Object-Oriented Software Engineering

Using UML, Patterns, and Java

Configuration & Build Management

Outline of the Lecture

- Purpose of Software Configuration Management (SCM)
- Some Terminology
- Software Configuration Management Activities
- Outline of a Software Configuration Management Plan
- Build and Release Management
- Continous Integration

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

Administering Software Configuration Management

- Software Configuration Management is a project function with the goal to make technical and managerial activities more effective
- Software Configuration Management can be administered in several ways:
  - Organization-wide
  - Project-specific
  - Distributed among the project members
  - Mixture of all of the above.

Configuration Management Activities (1)

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

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

Configuration Management Activities (2)

- Configuration item identification
  - Modeling the system as a *set of evolving components*
- Promotion management
  - the creation of *versions for other developers*
- Release management
  - the creation of *versions for clients and users*
- Change management
  - the handling, approval & *tracking of change requests*

This Lecture

- Branch management
  - the management of *concurrent development*
- Variant management
  - the management of *coexisting versions*

Bruegge-Dutoit Ch13,p.551ff

Configuration Management Roles

- Configuration Manager
  - Responsible for identifying configuration items
  - Also often 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 definitions for these terms follow the IEEE standard.

Terminology: Configuration Item

Configuration Item: An aggregation of hardware, software, or both, designated for configuration management and treated as a single entity in the configuration management process.

- Software configuration items are not only source files but all types of documents
- In some projects, not only software but also hardware configuration items (CPUs, bus speed frequencies) need to be put under control!

Tasks for the Configuration Managers

Define configuration items

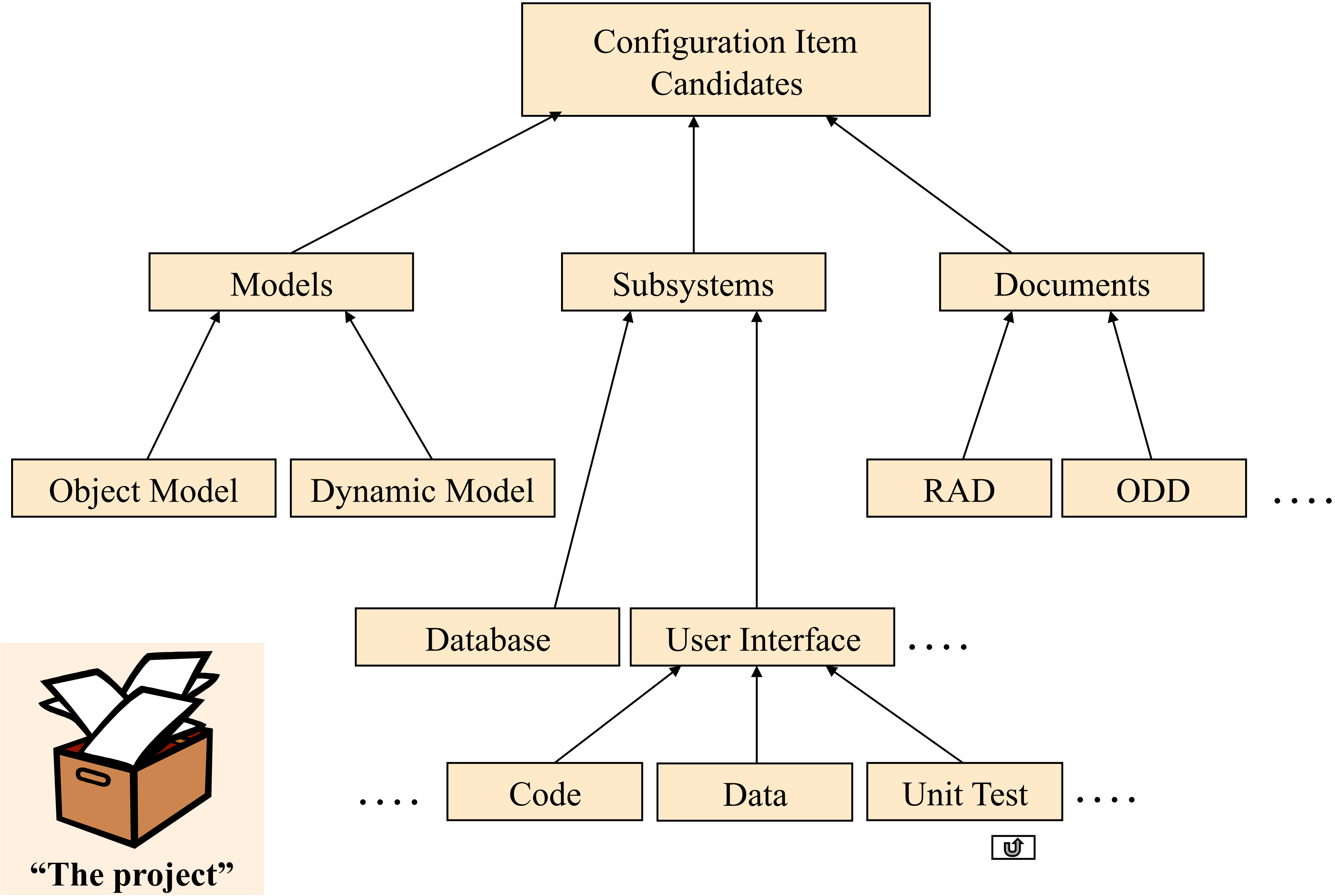
Define Configuration Items

- 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?
- Choices for the Project Manager:
  - Starting with Configuration Items too early introduces bureaucracy
  - Starting with Configuration Items too late introduces chaos.

Define Configuration Items (continued)

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

Configuration Item Candidates



Configuration Item Candidates

Models

Subsystems

Documents

Object Model

Dynamic Model

RAD

ODD

. . . .

Database

User Interface

. . . .

. . . .

Code

Data

Unit Test

. . . .

“The project”

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

Terminology: Version

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.

