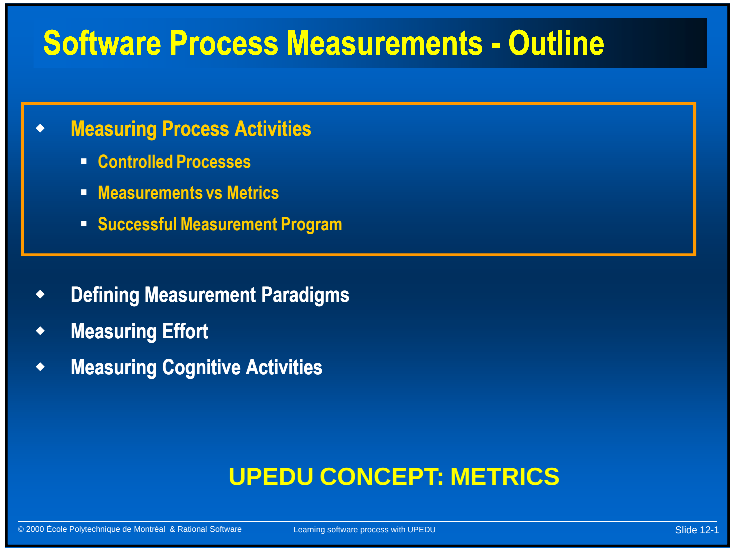
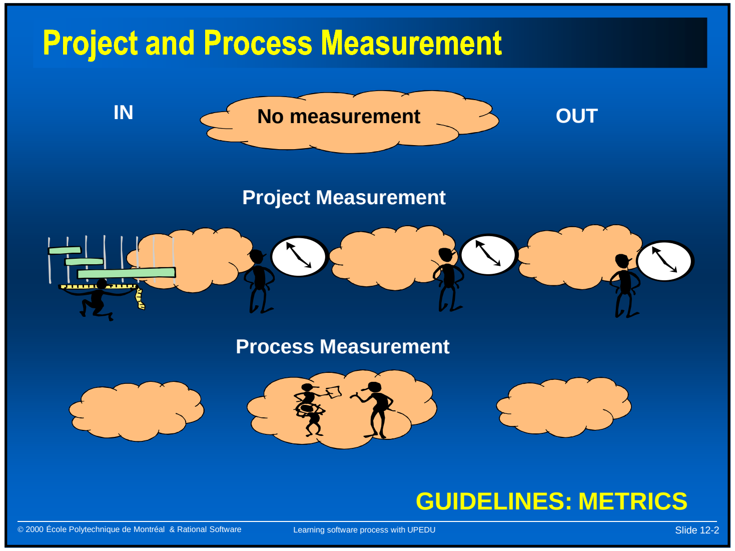


Software Process Measurements - Outline

- Measuring Process Activities
  - Controlled Processes
  - Measurements vs Metrics
  - Successful Measurement Program
- Defining Measurement Paradigms
- Measuring Effort
- Measuring Cognitive Activities

