Software Configuration Management

SIT 401: Software Project Management

Software Configuration Items

- Software configuration: All items that constitute the software while under the development (e.g., programs, data, documents such as the software requirements specification, test cases, etc.); they are referred to as Software Configuration Items or SCIs
- Most (or all) SCIs change during the development; such changes must be effectively controlled

Software Configuration Management

• Software Configuration Management (SCM) is the process of identifying and defining the SCIs in the system and coordinating the changes made to these items • A formal definition: “SCM is the process of identifying and defining the items in the system, controlling the change of these items throughout their life cycle, recording and reporting the status of items and change requests, and verifying the completeness and correctness of items” (IEEE)

Software Configuration Management (continued)

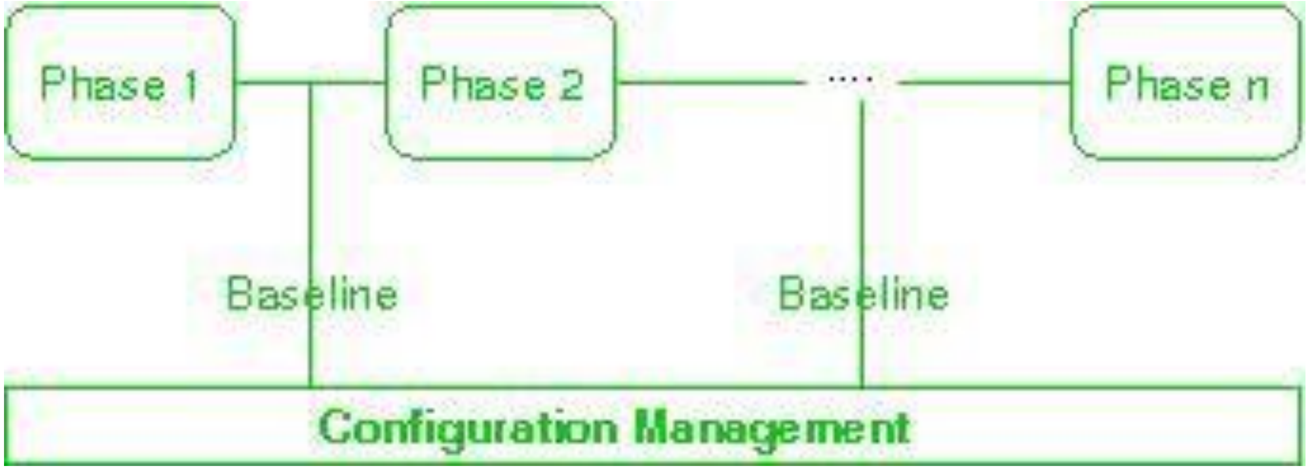
- SCM provides for systematic evolution of a software under development and provides for
  - Visibility
  - Controlled change
  - Traceability
  - Monitoring

CMM KPA on SCM

- Goals of Level 2 KPA (for SCM) – Goal 1: Software configuration management activities are planned
  - Goal 2: Selected software work products are identified, controlled, and available
  - Goal 3: Changes to identified software work products are controlled
  - Goal 4: Affected groups and individuals are informed of the status and content of software baselines

Baselines in the Context of SCM

- Primary task of SCM: control and monitor change



- Baseline: “A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed (IEEE)

Components of a SCM

- Software configuration identification
- Change control
- Configuration status accounting
- Configuration audit

Software Configuration Identification

- No fast rules: once a SCI has been identified, it should be given an identification name
- Examples of SCIs – The entire product of a software development phase (e.g., a requirements document)
  - Chapters or sections of a document
  - A separately compilable module

  - A file consisting of a number of modules
  - A file consisting of module definitions

Baselines of a Typical SDLC

- Phase/discipline: requirements analysis Baseline: software requirements specification
- Phase/discipline: software design Baseline: design specification
- Phase/discipline: coding and implementation Baseline: source code

- Phase/discipline: testing and integration
  Baseline: test plans and data
- Phase/discipline: acceptance testing/release
  Baseline: operational software

Software evolution

Software evolves over a period of time

- Many different items are produced over the duration of the project
- Different versions are produced
- Teams work in parallel to deliver the final product

Software evolution implies a constantly changing system

How software changes

The four aspects of software evolution are:

- Corrective changes
- Adaptive changes
- Perfective changes
- Preventive changes

i. Corrective changes

- Required to maintain control over the system’s day-to-day functions
- These changes are made as faults (or) bugs are found during the development time