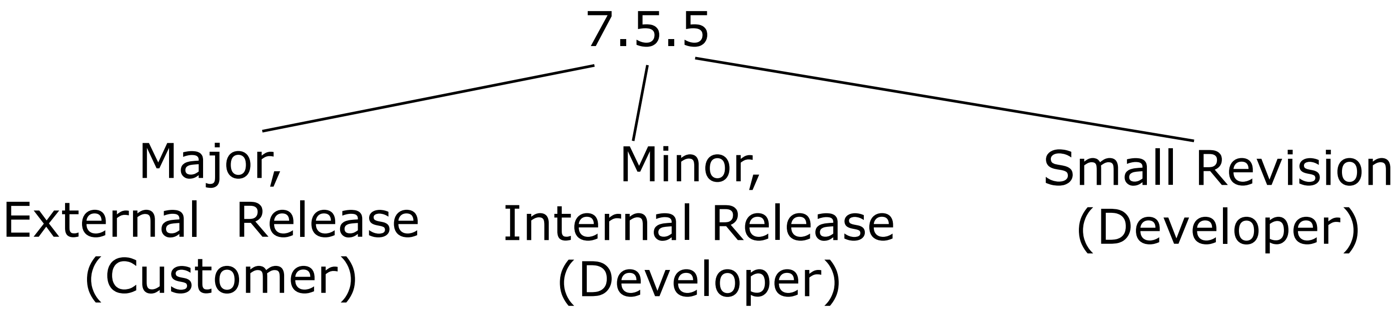
Terminology: Baseline

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:* The API has been 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.

Naming Schemes for Baselines

- Many naming scheme for baselines exist (1.0, 6.01a, ...)
- A 3 digit scheme is quite common:



7.5.5

Major,
External Release
(Customer)

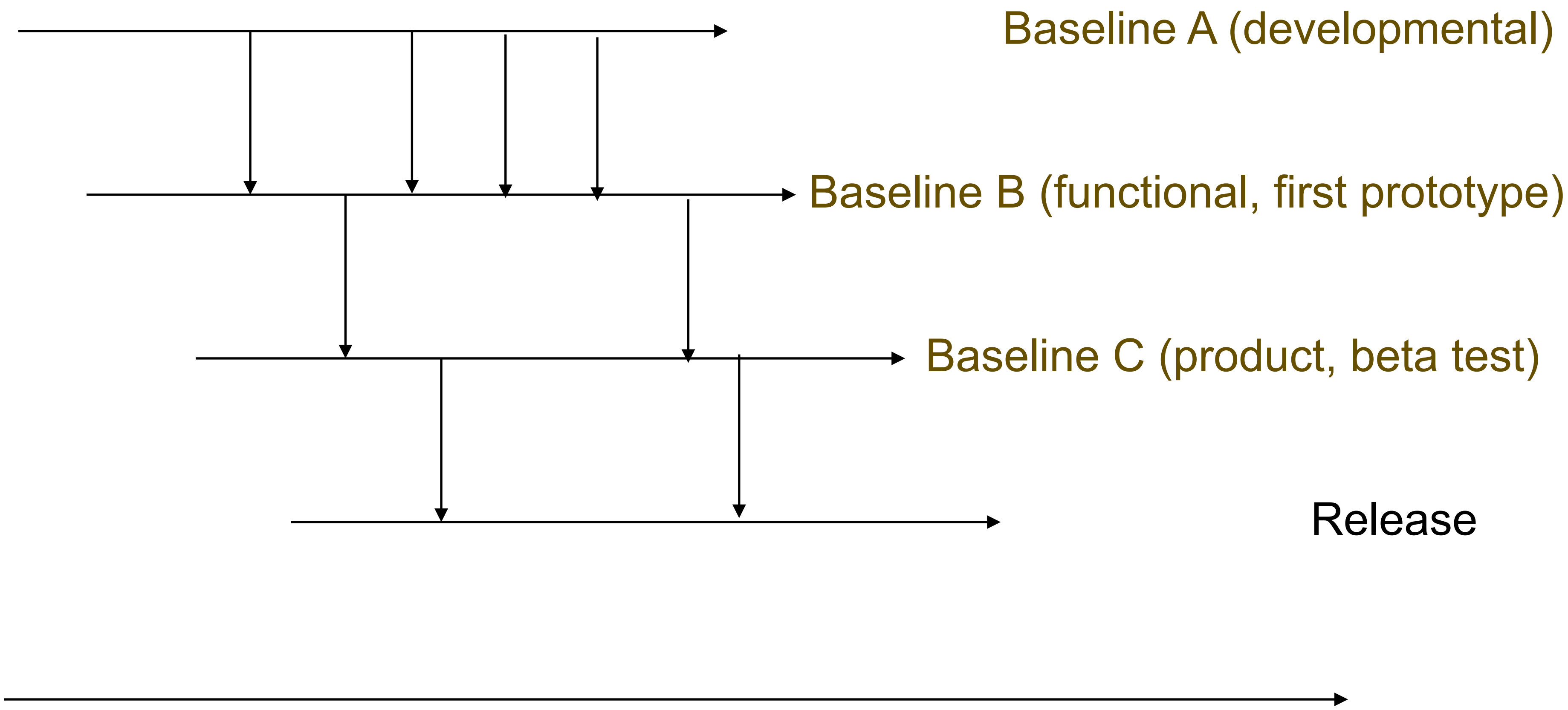
Minor,
Internal Release
(Developer)

Small Revision
(Developer)

Types of 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
  - Functional baseline
  - Product baseline

Transitions between Baselines


Baseline A (developmental)
Baseline B (functional, first prototype)
Baseline C (product, beta test)
Release

How do we manage changes in baselines?

