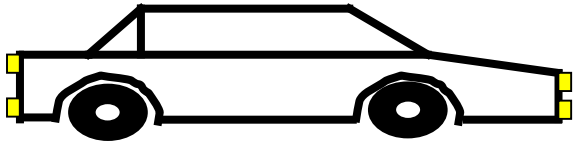

6
Software Configuration Management

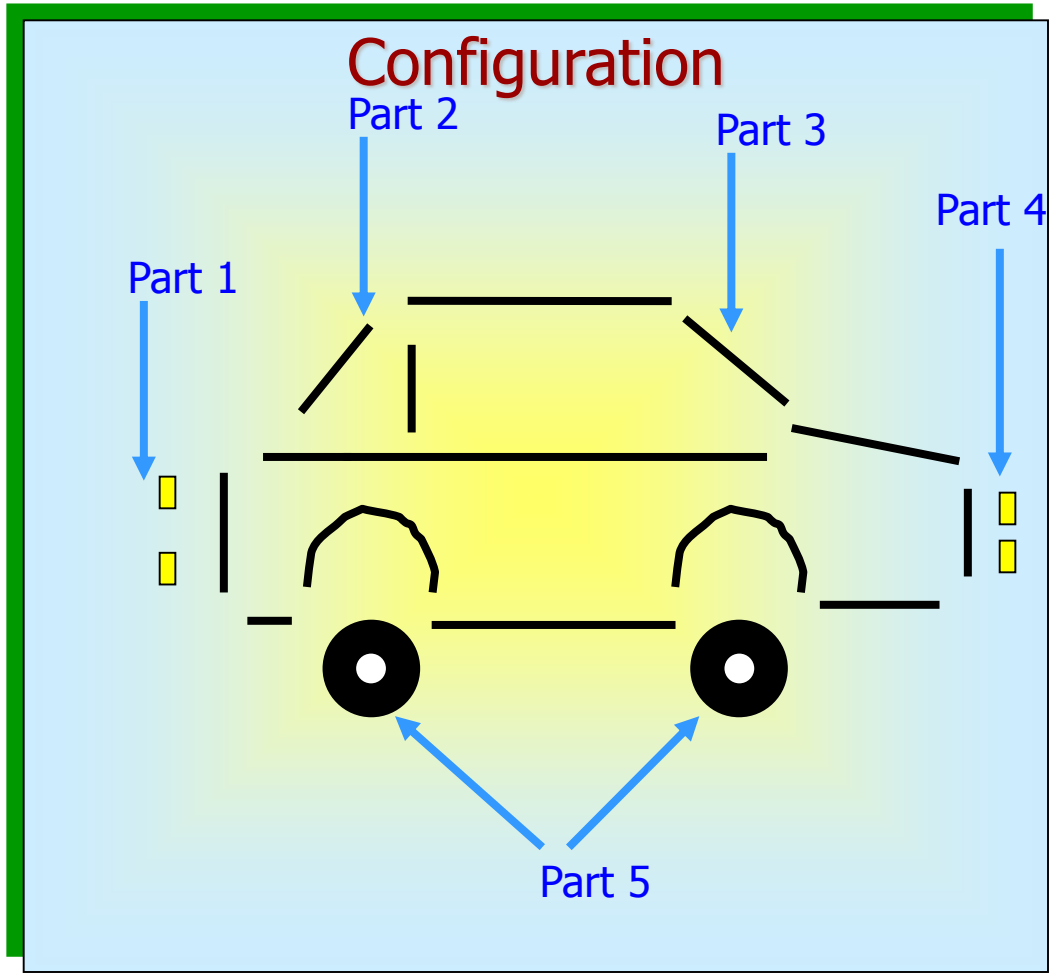
Objectives

After completing this lesson, participants will be able to:

- Define SCM
- Benefits of SCM
- Define Software Configuration
- Explain the SCM Workflow
- Work on Case study

Hardware Configuration & Parts List





PARTS LIST

Part 1 - Rear Lights

Part 2 - Rear Glass

Part 3 - Front Glass

Part 4 - Front Lights

Part 5 – Wheels.. etc.

Purpose:

- To know what parts are needed to assemble the car
- To know what parts are needed to repair car

CONFIGURATION: Is relative arrangement of parts

Purpose:

- To show how car is put together
- To show how car is taken apart

SCM?

- Change is inevitable when software is developed.
- Change increases the level of confusion .
- Confusion arises when…
  - Changes are not analyzed before they are made.
  - Not recorded before they are implemented.
  - Not reported to those with a need to know.
  - Not Controlled in a manner that will improve quality & reduce errors.

