BINUS UNIVERSITY

People
Innovation
Excellence

Course : COMP6100/Software Engineering

Effective Period : Desember 2017

Formal Modeling and Software Configuration Management

Session 18 (Lecture)

BINUS UNIVERSITY

People
Innovation
Excellence

Acknowledgement

These slides have been adapted from Pressman, R.S. (2015). *Software Engineering : A Practioner's Approach. 8th ed*. McGraw-Hill Companies.Inc, Americas, New York. ISBN : 978 1 259 253157. Chapter 28 and 29

Learning Objectives

LO 4 : Analyze the software project management and the proposed potential business project

People
Innovation
Excellence

BINUS UNIVERSITY

Contents

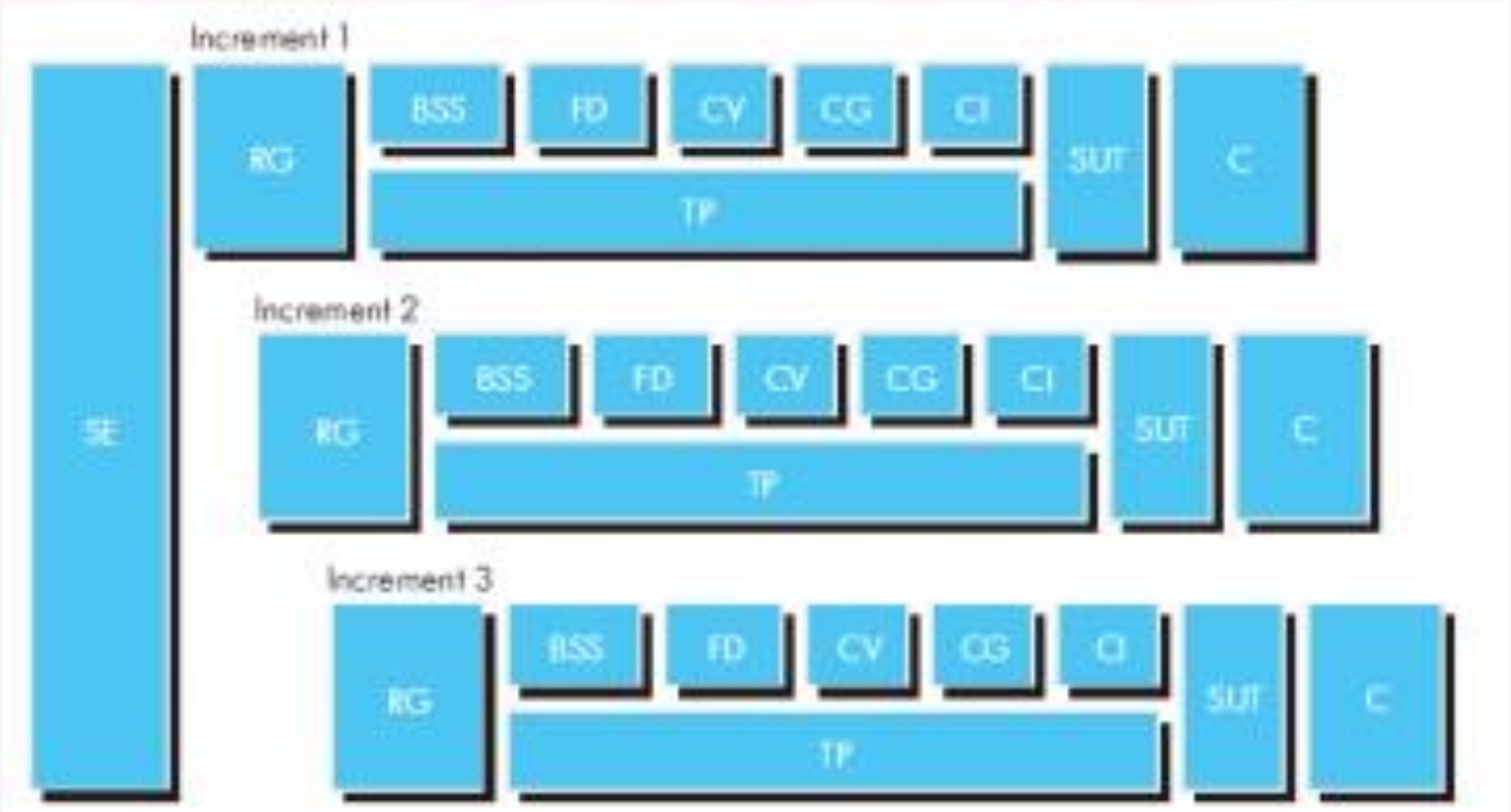
- Formal Modeling and Verification
- The Cleanroom Process Model
- Functional Specification
- Design Refinement & Verification
- Cleanroom Testing
- Formal Methods
- The Software Configuration
- The SCM Process
- Configuration Management for Web and MobilApps

People
Innovation
Excellence

Formal Modeling and Verification

- *Cleanroom software engineering* and *formal methods*
  - Both demand a specialized specification approach and each applies a unique verification method.
  - Both are quite rigorous and neither is used widely by the software engineering community.
- If you must build "bullet-proof" software, these methods can help immeasurably.

The Cleanroom Process Model



SE — system engineering
RG — requirements gathering
BSS — box structure specification
FD — formal design
CV — correctness verification
CG — code generation
CI — code inspection
SUT — statistical use testing
C — certification
TP — test planning

People
Innovation
Excellence

The Cleanroom Process Model

The Cleanroom Strategy-I

- Increment Planning
  - adopts the incremental strategy
- Requirements Gathering
  - defines a description of customer level requirements
- Box Structure Specification
  - describes the functional specification
- Formal Design
  - specifications (called “black boxes”) are iteratively refined (with an increment) to become analogous to architectural and procedural designs (called “state boxes” and “clear boxes,” respectively).
- Correctness Verification
  - verification begins with the highest level box structure (specification) and moves toward design detail and code using a set of “correctness questions.” If these do not demonstrate that the specification is correct, more formal (mathematical) methods for