=> Change Management

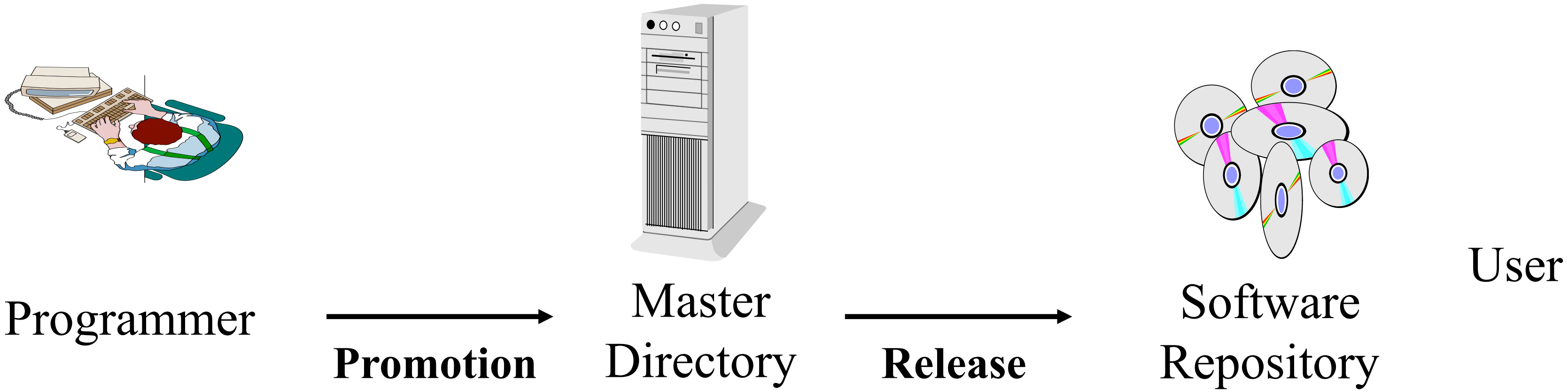
Time

Change management

- Change management is the handling of change requests
- The general change management process:
  - The change is requested
  - The change request is assessed against requirements and project constraints
  - Following the assessment, the change request is accepted or rejected
  - If it is accepted, the change is assigned to a developer and implemented
  - The implemented change is audited.

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.

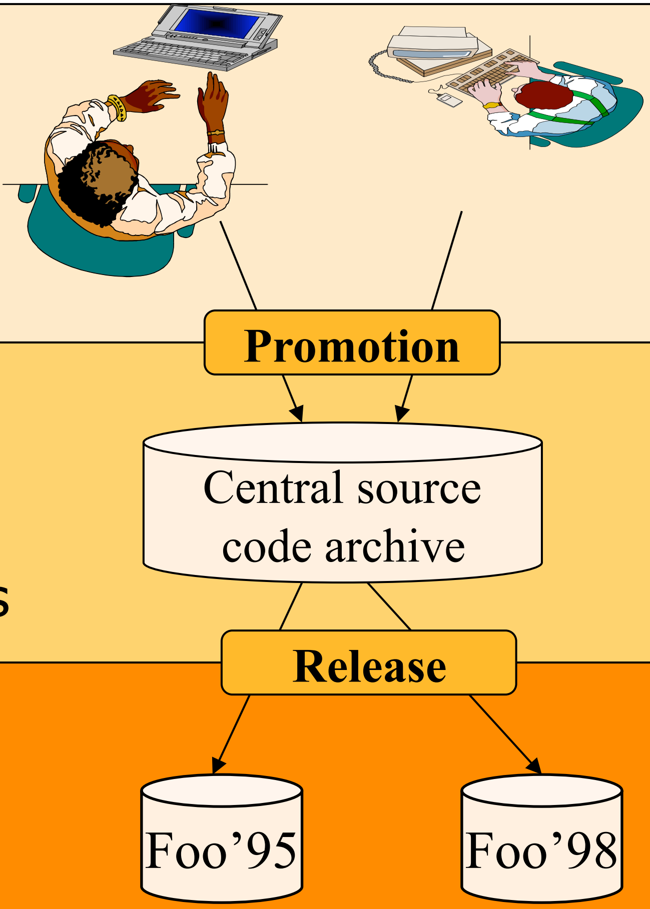


Terminology: SCM Directories

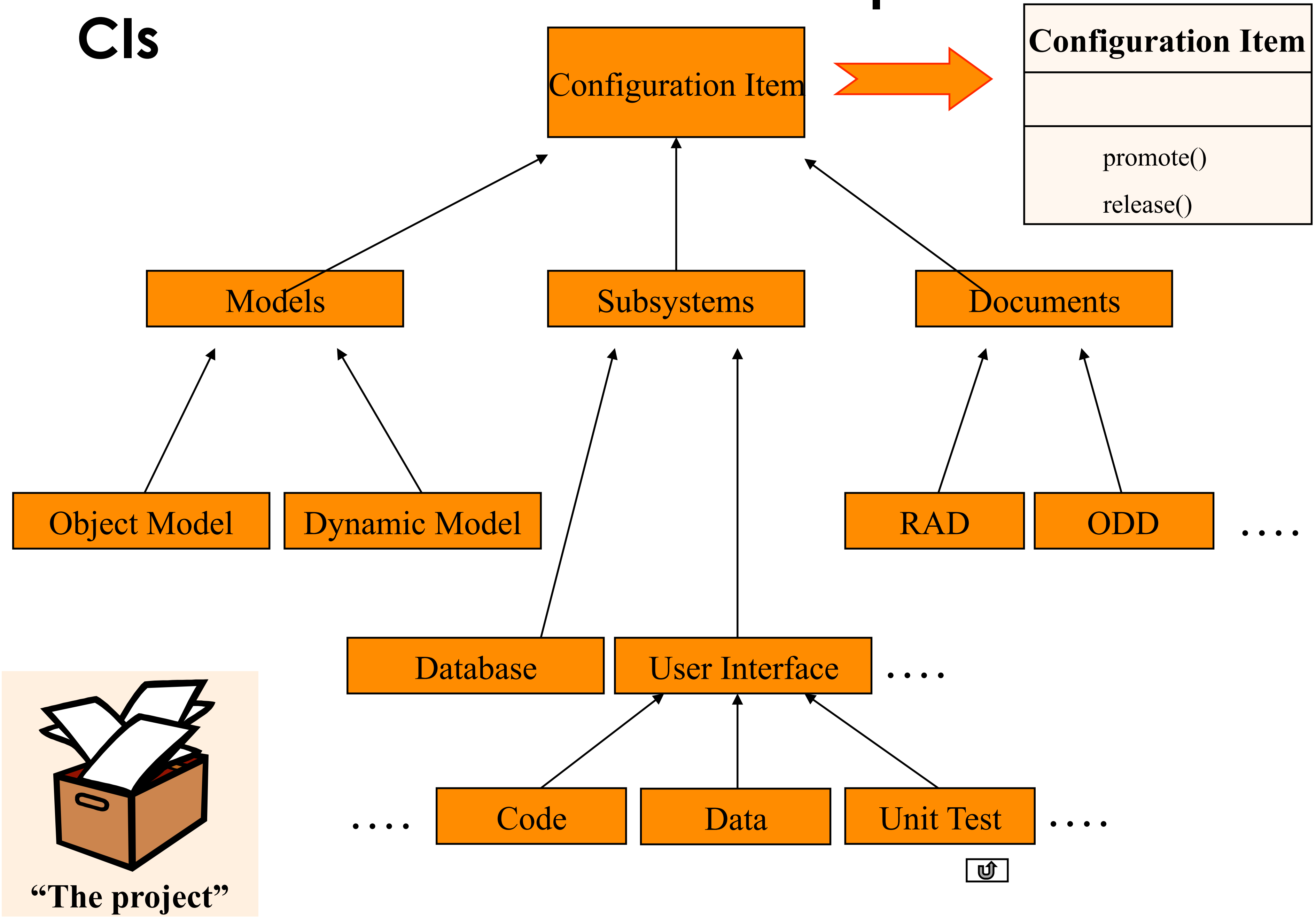
- Programmer's Directory (IEEE: Dynamic Library)
  - Library for holding newly created or modified software entities
  - The programmer's workspace is controlled by the programmer only
- Master Directory (IEEE: Controlled Library)
  - Manages the current baseline(s) and for controlling changes made to them
  - 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.



