Software Project Management II

Dr. Jong Yih Kuo
Dr. Chien-Hung Liu
Computer Science and Information Engineering
National Taipei University of Technology
# Course Schedule

<table>
<thead>
<tr>
<th>Time</th>
<th>Topic</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>18:00~19:30</td>
<td>Analysis and Design for Module Interface</td>
<td></td>
</tr>
<tr>
<td>(90 min)</td>
<td>- Use Case Modeling</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Scenario-based Test Case Design</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- System Sequence Diagram</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Operation Contract</td>
<td></td>
</tr>
<tr>
<td>19:40~20:05</td>
<td>Configuration Management</td>
<td>CM Plan</td>
</tr>
<tr>
<td>(25 min)</td>
<td>- Version Control Process</td>
<td></td>
</tr>
<tr>
<td>20:05~20:25</td>
<td>SVN</td>
<td></td>
</tr>
<tr>
<td>(20 min)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:25~20:40</td>
<td>Review Process</td>
<td></td>
</tr>
<tr>
<td>(15 min)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:40~21:00</td>
<td>Mantis</td>
<td>Bug Tracking Report</td>
</tr>
<tr>
<td>(20 min)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Configuration Management
Configuration Management Concepts

- Software Configuration Management activities
  - Configuration item identification
    - Identify changes
  - Configuration Control
    - Control changes
    - Version control: Make sure that changes are being properly implemented
  - Record and report changes to others who have an interest
    - Tracking change
    - Audit the products to verify the work products correct and complete.
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?

Source: R.S. Pressman, “Software Engineering: A Practitioner’s Approach, 6/e”
The SCM Process

- CIs identification
- Change control
- Version control
- Configuration audit
Configuration Item Identification (1)

- Configuration Item Identification
  - Select the configuration items
  - Assign unique identifiers
  - Recording their characteristics

- Configuration items (CI)
  - An aggregation of work products that is designated for configuration management and treated as a single entity (CMMI)
    - Basic object: a unit of information created during analysis, design, code, or test
    - Aggregate object: a collection of basics objects
Configuration Item Identification (2)

- **Configuration items**
  - Project Execution Plan (PEP), Software Requirement Specification (SRS) – Use case model, System Sequence Diagram, Operation Contract
  - Software modules, source codes, Compiler tools
  - Test Plans - Test procedures, test case, test results

- Diagram:
  - Design specification
    - data design
    - architecture design
    - module design
    - interface design
  - Test specification
    - Test plan
    - Test procedures
    - Test cases
  - Source code
  - Component N
    - Interface description
    - Algorithm description
    - PDL
## Configuration Item Identification (3)

- Each CI has a set of features
  - ID
  - Name
  - Description: version
  - Author
  - File type
  - Programming language for software code files
Configuration Item Identification (4)

- Purposes
  - Who has a particular system version?
  - What platform is required for a given version?
  - How many versions of a system been created? What were their created dates?
  - What versions are affected by a change to a particular component?
  - How many reported faults in a particular version?
Baselines

- IEEE (IEEE 610.12-1990)
  - A specification or product that has been formally reviewed and agreed upon, that serves as the basis for further development, and that can be changed only through formal change control procedures

- One or more the approval of Configuration Items (CIs) that is obtained through a formal technical review
Baselines

- software work products
- CIs
  - formal technical review
  - approved
  - modified
  - SCM controls

CM System
- stored
- CIs

BASELINES
- System Specification
- Software Requirements
- Design Specification
- Source Code
- Test Plans/Procedures/Data
- Operational System
Change Control Process (1)

- Identify the part to change
- Decide whether to change
- Perform change
- Test and release the change and update configuration management database
Change Control Process (2)

- Identify the part to change
  - Change request form (CRF)
    - change required
    - suggest of change
    - reason why change was suggested
    - urgency of change (from requester of the change)
    - change evaluation
      - impact analysis
      - change cost
      - recommendations
  - Change requests should be registered in the CM database
Sample of Change Request Form

Project: Proteus/PCL-Tools
Change requester: I. Sommerville
Number: 23/94
Date: 1/12/98
Requested change: When a component is selected from the structure, display the name of the file where it is stored.
Change analyser: G. Dean
Analysis date: 10/12/98
Components affected: Display-Icon.Select, Display-Icon.Display
Associated components: FileTable
Change assessment: Relatively simple to implement as a file name table is available. Requires the design and implementation of a display field. No changes to associated components are required.
Change priority: Low
Change implementation: Estimated effort: 0.5 days
Date to CCB: 15/12/98
CCB decision: Accept change. Change to be implemented in Release 2.1.
Change implementor: Date of change:
Date submitted to QA: QA decision:
Date submitted to CM:
Comments
Change Control Process (3)

