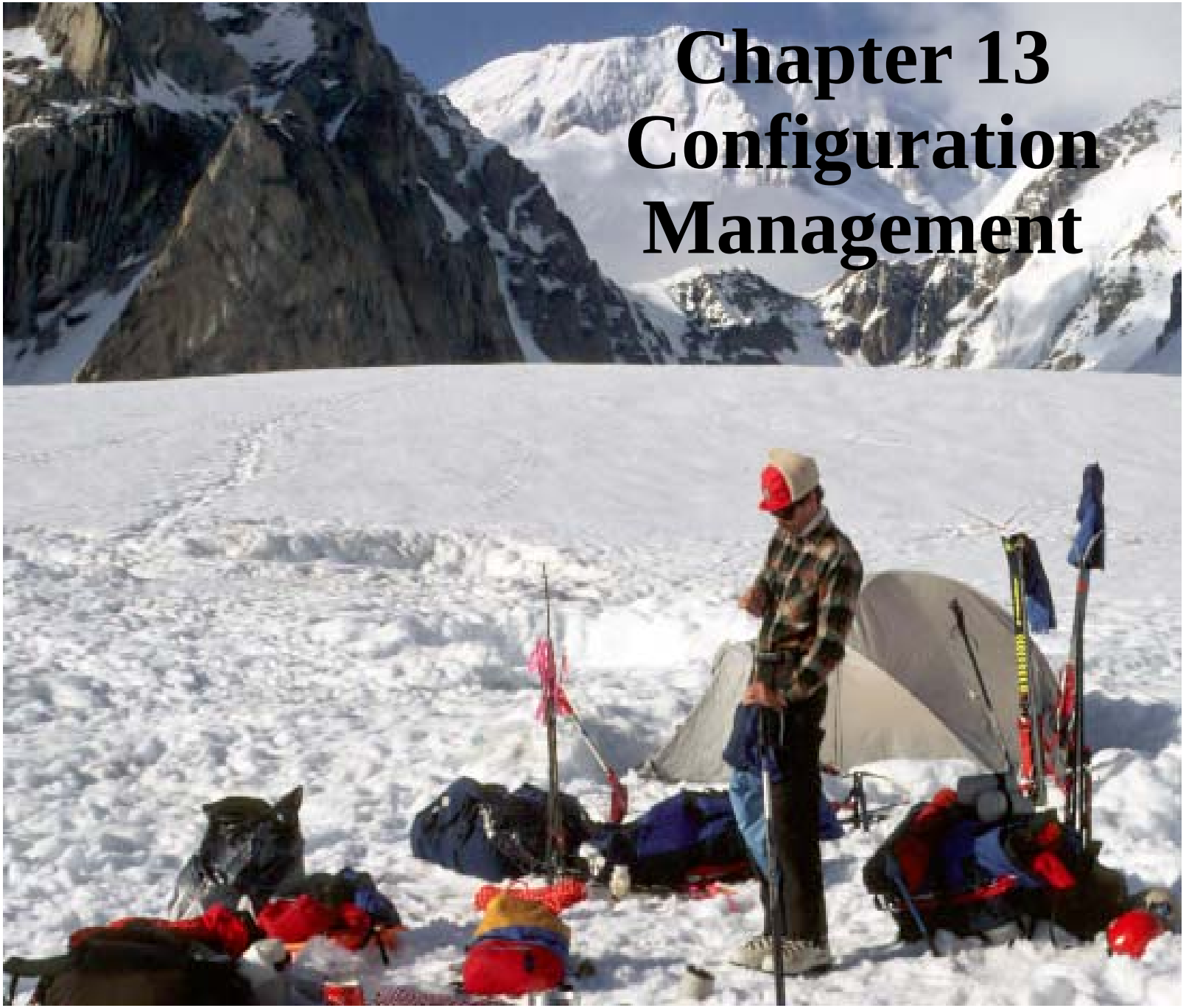
Object-Oriented Software Engineering
Using UML, Patterns, and Java

Chapter 13 Configuration Management



Outline of the Lecture

- Purpose of Software Configuration Management (SCM)
  - **Motivation: Why software configuration management?**
  - **Definition: What is software configuration management?**
  - **Activities and roles in software configuration management**
- Some Terminology
  - **Configuration Item, Baseline, SCM Directory, Version, Revision Release.**
- Software Configuration Management Activities
  - **Promotion Management, Release Management, Change Management**
- Outline of a Software Configuration Management Plans
  - **Standards (Example: IEEE 828-1990)**
  - **Basic elements of IEEE 828-1990**
- Configuration Management Tools

Why Software Configuration Management ?