The Cleanroom Process Model

The Cleanroom Strategy-II

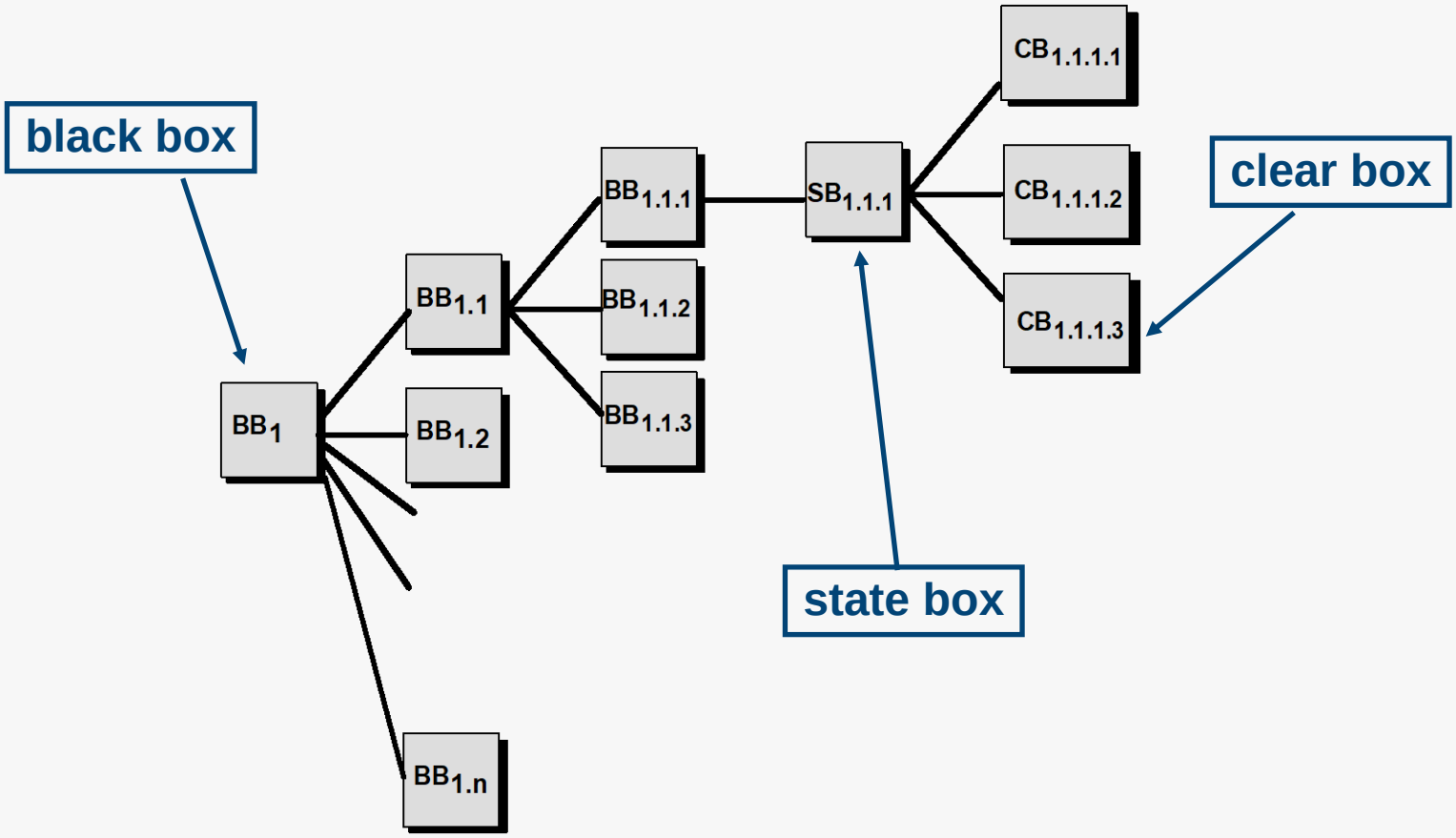
- Code Generation, Inspection and Verification
  - the box structure specifications, represented in a specialized language, are transmitted into the appropriate programming language.
- Statistical Test Planning
  - a suite of test cases that exercise of “probability distribution” of usage are planned and designed
- Statistical Use Testing
  - execute a series of tests derived from a statistical sample (the probability distribution noted above) of all possible program executions by all users from a targeted population
- Certification
  - once verification, inspection and usage testing have been completed (and all errors are corrected) the increment is certified as ready for integration.

BINUS UNIVERSITY

People
Innovation
Excellence

Functional Specification

Box Structure Specification

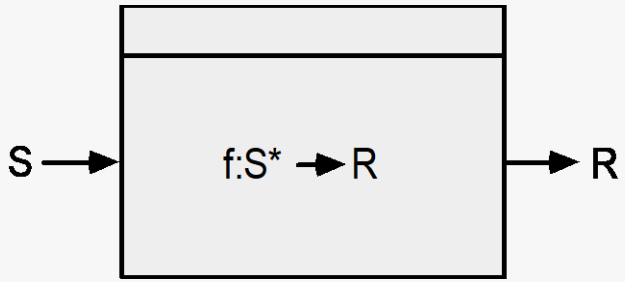


BINUS UNIVERSITY

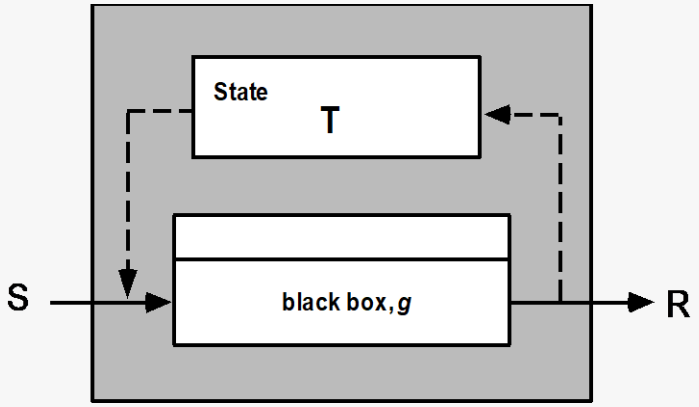
People
Innovation
Excellence

Functional Specification

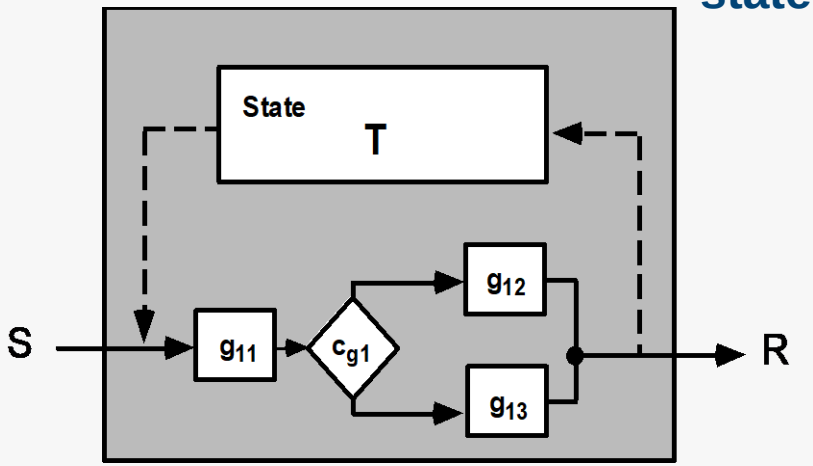
Box Structures



black box



state box



clear box

Design Refinement and Verification

If a function f is expanded into a sequence g and h, the correctness condition for all input to f is:

- Does g followed by h do f?

  When a function f is refined into a conditional (if-then-else), the correctness condition for all input to f is:

- Whenever condition <c> is true does g do f and whenever <c> is false, does h do f?

  When function f is refined as a loop, the correctness conditions for all input to f is:

- Is termination guaranteed?
- Whenever <c> is true does g followed by f do f, and whenever <c> is false, does skipping the loop still do f?