Promotion and Release are Operations on CIs



Configuration Item

Configuration Item

promote()

release()

Models

Subsystems

Documents

Object Model

Dynamic Model

RAD

ODD

. . . .

Database

User Interface

. . . .

. . . .

Code

Data

Unit Test

. . . .

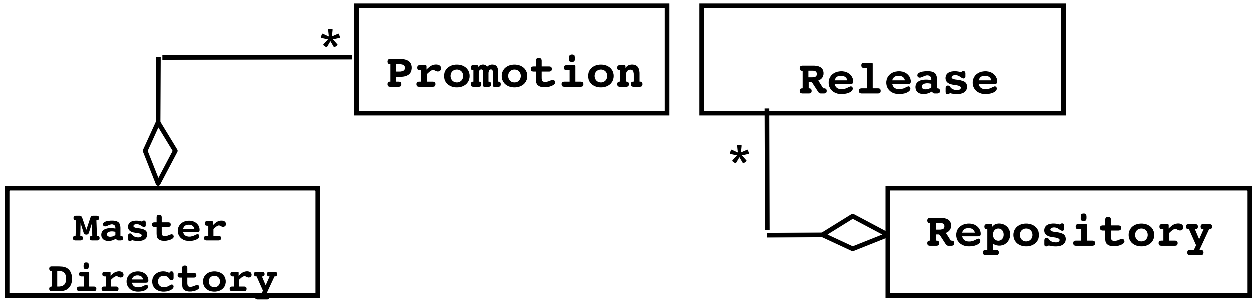
“The project”

Let‘s Create an Object Model for Configuration Management

„Promotions are stored in the master directory and releases are stored in the repository“