- The problem:
  - **Multiple people have to work on software that is changing**
  - **More than one version of the software has to be supported:**
    - **Released systems**
    - **Custom configured systems (different functionality)**
    - **System(s) under development**
  - **Software must run on different machines and operating systems**

↙ *Need for coordination*

- Software Configuration Management
  - **manages evolving software systems**
  - **controls the costs involved in making changes to a system**

What is Software Configuration Management?

- Definition:
  - **A set of management disciplines within the software engineering process to develop a baseline.**
- Description:
  - **Software Configuration Management encompasses the disciplines and techniques of initiating, evaluating and controlling change to software products during and after the software engineering process.**
- Standards (approved by ANSI)
  - **IEEE 828: Software Configuration Management Plans**
  - **IEEE 1042: Guide to Software Configuration Management**

Software Configuration Management is a Project Function

- SCM is a Project Function (as defined in the SPMP) with the goal to make technical and managerial activities more effective.
- Software Configuration Management can be administered in several ways:
  - **A single software configuration management team for the whole organization**
  - **A separate configuration management team for each project**
  - **Software Configuration Management distributed among the project members**
  - **Mixture of all of the above**

Configuration Management Activities

- Software Configuration Management Activities:
  - **Configuration item identification**
  - **Promotion management**
  - **Release management**
  - **Branch management**
  - **Variant management**
  - **Change management**

- No fixed rules:
  - **Activities are usually performed in different ways (formally, informally) depending on the project type and life-cycle phase (research, development, maintenance).**

Configuration Management Activities (continued)

- Configuration item identification
  - **modeling of the system as a set of evolving components**
- Promotion management
  - **is the creation of versions for other developers**
- Release management
  - **is the creation of versions for the clients and users**
- Change management
  - **is the handling, approval and tracking of change requests**
- Branch management
  - **is the management of concurrent development**
- Variant management
  - **is the management of versions intended to coexist**

Configuration Management Roles

- Configuration Manager
  - **Responsible for identifying configuration items. The configuration manager can also be responsible for defining the procedures for creating promotions and releases**
- Change control board member
  - **Responsible for approving or rejecting change requests**
- Developer
  - **Creates promotions triggered by change requests or the normal activities of development. The developer checks in changes and resolves conflicts**
- Auditor
  - **Responsible for the selection and evaluation of promotions for release and for ensuring the consistency and completeness of this release**

Terminology

- We will define the following terms
  - **Configuration Item**
  - **Baseline**
  - **SCM Directories**
  - **Version**
  - **Revision**
  - **Release**

- The definition of the terms follows the IEEE standard.
- Different configuration management systems may use different terms.
  - **Example: CVS configuration management system used in our projects uses terms differeing from the IEEE standard.**

Terminology: Configuration Item

"An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process."

- Software configuration items are not only program code segments but all type of documents according to development, e.g
  - **all type of code files**
  - **drivers for tests**
  - **analysis or design documents**
  - **user or developer manuals**
  - **system configurations (e.g. version of compiler used)**
- In some systems, not only software but also hardware configuration items (CPUs, bus speed frequencies) exist!

Finding Configuration Items

- Large projects typically produce thousands of entities (files, documents, data ...) which must be uniquely identified.
- Any entity managed in the software engineering process can potentially be brought under configuration management control
- But not every entity needs to be under configuration management control all the time.
- Two Issues:
  - **What: Selection of Configuration Items**
    - **What should be under configuration control?**
  - **When: When do you start to place entities under configuration control?**
- Conflict for the Project Manager:
  - **Starting with CIs too early introduces too much bureaucracy**
  - **Starting with CIs too late introduces chaos**

Finding Configuration Items (continued)

- Some items must be maintained for the lifetime of the software. This includes also the phase, when the software is no longer developed but still in use; perhaps by industrial customers who are expecting proper support for lots of years.
- An entity naming scheme should be defined so that related documents have related names.

