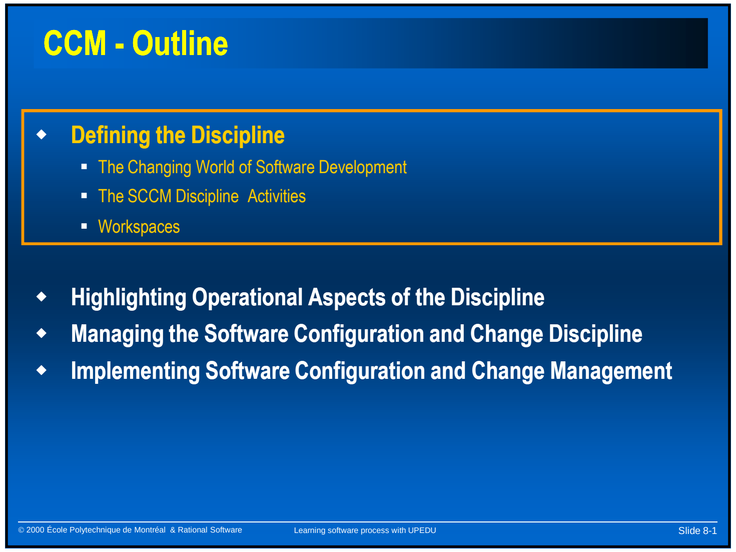
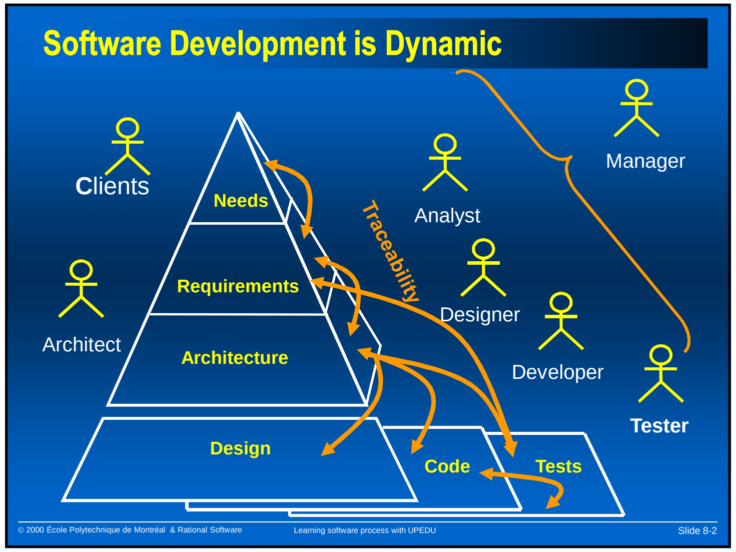


CCM - Outline

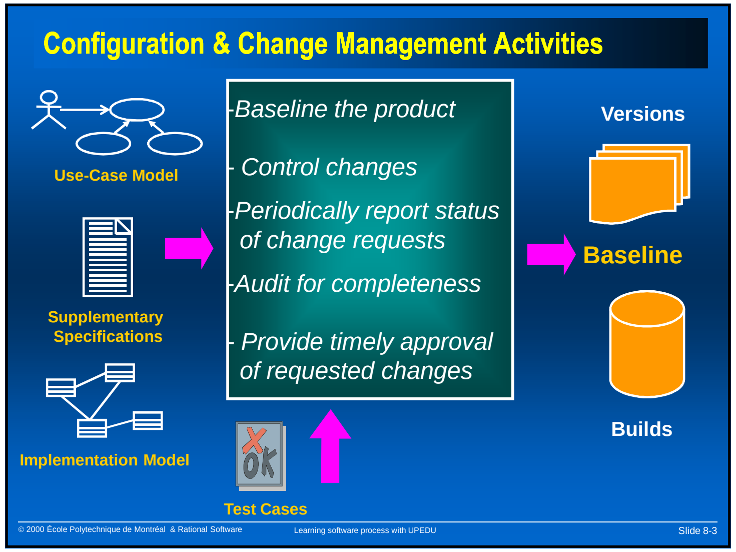
- **Defining the Discipline**
  - The Changing World of Software Development
  - The SCCM Discipline Activities
  - Workspaces
- **Highlighting Operational Aspects of the Discipline**
- **Managing the Software Configuration and Change Discipline**
- **Implementing Software Configuration and Change Management**