Problem:There can be many promotions and many releases

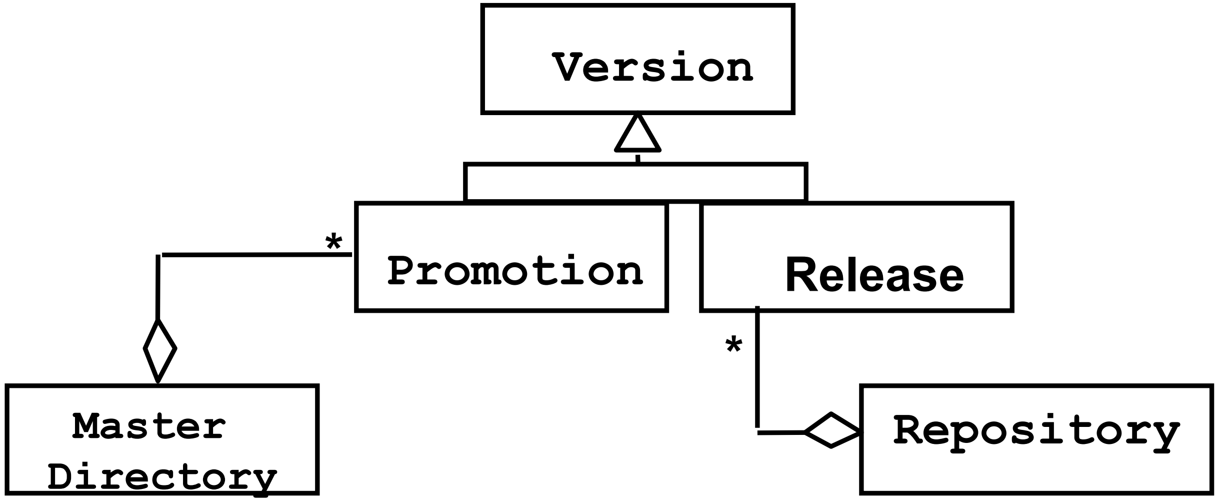
Solution: Use Multiplicity



Let‘s Create an Object Model for Configuration Management

Insight: Promotions and Releases are both versions

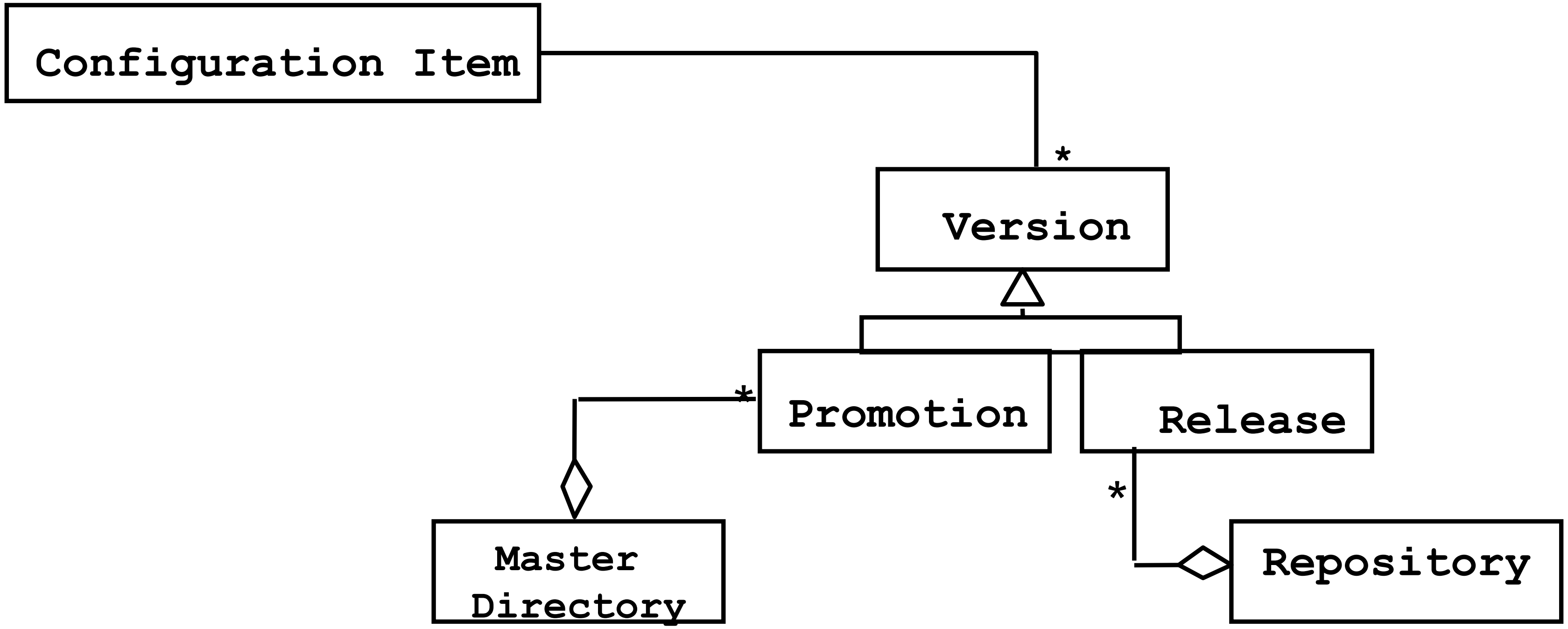
Solution: Use Inheritance



Let‘s Create an Object Model for Configuration Management

Problem: A configuration item can have several versions

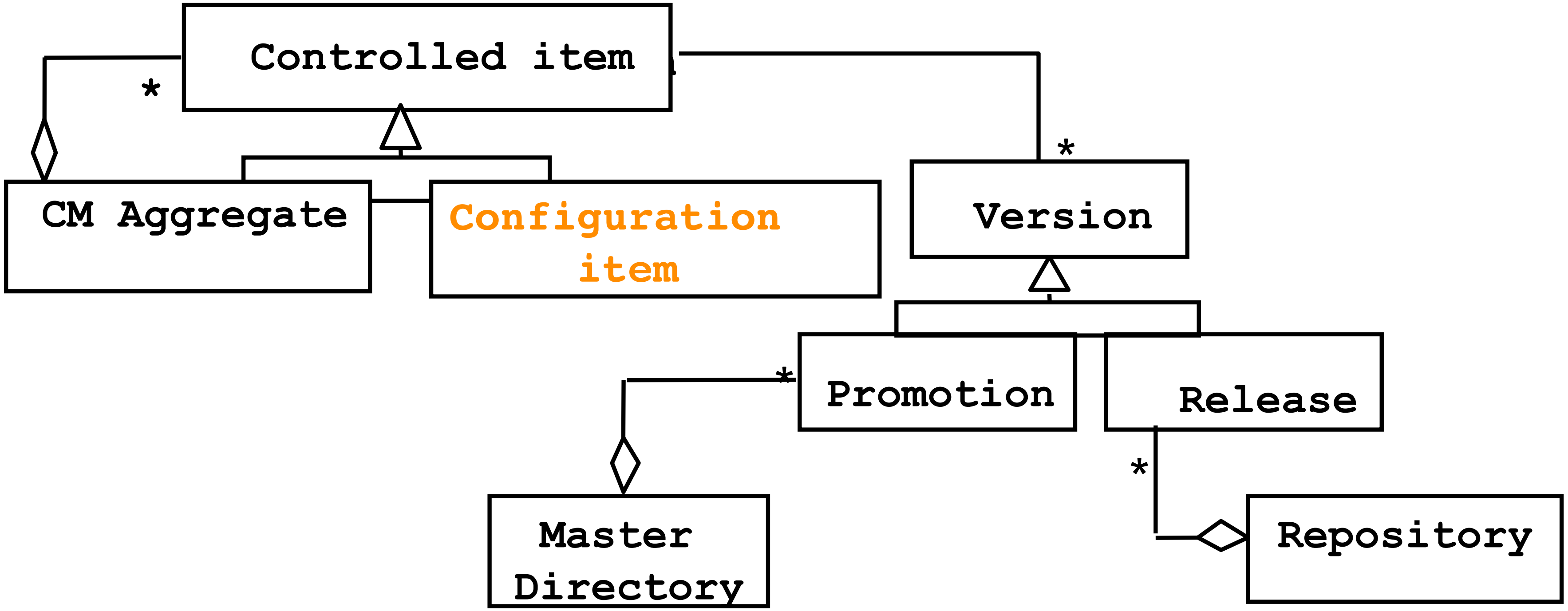
Solution: Create a 1-many association between Configuration Item and Version