- Decide whether to change - Change control authority
  - Change Control Boards (CCB)
    - A group of people responsible for evaluating and approving or disapproving proposed changes to configuration items, and for ensuring implementation of approved changes. [CMMI]

Evolution graph: describes the change history
Change Control Process (4)

1. Recognize the need for change
   - Users request for changes
   - Developers evaluate the changes
2. Generate the change report
3. Decide change
   - Request is granted and waits for action
   - Request is denied
   - Inform users
   - Assign people to SCI changes
   - Check out SCI from Repository
   - Implement the changes
   - Review the changes
   - Establish a baseline for testing
4. Perform Change
5. Testing and review activities
   - Check in the changed SCI into Repository
   - Include the changed SCI into new release
   - Rebuild a new version
   - Audit the changes
6. Test and Release Change
Versions and Releases

- **Version**
  - An instance of a system which is functionally distinct in some way from other system instances

- **Release**
  - An instance of a system which is distributed to users outside of the development team

- Version control
  - The establishment and maintenance of [baselines] and the identification of changes to baselines that make it possible to return to the previous baseline.
Version Tree
Version Control

- Version control combines procedures and tools to manage different versions of configuration objects that are created during the software process.
- Record changes made to codes or documents:
  - Changes made
  - Rationale or why the change made
  - Who made the change
  - When the change was made
Version Control

- The version control mechanisms, integrated within the change control process, implement two important elements of change management
  - **Access control**: governs which software engineers have the authority to access and modify a particular configuration item
  - **Synchronization control**: ensure that parallel changes, performed by two different people, don’t overwrite one another
Version identification

- **Version numbering**
  - Derivation structure, such as V1, V1.1, V1.2, V2.1, V2.2
  - Need to keep track of the differences between versions

- **Attribute-based identification**
  - Attributes associated with a version to identify that version
    - Date, Creator, Programming Language, Customer, Status, HW platform
    - Ex: AC3D(language=Java, platform=NT, date=Jan1999)
Release Management

- Release managers response for
  - Deciding when to release the system to customer
  - Managing the process of creating release and distribution media
  - Documenting the release to ensure the system can be re-created exactly as distributed
Release Management

- Releases are NOT just a set of executable programs, include
  - Configuration files defining how the release is configured for a particular installation
  - Data files needed for system operation
  - An installation program or shell script to install the system on target hardware
  - Electronic and paper documentation
  - Packaging and associated publicity
Configuration Audit

- Ensure the changes been properly implemented
  - Ensure that the correct CIs being incorporated into specific build and all documentation is up-to-date and consistent with the version that has been built
  - **Formal technical reviews**
    - Focus on the technical correctness of the configuration objects being modified
    - Access the CI to determine consistency with other CIs, omissions, or potential side effects
Configuration Audit

- **Software configuration audit** (Complements the formal technical review by addressing the following questions:)
  - Has the change specified in the ECO been made? Have any additional modifications been incorporated?
  - Has a formal technical review been conducted to access technical correctness?
  - Has the software process been followed, and have software engineering standards been properly applied?
  - Has the change been “highlighted” in the SCI? Have the change date and change author been specified? Do the attributes of the configuration object reflect the change?
  - Have SCM procedures for noting the change, recording it, and reporting it been followed?
  - Have all related SCIs been properly updated?
Status Reporting

- Configuration status reporting (CSR) (sometimes called status accounting) is a CM task that answers the following questions:
  - What happened
  - Who did it
  - When did it happen
  - What else will be affected
Status Reporting

- A CSR entry is made
  - Each time a SCI is assigned new or updated identification
  - Each time a change is approved by the CCA
- Each time a configuration audit is conducted, the results are reported as part of the CSR task
- CSR report is generated on a regular basis and is intended to keep management and practitioners appraised of import changes
CM Roles and Responsibilities

- A **project manager** is in charge of a software group
  - Auditing the work products
- A **configuration manager** is in charge of the CM procedure
  - Controlling, tracking the work products
- The **software engineers** are responsible for developing and maintaining the software product
  - Changing, building, and access control the work products
Configuration Management Plan

- IEEE 828-2005
- Overview
- Objectives/Purpose
- Scope
  - Configuration items
    - document configuration
    - Software configuration item
    - Hardware configuration
  - CM activities
    - Establish a configuration management system
    - Create and Release baseline
    - Track change
    - Change control
- Roles and Responsibilities
  - Project Manager
  - Configuration manager
  - Developer
- Schedule