BINUS UNIVERSITY

People
Innovation
Excellence

Design Refinement and Verification

Advantages of Design Verification

- It reduces verification to a finite process.
- It lets cleanroom teams verify every line of design and code.
- It results in a near zero defect level.
- It scales up.
- It produces better code than unit testing.

BINUS UNIVERSITY

People
Innovation
Excellence

Cleanroom Testing

- statistical use testing
  - tests the actual usage of the program
- determine a “usage probability distribution”
  - analyze the specification to identify a set of stimuli
  - stimuli cause software to change behavior
  - create usage scenarios
  - assign probability of use to each stimuli
  - test cases are generated for each stimuli according to the usage probability distribution

BINUS UNIVERSITY

Cleanroom Testing

Certification

1. Usage scenarios must be created.
2. A usage profile is specified.
3. Test cases are generated from the profile.
4. Tests are executed and failure data are recorded and analyzed.
5. Reliability is computed and certified.

People
Innovation
Excellence

Cleanroom Testing

Certification Models

Sampling model. Software testing executes *m* random test cases and is certified if no failures or a specified numbers of failures occur. The value of *m* is derived mathematically to ensure that required reliability is achieved.

Component model. A system composed of *n* components is to be certified. The component model enables the analyst to determine the probability that component *i* will fail prior to completion.

Certification model. The overall reliability of the system is projected and certified.

Formal Methods

- *"Formal methods used in developing computer systems are mathematically based techniques for describing system properties. Such formal methods provide frameworks within which people can specify, develop, and verify systems in a systematic, rather than ad hoc manner."*

  The Encyclopedia of Software Engineering [Mar01]

- **The Problem with conventional specs:**
  - **contradictions**
  - **ambiguities**
  - **vagueness**
  - **incompleteness**
  - **mixed levels of abstraction**

Formal Methods

Formal Specification

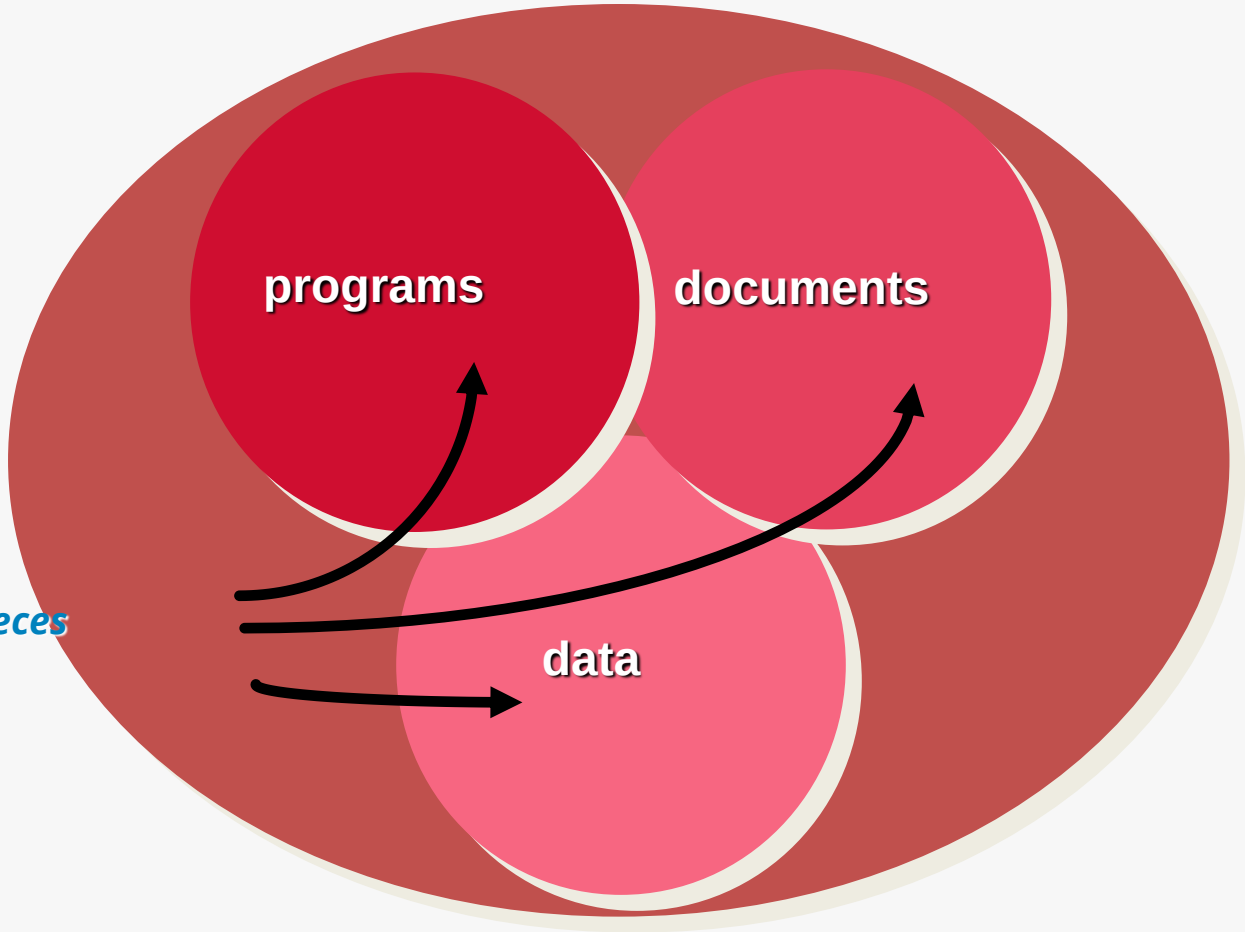
- Desired properties—consistency, completeness, and lack of ambiguity—are the objectives of all specification methods
- The formal syntax of a specification language enables requirements or design to be interpreted in only one way, eliminating ambiguity that often occurs when a natural language (e.g., English) or a graphical notation must be interpreted
  - The descriptive facilities of set theory and logic notation enable clear statement of facts (requirements).
- Consistency is ensured by mathematically proving that initial facts can be formally mapped (using inference rules) into later statements within the

Formal Methods

Formal Methods Concepts

- *data invariant*—a condition that is true throughout the execution of the system that contains a collection of data
- *state*
  - Many formal languages, such as OCL , that is, a system can be in one of several states, each representing an externally observable mode of behavior.
  - The Z language defines a *state* as the stored data which a system accesses and alters
- *operation*—an action that takes place in a system and reads or writes data to a state
  - *precondition* defines the circumstances in which a particular operation is valid
  - *postcondition* defines what happens when an operation has completed its action

