

1809ICT Assignment 2

Project Review, Testing Plan, Metrics, & Software Process



Student Name

Leon **Andov**
Joshua **Baker**
Scott **McMullan**

Student ID

s5088455
s5057501
s2916255

Date & Time of Laboratory

Friday, 1200
Friday, 1400
Thursday, 1600

Table of Contents

1 – Project Review & Reversion	3
Problems	3
Project Timeframe	3
Personnel	3
Equipment Costs	3
Improvements.....	3
Project Timeframe	3
Personnel	4
Equipment Costs	4
2 – Project Testing Plan & Test Cases	4
Test Plan.....	4
Test Cases.....	5
3 – Project Metrics	10
4 – Software Process.....	11
PM.O2	11
Project Assessment and Control Process:.....	11
Measurement Process	11
Software Acceptance Support Process:	11
Achieving PM.O2.....	11
PM.2 - Project plan execution.....	11
PM.3 - Project assessment and control.	11
PM.4 - Project Closure	11
PM Products, Processes, & Activities.....	11
References	12

1 – Project Review & Reversion

While the initial project was successful in meeting the standards of the marking criteria, many issues in the cost benefit analysis were discovered upon self-reflection. The three issues are as follows:

Problems

Project Timeframe

- Due to poor time-allocation, the initial expected development period of four months was not realistic. Many factors were oversighted in the planning stage, these included:
 - The working pace of the programmers; they should have more time to complete their tasks.
 - Code validation was not taken into consideration.
 - A testing plan and test cases were also not taken into consideration.

Personnel

- Team member salary was not allocated well, therefore the budget is not estimated thoroughly. This resulted in the following cost miscalculations:
 - An employee dedicated to the 3rd party database integration was not included.
 - Full-time wages;
 - The two maintenance employees would not realistically be paid a salary of \$80,000 and was miscalculated. A lack of salary research overall created this result.

Equipment Costs

- Individual hardware and software costs were partially unnecessary. The assumption for developmental hardware requirement is not essential because the developers have their own work computers.

Improvements

To address the problems identified above, the following improvements will be implemented:

Project Timeframe

- To ensure that the hired programmers can complete their work, instead of having a 4-month timeframe there will now be a 5-month timeframe.
 - The average application development process is 4 months. However, a timeframe of 5 months allows the hired programmers to produce higher quality code with more spaced out development stage deadlines.
- Increasing the project timeframe to 5 months will mean that there will be time for good software testing to take place.
 - Unit, component, integration, validation, and system testing.
 - Each of these areas cover the entire testing scope of the application.
 - Testing plan, test suite and test cases.
 - Each test case will cover a subset of function(s) within a specific area of the application that will collectively produce a test suite.
 - From a security standpoint, 3rd party database access will mean that user input sanitisation must be thoroughly analysed.

Personnel

- More research and planning has taken place in attempt to reduce the human labour costs;
 - The full-time maintenance employees will be moved from a full-time position to part time (from \$80,000 to \$40,000) which will result in a significant reduction of the budget.
 - An employee will be hired to manage and complete work regarding the 3rd party database API. There is plenty of room to hire another employee considering the cost reduction above.

Equipment Costs

- 3 computers were included in the cost benefit analysis with miscellaneous software licenses (\$4290 in total):
 - It can be assumed that programmers seeking jobs in 2017 have their own work computers, therefore the computers included in the budget can be cut out.
 - The software licenses may still be necessary, most programmers use free text editors to write their code but the application will be written in a premium software that will support the features that will be implemented on the application.

2 – Project Testing Plan & Test Cases

Test Plan

Software testing will take place before the release and after the software has been developed; end users and independent testers will be responsible for the testing. End users will administer the tests while the independent testers will perform the testing. In addition to acceptance testing, system level tests, unit tests and integration tests will be performed by the independent testers. It is projected that the total cost of testing will equal \$3000, this covers the human resource cost, office hire and testing hardware including computer hire. In Health RPG the user will be able to swipe the screen left, right and upwards to access the various aspects of the app, this means graphical user interface testing is essential as it will ensure functionality is not overlooked. Stress testing will be used for the database to ensure that all food and vitamin types are accessible with an emphasis on popular products such as Vitamin C. Users will have no reason to use the app if the food and vitamins they consume do not exist virtually for tracking therefore the database is a mission critical component of the app and acceptance testing will be arranged. All unit testing to take place will follow the black box testing technique as it is best applicable for system and acceptance testing, equivalence partitioning and boundary value analysis. Using this method, the testers are not required to know programming languages and can focus on interface error finding, errors within the data structures and database access.

