UECS3383 Software Quality Assurance

Lecture 07 – Configuration Management

The Need for Configuration Management

- “What is the correct version of the software module that I have to continue its coding?”
- “Who can provide me with an accurate copy of the last year’s version 4.1 of the TMY software package?”
- “What version of the design document matches the software version we are adapting to a new customer?”
- “What version of the software system is installed at ABC Industries?”
- “What changes have been introduced in the version installed at the ABC Industries’ site?”
- “What changes have been introduced in the new version of the software?”
- “Where can I find the full list of customers that use version 6.8 of our software?”
- “Can we be sure that the version installed at Top Com Ltd. does not include undocumented changes?”

Definition of Software Configuration Management

- **Software configuration item (SCI)**
  - An approved unit of software code, a document or piece of hardware that is designed for configuration management and treated as a distinct entity in the software configuration management process.
- **Software configuration item version (SCI version)**
  - The approved state of an SCI at any given point of time during the development or maintenance process
- **Software configuration version**
  - An approved selected set of documented SCI versions, that **constitute** a software system or document at a given point of time, where the activities to be performed are controlled by *software configuration management* procedures.

Software Configuration

- A software configuration is composed of as many SCIs as the developers assume will be needed in the future, with each SCI approved, identified and registered
- The SCIs aggregated in each software configurations version naturally correspond to the software components and others software definitions

Common Types or Classes of Software Configuration Items (SCI)

- **Design documents**
- **Software code**
  - Source code
  - Object code
  - Prototype software
- **Data files**
  - Test cases and test scripts
  - Parameters, codes, etc.
- **Software development tools** (the versions applied in the development and maintenance stages)
  - Compilers and debuggers
  - Application generators
  - CASE tools

Example: Different Software Configuration Versions of PMT Software

	Release and release date	
SCI Version	**PMT Version 6.0 January 6, 2002 SCI Version in the Release**	**PMT Version 7.0 January 22, 2003 SCI Version in the Release**
SRD (Software Req Doc)	**Ver. 1**	**Ver. 1**
CDD (Critical Design Doc)	**Ver. 3**	**Ver. 4**
STP (Software Test Plan)	**Ver. 3**	**Ver. 4**
SIP (Software Installation Plan)	**Ver. 2**	**Ver. 2**
VDD (Version Desc Doc)	**Ver. 6**	**Ver. 7**
Code Module 1	**Ver. 3**	**Ver. 5**
Code Module 1	**Ver. 8**	**Ver. 8**
Code Module 1	**Ver. 2**	**Ver. 2**
Test cases file	**Ver. 3**	**Ver. 4**
CL compiler	**Ver. 5**	**Ver. 7**
Software user manual	**Ver. 6**	**Ver. 7**

Definition of Software Configuration Management

- An SQA component responsible for applying (computerized and non-computerized) technical tools and administrative procedures that enable completion of the tasks required to maintain SCIs and software configuration versions.

Tasks of Software Configuration Management

- Control software change
- Release of SCI and software configuration versions
- Provision of SCM information services
- Verification of compliance to SCM procedures

Tasks of Software Configuration Management

- **Control software change**
  - Grant approval to carry out changes
  - Control the changes and assure the quality of approved changes
  - Document the approved changes
  - Apply mechanisms that coordinate the changes made to the SCI by preventing more than one team from simultaneously introducing changes into the same SCI

Tasks of Software Configuration Management

- **Release of SCI and software configuration versions**
  - Approve the release of new versions
  - Document the configuration of each released software configuration version
  - Document the sites where software configuration versions are installed
  - Secure the version source and documentation files from changes, deletions and other damages

Tasks of Software Configuration Management

- **Provision of SCM information services**
  - Information about the status of changes
  - Information about versions installed at a site as well as about the site itself
  - Version history list
  - Accurate copies of given versions
  - Supply copies of documentation
- **Verification of compliance to SCM procedure**
  - Audit compliance to SCM procedure
  - Initiate updating and change of SCM procedure

The Software Configuration Authority

- It is practically self-evident that an authority to oversee implementation of the above tasks is vital in software developing and/or maintaining organizations
- SCM procedures specify who is responsible for SCM issues
- This responsibility is usually assigned to a senior professional or a committee dedicated to SCM issues
- In many organization, software change control is dealt with by a special committee called the “Software Change Control Authority” (SCCA) or the software change control board (SCCB). This body is frequently called the change control authority (CCA) or the change control board (CCB)