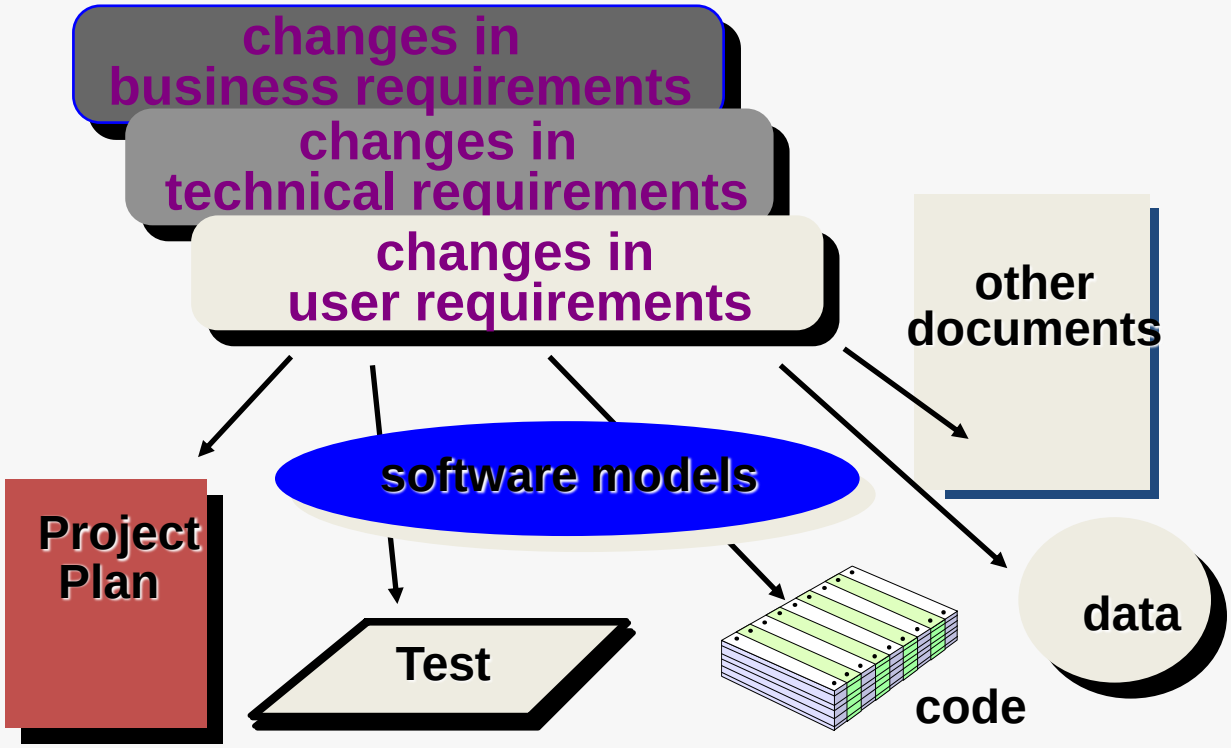
The Software Configuration



The pieces

The Software Configuration

What Are These Changes?



People
Innovation
Excellence

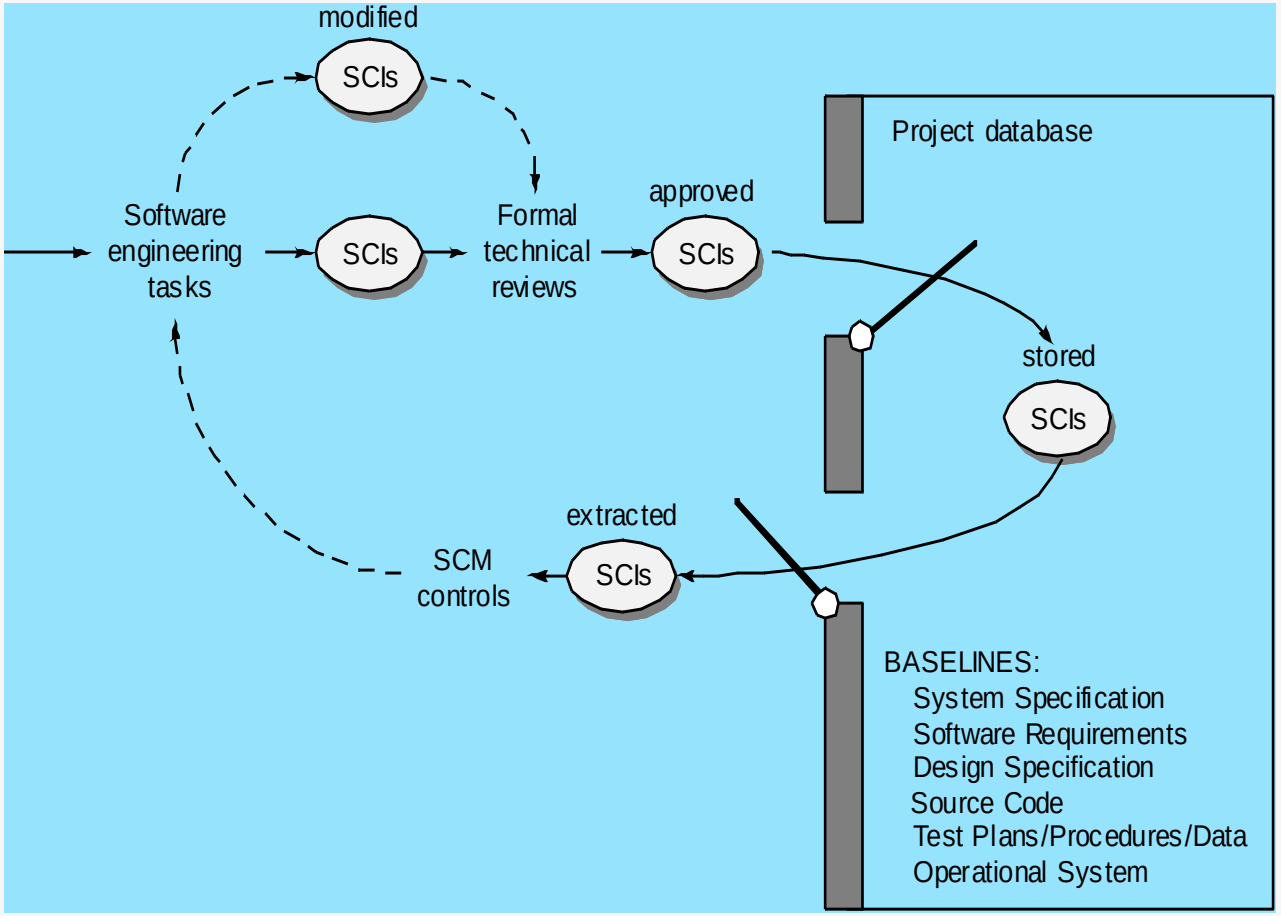
The Software Configuration

Baselines

- The IEEE (IEEE Std. No. 610.12-1990) defines a baseline as:
  - A specification or product 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 control procedures.
- a baseline is a milestone in the development of software that is marked by the delivery of one or more software configuration items and the approval of these SCIs that is obtained through a formal technical review

