QA 10 –Producing a Final Report

**Introduction**

The aim of this document is to make clear to project teams what is expected from the end of project delivery. It gives guidelines for the production of the following documents.

* The end-of-project report
* The project test report
* The project maintenance manual report
* The personal reflective report
* Also gives a checklist for other documents and code which should be included.

**Extra Documents Produced for the Final Delivery**

*The End-Of-Project Report*

This document should state clearly how much has been accomplished on the project. Typically the report will be written by the Project Leader with contributions from the rest of the team. It should be written as a coherent submission, and should provide the reader with the following items:

*A management summary:* This should sum up in one page what the project achieved. It should not take more than one page of A4.

*A historical account of the project:* This should outline the main events over the lifetime of the project. It should take no more than 2 a4 pages.

*Final state of the project:* This should give a summary of which parts of the project are perceived as correct and which are not. It is good to be as accurate as possible. Also it needs to address any missing features.

*Performance of each team member.* The project leader should write a description of up to half a page of the duties and performance of each group member, including himself.

*Critical evaluation of the team and the project:* No more than a page and should address the following:

* How did the team perform as a whole and how could that have been improved?
* How could the project hat was set for you have been improved.
* What were the most important lessons learned about software projects and about working in teams?

*The project test report:*

This document will be a list of the named/numbered tests planned for the project, plus for failed tests, an explanation of why the test failed if available. In some cases, it may be possible to group failed tests with a single explanation. In such a case, the explanation should be given once.

The document should report on both the results for the tests in your test plan, and the acceptance tests given by the client. This report should have a standard document header page, but does not need *purpose/scope/objectives.*  The body of the document can just be two tables of tests.

*The project maintenance manual:*

Program maintainers pick up the maintenance documentation because they have a specific question in mind. The goal of your program maintenance documentation should be to answer all of the likely questions, or at least to show the installer and maintainers of your software which part of the program source is likely to provide the answers.

Most programs contain bugs. It is almost impossible to develop a software without any bugs. Normal commercial practice especially with minor bugs is to document them and correct them at a convenient time, rather than rushing to fix them and releasing a new version immediately. Making immediate fixes is costly in distribution and reinstallation of a new versions.

A checklist for the structure of a maintenance manual is:

*Program description –* gives a brief description of what the program does.

*Program structure –* Describes the design of the program

*Algorithms –*Description in detail of the significant algorithms used in the program.

*The main data areas –* It specifies the data structures.

*Files -* It may be that the program accesses certain fixed files or needs files of a certain type to be available.

*Interfaces –*Many programs control or read devices such as measuring instruments.

*Suggestions for improvements –*Most programs are a compromise between what one would like to do and what one has time to do. It is worth giving suggestion for future improvement of the software if it wasn’t possible to do it at the developing time because of various problems.

*Things to watch for when making changes –* It is desirable to make only changes or to be careful when making changes to the software. The developer must be aware that when they make changes these changes shouldn’t change any other working part of the program.

*Physical limitations of the program –* A computer installation is a finite environment. It can only have so much memory. Some programs will come against these constraints. It is important to list the requirements, where known, because all environments impose the same constraints.

*Rebuilding and testing -* Maintainers need to know what to do when rebuilding the program. What is the location of all the files? What should be done in order to rebuild the software. How do they add a test when a new problem is discovered. If documents are not in standard format, then it is worth to describe how to build them.

Feedback will have been given on the documents delivered before Easter (for user interface plan, the test plan and the design specification.) Groups are expected to use the feedback along with experience of trying to implement their design to produced improved version of the three documents.

**Structure of the Final Delivery**

*Reports:* All of the new reports plus the revised reports should be submitted for the group on Blackboard:

* The End-of-Project Report
* The Project Test Report
* The Project Maintenance Manual
* The User Interface Hand-In
* The Test Plan
* The Design Specification

All documents should be provided in a format which can be examined on screen. MS Word is acceptable.

The group project team should make sure that the state of their project repository at the end of Coding Week is such that it contains a working version of the project, and that it is accessible and easy to run. The client will meet with the team on the following week for demonstration of the software.

**Personal Reflective report**

In addition to previous documentations, each person must submit a personal reflective document. It should review the roles and tasks undertaken by the student during the project. Here individuals may include any issues relevant to the assessment of the project which might otherwise go unreported. An example would be problems that arose from personality clashes and led to seemingly unfair allocation of duties. This report should not be more than 2 pages A4. This report should be submitted in blackboard. A link to the student’s blog should also be included, and the blog should be kept available until the end of May.