When is SCM used?

- IF
- …we can't identify the document(s) that corresponds to a given source code.
- …we can't determine which version of the compiler generated the executables.
- …a bug that was corrected last month suddenly reappears.
- …we can't identify the document & source code changes made to a particular version of the product.
- …shared source code is changed, and all developers sharing the code are not notified.
- **THEN what is lacking is Configuration Management (CM)**

What does SCM Answer?

- What is our current software configuration?
- What is its status?
- How do we control changes to the configuration?
- What changes have been made to the software?
- Does anyone else's change affect our software?

Definition

- ***The art of coordinating software development to minimize….confusion is called Configuration Management***

- *Babich – Software Configuration Management*

IEEE 828 Definition of SCM

- SCM is the process of identifying and defining items in the system, controlling the change of these items throughout their lifecycle, recording and reporting the status of items and change requests, and verifying the completeness and correctness of the items

SCM Addresses

- What constitutes the software product at any point of time
- What changes have been made to the software product

Why Use SCM

To Ensure That:

- Software conforms to requirements
- Software has undergone all reviews and tests
- Right version of any work product is used at any point of time
- All components of the delivered software are consistent with each other
- Right version of software components are shipped to the customer site

The Configuration Management plan

- Defines the types of documents to be managed in a document naming scheme.
- Defines who takes responsibility for the CM procedures and creation of baselines.
- Defines policies for change control and version management.
- Defines the CM records which must be maintained.
- Describes the tools which should be used to assist the CM process and any limitations on their use.
- Defines the process of tool use.
- Defines the CM database used to record configuration information.
- May include information such as the CM of external software, process auditing, etc

Four Part Discipline

CM is a four-part discipline applying technical and administrative direction, control and surveillance for :

- Configuration Identification / Baseline.
- Configuration Change Control
- Configuration Status Accounting
- Configuration Audit

Software Configuration

- The items comprising all the information produced as a part of software development process are software configuration
- Examples
  - Programs
  - Documents
  - Data

Configuration Item (CI)

- The item that comprises information produced as part of the software process activity is called a configurable item
- The key is to identify all such items that have an influence on the final behavior of the system
- This can be a single section of a large specification or a suite of test cases or a program component

Configuration Item Identification

- Large projects typically produce thousands of documents which must be uniquely identified.
- Some of these documents must be maintained for the lifetime of the software.
- Document naming scheme should be defined so that related documents have related names.
- A hierarchical scheme with multi-level names is probably the most flexible approach.

Typical CIs for Development Projects

- Requirements Document
- Design Documents
- Source Code
- Test plans with Test cases
- Test data
- User Manuals

Typical CIs for Maintenance Projects

- Application transition document
- Application design documents
- Application Code
- Test plan with Test cases
- Test data