Software Change Control

- Software change management controls the process of introducing changes mainly by doing the following:
  - Examining change requests and approving implementation of appropriate requests
  - Assuring the quality of each new version of software configuration before it becomes operational

Factors Affecting Approval of Proposed Change

- Expected contribution of the proposed change
- Urgency of the change
- Effect of the proposed change on project timetables, level of service, etc.
- Efforts required in making the change operational
- Required software quality assurance efforts
- Estimated required professional resources and cost of performing the change

Template – Software Change Request (SCR) Document

1. **Change Principles**
   - The initiator
   - The date the SCR was presented
   - The character of the change
   - The goals
   - The expected contribution to the project/system
   - The urgency of performance
2. **Change Details**
   - Description of the proposed change
   - A list of the SCIs to be changed
   - Expected effect on other SCIs
   - Expected effect on interfaces with other software systems and hardware firmware
   - Expected delays in development completion schedules and expected disturbances to services to customers
3. **Change Timetable and Resource Estimates**
   - Timetable for implementation
   - Estimated required professional resources
   - Other resources required
   - Estimated total cost of the requested change

Quality Assurance of Software Changes

- While change efforts are directed to one or several SCIs, the user experiences the changes indirectly, through application of the revised version of the software system
- The goal of software quality assurance is to assure that the quality of the new software system version does not fall below that of the previous version
- Quality assurance efforts are required at two levels:
  - Quality assurance of each of the changed SCIs
  - Quality assurance of the entire new software system version (that includes changed SCIs)

Quality Assurance of Software Changes

- **Quality assurance of the changed SCIs**
  - Requires preparation of a reviews and testing plan at a magnitude appropriate to the character of the change
  - The process of reviews and testing, corrections and retesting (regression testing) the changed SCIs is expected to conclude with their approval

- **Quality assurance of the entire new software system version**
  - A new version of the software is considered to have been completed once the changed SCIs replace the former SCIs
  - Although one might expect the new version of the software system to function perfectly and certainly better than the old original version, many new versions, especially of complex software systems, actually fails
  - These system failures generally occur as a result of damage done to interfaces between the changed SCIs and other SCIs left unchanged and not retested because they were not expected to be affected by the changes performed

The Need to Release a New Software Configuration Version

- Defective SCIs
- Special features demanded by new customers
- Team's initiatives to introduce SCI improvements

Types of Software Configuration Releases

- **Baseline versions** – are configuration versions that are planned ahead, during a system's development or operating stage. As part of the process, baseline versions are also reviewed and approved. As a rule, they serve as milestones in the software system's life cycle

- **Intermediate versions** – are software configuration versions released, in most cases, to respond to immediate needs. These may range from correction of defects identified in an important SCI to swift introduction of adaptations to meet a new customer's requirements. As expected, intermediate versions will not receive the attention and efforts typically invested in baseline versions

Types of Software Configuration Releases

- **Revisions** – introduce minor changes and corrections to a given software configuration version. In some cases, several successive revisions are released before a new baseline version is released

- **Numeration conventions for identification of SCI and software versions**
  - Have been formulated to identify SCIs; the most commonly used is decimal numeration, which indicates the successive version and revision numbers and is registered accordingly
  - Example: An SCI design document captioned DD-7 may have several versions and revisions, identified as DD-7 Ver. 1.0; DD-7 Ver. 1.1; DD-7 Ver. 2.0; DD-7 Ver. 2.0; etc., where the first number represents the version and the second the revision. In short, an SCI is identified by its name in combination with its version and revision numbers

Software Configuration Management Plan (SCMP)