Test Cases

PROJECT:		Health RPG		
MODULE:		Create account		
REQUIREMENT:		Analytics R1		
TEST CASE ID:		Test 1		
TEST OBJECTIVE:		To check whether the entered health analytics are valid or not		
TEST DATE and TIME				
<u>Step No</u>	<u>Steps</u>	<u>Data</u>	<u>Expected Results</u>	<u>Actual Results</u>
1	Enter Medicare ID and press CONTINUE Button	ID = 4608 51230 22	Should Display Warning Message Box "Please enter a valid 10-digit ID"	
2	Enter Medicare ID and press CONTINUE Button	ID = dumbledore	Should Display Warning Message Box "Please enter a valid 10-digit ID"	
3	Enter Medicare ID and press CONTINUE Button	ID = 4608 51240 4	Should navigate to age window	

PROJECT:		Health RPG		
MODULE:		Create account		
REQUIREMENT:		Analytics R1		
TEST CASE ID:		Test 2		
TEST OBJECTIVE:		To check whether the entered health analytics are valid or not		
TEST DATE and TIME				
<u>Step No</u>	<u>Steps</u>	<u>Data</u>	<u>Expected Results</u>	<u>Actual Results</u>
1	Enter age and press CONTINUE Button	Age = Banana	Should Display Warning Message Box "Please enter a valid age"	
2	Enter age and press CONTINUE Button	Age = -4	Should Display Warning Message Box "Please enter a valid age"	
3	Enter age and press CONTINUE Button	Age = 18	Should navigate to height window	

PROJECT:		Health RPG		
MODULE:		Create account		
REQUIREMENT:		Analytics R1		
TEST CASE ID:		Test 3		
TEST OBJECTIVE:		To check whether the entered health analytics are valid or not		
TEST DATE and TIME				
<u>Step No</u>	<u>Steps</u>	<u>Data</u>	<u>Expected Results</u>	<u>Actual Results</u>
1	Enter height in CM and press CONTINUE Button	Height = 450	Should Display Warning Message Box "Please enter a valid height in CM"	
2	Enter height in CM and press CONTINUE Button	Height = 1.92	Should Display Warning Message Box "Please enter a valid height in CM"	
3	Enter height in CM and press CONTINUE Button	Height = 1920	Should Display Warning Message Box "Please enter a valid height in CM"	
4	Enter height in CM and press CONTINUE Button	Height = 192	Should navigate to body weight window	

PROJECT:		Health RPG		
MODULE:		Create account		
REQUIREMENT:		Analytics R1		
TEST CASE ID:		Test 4		
TEST OBJECTIVE:		To check whether the entered health analytics are valid or not		
TEST DATE and TIME				
<u>Step No</u>	<u>Steps</u>	<u>Data</u>	<u>Expected Results</u>	<u>Actual Results</u>
1	Enter weight in KG and press CONTINUE Button	Weight = 3	Should Display Warning Message Box "Please enter a valid weight in KG"	
2	Enter weight in KG and press CONTINUE Button	Weight = Five hundred kilograms	Should Display Warning Message Box "Please enter a valid weight in KG"	
2	Enter weight in KG and press CONTINUE Button	Weight = 641	Should Display Warning Message Box "Please enter a valid weight in KG"	
3	Enter weight in KG and press CONTINUE Button	Weight = 67	Should navigate to dietary plan window	

PROJECT:		Health RPG		
MODULE:		Create diet plan		
REQUIREMENT:		Dietary Plan R2		
TEST CASE ID:		Diet 1		
TEST OBJECTIVE:		To check whether the entered vitamin / food is in database		
TEST DATE and TIME				
Step No	Steps	Data	Expected Results	Actual Results
1	Enter vitamin name and press ADD Button	Vitamin = B17	Should Display Warning Message Box "Vitamin B17 is Illegal, please enter a legal vitamin"	
2	Enter vitamin name and press ADD Button	Vitamin = Smith Street	Should Display Warning Message Box "Vitamin Smith Street is not found, please enter a legal vitamin"	
3	Enter vitamin name and press ADD Button	Vitamin = C	Should navigate to QUANTITY window	

