy with and agreed upon the final checklist.