Learning software process with UPEDU Slide 8-1



Software Development is Dynamic

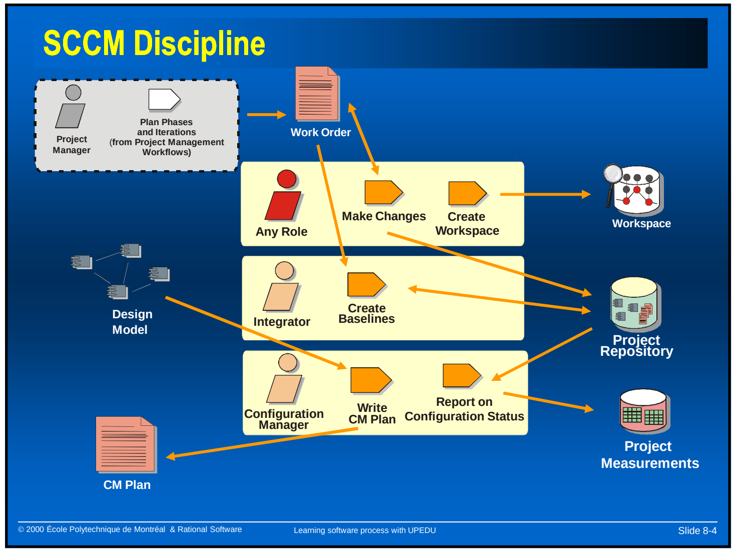
Learning software process with UPEDU Slide 8-2



Configuration & Change Management Activities

- Baseline the product
- Control changes
- Periodically report status of change requests
- Audit for completeness
- Provide timely approval of requested changes

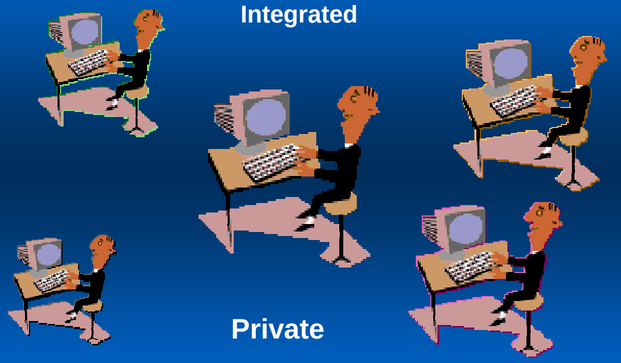
Learning software process with UPEDU Slide 8-3



SCCM Discipline

Learning software process with UPEDU Slide 8-4

Workspace



UPEDU Concept: Workspaces

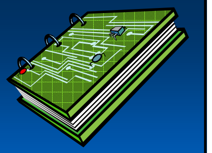
CCM - Outline

- Defining the Discipline
- **Highlighting Operational Aspects of the Discipline**
  - Identification of Software Configuration Items
  - Control of Baseline and Changes
  - Status Accounting of Components and Changes
  - Functional and Physical Audit
- Managing the Software Configuration and Change Discipline
- Implementing Software Configuration and Change Management

UPEDU GUIDELINES:
Important decisions in configuration and Change Management

Baselines are Stable Product Versions

- Major components of SCCM to enable control.
- Identify collections of configuration items referring to a unique version of each artifact
- 'Snapshot' in time of the development artifacts in the Implementation Model
- Stable product versions against which error reports and change requests are filled
- Official reference on which subsequent work is to be based
- Formal or informal baseline