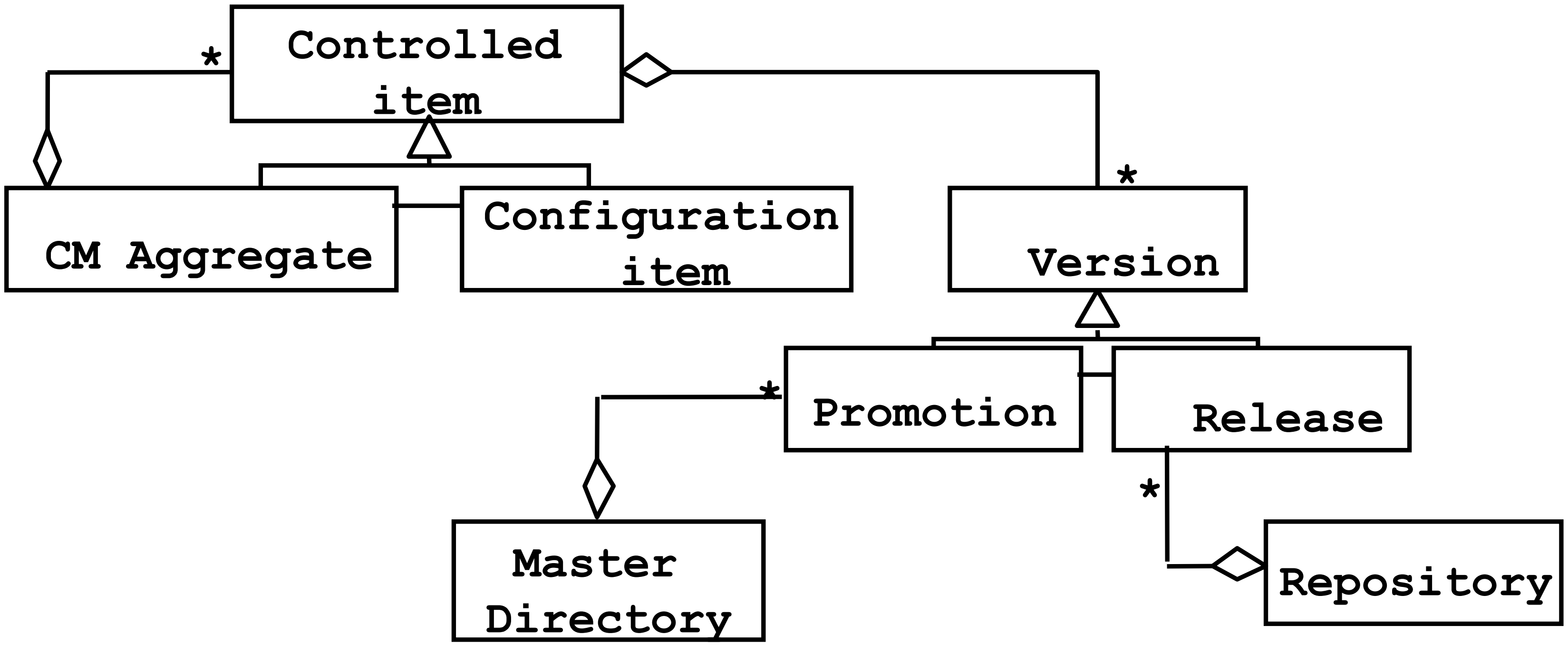
Let's Create an Object Model for Configuration Management

Problem: Configuration items can themselves be grouped

Solution: Use the composite design pattern



Final Object Model for Configuration Management (UML Class Diagram)



Software Configuration Management Tools

- RCS: The first on the block [Tichy 1975]
- CVS (Concurrent Version Control)
  - based on RCS, allows concurrent working without locking
  - http://www.cvshome.org/
- 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/
- Subversion
  - See next slide.

Subversion

- Open Source Project (http://subversion.tigris.org/)
- Based on CVS
  - Subversion interface and features similar to CVS
  - Commands: `checkout, add, delete, commit, diff`
- Differences to CVS
  - Version controlled moving, renaming and copying of files and directories
  - Version controlled metadata of files and directories
- Server Options
  - Standalone installation
  - Integrated into the Apache webserver
- The time for branch management is independent of the size of the system (unlike CVS, which creates physical copies of the files, Subversion uses only tags)

Tasks for Configuration Managers

Define configuration items

Define promote /release policies

Change Policies

- The purpose of change policies is to guarantee that each promotion or release conforms to commonly accepted criteria.

- Examples for change policies:

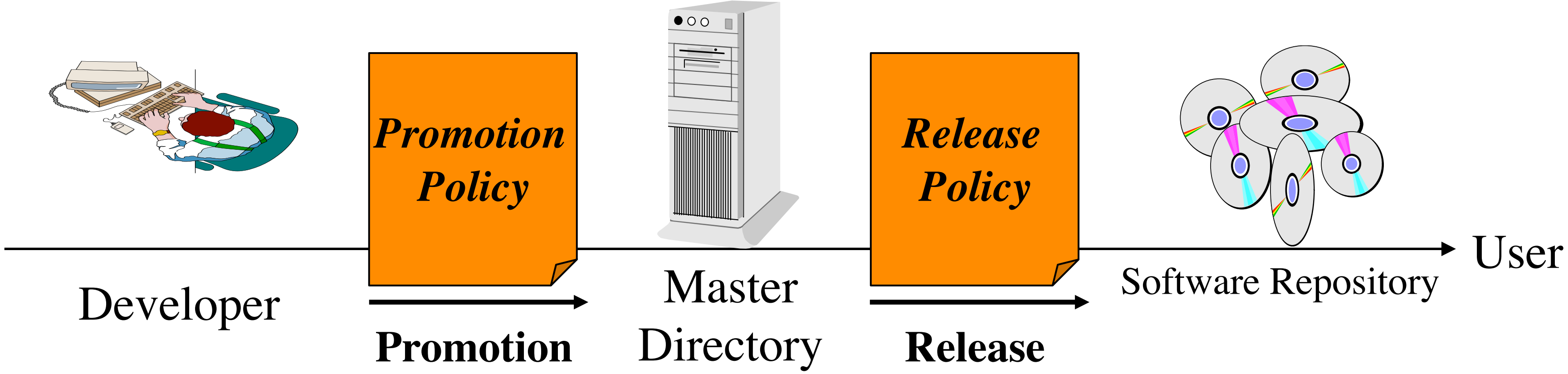
  “No developer is allowed to promote source code which cannot be compiled without errors and warnings.”