- Selecting the right configuration items is a skill that takes practice
  - **Very similar to object modeling**
  - **Use techniques similar to object modeling for finding CIs!**
    - **Find the CIs**
    - **Find relationships between CIs**

Which of these Entities should be Configuration Items?

- Problem Statement
- Software Project Management Plan (SPMP)
- Requirements Analysis Document (RAD)
- System Design Document (SDD)
- Project Agreement
- Object Design Document (ODD)
- Dynamic Model
- Object model
- Functional Model
- Unit tests
- Integration test strategy
- Source code
- API Specification
- Input data and data bases
- Test plan
- Test data
- Support software (part of the product)
- Support software (not part of the product)
- User manual
- Administrator manual

Possible Selection of Configuration Items

- Problem Statement
- Software Project Management Plan (SPMP)
- Requirements Analysis Document (RAD)
- System Design Document (SDD)
- Project Agreement
- Object Design Document (ODD)
- Dynamic Model
- Object model
- Functional Model
- Unit tests
- Integration test strategy
- Source code
- API Specification
- Input data and data bases
- Test plan
- Test data
- Support software (part of the product)
- Support software (not part of the product)
- User manual
- Administrator manual

Once the Configuration Items are selected, they are usually organized in a tree

Terminology: Version

- The initial release or re-release of a configuration item associated with a complete compilation or recompilation of the item. Different versions have different functionality.

Terminology: Baseline

"*A specification or product that has been formally reviewed and agreed to by responsible management, that thereafter serves as the basis for further development, and can be changed only through formal change control procedures*."

Examples:

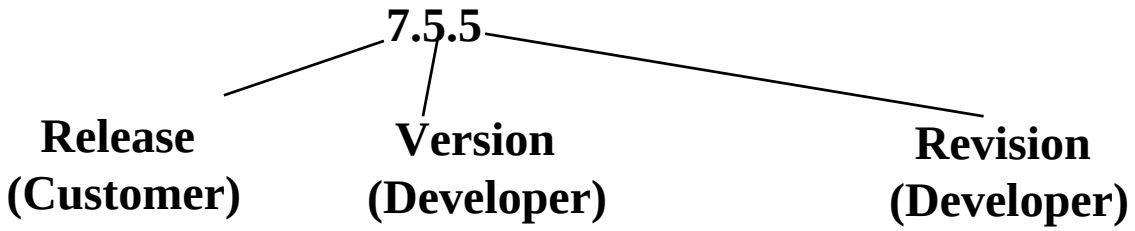
Baseline A: All the API have completely been defined; the bodies of the methods are empty**.**

Baseline B: All data access methods are implemented and tested**.**

Baseline C: The GUI is implemented.

More on Baselines

- As systems are developed, a series of baselines is developed, usually after a review (analysis review, design review, code review, system testing, client acceptance, ...)
  - ***Developmental baseline*** **(RAD, SDD, Integration Test, ...)**
    - **Goal: Coordinate engineering activities.**
  - ***Functional baseline*** **(first prototype, alpha release, beta release)**
    - **Goal: Get first customer experiences with functional system.**
  - ***Product baseline*** **(product)**
    - **Goal: Coordinate sales and customer support.**
- Many naming scheme for baselines exist (1.0, 6.01a, ...)
- A 3 digit scheme is quite common:

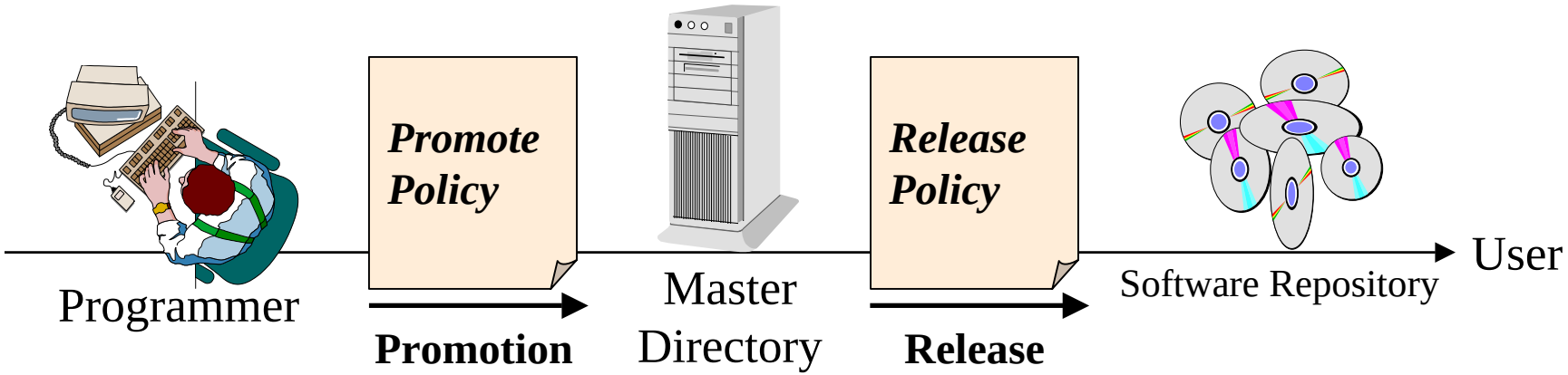


Change management

- Change management is the handling of change requests
  - **A change request leads to the creation of a new release**
- General change process
  - **The change is requested (this can be done by anyone including users and developers)**
  - **The change request is assessed against project goals**
  - **Following the assessment, the change is accepted or rejected**
  - **If it is accepted, the change is assigned to a developer and implemented**
  - **The implemented change is audited.**
- The complexity of the change management process varies with the project. Small projects can perform change requests informally and fast while complex projects require detailed change request forms and the official approval by one more managers.

Controlling Changes

- Two types of controlling change:
  - ***Promotion:*** **The internal development state of a software is changed.**
  - ***Release:*** **A changed software system is made visible outside the development organization.**



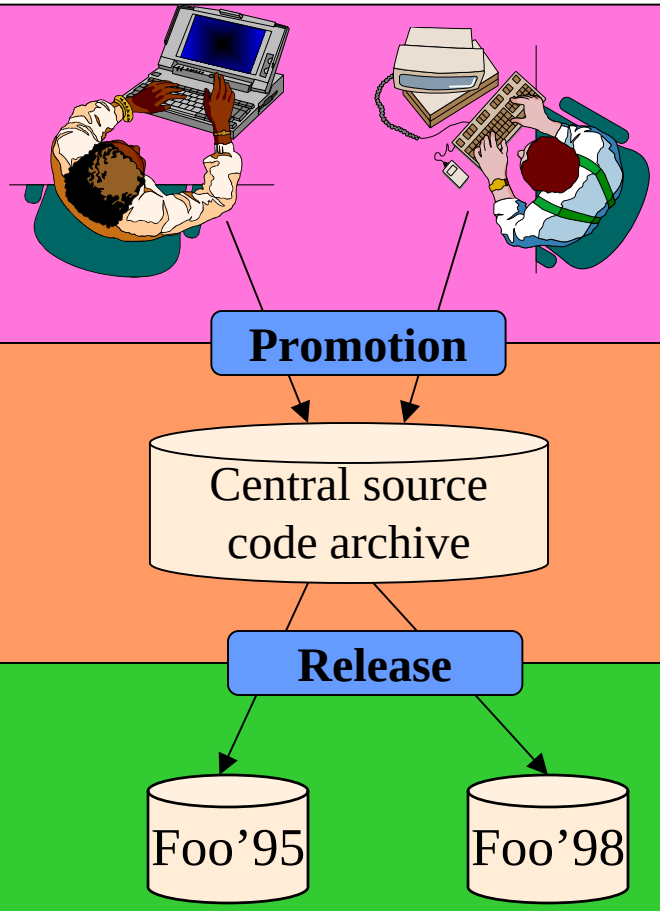
- Approaches for controlling change (Change Policy)
  - **Informal (good for research type environments and promotions)**
  - **Formal approach (good for externally developed CIs and for releases)**

Terminology: SCM Directories

- Programmer's Directory (IEEE: Dynamic Library)
  - **Library for holding newly created or modified software entities.**
  - **The programmer's workspace is controlled by the programmer only.**
- Master Directory (IEEE: Controlled Library)
  - **Manages the current baseline(s) and for controlling changes made to them.**
  - **Entry is controlled, usually after verification.**
  - **Changes must be authorized.**
- Software Repository (IEEE: Static Library)
  - **Archive for the various baselines released for general use.**
  - **Copies of these baselines may be made available to requesting organizations.**