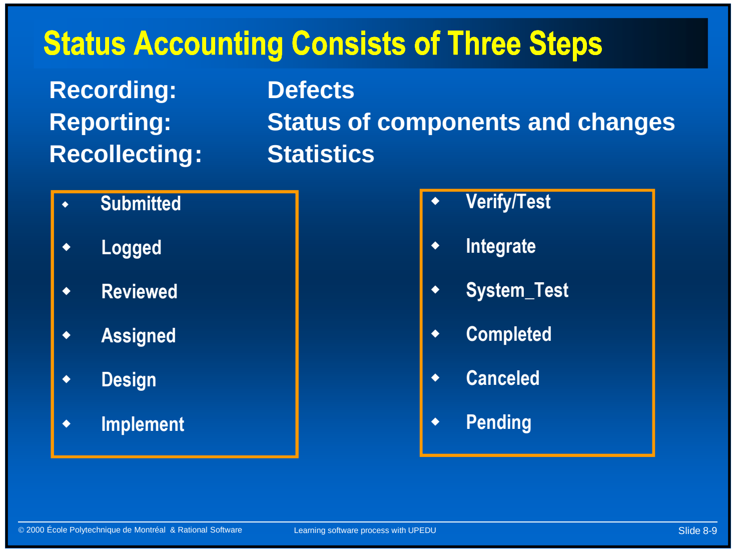


UPEDU Concept: Product Directory Structure
UPEDU Concept: Baselining

Control Helps Avoid Confusion

- Simultaneous update
- Multiple developers
- Multiple teams
- Multiple sites
- Multiple variants and versions
- Multiple iterations
- Multiple releases
- Multiple projects
- Multiple platforms
- Limited Notification
  - some developers are not notified of fix in shared artifacts

Without explicit control, parallel development degrades to chaos

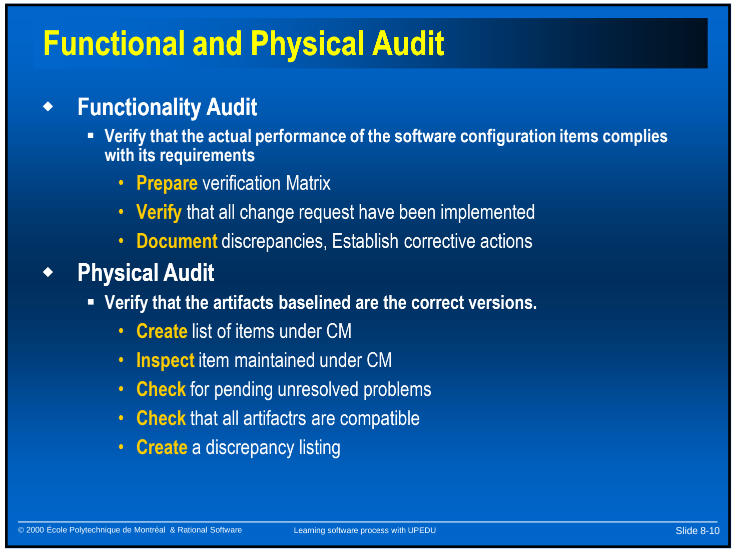


Status Accounting Consists of Three Steps

Recording:	Defects
Reporting:	Status of components and changes
Recollecting:	Statistics

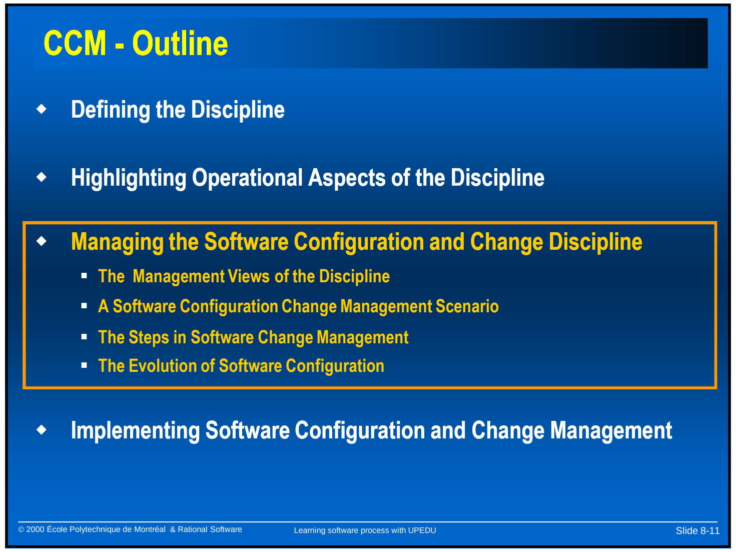
- Submitted
- Logged
- Reviewed
- Assigned
- Design
- Implement

