

## Several Techniques for Reviewing Code

- ♦ **Code inspection**
  - Formal evaluation technique
  - Team consensus on code quality
- ♦ **Walkthrough**
  - Author leads reviewers through the code
  - Cognitive synchronization of team members
- ♦ **Code reading**
  - Individual read the code according to standard practices
  - Code compliance to programming guide

UPEDU: Work GUIDELINE: REVIEW

© 2000 École Polytechnique de Montréal & Rational Software

Learning software process with UPEDU

Slide 6-9

## Several Techniques for Reviewing Code

- ♦ **Code inspection**
  - Formal evaluation technique
  - Team consensus on code quality
- ♦ **Walkthrough**
  - Author leads reviewers through the code
  - Cognitive synchronization of team members
- ♦ **Code reading**
  - Individual read the code according to standard practices
  - Code compliance to programming guide

UPEDU: Work GUIDELINE: REVIEW

© 2000 École Polytechnique de Montréal & Rational Software

Learning software process with UPEDU

Slide 6-10

## Components are Divided into Two Types

### Deliverables

<b>Executables</b>	.exe files
<b>Load libraries</b>	.dll files
<b>Applets</b>	.class for Java
<b>Web pages</b>	.htm and .html files
<b>Database tables</b>	

### Deliverables produced from components

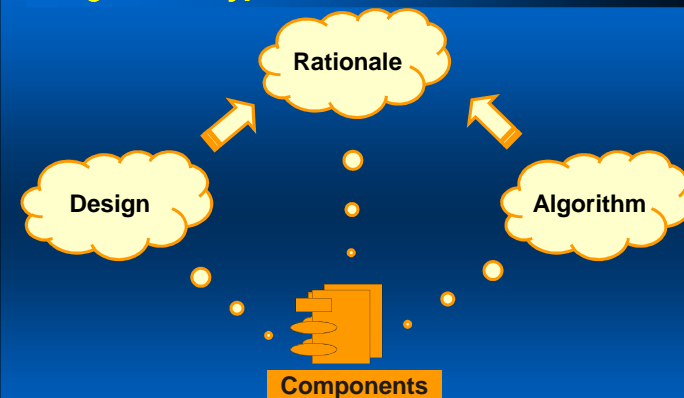
<b>Source code files</b>	.h, .cpp and .hpp files for C++
<b>Binary files</b>	.o and .a files linked into exec
<b>Build files</b>	makefiles

© 2000 École Polytechnique de Montréal & Rational Software

Learning software process with UPEDU

Slide 6-11

## Merge Three Types of Information

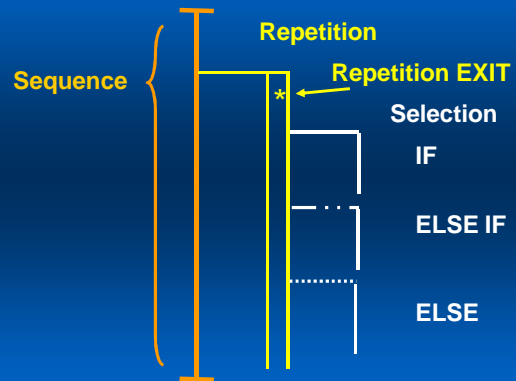


© 2000 École Polytechnique de Montréal & Rational Software

Learning software process with UPEDU

Slide 6-12

## Schematic Pseudocode



© 2000 École Polytechnique de Montréal &amp; Rational Software

Learning software process with UPEDU

Slide 6-13

## Programming Guidelines

- ♦ Establish under the responsibility of the organization
- ♦ Adapt to the organizational needs
- ♦ Define level of source code documentation
- ♦ Specify naming convention for file and variable names
- ♦ Explain restricted use of programming features
- ♦ Create iteratively in collaboration with the team

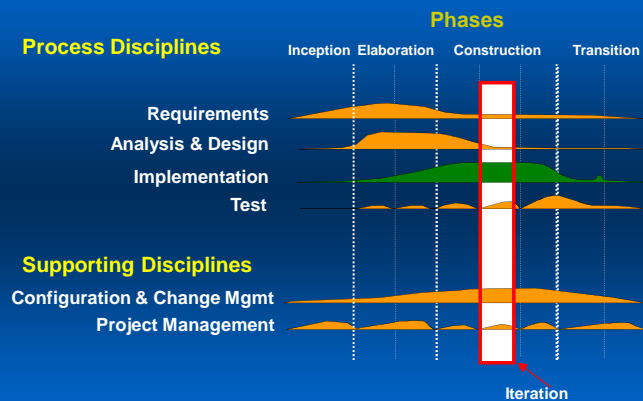
**UPEDU Concept: Mapping from design to Code**  
**UPEDU Guideline: Programming Guide**

© 2000 École Polytechnique de Montréal &amp; Rational Software

Learning software process with UPEDU

Slide 6-14

## Iteration Plan of the Construction Phase



© 2000 École Polytechnique de Montréal &amp; Rational Software

Learning software process with UPEDU

Slide 6-15

## Artifacts of the Construction Phase

- ♦ **Software Development Plan**
    - (updated for the iteration)
  - ♦ **Use-Case Model**
    - (updated)
  - ♦ **Design Model**
  - ♦ **Implementation Model**
  - ♦ **Test Model**
  - ♦ **Change Requests**
- For each iteration:**
- Test Plan
  - Iteration Plan
  - End-User Support Material (preliminary)
  - Installation Artifacts
  - Release Notes

© 2000 École Polytechnique de Montréal &amp; Rational Software

Learning software process with UPEDU

Slide 6-16

### Objectives of this Iteration

- ♦ Requirements are stable.
- ♦ The architecture is fully implemented.
- ♦ Many functionalities have been implemented and integrated
- ♦ Most of the effort will be spent in the implementation and test disciplines
- ♦ The project is approaching its first "beta" release

**UPEDU Concepts: Development and  
Integration Workspaces**

© 2000 École Polytechnique de Montréal & Rational Software

Learning software process with UPEDU

Slide 6-17