UPEDU CONCEPT: METRICS

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 12-1



Project and Process Measurement

IN

No measurement

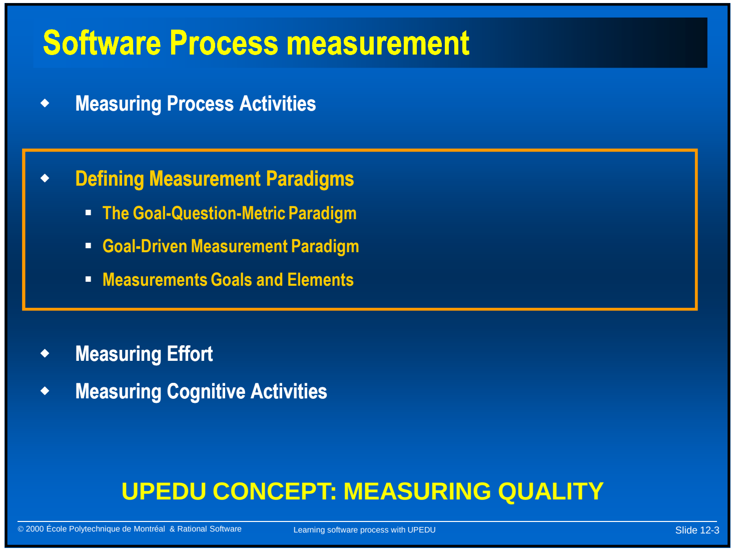
OUT

Project Measurement

Process Measurement

GUIDELINES: METRICS

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 12-2

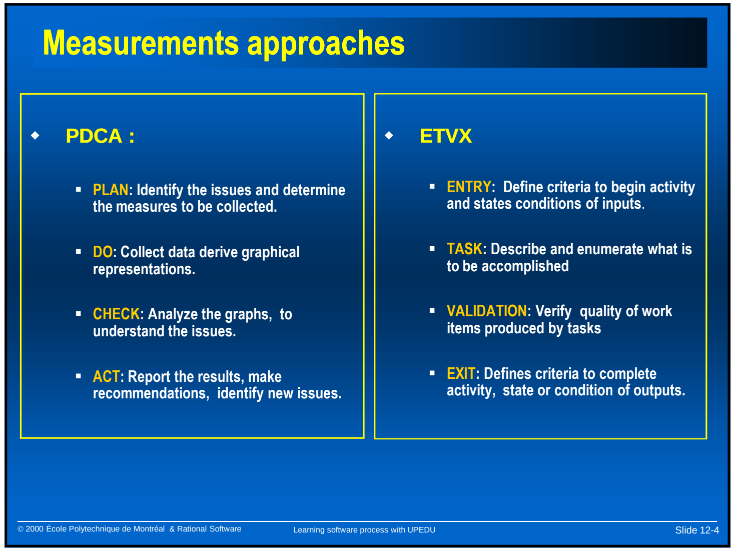


Software Process measurement

- Measuring Process Activities
- Defining Measurement Paradigms
  - The Goal-Question-Metric Paradigm
  - Goal-Driven Measurement Paradigm
  - Measurements Goals and Elements
- Measuring Effort
- Measuring Cognitive Activities

UPEDU CONCEPT: MEASURING QUALITY

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 12-3

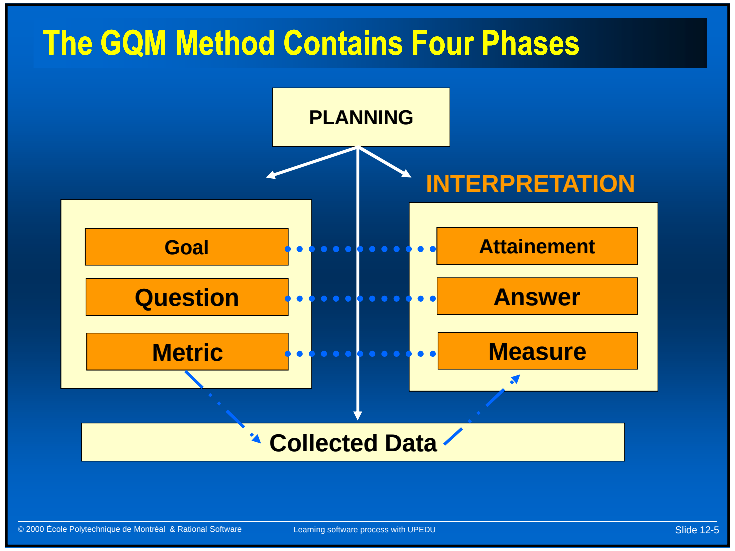


Measurements approaches

- PDCA :
  - PLAN: Identify the issues and determine the measures to be collected.
  - DO: Collect data derive graphical representations.
  - CHECK: Analyze the graphs, to understand the issues.
  - ACT: Report the results, make recommendations, identify new issues.

- ETVX
  - ENTRY: Define criteria to begin activity and states conditions of inputs.
  - TASK: Describe and enumerate what is to be accomplished
  - VALIDATION: Verify quality of work items produced by tasks
  - EXIT: Defines criteria to complete activity, state or condition of outputs.

