Software Engineering Essentials

Software Configuration Management 1

Bernd Bruegge, Stephan Krusche, Andreas Seitz, Jan Knobloch
Chair for Applied Software Engineering — Faculty of Informatics

Learning goals

1) Understand why software configuration management is required

2) Explain the differences between a promotion, a release and a revision

Motivation

- Multiple people work on artifacts that are changing
- More than one version of the artifact has to be supported:
  - Source code
  - Documents
  - Released software systems
- Software is running on different machines & operating systems in the target environment

➡ Need for coordination: Software configuration management (SCM)

  - Manages evolving software systems
  - Controls the costs involved in making changes to the system

Definition

Software configuration management (SCM) encompasses disciplines and techniques for

- Initiating, evaluating and controlling change to work products
- During and after a software project

It is a project function with the goal to make technical and managerial activities more effective

IEEE 828-2012: Standard for configuration management in systems and software engineering

Terminology

Version: A specific instance of configuration items (e.g. files, hardware)

Release: The (formal) distribution of an approved version

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

Baseline: A work product that has been (formally) reviewed and that thereafter can be changed only through change control procedures

Six main activities

1) Configuration item identification
- Modeling the system as set of evolving components

2) Promotion management
- Creation of versions for other developers

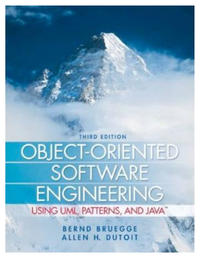
3) Build and release management
- Creation of versions for clients and users

4) Change management
- Handling, approval & tracking of change requests

Covered in this course

5) Branch management
- Management of concurrent development

6) Variant management
- Management of coexisting versions



Bruegge, Dutoit: Object-Oriented Software Engineering Using UML, Patterns, and Java (Chapter 13)