The Software Configuration

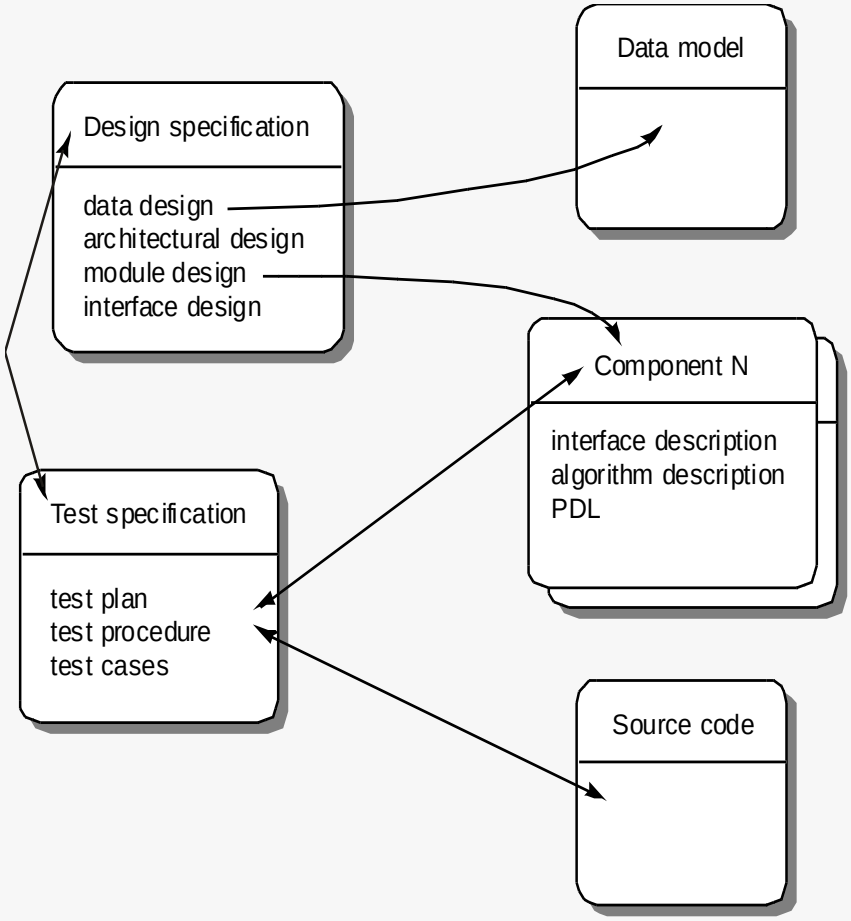
Baselined SCIs and the project database



People
Innovation
Excellence

The Software Configuration

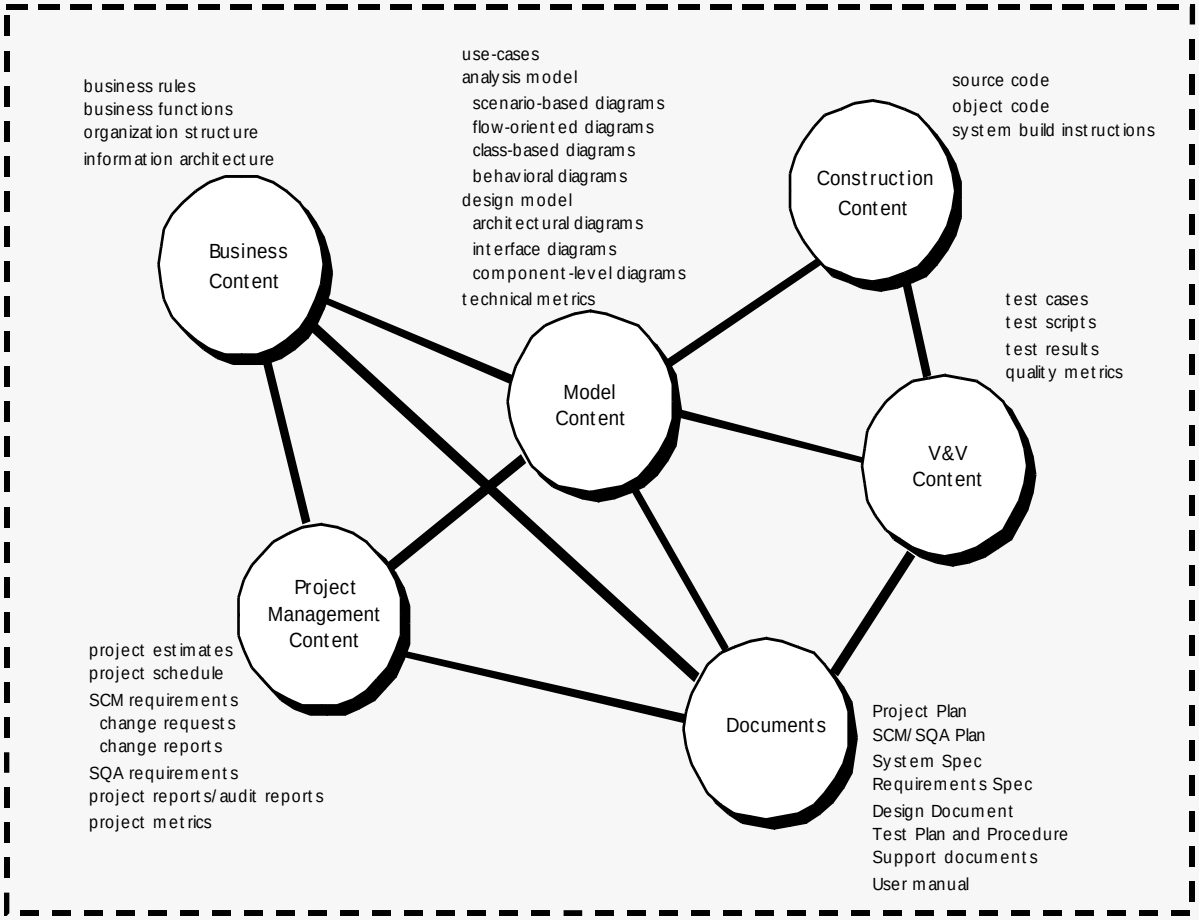
Configuration Objects



People
Innovation
Excellence

The Software Configuration

Repository Content



People
Innovation
Excellence

The Software Configuration

Repository Features

- **Versioning.**
  - saves all of these versions to enable effective management of product releases and to permit developers to go back to previous versions
- **Dependency tracking and change management.**
  - The repository manages a wide variety of relationships among the data elements stored in it.
- **Requirements tracing.**
  - Provides the ability to track all the design and construction components and deliverables that result from a specific requirement specification
- **Configuration management.**
  - Keeps track of a series of configurations representing specific project milestones or production releases. Version management provides the needed versions, and link management keeps track of interdependencies.
- **Audit trails.**
  - establishes additional information about when, why, and by whom changes are made

The Software Configuration

SCM Elements

- ***Component elements*—a set of tools coupled within a file management system (e.g., a database) that enables access to and management of each software configuration item.**
- ***Process elements*—a collection of procedures and tasks that define an effective approach to change management (and related activities) for all constituencies involved in the management, engineering and use of computer software.**
- ***Construction elements*—a set of tools that automate the construction of software by ensuring that the proper set of validated components (i.e., the correct version) have been assembled.**
- ***Human elements*—to implement effective SCM, the software team uses a set of tools and process features (encompassing other CM elements)**

