

M24: Assignment #1

Due on Monday, February 19, 2015

Ben Mora

SwanTech

Contents

Team Organisation and Collaboration	3
Team Name	3
Team Organisation	3
Team Collaboration	3
Team Role : Customer Interface Manager	4
Team Role : Design Manager	4
Team Role : Implementation Manager	4
Team Role : Test Manager	5
Team Role : Planning and Quality Manager	6
PQM - Team Management	6
PQM - Version Control and Code Inspection	6
PQM - Documentation	6
PQM - Code Quality	6
Schedule, Format and Quality	7
Initial Schedule of Work	7
Environment	8

Team Organisation and Collaboration

Team Name

The team agreed unanimously to **SwanTech** as our team / company name.

We will use Java package name **com.swantech** in all Java classes.

Team Organisation

The M24 module 2015 has only 4 people attending, but 5 roles are identified in the lecture notes, so we have to split one role. The roles assigned are:

Customer Interface Manager	Simon Hewitt
Design Manager	David Tacey
Implementation Manager	Ifetayo Agunbiade
Test Manager	Mohamad Khaleqi
Planning and Quality Manager (PQM)	<i>split between team members:</i>
PQM - Team management	David Tacey
PQM - Version Control and Code Inspections	Simon Hewitt
PQM - Documentation	Ifetayo Agunbiade
PQM - Coded Quality	Mohamad Khaleqi

The team will meet formally each Monday at 15:00, in the Computer Science lab at Faraday 206, and will meet informally to check progress after the Wednesday lectures.

Ifetayo is investigating team time management tools that support work breakdown and delivery timelines (WBS and Gantt chart capabilities are needed). This will be an online tool allowing all team members to share and update progress.

Team Collaboration

The team will collaborate informally through a Facebook group, which has now been set up. Formal documents that are part of the development and delivery will be maintained in GitHub, chosen for its strong team collaboration capabilities, version control and multi-platform support.

Team Role : Customer Interface Manager

Simon Hewitt will be the Customer Interface Manager, as defined in the lecture slides "Team Roles and Group Work" From the lecture notes, the Customer Interface Manager is responsible for:

1. The team's relationship with its customers
2. Resolving ambiguities in requirements specifications

To do this I propose the following actions and deliverables:

- Schedule regular meetings with the customers:- Ben Mora and Bob Laramee, to be agreed with them
- Create populate and manage a user requirements document
- Ensure non-functional requirements meet customer expectations
- Create and manage the assumptions and issues list in the project control structure
- Ensure the team has understanding of the requirements through formal requirements specification reviews
- And ensure team understands requirements by informal discussions at team meetings
- Review the designs to ensure they meet the requirements
- Review the test plans and test cases to ensure user requirements are tested and validated
- Responsible for managing each stage sign-off, ensuring the team deliverables meet the requirements and the customer accepts these deliverables as fit for purpose

As we will be following a RAD approach rather than a traditional waterfall, the requirements document will not be signed off and frozen, but will be the guide for each RAD development cycle. The requirements specification will be updated as necessary from feedback from RAD delivery and review. We believe that allowing the customer to see, use and feedback on developing work ensures that we are on track to deliver what is really needed rather than what was written down (with the possible ambiguities that can arise from the complex process of documenting requirements). Furthermore, this RAD approach enables the development team to propose improvements and design alternatives that may deliver a better product.

We will investigate requirement documentation alternatives over the next 14 days. This can range from an Excel spreadsheet to high cost, high complexity commercial offerings designed for huge teams and multi year projects. We will be looking for small scale, preferably open source solutions.

Team Role : Design Manager

Team Role : Implementation Manager

Team Role : Test Manager

Test manager role will be *Mohamad Khaleqi* responsibility. The test manager roles is a task with responsibility for the test effort's¹ success. This role involves codes quality and codes testing. For test manager there are some roles:

- understanding the testing by analyzing the project requirements.
- Organize the testing meeting.
- Develop the test plan.
- Arrange the software and hardware requirement for the test.
- Document all testing result and prepare a report.
- Communicate with team's members.

In order to do these, I should:

- Prepare Unit testing e.g. JUnit²
- Integration testing³There are two types of testing:
 - Button-up integration testing.
begin with Unit testing and then moving to higher level combinations of units called modules or builds.
 - Top-Down integration testing.
In this test, first higher level modules are tested and then low level module.

In developing software process, usually Button-up integration test happens first and then Top-Down integration test. [?]

- Software testing (Testing and debugging)
- Black Box and white Box
- Alpha and Beta testing
- Regression testing
- Ensure the team has understanding of the testing methods and also terms in testing unit.

¹Testing is the process of evaluating a system and the components to check if they are working correctly or not. Basically, it means to check the system to find errors, gaps, bugs and missing requirement or on the other words testing is a process of analysing a software to detect the difference between existing and required conditions.

