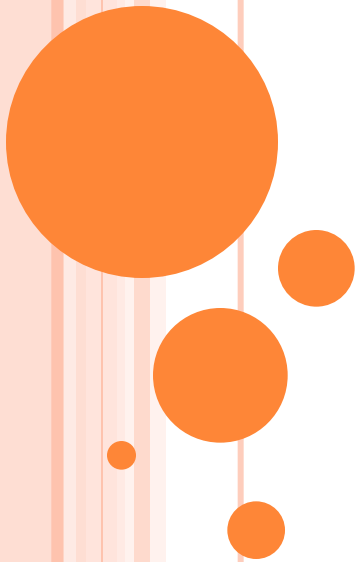
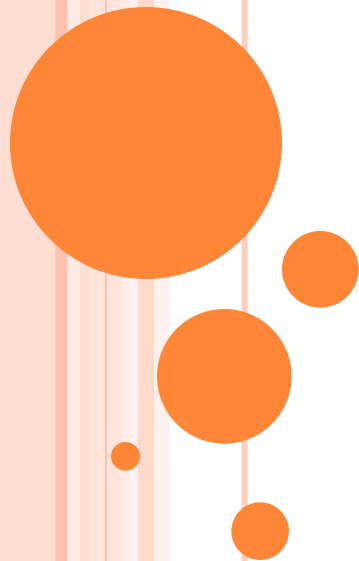


# SOFTWARE ENGINEERING



# LECTURE-23

**Configuration Management**



# TOPICS COVERED

- Software Configuration Management Basics
- Version Control
- Change Control
- Baseline Management



# WHY IS SOFTWARE DIFFICULT TO BUILD?

- Lack of control
- Lack of monitoring
- Lack of traceability
- Uncontrolled changes



# CONFLICTS IN TEAM SOFTWARE DEVELOPMENT

- Simultaneous updates – how to prevent one person from undoing the changes of another
- Shared and common code – how to notify everyone who needs to know about a change
- Versions – how to make changes to all affected versions when needed



# SOFTWARE CONFIGURATION MANAGEMENT BASICS

- Identification – identifying software configuration items in a baseline
- Control – controlling the release of a product and changes to it throughout its lifecycle
- Status Accounting – recording and reporting of the status of components and changes
- Auditing and Reviewing – Validating the completeness of a product and that SCM procedures are being followed



# SCM DEFINITIONS

- Baseline – One or more software configuration items that have been formally reviewed and agreed upon and serve as a basis for further development
- Software Configuration Item – A collection of software elements treated as a unit for the purposes of SCM
- Configuration – A collection of all the elements of a baseline and a description of how they fit together

## SCM DEFINITIONS (CONT'D)

- Configuration Control Board – Group with the responsibility for reviewing and approving changes to baselines
- Software – All of the code, specifications, plans, descriptions, processes, and documents associated with a software development effort
- Version – A specific instance of a baseline or configuration item





# EXAMPLES OF CONFIGURATION ITEMS

- Product concept specification
- Software project plans
- Software requirements specifications
- Software design descriptions
- Source code
- Database descriptions
- SCM procedures
- Software release processes
- Software test documents
- User documentation
- Maintenance documentation



# VERSION CONTROL

- Allows different projects to use the same source files at the same time
- Isolates work that is not ready to be shared by the rest of the project
- Isolates work that should never be shared
- Allows software engineers to continue development along a branch even when a line of development is frozen



# CHANGE CONTROL

- Proposed changes to baselines must have some level of review
- The impact of proposed changes must be identified and understood.
- When appropriate the approval of the CCB, key managers and project members must be obtained
- Approved changes must be properly implemented
- After changes are made all affected parties must be notified

# BASELINE MANAGEMENT

- What baselines are required to be defined and managed?
  - Typically aligned with major milestones
  - Applies to documents as well as code
- How is the current software configuration defined?
  - A snapshot of everything the product has produced at some point in time



# BASELINE MANAGEMENT (CONT'D)

- Who must approve changes to baselines?
  - Usually the Change Control Board (CCB)
- How and when are baselines created and physically controlled?
  - Through the use of document control systems, code control tools, and procedures to prevent the making of unapproved changes



# BASELINE MANAGEMENT (CONT'D)

- How are people informed of changes?
  - The CCB disseminates change information
- How are baselines verified?
  - By reviews, inspections, and the testing of code
- Are baselines tied to project milestones?
  - Many are, but during coding many may not be



# BASELINE MANAGEMENT (CONT'D)

- What information is required to process a change to a baseline?
  - A description of the proposed changes
  - Reasons for making the changes
  - List of other items affected by the changes
- What tools, resources, and training are required to perform baseline change assessment?
  - File comparison tools to identify changes
  - Resources and training depend on size and complexity of project



# BASELINE MANAGEMENT (CONT'D)

- How are unauthorized changes to source code prevented, detected, and corrected?
  - No way to prevent unauthorized changes
  - Provide software engineers with training
  - A commercial available SCM systems provide adequate protection
  - Unauthorized changes should be caught during assessment procedures