The SCM Process

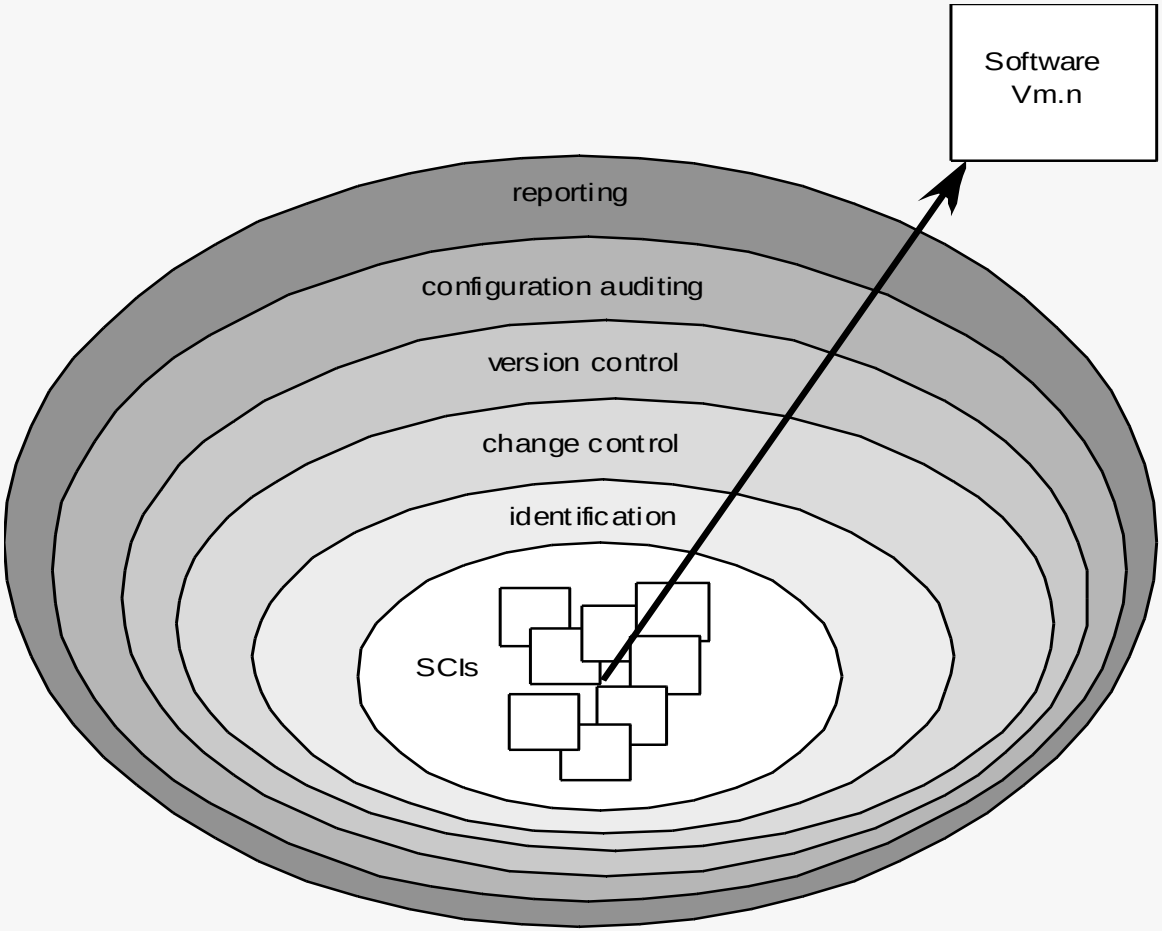
Addresses the following questions …

- How does a software team identify the discrete elements of a software configuration?
- How does an organization manage the many existing versions of a program (and its documentation) in a manner that will enable change to be accommodated efficiently?
- How does an organization control changes before and after software is released to a customer?
- Who has responsibility for approving and ranking changes?
- How can we ensure that changes have been made properly?
- What mechanism is used to appraise others of changes that are made?

BINUS UNIVERSITY

People
Innovation
Excellence

The SCM Process



The SCM Process

Version Control

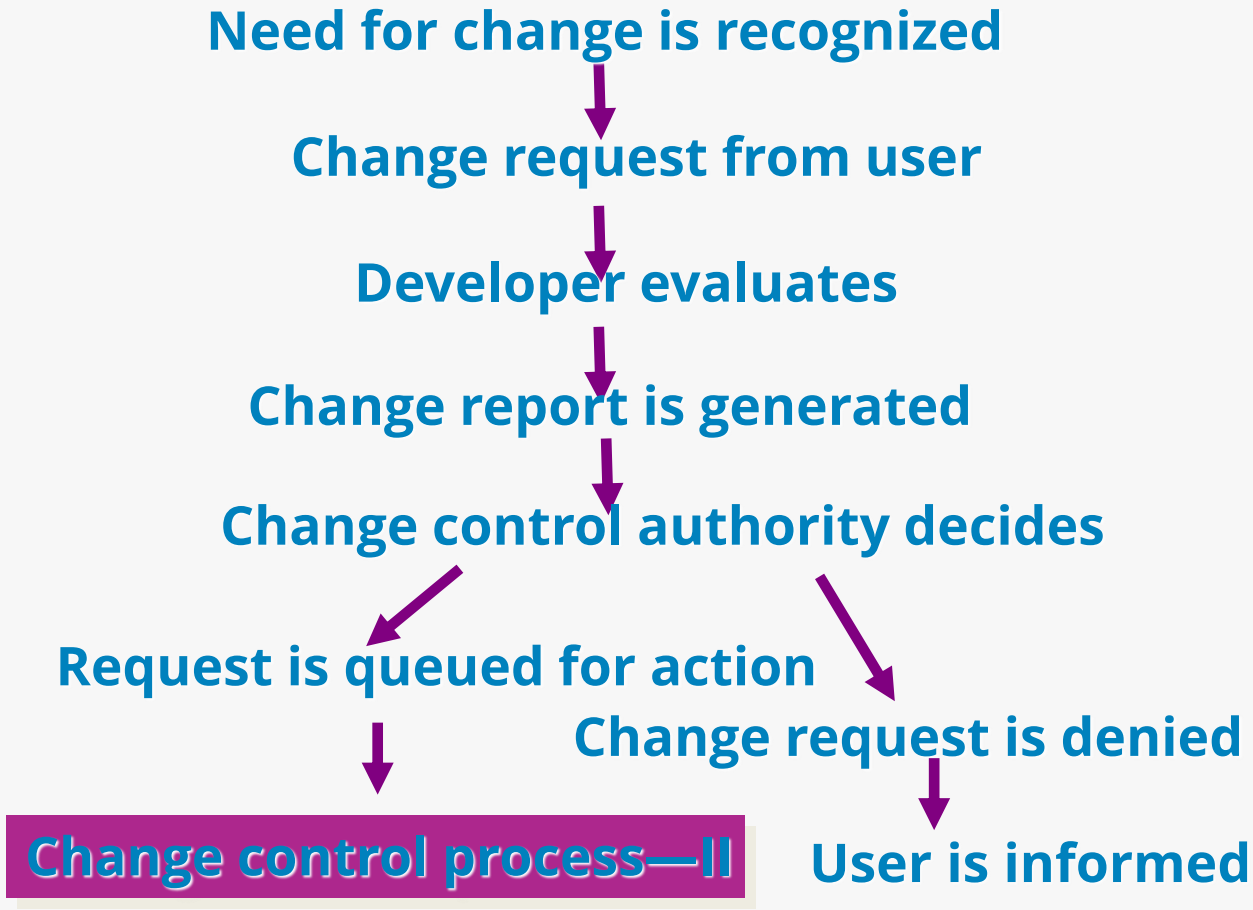
- Version control combines procedures and tools to manage different versions of configuration objects that are created during the software process
- A version control system implements or is directly integrated with four major capabilities:
  - a *project database (repository)* that stores all relevant configuration objects
  - a *version management* capability that stores all versions of a configuration object (or enables any version to be constructed using differences from past versions);
  - a *make facility* that enables the software engineer to collect all relevant configuration objects and construct a specific version of the software.
  - an *issues tracking* (also called *bug tracking*) capability that enables the team to record and track the status of all outstanding issues associated with each configuration object.