²Testing will not catch every error in the program, because it cannot evaluate every execution path in any but the most trivial programs. The same is true for unit testing. Additionally, unit testing by definition only tests the functionality of the units themselves. Therefore, it will not catch integration errors or broader system-level errors (such as functions performed across multiple units, or non-functional test areas such as performance). Unit testing should be done in conjunction with other software testing activities, as they can only show the presence or absence of particular errors; they cannot prove a complete absence of errors. In order to guarantee correct behavior for every execution path and every possible input, and ensure the absence of errors, other techniques are required, namely the application of formal methods to proving that a software component has no unexpected behavior.

³In some books it is called as Component testing.

Team Role : Planning and Quality Manager

PQM - Team Management

PQM - Version Control and Code Inspection

This part role will be handled by Simon Hewitt. Version Control is managed by GitHub, so I will ensure the team understand GitHub and how to use it, that they properly use check-out and check-in. I will have control over final deliverable versions, and managing branch merging. Code inspection will be peer review across the group. I will investigate mark up tools. The MicroSoft Word document review tools are a powerful tool for team review, I will seek something similar for plain-text code files.

PQM - Documentation

PQM - Code Quality

This part will be Mohamad Khaleqi responsibility. In order to check the code quality in Java, First I will use Eclipse IDE because of automatic code indentation and auto formatting. Then I will make sure the code is following Bob's Concise Coding Conventions and also Doxygen. In addition to have better and tidy code, I will use CodePro Analytix ⁴ and PMD⁵ which they are both Eclipse plugin. They are both perfectly fine with any problem in the code and both are compatible with Java code. For code quality I will use LTFCE definition. The code over its lifetime will be read many more times than written. So, good code is:

- **Legible:** Only the code(Not comment) should clearly state the indent and reader should easily make sense of the code.
- **Testable:** The code will be in the way that unit tester can do its job.
- **Felxible:** Dependencies, both on other code in the code base and arbitrary implementation choices, should be minimized.
- **Complaint:** The code itself should be correct and meet the system requirements.
- **Economical:** The code should be responsible in terms of using system resources e.g. memory and CPU.

⁴CodePro Analytix is the premier Java software testing tool for Eclipse developers who are concerned about improving software quality and reducing developments costs and schedules. More info: <https://developers.google.com/java-dev-tools/codepro/doc/>

⁵PMD is a source code analyzer. It finds common programming flaws like unused variables, empty catch blocks, unnecessary object creation, and so forth. More info: <http://pmd.sourceforge.net/>

Schedule, Format and Quality

After searching for suitable document templates, we believe the attached document from the Swiss Federal Institute of Technology, Zurich, provides an excellent template. It is in Microsoft Word format, so we will reformat into LaTeX, taking the opportunity to omit sections that are not relevant and ensure it fits our needs. This answers the question on format, and contains the necessary guidance on quality management as well.

6

Initial Schedule of Work

When we receive the project requirements on 18/2/15, the initial actions are:

- Arrange a team meeting as soon as possible to review the document
- Set up and populate an open issues register, add all open questions to the register (Simon)
- Initial decomposition into work domains, arranged by team strengths as far as possible
- Create a first draft WBS ⁷ structure and assign names and target dates to each work package.
- Customer Interface Manager (SH) to arrange meeting(s) with Ben Mora to resolve all open questions.
- Test Manager (MK) to create outline test plan, showing what elements will be tested by which techniques, and setting up any necessary infrastructure.
- Design Manager (DT) to create high level design
- Implementation Manager (IA) to create high level implementation plan, and take ownership and manage the WBS schedule.

All of these actions to be completed by Monday 2 March 2015.

⁶se.inf.ethz.ch/old/.../Project Plan wo QA, Transition.doc

⁷Work Breakdown Structure, a simple way to manage deliverables in a project (http://en.wikipedia.org/wiki/Work_breakdown_structure)

Environment

It is assumed that the project will be developed in Java.

The team members use different desktop OS including Linux, OS X and Windows, so tools must support each of these. We have agreements or working assumptions or proposals for all identified software components:

Language	Java	Assumption for the project
IDE	Eclipse or NetBeans	Individual choice
Desktop	OS X, Windows, Linux	Individual Choice
JVM	TBC	Awaiting input from Ifetayo
Source and version control	GitHub	Agreed
Desktop source and version control	none specified, default is Git command line	Individual choice
Documentation	LaTeX	works well with GIT and is a academic standard
Informal Collaboration	Facebook	De facto standard
Testing	JUnit	The most widely used Java testing framework, simple to adopt
Test runner	To be decided	Nightly build and test sequence is desirable
Time Management	TBC (Ifetayo)	Gantt chart, WBS