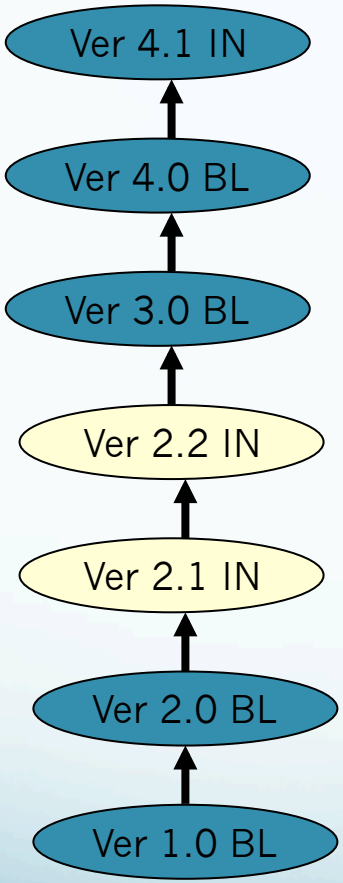
- **The plan includes:**
  - An overview of the software development project or existing software system
  - A list of scheduled baseline version releases.
  - A list of SCIs (documents, code, etc.) to be included in each version.
  - A table identifying the relationship of software development project plans and maintenance plans to scheduled releases of new SCIs or SCI versions.
  - A list of assumptions about the resources required to perform the SCMP.
  - Estimates of the human resources and budget needed to perform the SCMP.

Software Configuration Evolution Models

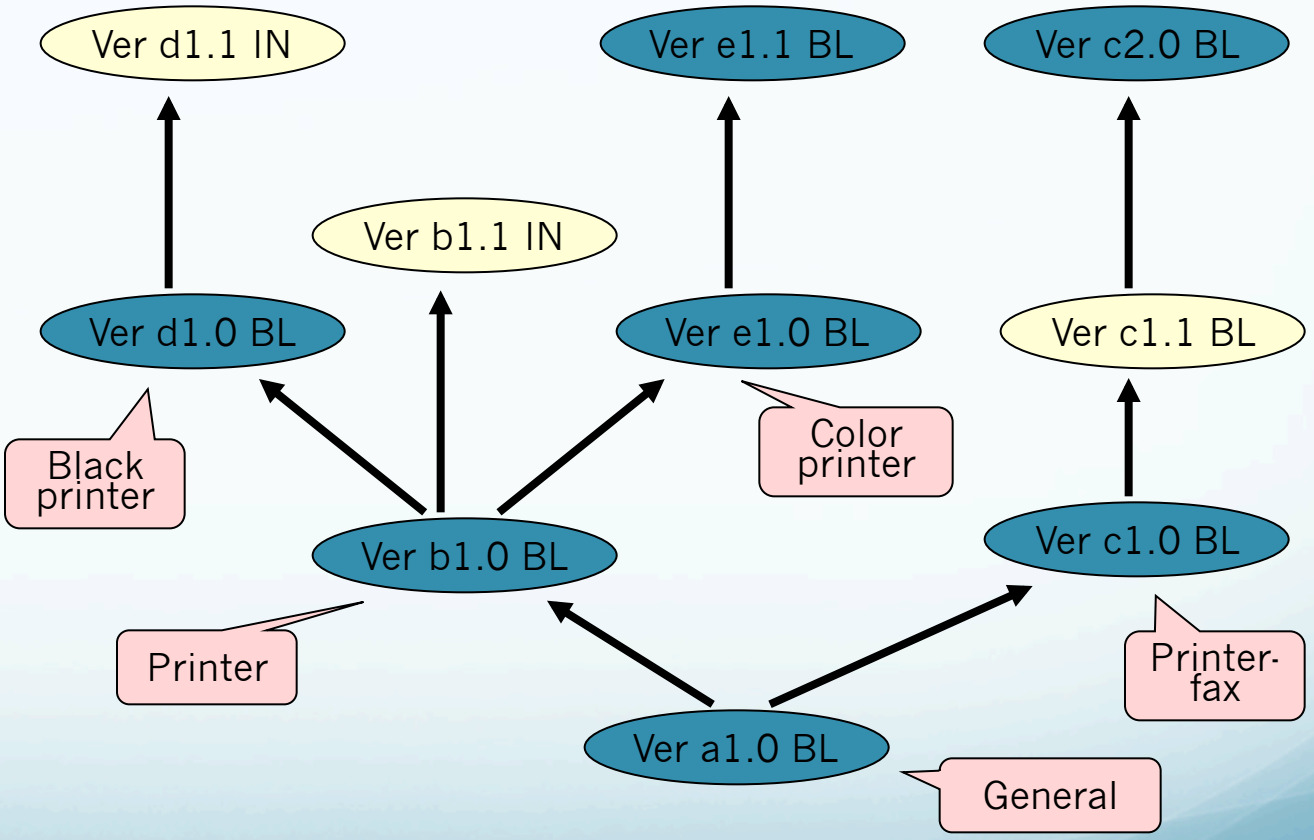
- Successive development or evolution of a software system's configuration versions should be undertaken according to a route that is planned in advance by the system's developer