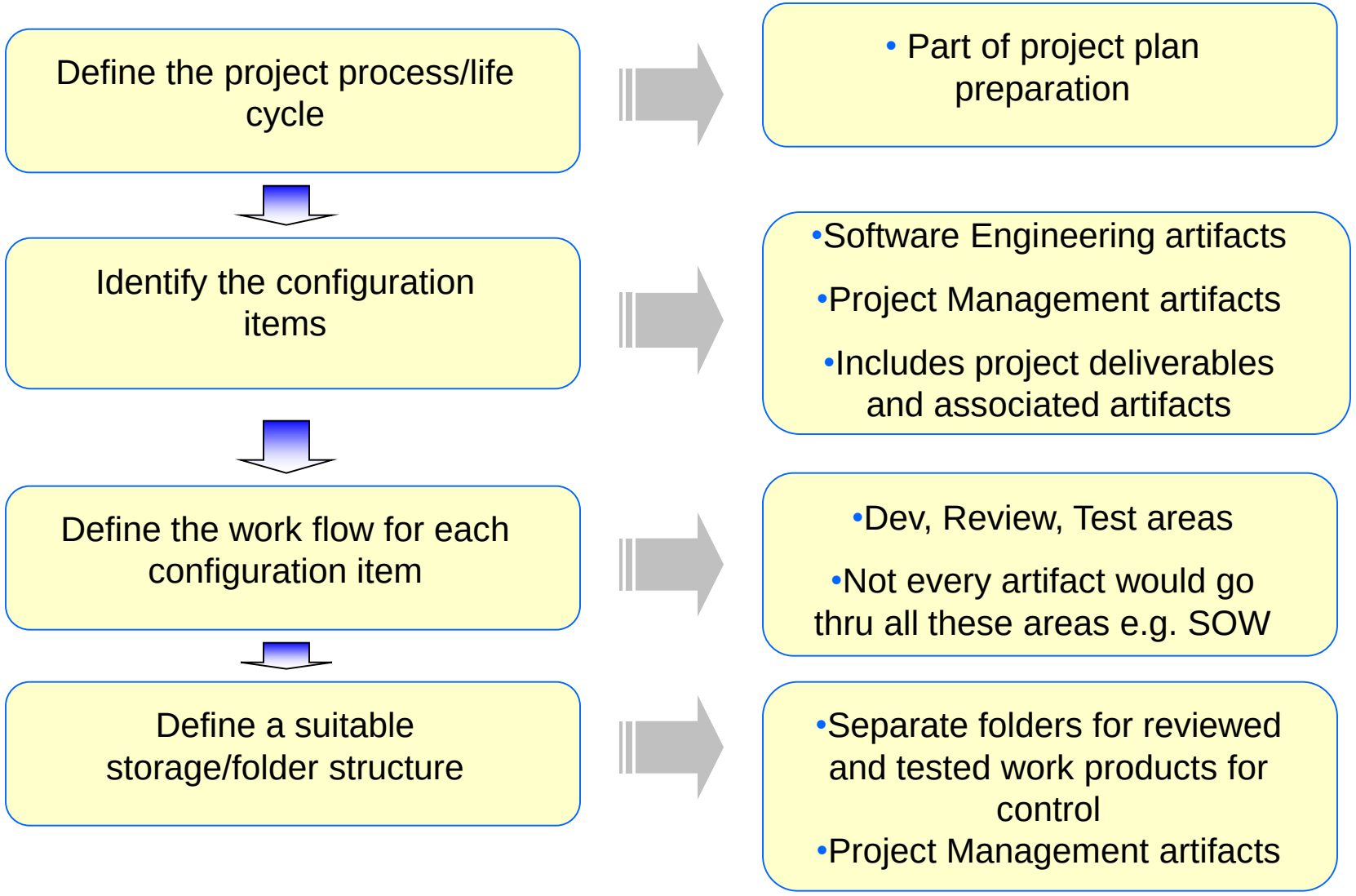
Typical CIs for Testing projects

- Test Strategy / Plan
- Test cases
- Test data
- Test Conditions
- Requirements traceability Matric