- Some changes may be long-term and fundamental, some may be patches to keep the system in operation (emergency fixes)

ii. Adaptive Changes

- Essentially maintaining control over system modifications
- As one part of the system changes, other impacted areas will need to be updated
- Examples Database upgrades, Use of a new compiler or development tool

iii. Perfective Changes

- Perfecting existing acceptable functions

- The domain of Refactoring designs falls into this category
- Perfective changes are done to increase the long-term maintainability or elegance of the solution
- Involves changes to design or data structures for better efficiency, Updates to documentation to improve their quality, Enhancing the code to make it more readable.

iv. Preventive Changes

- Preventing the system performance from degrading to unacceptable levels.
- Involves alterations made to ensure that the system has a defence against potential failures. Example
- Adding extra redundancy module to ensure that all transactions are properly logged

Change Control

- Change control is the process of controlling the changes to the SCIs
- A formal definition: “An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes

to configuration items after formal establishment of their configuration identification” (IEEE)

Change Control Process

- The decision making process is carried out by the Configuration Control Board (CCB) – Consists of one or more individuals led by the Configuration Manager (CM)

- The change control process

  (1) While a SCI is under development (working state), it can change freely since it is not under SCM yet

  (2) When a SCI is in a stable condition, it is submitted to the CM for review (under review state)

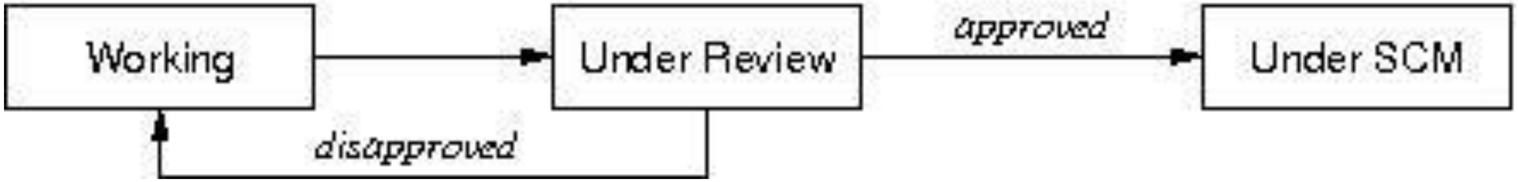
Change Control Process [continued]

- The change control process [continued]

  (3) While under review, the SCI is considered

“frozen”

(4) Once the SCI is reviewed (for satisfactory quality) and approved, it is entered into a library; the SCI is now formally under SCM



Change Control Process [continued]

- The change control process [continued]

(5) Once in the library (i.e., under the SCM), the SCI cannot be changed unless the change is approved by the CCB

(6) Request for change must be formally initiated via a Change

(7) Check-in, check-out process: Once the change has been approved, the SCI has to be checked-out from the library for change; once the change has been implemented, the SCI must be checked-in to the library

Change Request Form

- Proposer information
- Description of change
- Reasons for the change
- SCIs affected by the change
- Priority of the change (fault error?)

CCB’s Decision

- CCB makes an approval/disapproval decision (based on cost/benefit analysis, quality issues, scheduling)
- CCB assigns a unique number to the proposed change for future references

- CM recommends certain action (e.g., no action, change the SCI, changes to programs, changes to the documents)

Configuration Status Accounting

• Implementing a change may take a long time • The objective of Configuration Status Accounting is to answer question such as:

  - What is the status of a CR?
  - What is the status of an approved CR?

  - Scheduled/not scheduled? Active? Completed?
  - Who is in charge of implementing an approved CR?
  - What is the average time to process a CR?
  - What is the average efforts needed to process a CR?
  - What is the number of CRs per SCI?
  - Have all related SCIs been properly updated?

Configuration Auditing

- Configuration Auditing Objective: to verify compliance with configuration control standards

– Configuration auditing is performed by auditors (external to the development team) who are in charge of determining if the defined processes are being followed and to ensure that the SCM goals are satisfied – Determine whether software engineering and organizational standards (e.g., documentation standards, coding conventions) been properly followed?

Resources for SCM

- `make` utility: operates on a `makefile` (includes a definition of the SCIs and their dependencies and a procedure for re-building the software if any of the components of the system has changed since the last built); originally for the UNIX but migrated to the PC environments

- SCCS utility: administration programs for Source Code Control System (SCCS)
- `rcs` utility: a revision control that creates new revision control files or changes attributes of existing ones; an RCS file contains multiple revisions of text, an access list, a change log, descriptive text, and some control attributes
- Many other similar tools, e.g., `diff`
- Many commercial tools

A Plan for SCM

1. Introduction: Purpose, Scope, Definition, Acronyms

2. Management: organization of the project and baselines, SCM responsibilities, who does what
3. CM activities: how the organization will perform the CM activities (identification, control, status accounting, audit)
4. Tools, Techniques and Methodologies: technical details of implementing CM
5. Supplier Control (sub-contractor software; vendor software)
6. Records Collection and Retention