- Verify/Test
- Integrate
- System_Test
- Completed
- Canceled
- Pending



Functional and Physical Audit

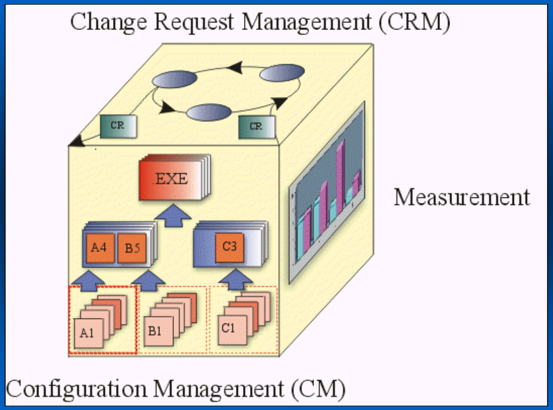
- Functionality Audit
  - Verify that the actual performance of the software configuration items complies with its requirements
    - Prepare verification Matrix
    - Verify that all change request have been implemented
    - Document discrepancies, Establish corrective actions
- Physical Audit
  - Verify that the artifacts baselined are the correct versions.
    - Create list of items under CM
    - Inspect item maintained under CM
    - Check for pending unresolved problems
    - Check that all artifactrs are compatible
    - Create a discrepancy listing

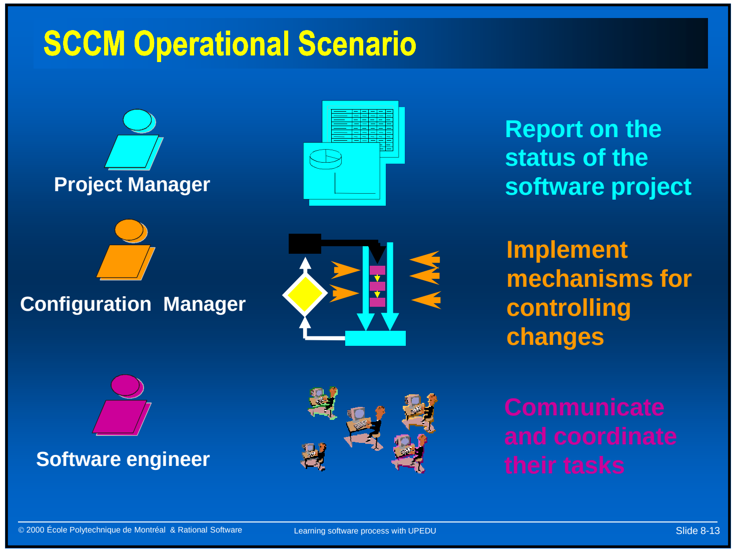


CCM - Outline

- Defining the Discipline
- Highlighting Operational Aspects of the Discipline
- Managing the Software Configuration and Change Discipline
  - The Management Views of the Discipline
  - A Software Configuration Change Management Scenario
  - The Steps in Software Change Management
  - The Evolution of Software Configuration
- Implementing Software Configuration and Change Management

Configuration Management Cubic View


Change Request Management (CRM)
CR
CR
EXE
A4
B5
C3
A1
B1
C1
Measurement
Configuration Management (CM)