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 12-4



The GQM Method Contains Four Phases

PLANNING

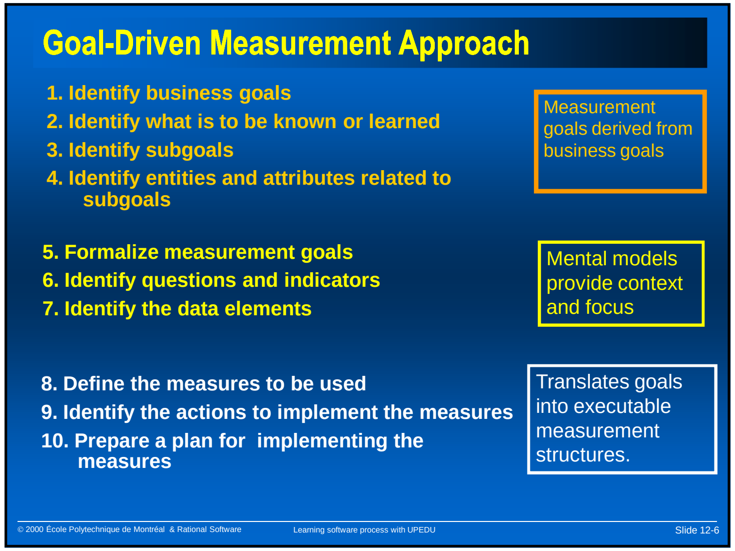
INTERPRETATION

Goal — Attainement

Question — Answer

Metric — Measure

Collected Data



Goal-Driven Measurement Approach

1. Identify business goals
2. Identify what is to be known or learned
3. Identify subgoals
4. Identify entities and attributes related to subgoals

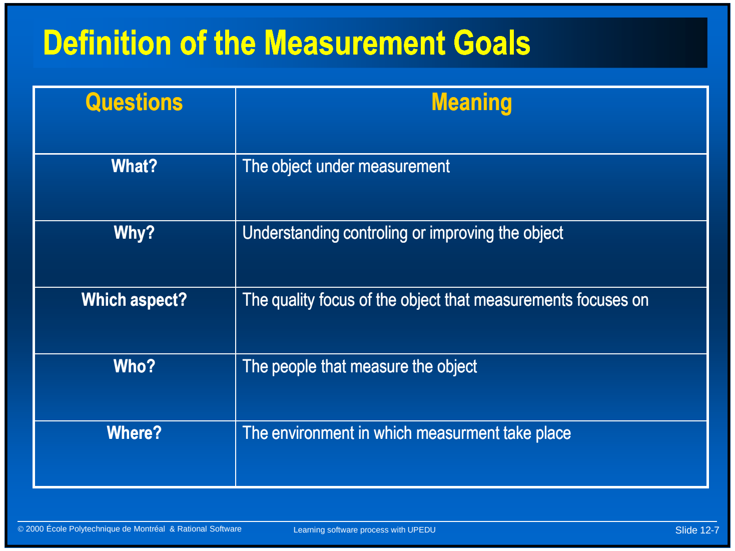
Measurement goals derived from business goals

5. Formalize measurement goals
6. Identify questions and indicators
7. Identify the data elements

Mental models provide context and focus

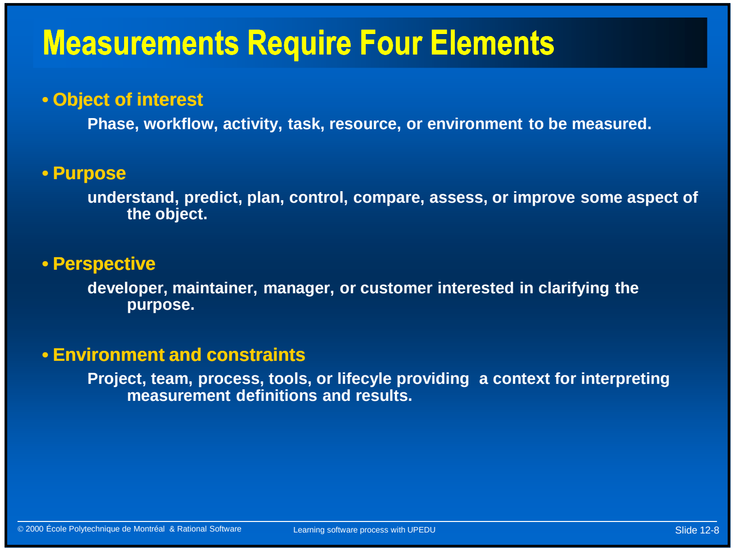
8. Define the measures to be used
9. Identify the actions to implement the measures
10. Prepare a plan for implementing the measures

Translates goals into executable measurement structures.