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

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: Version vs. Revision

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

Tasks for Configuration Managers

Define configuration items

Define promote /release policies

Define activities and responsibilities

Define Activities and Responsibilities

Configuration Item Identification

Configuration Control

Configuration Status Accounting

Configuration Audits and Reviews

Interface Control

Configuration Control

- Defines the following steps
  - How to identify the need for a change (layout of change request form)
  - Analysis and evaluation of a change request
  - Approval or disapproval of a request
  - Verification, implementation and release of a change

Change Request

- Specifies the procedures for requesting a change to a baselined configuration item and the information to be documented:
  - Name(s) and version(s) of the configuration item(s) where the problem appears
  - Originator’s name and address
  - Date of request
  - Indication of urgency
  - The need for the change
  - Description of the requested change

Evaluation of a Change

- Specifies the analysis required to determine the impact of proposed changes and the procedure for reviewing the results of the analysis.

Change Approval or Disapproval

- Describes the organization of the configuration control board (CCB)
  - Can be an individual or a group
  - Multiple levels of CCBs are also possible, depending on
  - In small development efforts one CCB level is sufficient
- Also indicates the level of authority of the CCB and its responsibility
  - In particular, it must be specified when the CCB is invoked.

Implementing Change

- Specifies the activities for verifying and implementing an approved change
- A completed change request must contain this information:
  - The original change request(s)
  - The names and versions of the affected configuration items
  - Verification date and responsible party
  - Identifier of the new version
  - Release or installation date and responsible party.

Implementing Change (cont'd)

- Specifies also activities for
  - Archiving completed change requests
  - Planning and control of releases
  - How to coordinate multiple changes
  - How to add new configuration items to the configuration
  - How to deliver a new baseline.

Configuration Status Accounting

- Answers the following questions:
  - What elements are to be tracked and reported for baselines and changes?
  - What types of status accounting reports are to be generated? What is their frequency?
  - How is information to be collected, stored and reported?
  - How is access to the configuration management status data controlled?

Configuration Audits and Reviews

- Identifies audits and reviews for the project
  - An audit determines for each configuration item if it has the required physical and functional characteristics
  - A review is a management tool for establishing a baseline.

Configuration Audits and Reviews (cont’d)

- For each audit or review the plan has to define:
  - Objectives
  - The Configuration Items under review
  - The schedule for the review
  - Procedures for conducting the review
  - Participants by job title
  - Required documentation
  - Procedure for recording deficiencies and how to correct them
  - Approval criteria.

Tasks for Configuration Managers

Write the SCMP

Define configuration items

Define promote /release policies

Define activities and responsibilities

Outline of the Lecture

✓ Purpose of Software Configuration Management (SCM)

✓ Some Terminology

✓ Software Configuration Management Activities

- Outline of a Software Configuration Management Plan
- Build and Release Management
- Continous Integration

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 configuration management procedures and creation of baselines
- Defines *policies for change* control and version management
- Describes the *tools* which should be used to assist the configuration management process and any limitations on their use
- Defines the *configuration management database* used to record configuration information.

Outline of a Software Configuration Management Plan (SCMP, IEEE 828-2005)

1. Introduction
   - Describes the Plan’s purpose, scope of application, key terms, and references
2. SCM management (WHO?)
   - Identifies the responsibilities and authorities for managing and accomplishing the planned SCM activities
3. SCM activities (WHAT?)
   - Identifies all activities to be performed in applying to the project
4. SCM schedule (WHEN?)
   - Establishes required coordination of SCM activities with other activities in the project
5. SCM resources (HOW?)
   - Identifies tools and physical and human resources required for the execution of the Plan
6. SCM plan maintenance
   - Identifies how the Plan will be kept current while in effect

SCMP Section 1: Introduction

- Overview description of the software project
- Identification of the software CI(s) to which SCM will be applied
- Identification of other software to be included as part of the Plan (e.g., support or test software)
- Relationship of SCM to the hardware or system configuration management activities for the project
- The degree of formality, depth of control, and portion of the software life cycle for applying SCM on this project
- Limitations, such as time constraints, that apply to the Plan
- Assumptions that might have an impact on the cost, schedule, or ability to perform defined SCM activities (e.g., assumptions of the degree of customer participation in SCM activities or the availability of automated aids).

SCMP Section 2: SCM management

2.1 Organization

- Organizational context (technical and managerial) within which the configuration management activities are implemented

2.2. Responsibilities

- List name or job title of people how perform activities
- For each board, list
    - purpose and objectives
    - membership and affiliations
    - period of effectivity
    - scope of authority
    - operational procedures

2.3. Applicable policies, directives and procedures:

- External constraints placed on the SCMP.

SCMP Section 3: SCM activities (1/2)

3.1 Configuration identification

- Identify configuration items (events, items, procedures)
- Name configuration items (unique identifiers)
- Acquiring configuration items (physical procedures)

3.2. Configuration control

- Requesting changes
- Evaluating changes
- Approving or disapproving changes
- Implementing changes

3.3. Configuration status accounting

- Metrics to be tracked and reported and type of report
- Storage and access control of status data

SCMP Section 3: SCM activities (2/2)

3.4. Configuration evaluation and reviews

- At minimum an audit on a CI prior to its release
- Defines objective, schedule, procedures, participants, approval criteria etc.