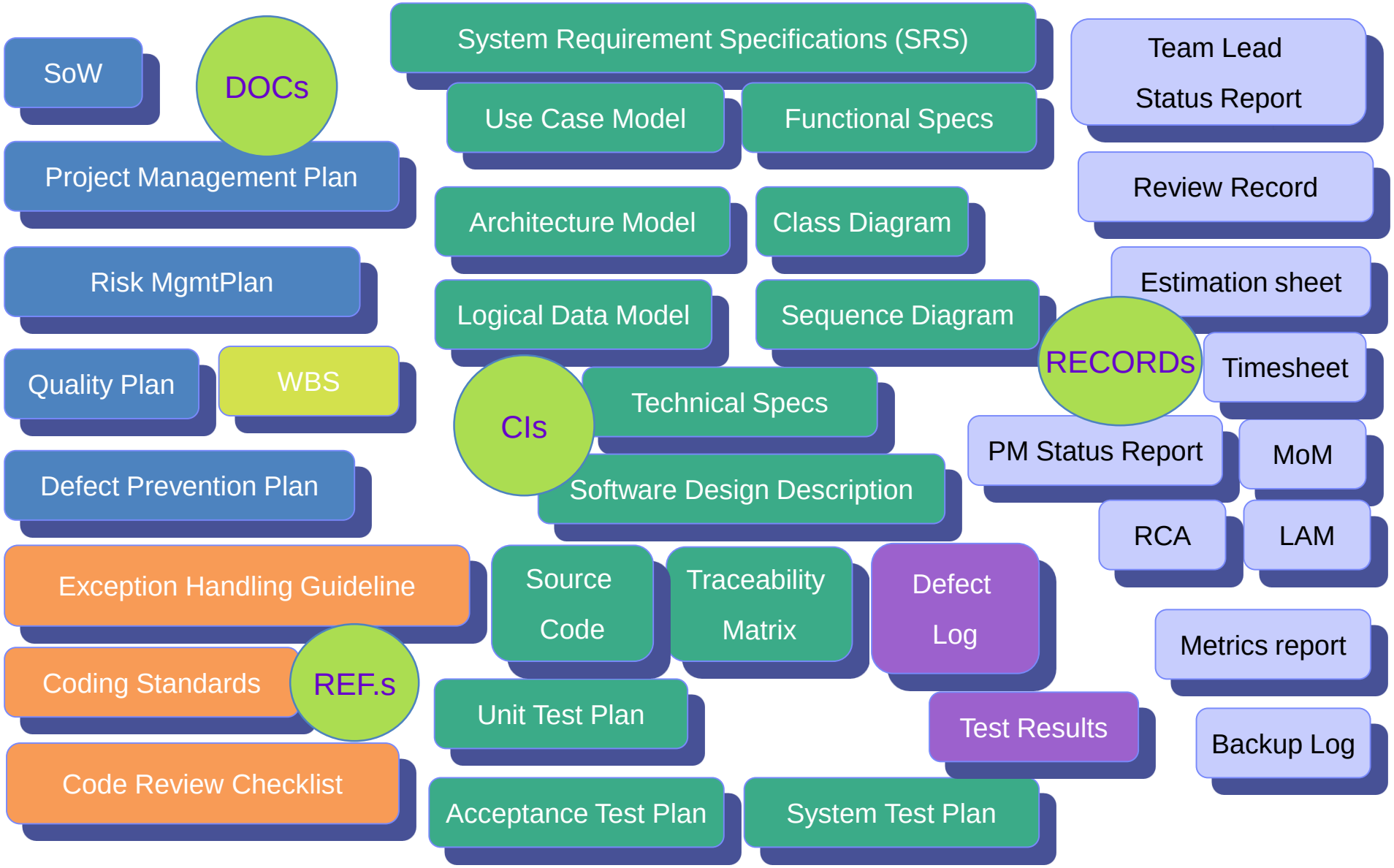
Typical CIs for Porting Projects

- Test Strategy/Plan with Test cases, data and results
- Database structures
- Rules for data extraction, transformation and loading

Plan for SCM : The Concept



Project Artifacts



What Is a RECORD

Records are a special type of document which are maintained to provide evidence of conformity to requirements and project processes

Identify storage location for records in SCM plan

Tools like Team room, SDMS, PTMS or Rational Portfolio Manager may be used for record maintenance

Typical Records

- Review Log
- Defect Log
- Configuration Audit Report
- Status Reports
- Minutes of Meeting
- Metrics report
- Backup log
- Media register

Plan when Baselines will be Established

Baseline : Any work product (code, test plans, design document, requirements document) that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change procedure

It is a document or a set of documents formally designated and fixed at a specific time during the development

Baselines are normally established at specific stage of a Project

- During Project Planning: At the start of the project - input baseline
- Baselines are done at the end of all phases of SDLC
- During Development: At the end of reviews and testing
- During Release: When deliverables are sent to the customer
- Before a change is made on a CI

Types of Baseline

- Initial Baseline – Established at the start of the project – after project initiation
- Requirements Baseline - established when the Requirements Specifications are completed and signed off
- Design Baseline – established when the Requirements Specifications are transformed completely into design specifications

Types of Baseline

Product Baselines

- Initial Product Baseline - Is established after the programs are coded, reviewed, unit tested, and accepted
- Release Baseline - Is established after the programs are integrated / system tested and accepted
- Acceptance Baseline - Is established when the final product is accepted by the Customer

Change Management

No matter where you are in the system lifecycle, the system will change, and the desire to change it will persist throughout the lifecycle

First Law of System Engineering

Bersoff & Henderson – Software Configuration Management

Change management

Software systems are subject to continual change requests:

- From users
- From developers
- From market forces

Change management is concerned with keeping track of these changes and ensuring that they are implemented in the most cost effective way.

Change Management (Contd.)

- To ensure that any changes to an approved baseline or end-product are controlled and authorized. The integrity of the end-product and traceability of the changes must be maintained at all times
- No change should be implemented until it has been formally approved
- Each project should define the authorization procedures for the change control. There could be number of levels of authorization and they must be specified
- Version Number must be clearly specified and maintained for all configuration items under change control

Change Management (Contd.)

