

Quality Group Practice Guides

Configuration Management Guidelines

Abstract

Configuration Management (CM) is about controlling items and change on a project (i.e. the Management of Configurable Items). The purpose of this note is to introduce the concept of configuration management and point to specific texts that will aid a complete understanding

Introduction

Configuration Management (CM) is about controlling items and changes on a project i.e. the Management of Configurable Items. This includes:-

- Managing requests for change made internally and by the customer;
- Version control on documents (a document is an example of a configurable item);
- Version control on code (a 'C' program is an example of a configurable item);
- Maintaining relationships between configurable items. At its simplest, this means - version 5.0 of a release of a system consists of version 1.1 of program A, version 2.3 of program B, and so on. For example, a build state log and may even go as far as including other configurable items such as version 8 of design document C, version 3.8 of test script etc;

Management of work areas (e.g. people should develop code in one area, progress it to its next state i.e. unit test, then onto system test, release to UAT, live, and so forth. These areas should be properly controlled); .

Configuration Management should be in place from the very beginning of a project when the system requirement is first specified. If the correct information is maintained about relationships between configurable items then this can greatly help with impact analysis of changes, it provides traceability of changes throughout the life cycle of a project

A good configuration management tool can help with version control, building code, maintaining relationships etc. However a proper understanding of the principles of Configuration Management and good working procedures are still necessary.

This document should be read in conjunction with the following Quality Group Practice Guides:-

- Project Life Cycle Guidelines Stages 2, 3 & 4

Scope

The purpose of this section is to introduce the concept of configuration management and point to specific texts that will aid a complete understanding.

Glossary of Terms

Software Configuration Management	The practice of applying administrative and technical procedures throughout the software life cycle to:- <ul style="list-style-type: none">• Identify, define and baseline software items and modification requests;• Control modifications and releases of the items;• Record and report the status of the items and modification requests;• Ensure the completeness, consistency and correctness of the items;• Control storage, handling and delivery of the items.
------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Process

ISO/IEC 12207:1995 defines the configuration management process as:-

- Process implementation;
- Configuration identification;
- Configuration control;
- Configuration status accounting;
- Configuration evaluation;
- Release management and delivery.

Potential Problem Areas

The following are areas in which problems may arise, and where Configuration Management could help:-

- R&D changing code that should have been owned by project - invalidated testing, made code suddenly stop working;
- Lack of maintenance of proper build state logs;
- Management of DB changes - SQLs (Structured Query Language) scripts for example should be under version control;
- Proper management of all configurable items e.g. the directory structure, database versions, environment variables etc;
- Maintenance of multiple versions of installations. A proper shadow environment of customer environment should exist;
- Ensuring adequate regression testing has been performed;

Cost Effectiveness

Various factors must be considered when evaluating the cost effectiveness of the chosen CM solution:-

- Think of building code and maintaining different versions, environments etc. This is where a tool would make the most immediate positive impact;
- Impact analysis;
- Product management (rather than project management)

Think of cost not only in terms of rework/money but also in terms of quality, customer satisfaction.

Areas Where a Tool Would Be Of Great Benefit

The following are areas of the life-cycle where the use of a good CM tool would be of benefit:-

- Controlling code e.g. building code - we could know what is there and easily rebuild if required;
- The correct tool could help speed up making builds and releases;
- Impact analysis for changes;
- Use of a CM tool means we could manage the product rather than just individual projects, thus making it easier to share code changes, assess bugs across projects etc;
- The correct tools could help merge changes so that one project makes the change and the others can merge those changes into their code - thus saving time.

Responsibility

Depending on the nature and complexity of a project, an independent configuration manager/controller should be appointed whose responsibility will be to define and develop a configuration management plan where the detailed statement of intent in a project is described with the purpose of creating a base line and strategy.

Configuration Identification, Base Line And Status Accounting

Identification of configuration items will be a continuous process, which will start with the creation of initial sets of project development items, forming the base line for the configuration management. The typical development items in a project could be requirement specification, design specification, environment tools, test plan, test cases, and software items with version numbers for modules and sub-modules or customer property (not an exhaustive list).

The status accounting is an important activity to maintain the status of each development items with different status/stages information until the time when the base line is formed with the first product of a particular stage reaching configuration item status. Different status indicators might include 'uncontrolled items', 'under review', 'review complete' and 'configuration item'.

Further Reading

1. ISO/IEC 12207:1995 Information Technology - Software life cycle processes
www.bsi.org.uk
2. ISO/IEC TR 15846:1998, Information technology - Software life cycle processes - Configuration management; www.bsi.org.uk
3. ISO 10007:2003 Quality management - Guidelines for configuration management; www.bsi.org.uk
4. ITIL - Information Technology Infrastructure library - Configuration Management www.itil.co.uk