3.5. Interface control

- Coordination of changes to CIs with changes to interfacing items outside of the scope of the Plan

3.6. Subcontractor/vendor control

- Incorporation of items developed outside the project environment into the project CIs

3.7. Release Management and Delivery

- Description of the formal control of build, release and delivery of software products

SCMP Section 4: SCM schedules

- Sequence and coordination of SCM activities
- Relationship of key SCM activities to project milestones or events, such as
  - Establishment of configuration baseline
  - Implementation of change control procedures
  - Start and completion dates for a configuration audit
- Schedule either as absolute dates, relative to SCM or project milestones or as sequence of events
- Graphical representations can be used here

SCMP Section 5: SCM resources

- Identifies environment, infrastructure, software tools, techniques, equipment, personnel, and training
- Key factors for infrastructure: functionality, performance, safety, security, availability, space requirements, equipment, costs, and time constraints
- Identify which tools etc. are used in which activity

SCMP Section 6: SCM plan maintenance

- This section answers the following questions
  - Who is responsible for monitoring the Plan
  - How frequently updates are to be performed
  - How changes to the Plan are to be evaluated and approved
  - How changes to the Plan are to be made and communicated
- Also includes history of changes made to the plan

Tailoring the SCMP

- The IEEE standard allows quite a bit flexibility for preparing an SCMP
- The SCMP may be
  - tailored upward:
    - to add information
    - to use a specific format
  - tailored downward
    - Some SCMP components might not apply to a particular project.
- Always state the reasons for diverting from the standard in the Introduction
- It is not possible to omit any of the six major classes of information

Conformance to IEEE Standard 828-2005

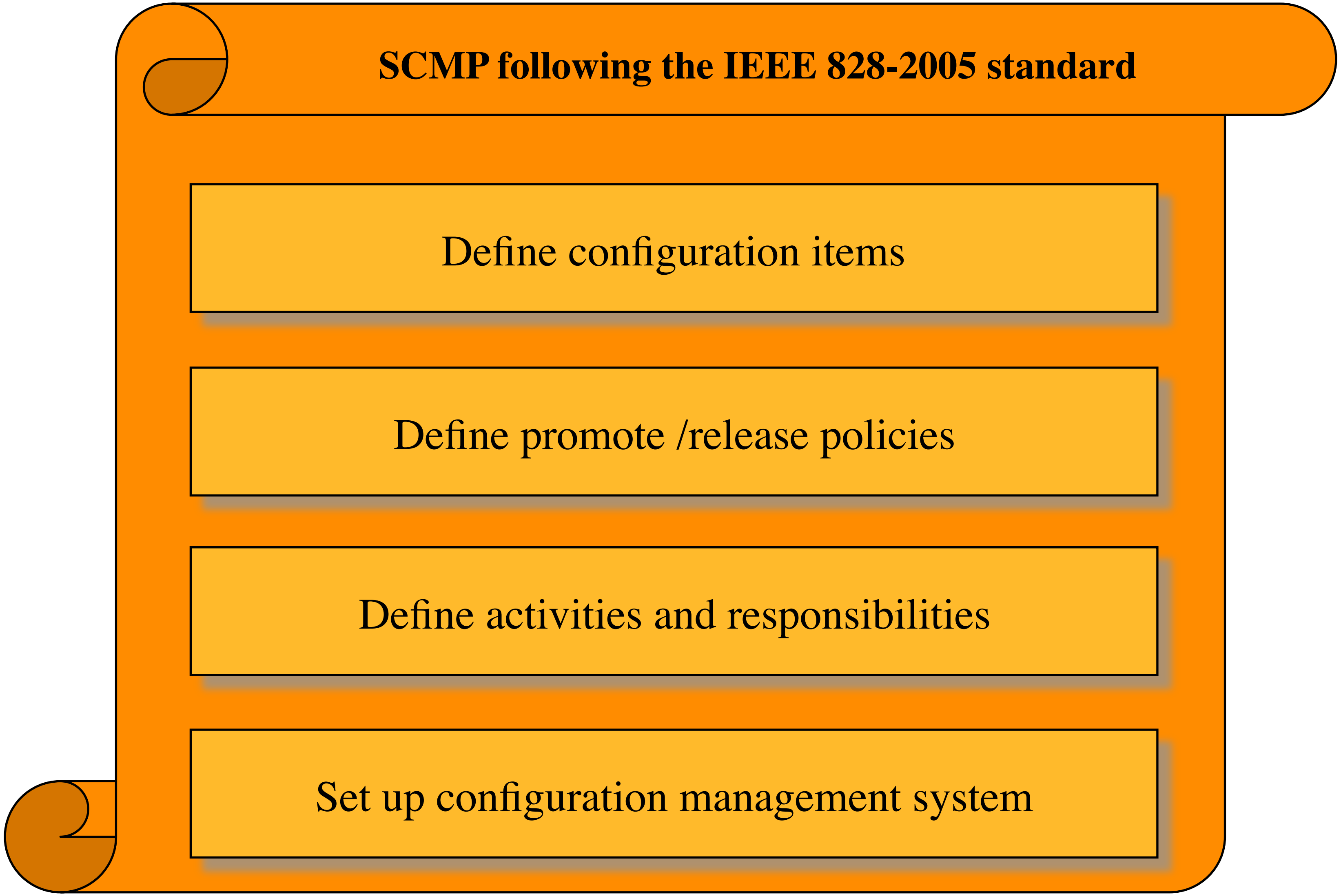
- Presentation format & Minimum information
  - A separate document or a section embedded in another document titled “Software Configuration Management Plan”
- Consistency Criteria:
  - All activities defined in the Plan shall be assigned to an organizational unit.
  - All activities defined shall have resources identified to accomplish the activities.
  - All CIs identified in the Plan shall have defined processes for baseline establishment and change control.
- If the above criteria are met, we can write:

  “This SCM Plan conforms with the requirements of IEEE Std 828-2005.”

Example SCM Plans
(from IEEE 1042.1987 Guide)

Life-cycle Phase	Project Type	Size	SCM Tools	Life Span	Writing	Character of Project
A Development	Critical	Medium	Advanced	Short	Highly Structured	Complex system contracted to another company
B Concept	Prototype	Small	Basic	Short	Informal	Small software development project
C Maintenance	Support Software	Large	On-line	Full Life-Cycle	Structured	SCMP used by organization using contracted SW
D All	Commercial	Small	Integrated	Full Life-Cycle	Informal	Development of embedded applications

Tasks for Configuration Managers (Summary)



SCMP following the IEEE 828-2005 standard

Define configuration items

Define promote /release policies

Define activities and responsibilities

Set up configuration management system

Add-on Exercise

- Write a Software Configuration Management Plan (SCMP) that conforms with the requirements of IEEE Std 828-2005 for the project of your choice
- Add this to the SPMP to the project of your choice (Exercise 1)