# BASELINE MANAGEMENT (CONT'D)

- What tools, resources, and training are required to perform baseline management?
  - A fully featured SCM tool
  - On large projects a separate SCM group may be needed
  - SCM training is required for all involved in the process



# WORKSPACE MANAGEMENT

- Software engineers need a consistent and reproducible workspace area (a sandbox) in which they can develop and debug their code
  - They need to be able to share project files while shielding the project from the instability of their evolving code
  - SCM tools should provide such a capability
- 

# BASELINE CHANGE ASSESSMENT

- Helps to identify recent changes that may be responsible for problems
- Helps to ensure that only authorized changes are made

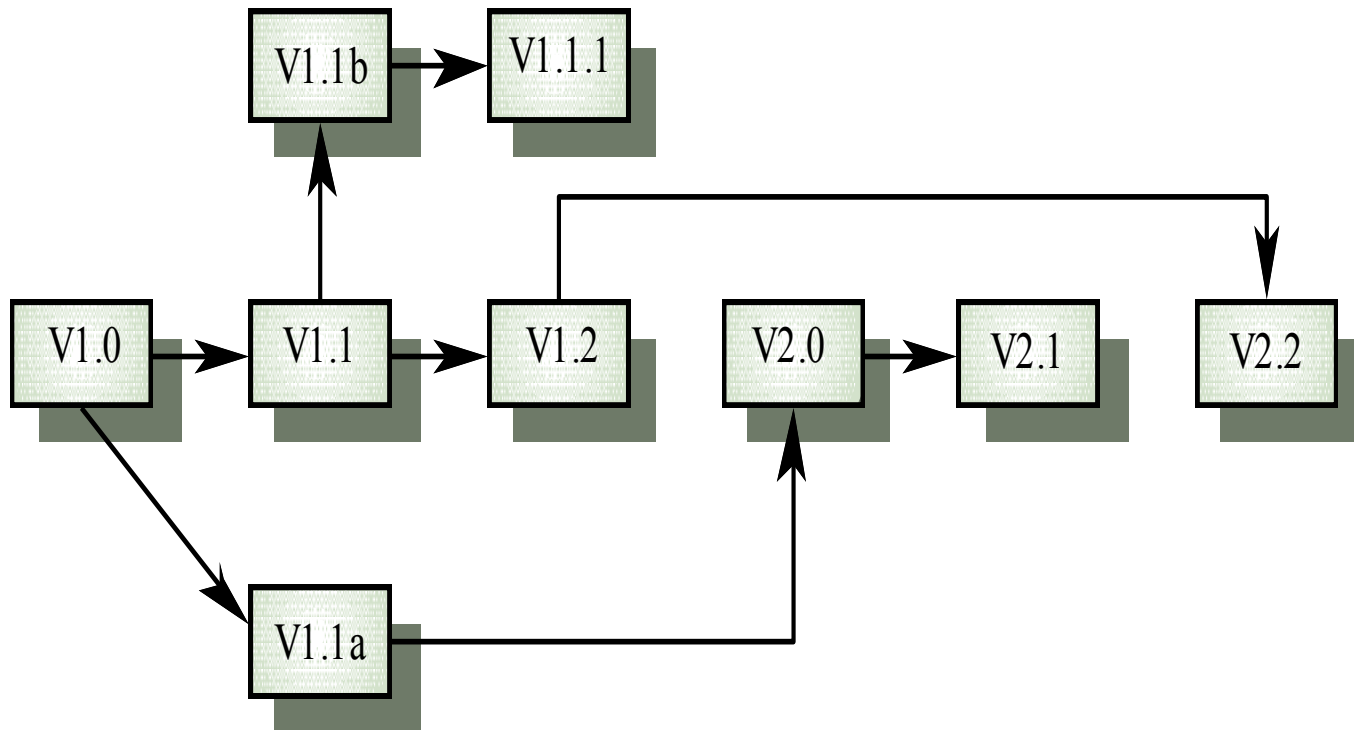


# VERSION MANAGEMENT


- Being able to reliably build and recreate versions of a product as it evolves and after it is released.
- Being able to retreat to a previous version if necessary
- Being able to recreate all versions of the product that customers have



# VERSION DERIVATION STRUCTURE



# TYPES OF AUDITS

- In-process audits – verify the consistency of the design as it evolves through the development process
  - Functional audits – verify that functionality and performance are consistent with requirements defined in the SRS
  - Physical audits – verify that the as-built version of software and documentation are internally consistent and ready for delivery
  - Quality system audits – independent assessment of the compliance to the software QA plan
- 

# CONFIGURATION STATUS ACCOUNTING REQUIREMENTS

- Identifying the types of information that project managers need
- Identifying the degree of control needed by project management
- Identifying the reports required and the different audiences for each report
- Identifying the information required to produce each report



# SCM SUMMARY

- Change is inevitable
- Defined procedures are required to manage change without preventing change
- Software presents many challenges from a control, management, and tracking perspective
- Knowing what you have and how you got there is very important
- Being able to recreate exactly what is delivered to customers is essential