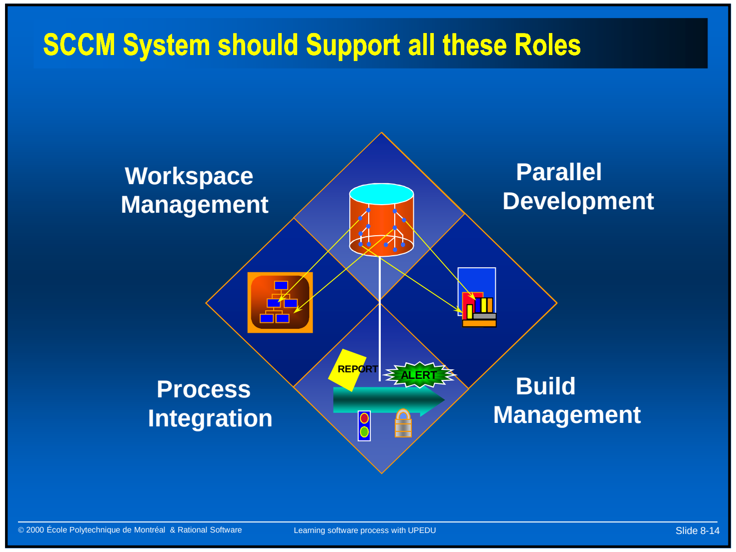
SCCM Operational Scenario

Project Manager — Report on the status of the software project

Configuration Manager — Implement mechanisms for controlling changes

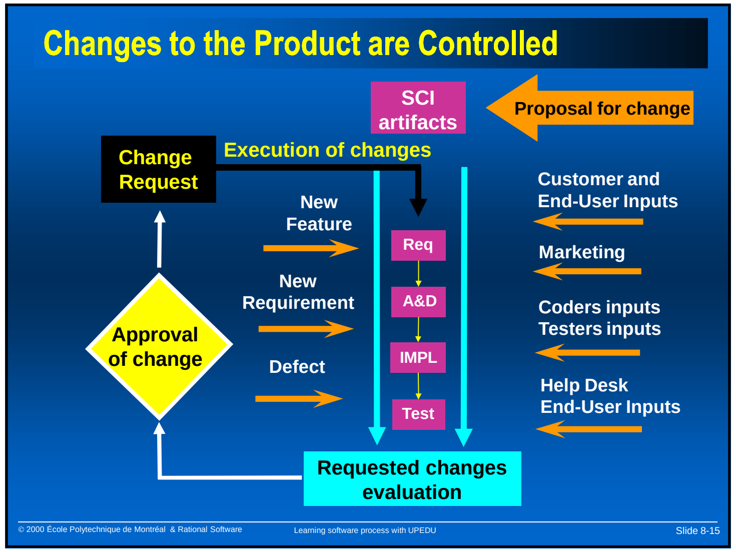
Software engineer — Communicate and coordinate their tasks

Learning software process with UPEDU



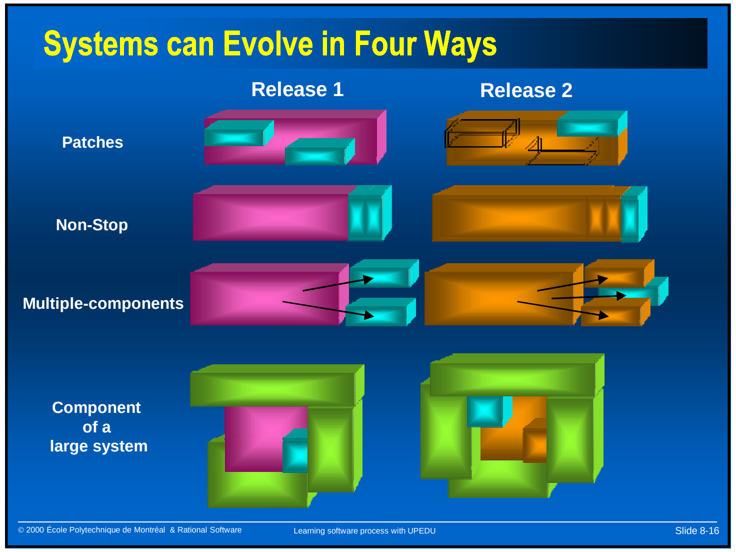
SCCM System should Support all these Roles