BINUS UNIVERSITY

People
Innovation
Excellence

The SCM Process

Change Control Process—I



The SCM Process

Change Control Process-II

Request is queued for action

↓

Assign individuals to configuration objects

↓

"Check-out" configuration objects

↓

Make the change

↓

Review (audit)the change

↓

"Check in" the configuration items that have been changed

↓

Establish a baseline for testing

↓

change control process—III

People
Innovation
Excellence

The SCM Process

Change Control Process-III

Perform quality assurance and testing activities

"Promote" changes for inclusion in next release (revision)

Rebuild appropriate version of software

Review (audit) the change to all configuration items

Include changes in new version

Distribute the new version

People
Innovation
Excellence

Configuration Management for Web and MobileApps

- Content.
  - A typical Web an MobileApp contains a vast array of content—text, graphics, applets, scripts, audio/video files, forms, active page elements, tables, streaming data, and many others.
  - The challenge is to organize this sea of content into a rational set of configuration objects and then establish appropriate configuration control mechanisms for these objects.
- People.
  - Because a significant percentage of Web and MobileApp development continues to be conducted in an ad hoc manner, any person involved in the Web and MobileApp can (and often does) create content

Configuration Management for Web and MobileApps

- **Scalability.**
  - As size and complexity grow, small changes can have far-reaching and unintended affects that can be problematic. Therefore, the rigor of configuration control mechanisms should be directly proportional to application scale.
- **Politics.**
  - Who 'owns' a Web and MobileApp?
  - Who assumes responsibility for the accuracy of the information on the Web and MobileApp?
  - Who assures that quality control processes have been followed before information is published to the site?
  - Who is responsible for making changes?
  - Who assumes the cost of change?

Configuration Management for Web and MobileApps

Content Management-I

- The collection subsystem encompasses all actions required to create and/or acquire content, and the technical functions that are necessary to
  - convert content into a form that can be represented by a mark-up language (e.g., HTML, XML
  - organize content into packets that can be displayed effectively on the client-side.
- The management subsystem implements a repository that encompasses the following elements:
  - *Content database*—the information structure that has been established to store all content objects
  - *Database capabilities*—functions that enable the CMS to search for specific content objects (or categories of objects), store and retrieve objects, and manage the file structure that has been established for the content
  - *Configuration management functions*—the functional elements and associated workflow that support content object identification, version control, change management, change auditing, and reporting.

Configuration Management for Web and MobileApps

Content Management-II

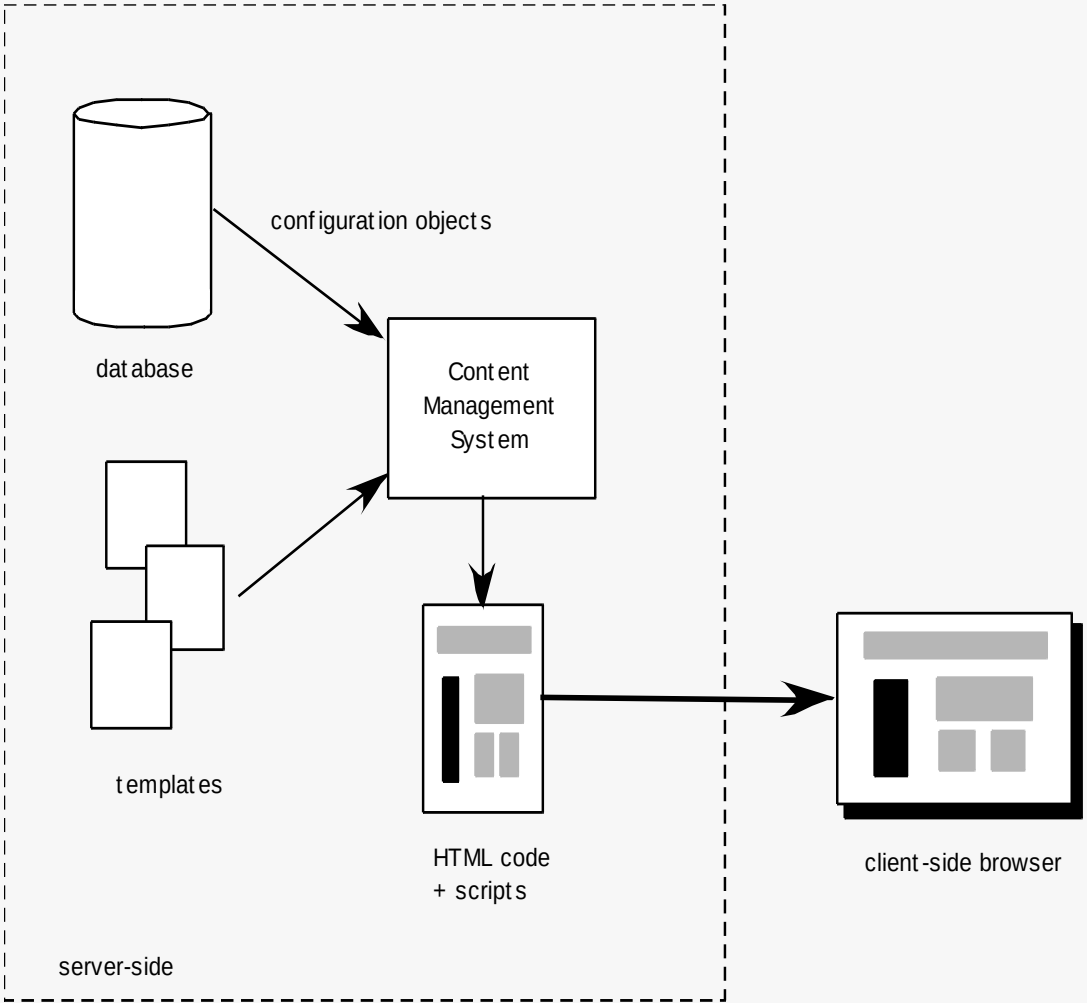
- The publishing subsystem extracts from the repository, converts it to a form that is amenable to publication, and formats it so that it can be transmitted to client-side browsers. The publishing subsystem accomplishes these tasks using a series of templates.
- Each *template* is a function that builds a publication using one of three different components [BOI02]:
  - *Static elements*—text, graphics, media, and scripts that require no further processing are transmitted directly to the client-side
  - *Publication services*—function calls to specific retrieval and formatting services that personalize content (using predefined rules), perform data conversion, and build appropriate navigation links.
  - *External services*—provide access to external corporate information infrastructure such as enterprise data or "back-

