QA 08 –Operating procedures and Configuration Management Standards

**Software Configuration management**

Configuration management is the management and control of all the changes in the system so that the state of each component is always known. Test specification and the User Interface document must be kept under control of configuration management system.

It will allow user to

1. Retrieve copies of any version of a *file,* enabling recovery of previous or old versions.
2. Retrieve copies of any version ofdirectory structure.
3. Check in changes to the file.
4. Inquire about differences between versions, obtain a log summarising the changes checked in for a particular version and produce a history of file showing all changes and the user responsible.

If the following items are produced they must be kept under control of the configuration management system:

1. List of project deliverables
2. Requirements Specification
3. Design Spec
4. Test Spec
5. User Interface document
6. Maintenance Manual
7. End-of-Project Reports
8. Source code and tools (e.g. build files)

The QA manager will be responsible for adherence to the configuration management procedures.

*Configuration* items are project items which are controlled by the configuration management system.

QA manager is responsible of the allocation of configuration references and for the maintenance of a file *config\_refs* in the configuration directory.

Any document must have some of these statuses: *Draft, for review, Release*

**The Structure of directories.**

One aspect of config management is the *standardisation* of the directory structure. This structure is managed by the version control system.

QA manager ensures that an initial structure is created. The following directories should be present + additional if required:

*Docs –*contains submitted documents produced by the group

*Man –*management documents are stored in this directory

*Src –*includes the source code for the project

*Dev –*This directory should contain a folder of the date of each tutorial. If student creates a draft document or some code as part of their duties for that week then it should be submitted to this folder.

*IDEs will typically manage file directories at an appropriate level for including your emerging product in* ***src.*** *For example, NetBeans will use a directory called* ***nbproject.*** *Eclipse will have a number of “dot” files. These must be under version control. Unit test code will be in the* ***src*** *directory. The IDE will normally manage this. Directories generated by the IDE from source must not be under version control.*

**Managing documents**

Keep a proper status level for each document. When updating a document the responsible must ensure that they have the correct version. Major changes must take place through version control branch.

**Managing minutes of meeting**

They will be stored in the **man** minutes directory. There must be only 1 file per meeting and it has to have the date text plus any file extension where needed

**Problem reporting and corrective action**

Problems relating to items in **docs** directory must go through a formal problem reporting and action process.

A form called Change Control Form is necessary to record the progress of changing released documents. There is an example of this CCF document in GPDocs. Completed forms should be kept in a ring binder know as Change File. This binder must contain dividers with these headings: ***Outstanding CCFs****,* ***CCFs Dealt with****.* The *change file* should be maintained by the Project leader. The Quality assurance manager must make sure these files are completed and used correctly. The following is a problem reporting and corrective action procedure:

* When we discover a problem in a configuration item that has been allocated a vs number and has a status of *draft, or release,* the details of the problem must be noted on a Change control Form
* The CCF must be given to the QA Manager, who will assign a unique CCF number to the form and who will then investigate the problem with the aid of other team members as appropriate.
* When a change has been completed whoever made the change must report to the QA manager. Then QA manager may initiate formal testing to confirm the change fixed the problem.
* If the change is accepted by the QA manager, the configuration item must have its version number incremented. The CCF corresponding to the reported problem will be then moved from the ***Outstanding CCFs***to ***CCFs Dealt With****.*
* If the change is not accepted, then the QA manager will advise on further action if necessary to complete a new CCF.

Project teams can choose to run all reporting and corrective action online, In order to do so they have to:

* Create a directory *CCFs* at the top level. A copy of each raised CCF should be kept in the CCFs directory. The first should be named CCF001.doc.
* Two files must be maintained by the Project Leader in the CONFIG directory. These files are called *Outstanding\_CCFs* and *CCFs\_Dealt\_With.*