Learning software process with UPEDU



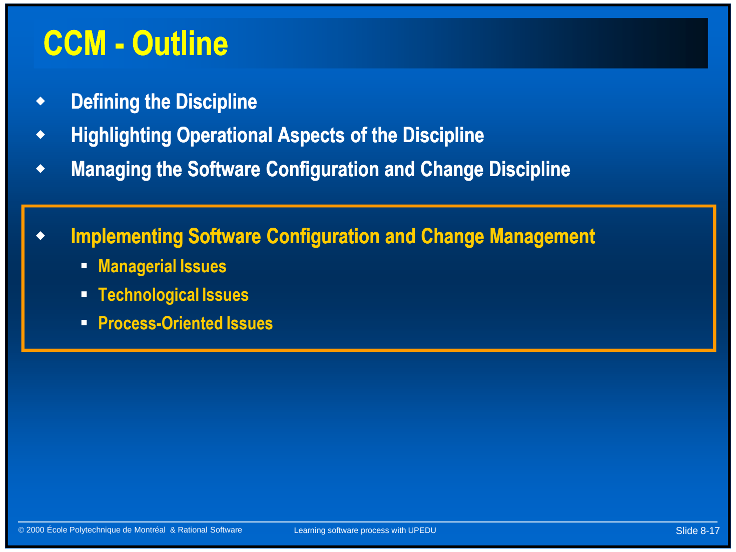
Changes to the Product are Controlled

Learning software process with UPEDU



Systems can Evolve in Four Ways

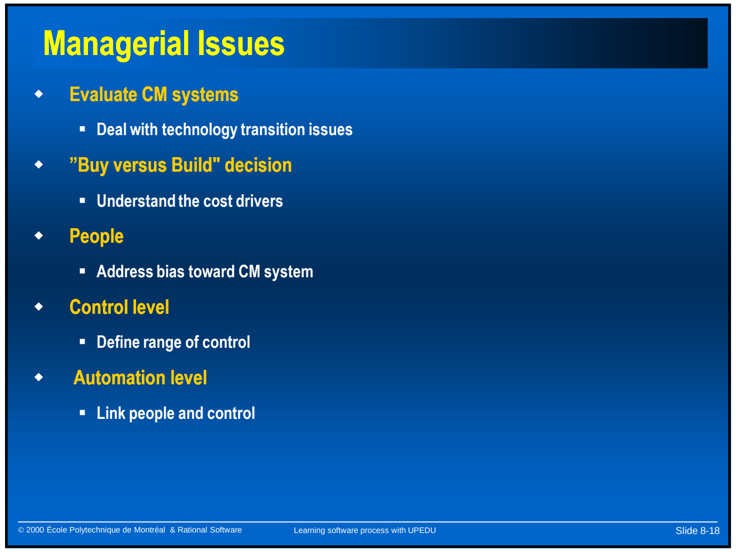
Learning software process with UPEDU



CCM - Outline

- Defining the Discipline
- Highlighting Operational Aspects of the Discipline
- Managing the Software Configuration and Change Discipline
- Implementing Software Configuration and Change Management
  - Managerial Issues
  - Technological Issues
  - Process-Oriented Issues

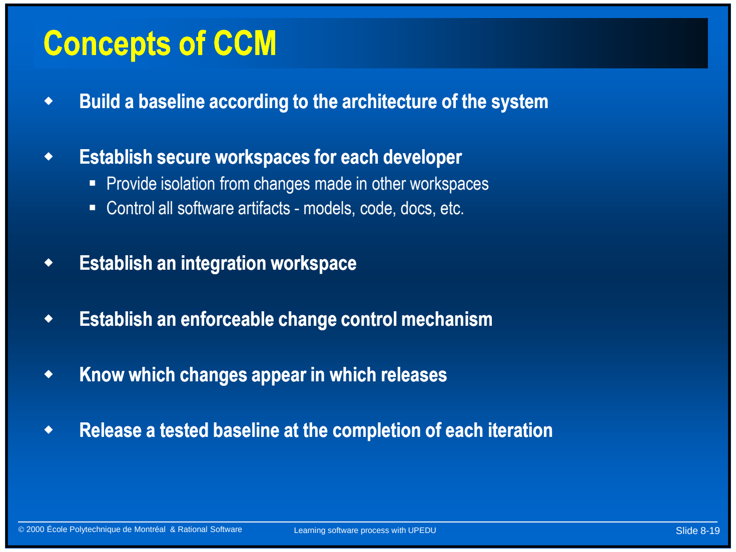
© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 8-17