PROJECT:		Health RPG		
MODULE:		Create diet plan		
REQUIREMENT:		Dietary Plan R2		
TEST CASE ID:		Diet 2		
TEST OBJECTIVE:		To check whether the entered vitamin / food is in database		
TEST DATE and TIME				
Step No	Steps	Data	Expected Results	Actual Results
1	Enter food name and press ADD Button	Food = 51	Should Display Warning Message Box "Please enter a valid food"	
2	Enter food name and press ADD Button	Food = Fogu	Should Display Warning Message Box "Fogu food is illegal, please enter a legal food"	
3	Enter food name and press ADD Button	Food = Broccoli	Should navigate to QUANTITY window	

PROJECT:		Health RPG		
MODULE:		Preview		
REQUIREMENT:		Preview Plan Ability R3		
TEST CASE ID:		Prev 1		
TEST OBJECTIVE:		To check if user can preview their own diet plan		
TEST DATE and TIME				
<u>Step No</u>	<u>Steps</u>	<u>Data</u>	<u>Expected Results</u>	<u>Actual Results</u>
1	Press PREVIEW Button (diet plan existing)		Should Display Warning Message Box "Please create a diet plan first"	
2	Press PREVIEW Button (diet plan not existing)		Should navigate to DIET PLAN window	

PROJECT:		Health RPG		
MODULE:		Preview		
REQUIREMENT:		Preview Dietary Progress R4		
TEST CASE ID:		Prev 2		
TEST OBJECTIVE:		To check if user can swipe screen right and examine dietary progress		
TEST DATE and TIME				
<u>Step No</u>	<u>Steps</u>	<u>Data</u>	<u>Expected Results</u>	<u>Actual Results</u>
1	Swipe screen LEFT from main window		Should navigate to ADD window	
2	Swipe screen UP from main window		Should navigate to SOCIAL window	
3	Swipe screen DOWN from main window		Window should stay the same	
4	Swipe screen RIGHT from main window (diet plan not existing)		Should Display Warning Message Box "Please create a diet plan first"	
5	Swipe screen RIGHT from main window (diet plan existing)		Should navigate to DIET PROGRESS window	

PROJECT:		Health RPG		
MODULE:		Create Dietary Task Reminder		
REQUIREMENT:		Dietary Task Reminder R5		
TEST CASE ID:		Diet 3		
TEST OBJECTIVE:		To check whether the user can create dietary tasks as text and set reminders		
TEST DATE and TIME				
<u>Step No</u>	<u>Steps</u>	<u>Data</u>	<u>Expected Results</u>	<u>Actual Results</u>
1	Enter dietary task and press ADD Button (diet plan not existing)	Task = Vitamin C 500mg intake	Should Display Warning Message Box "Please create a diet plan first"	
2	Enter dietary task and press ADD Button (diet plan existing)	Task = Vitamin C 500mg intake	Should navigate to SET REMINDER window	
3	Select reminder date and time (dietary task not existing)	Date and time = 29/4/17 08:30	Should Display Warning Message Box "Please create a dietary task first"	
4	Select reminder date and time (dietary task existing) + (user has not agreed to allow the app to make reminders)	Date and time = 29/4/17 08:30	Should Display Warning Message Box "Please accept the user agreement"	
5	Select reminder date and time (dietary task existing) + (user has agreed to allow the app to make reminders)	Date and time = 29/4/17 08:30	Should navigate to main window	

3 – Project Metrics

The two project goals that have been formulated are:

1. Health RPG will reach 100,000 concurrent users within a year of its release.
2. Health RPG will have a 5-star rating on the google and apple store.