BINUS UNIVERSITY

People
Innovation
Excellence

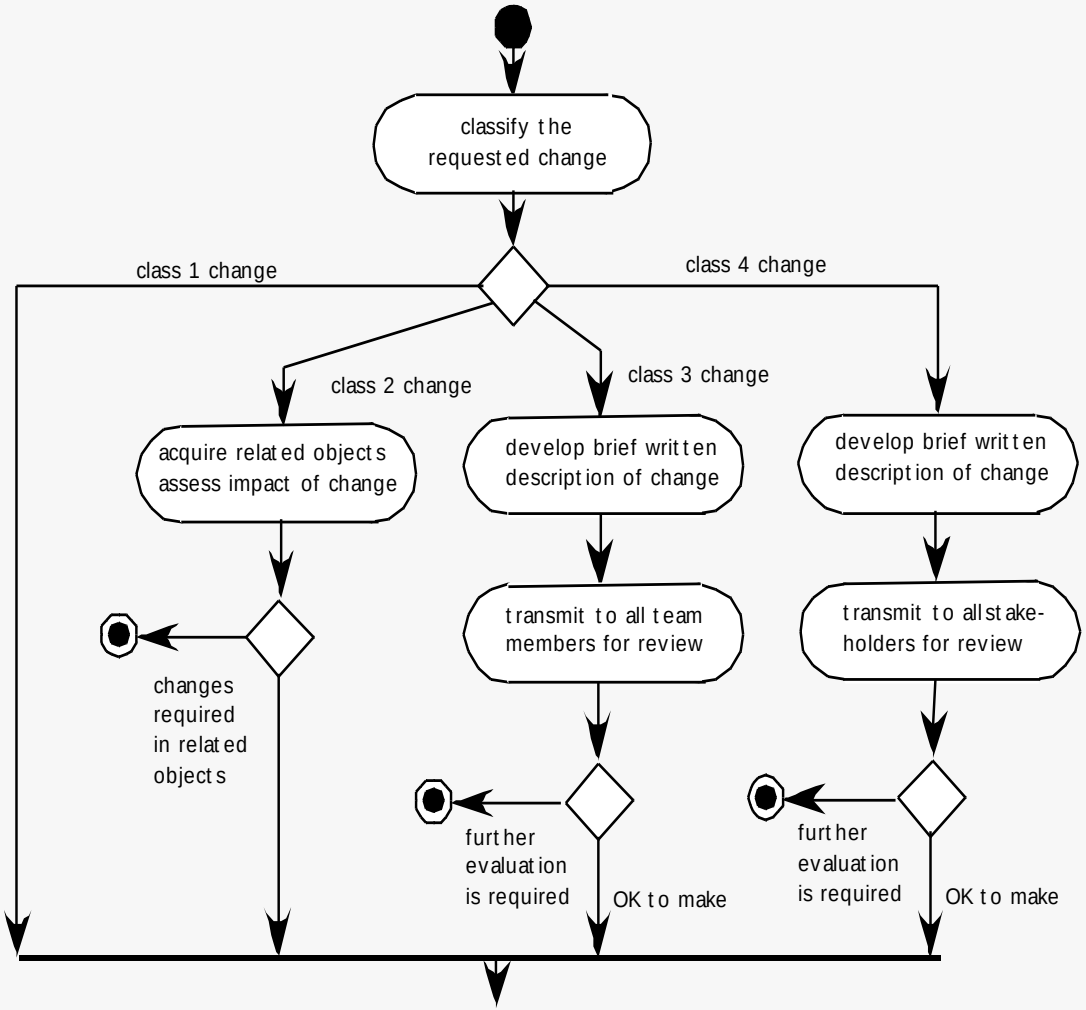
Configuration Management for Web and MobileApps

Content Management



Configuration Management for Web and MobileApps

Change Management for WebApps-I



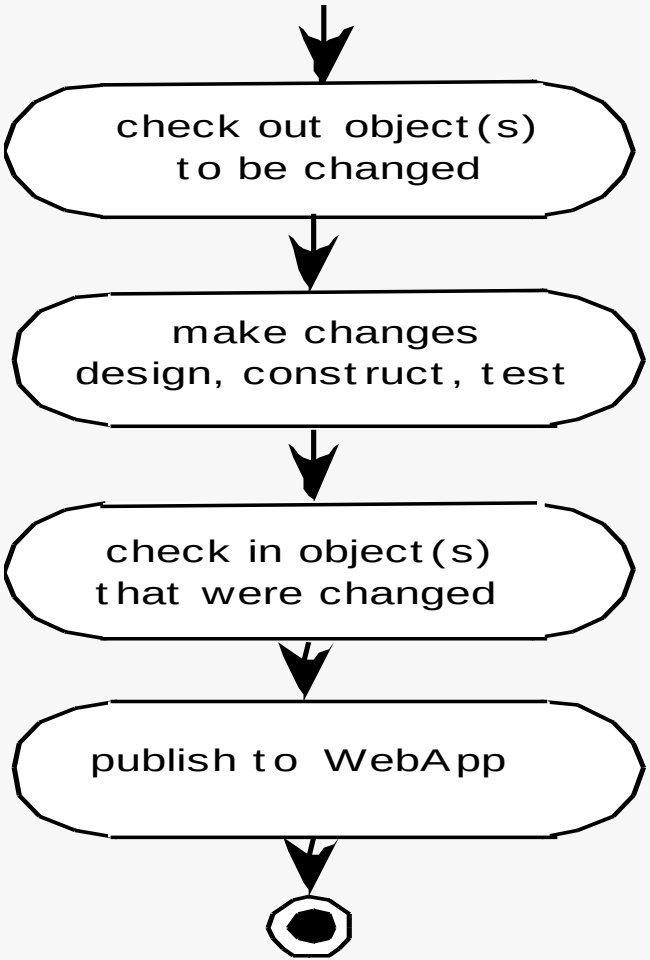
People
Innovation
Excellence

BINUS UNIVERSITY

People
Innovation
Excellence

Configuration Management for Web and MobileApps

Change Management for WebApps-II



References

- Pressman, R.S. (2015). ***Software Engineering : A Practioner's Approach. 8th ed***. McGraw-Hill Companies.Inc, Americas, New York. ISBN : 978 1 259 253157
- Formal modelling analysis
  http://www.springerlink.com/content/2xedpxdgg209gdb7/
- Sw configuration Management
  http://www.sei.cmu.edu/reports/87cm004.pdf
- Change Configuration Management
  http://www.serena.com/solutions/change-and-configuration-management

BINUS UNIVERSITY

People
Innovation
Excellence

Q & A