Managerial Issues

- Evaluate CM systems
  - Deal with technology transition issues
- "Buy versus Build" decision
  - Understand the cost drivers
- People
  - Address bias toward CM system
- Control level
  - Define range of control
- Automation level
  - Link people and control

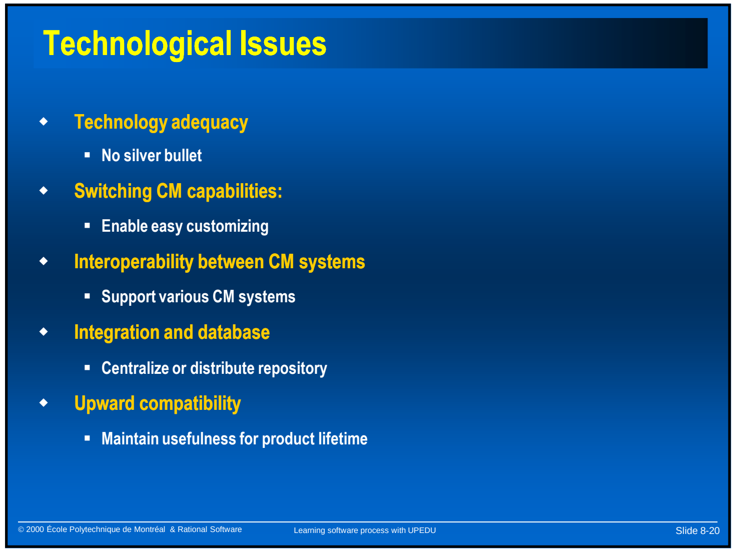
© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 8-18



Concepts of CCM

- Build a baseline according to the architecture of the system
- Establish secure workspaces for each developer
  - Provide isolation from changes made in other workspaces
  - Control all software artifacts - models, code, docs, etc.
- Establish an integration workspace
- Establish an enforceable change control mechanism
- Know which changes appear in which releases
- Release a tested baseline at the completion of each iteration

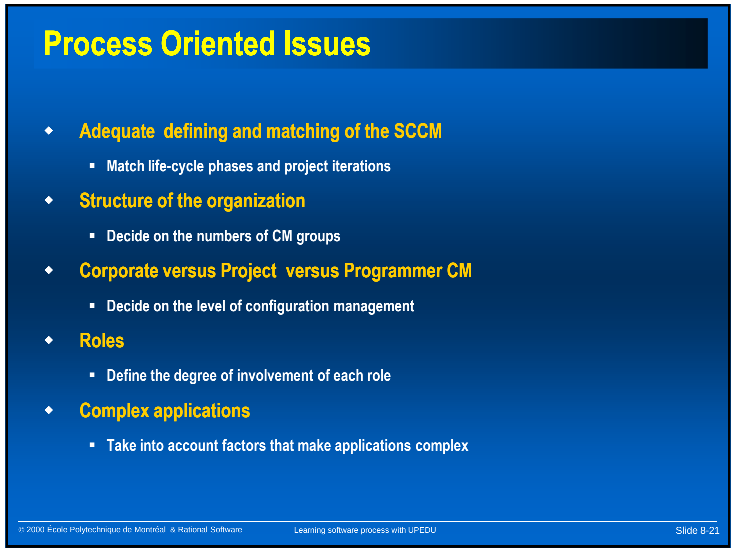
© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 8-19



Technological Issues

- Technology adequacy
  - No silver bullet
- Switching CM capabilities:
  - Enable easy customizing
- Interoperability between CM systems
  - Support various CM systems
- Integration and database
  - Centralize or distribute repository
- Upward compatibility
  - Maintain usefulness for product lifetime

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 8-20



Process Oriented Issues

- Adequate defining and matching of the SCCM
  - Match life-cycle phases and project iterations
- Structure of the organization
  - Decide on the numbers of CM groups
- Corporate versus Project versus Programmer CM
  - Decide on the level of configuration management
- Roles
  - Define the degree of involvement of each role
- Complex applications
  - Take into account factors that make applications complex