Standard SCM Directories

- Programmer's Directory
  - **(IEEE Std: "Dynamic Library")**
  - **Completely under control of one programmer.**
- Master Directory
  - **(IEEE Std: "Controlled Library")**
  - **Central directory of all promotions.**
- Software Repository
  - **(IEEE Std: "Static Library")**
  - **Externally released baselines.**



Change Policies

- Whenever a promotion or a release is performed, one or more policies apply. The purpose of change policies is to guarantee that each version, revision or release (see next slide) conforms to commonly accepted criteria.

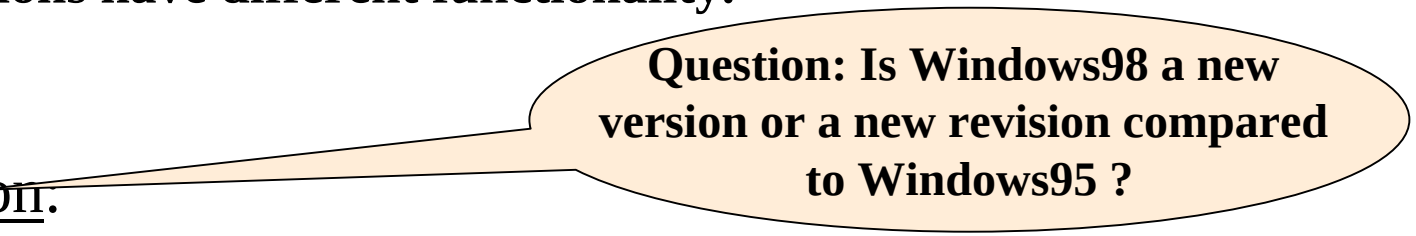
- Examples for change policies:

  “No developer is allowed to promote source code which cannot be compiled without errors and warnings.”

  “No baseline can be released without having been beta-tested by at least 500 external persons.”

Terminology: Version vs. Revision vs. Release

- Version:
  - **An *initial* release or re-release of a configuration item associated with a *complete compilation* or recompilation of the item. Different versions have different functionality.**



- Revision:
  - ***Change* to a version that corrects only errors in the design/code, but does not affect the documented functionality.**

- Release:
  - **The *formal distribution* of an approved version.**

Software Configuration Management Planning

- Software configuration management planning starts during the early phases of a project.

- The outcome of the SCM planning phase is the

  Software Configuration Management Plan (SCMP)

  which might be extended or revised during the rest of the project.

- The SCMP can either follow a public standard like the IEEE 828, or an internal (e.g. company specific) standard.

The Software Configuration Management Plan

- Defines the *types of documents* to be managed and a document naming scheme.
- Defines *who takes responsibility* for the CM procedures and creation of baselines.
- Defines *policies for change* control and version management.
- Describes the *tools* which should be used to assist the CM process and any limitations on their use.
- Defines the *configuration management database* used to record configuration information.

Tools for Software Configuration Management

- Software configuration management is normally supported by tools with different functionality.
- Examples:
  - **RCS**
    - **very old but still in use; only version control system**
  - **CVS (Concurrent Version Control)**
    - **based on RCS, allows concurrent working without locking**
    - **http://www.cvshome.org/**
    - **CVSWeb: Web Frontend to CVS**
  - **Perforce**
    - **Repository server; keeps track of developer's activities**
    - **http://www.perforce.com**
  - **ClearCase**
    - **Multiple servers, process modeling, policy check mechanisms**
    - **http://www.rational.com/products/clearcase/**

Summary

- Software Configuration Management: Important part of project management to manage evolving software systems and coordinate changes to them.
- Software Configuration Management consists of several activities:
  - **Promotion and Release management (Covered today)**
  - **Branch, Variant and Change Management ([Bruegge-Dutoit])**
- Public standard for SCM plans: IEEE 828.
- The standard can be tailored to a particular project:
  - **Large projects need detailed plans to be successful**
  - **Small projects should not be burdened with the bureaucracy of detailed SCM plans**
- SCM should be supported by tools. These range from
  - **Simple version storage tools**
  - **Sophisticated systems with automated procedures for policy checks and support for the creation of SCM documents.**