- Safeguards defined and approved baselines
- Prevents unnecessary changes
- Expedites worthwhile changes
- Maintains product integrity
- Plan, control and track changes
- Avoids "*the left hand not knowing what the right hand doing*" syndrome

The change management process

```
Request change by completing a change request form
Analyze change request
if change is valid then
   Assess how change might be implemented
    Assess change cost
   Submit request to change control board
    if change is accepted then
        repeat
        make changes to software
        submit changed software for quality approval
       until software quality is adequate
   create new system version
 else      endif
   reject change request
else
    reject change request
      endif
```

Change Control Board

- A central controlling mechanism to ensure that every change is properly considered, evaluated, analyzed and implemented
- The objective is to review and approve the change reports and to manage the change control process in a fair and stable manner
- Ideally consists of *Project Manager, Project Leader ,Customer SPOC*

CCB Responsibilities

- Receive all changes
- Ensure impact analysis is done
- Assess the additional effort, cost and schedule changes required
- Discuss / Negotiate with the Customer
- Plan for change implementation
- Identify resources for implementing the change
- Track the change to closure

Configuration status reporting

- Reports on status of change requests
- Reports on changes due to new requirements
- Status of baseline library

Configuration Status Accounting (CSA)

The Configuration Status Accounting system will provide for you a record of the following information for each application software configuration item (CI) for example:

1. The planned and actual dates the:

- Application software specification was released
- Allocated baseline was established
- Application software design document was approved, design document was released
- Application software Test Plan for the CI was approved, Test Plan for the CI was released
- Application software CI testing took place, test report for the CI was approved
- Application software Product Baseline was established

2. The Configuration Status Accounting system will record all change requests made to the application software specification (to the allocated baseline):

- The date, title, status, approval date of the change request
- Also the planned and actual dates of change implementation

This will ensure you have the latest and greatest version of your software within your archive and deployed to your customers

Conduct SCM Audit

Why Configuration Audit

- Check to ensure that configuration management activities are conducted as per plan
- Compliance to the applicable standards is maintained
- Integrity of the baseline is maintained
- Baseline contents are complete & correct
- Status of each of the configuration items is correctly available
- To reduce the Project Risk

Conduct SCM Audit

Types of Audit :

- Physical Configuration Audit (PCA)
- Functional Configuration Audit (FCA)

What is Physical Configuration Audit ?

An audit to verify that the software work product is being developed as per the customer specified requirements

- E.g. During the Design, Coding, Testing phases

What is Functional Configuration Audit ?

An audit to verify that the development of a configuration item (CI) has been completed as per

- Functional requirements
- Non-functional requirements e.g. performance, scalability, reliability

Conducted before the deliverables are shipped to the customer

☝*In IGSI there is no distinction of Physical & Functional Audits*

Prepare Software Build Plan

- Define the build procedure e.g. incremental, sequence of integration of programs/modules
- Specify the entry criteria for build e.g. completion of a set of programs/modules, readiness of test cases
- Specify the build environment – hardware, software, tools

System Building

- The process of compiling and linking software components into an executable system
- Different systems are built from different combinations of components.
- This process is now always supported by automated tools that are driven by ‘build scripts’

Release management

- Releases must incorporate changes forced on the system by errors discovered by users and by hardware changes
- They must also incorporate new system functionality
- Release planning is concerned with when to issue a system version as a release

System Releases

- Not just a set of executable programs
- May also include:
  - Configuration files defining how the release is configured for a particular installation
  - Data files needed for system operation
  - An installation program or shell script to install the system on target hardware
  - Electronic and paper documentation
  - Packaging and associated publicity
- Systems are now normally released on optical disks (CD or DVD) or as downloadable installation files from the web.

Software Configuration Management Tools

- Visual source safe from Microsoft
- Rational clearcase from IBM
- Tortoise SVN / SVN from Apache (Open Source)
- CVS Concurrent Version System (Open Source)

Case study

- A project is in the User Acceptance Testing Phase and while testing the customer realized that an important feature is missed while doing the requirement analysis, also without this feature the application remains incomplete.
- In the above scenarios what should be done to accommodate the requirement.

Answer: The request should be taken to the change control board and a through impact analysis should be done and on approval, should be taken for further implementation.

Benefits of SCM

- Increased software development productivity
- Lower software maintenance costs
- Better quality assurance
- Reduction of defects / bugs
- Faster problem identification and bug fixes
- Process dependent development rather than person dependent development
- Assurance that the correct system is built

Summary

- Configuration management is the management of system change to software products
- A formal document naming scheme should be established and documents should be managed in a database.
- The configuration data base should record information about changes and change requests
- A consistent scheme of version identification should be established using version numbers, attributes or change sets
- System releases include executable code, data, configuration files and documentation