- Two fundamental software configuration evolution models:
  - **Linear evolution model** – only one unique software system's configuration version serves all customers at any given time. Each new configuration version then replaces the prior version. This model can be applied to software which tend to be uniform in structure, where the need to meet a wide range of maintenance demands for a single version
  - **Tree evolution model** – several parallel versions of the software are developed to serve the needs of different customers simultaneously throughout the system's life cycle, where each branch serves a different product or product line

Software Configuration Evolution Models



Linear evolution model



Tree evolution model

Documentation of Software Configuration Versions

- Two main types of tasks to be completed within the framework of software configuration management:
  - **Documentation of Software Configuration Releases** (versions and revisions)
  - **Documentation of SCI versions**
    - **Identification**
      - SCI version number
      - Name(s) of software engineer(s) who implemented the change
      - Date the new version was completed and approved
    - **Changes in the new version**
      - Former SCI version number
      - Short description of the introduced changes
      - List of other SCIs that had to be changed as a result of the current changes
      - List of SCOs included in the new version
      - List of software problem reports resolved by the new version
      - Operational as well as other implications of the changes introduced in the new version

Software Configuration Release Documentation (Version Description Doc (VDD)) Template

- **Identification and installations**
  - Release version and revision number, including date
  - List of installations where the release was installed

- **Configuration of the released version**
  - List of SCIs (including SCI's version) in the released software version
  - List of hardware configuration items required for operating the specified version
  - List of interfacing software and hardware systems
  - Installation instructions for the new release

Provision: Software Configuration Management (SCM) Services

- **Information related to software change control:**
  - Change request status information
  - Change order progress information
- **Information about SCIs and software configuration versions:**
  - Accurate copies of SCI versions (code SCIs, document SCIs, etc.) and entire software configuration versions.
  - Full reports of changes between successive releases (versions and/or revisions) of code SCIs and between successive releases of other types of SCIs.
  - Copies of SCI version documentation and software configuration version documentation (VDDs).
  - Detailed version and revision history for SCIs and software configurations.
  - Progress information about planned versions and releases
  - Information correlated about versions installed at a given site and about the site itself.
  - List where a given software configuration version is installed.

Software Configuration Management Audits

- SCM involves the execution of a great variety of tasks by the SCM authority, the CCB and many others involved in software development and maintenance
- SCM audits are performed by the SCM authority and the CCB in order to control compliance with SCM procedure
- SCM audits may combine with internal quality issues, and are expected to initiate updates and changes of SCM procedures and instructions
- SCM audits check whether and how these tasks were performed for samples of change requests, SCIs and software configuration versions
- SCM audits may also be performed for a sample of planned releases, as specified in the SCMP
- However, although we expect SCM audits to yield information regarding the level of compliance to SCM procedures (including typical failures of those procedures), they cannot serve as compliance enforcement tools

Software Configuration Management Audits

- Listing of control Information of SCM audits in discovering and transmitting to management:
  - Percentage of unapproved changes introduced in the system during development or operation
  - Percentage of SCOs not carried out according to instructions and not fully complying with procedures
  - Percentage of design reviews and software tests of changed SCIs that have not been performed according to the relevant procedures
  - Percentage of SCOs that have been completed on schedule
  - Percentages of cases where SCIs affected by changes have not been checked, with some necessary changes not implemented
  - Percentages of properly documented new SCIs and software configuration versions
  - Percentage of properly documented installations of new software configuration versions
  - Percentage of cases of failure to transmit all version-related information to the customer
  - Number of cases recorded annually where the SCI work coordination mechanisms failed (i.e. did not prevent different teams from simultaneously introducing changes in the same SCI)

Computerized Tools for Managing Software Configuration

- Ability to comply with the required level of accuracy and completeness of information, with the required level of availability (measured by the response time from request of information to its provision)
- Should operate the mechanisms coordinating the work on an SCI's changes and prevent different teams from simultaneously introducing changes in the same SCI
- With the high security level, it is able to provide:
  - It secures the code version and documentation files versions by protecting them from any changes, deletions and other damages
  - It activates back-up procedures required for safe SCM file storage
- Current enhanced tools are characterized by easier input capacities, coordination of SCM support teams operating in different development environments, including geographically distributed teams, and provision of an expanded variety of reporting options