It is endeavoured that the application will reach 100,000 concurrent users within a year of its release. Many users would mean that the application is a huge success. This success can be measured by the amount of user activity in a single 24-hour period (up to 12 months). The application is designed to be lightweight and suitable for large amounts of traffic. If there was an insignificant amount of traffic then the servers backing the application would be wasteful and a downgrade would have to be taken into consideration. The application will have leader boards on major social media platforms which users can choose to compete on. Competitive users are more motivated to compete for a top position on the leader board by increasing their activity on the application. Hence it can support active returning users. An active community around the app would create more exposure and encourage new users to join. It is recognised that the user base has a mix of competitive and non-competitive users. While there is a leader board, the main purpose of the application is to promote a better lifestyle for a single user with the help of this app. Health RPG will set a new standard for health apps and encourage high activity with its features, hence it should produce many concurrent users.

A 5-star rating on both the google and apple store would indicate that the application has performed well and has great customer satisfaction. The rating system on each app store is an easy way for customers to define which applications are low or high quality. 4-5 star rated applications are always promoted more than low quality applications, therefore producing a high-quality application is necessary to achieve a 5-star rating. Unique and innovative applications perform the best on the market and tend to be the key factor in producing an application that reaches a 5-star rating. Health RPG provides an efficient way for users to produce and send diet plans, users will have access to a large 3rd party database to search for the nutritional value of most available products and it has a RPG level system driven by user achievements. This sets Health RPG apart from a standard health application because these features together have not been implemented into a single application to this date. Users are more likely to post good reviews about the application for this reason. Errors within apps are a common way to lose ratings, therefore it is crucial that the testing carried out in the testing stage of development irons out at least 99% of all the bugs found. New applications on the market that are attractive to a user usually have some leeway to minor errors if they are fixed quickly. This is something that the app should not rely on because good testing and debugging will ensure that the application will meet the standards of the stakeholders.

4 – Software Process

All information in this section is gathered from (ISO/IEC, 2012).

PM.O2

Progress of the project is monitored against the Project Plan and recorded in the Progress Status Record. Closure of the project is performed to get the Customer acceptance documented in the Acceptance Record.

Project Assessment and Control Process:

- Progress of the project is monitored and reported.
- Project objectives are achieved and recorded.

Measurement Process

- The required data are collected, stored, analysed, and the results interpreted.

Software Acceptance Support Process:

- The product is completed and delivered to the acquirer.

Achieving PM.O2

To achieve PM.O2 completely, one must perform the following Project Management Process activities:

PM.2 - Project plan execution.

- The Project Plan Execution activity implements the documented plan on the project. The activity provides:
 - Monitoring the project against the project plan.
 - Status of the project plan execution.
 - Change request accepted by the customer.
 - Reviews and agreements with the customer.

PM.3 - Project assessment and control.

- The Project Assessment and Control activity evaluates the performance of the plan. The activity provides:
 - Evaluation of actual plan performance and progress against targets.
 - Track change requests.
 - Documented problem, corrective action defined, and [stuck on] to closure.

PM.4 - Project Closure

- This activity provides the project's documentation and products in accordance with contract requirements. The activity provides:
 - Support of Customer product acceptance.
 - Completion of the project and sign of the Acceptance Record.
 - Summary and updated project repository for project closure.

PM Products, Processes, & Activities

<u>Work Products</u>	<u>Process Activities</u>	<u>Process Objectives.</u>
Project Plan.	Project Planning, Project Plan Execution, Project Assessment & Control, and Project Closure.	PM.O1, PM.O2, PM.O3, PM.O4, PM.O5, PM.O6, PM.O7.
Change Request.	Project Plan Execution and Project Assessment & Control.	PM.O2, PM.O3, PM.O4, PM.O5, PM.O7.
Acceptance Record.	Project Closure.	PM.O2.
Meeting Record.	Project Plan Execution.	PM.O2, PM.O3, PM.O4, PM.O5, PM.O7.
Progress Status Record.	Project Plan Execution and Project Assessment & Control.	PM.O2, PM.O3, PM.O4, PM.O5, PM.O7.
Statement of Work.	Project Planning.	PM.O1, PM.O5, PM.O6, PM.O7.
Project Repository.	Project Planning and Project Plan Execution.	PM.O1, PM.O5, PM.O6, PM.O7.
Software Configuration.	Project Closure.	PM.O2.

References

ISO/IEC. (2012, 09 15). *Freely Available Standards*. Retrieved from International Organization for Standardization: <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>