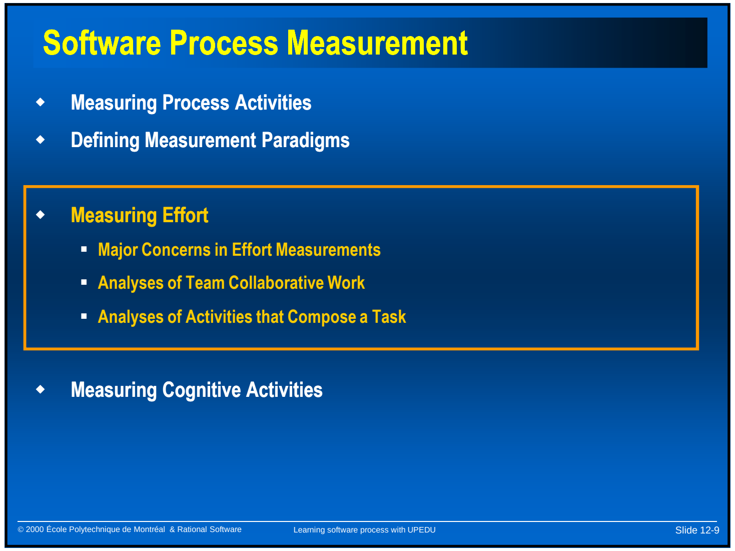
Definition of the Measurement Goals

Questions	Meaning
What?	The object under measurement
Why?	Understanding controling or improving the object
Which aspect?	The quality focus of the object that measurements focuses on
Who?	The people that measure the object
Where?	The environment in which measurment take place



Measurements Require Four Elements

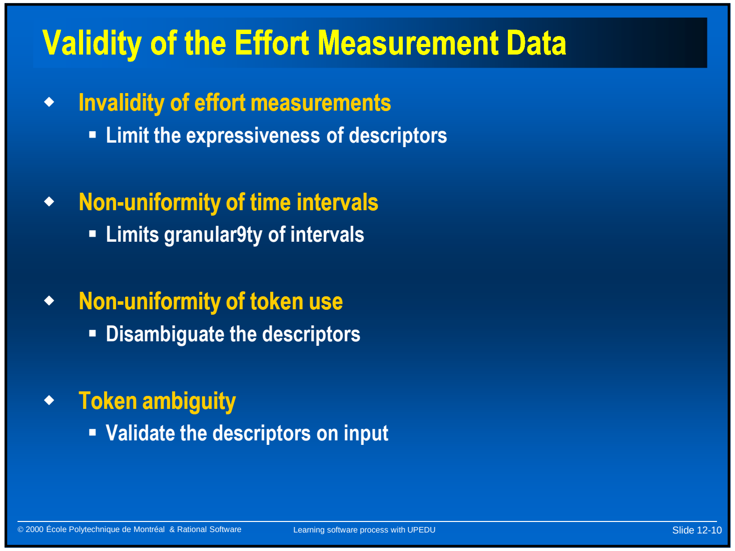
- Object of interest
  Phase, workflow, activity, task, resource, or environment to be measured.
- Purpose
  understand, predict, plan, control, compare, assess, or improve some aspect of the object.
- Perspective
  developer, maintainer, manager, or customer interested in clarifying the purpose.
- Environment and constraints
  Project, team, process, tools, or lifecycle providing a context for interpreting measurement definitions and results.



Software Process Measurement

- Measuring Process Activities
- Defining Measurement Paradigms
- Measuring Effort
  - Major Concerns in Effort Measurements
  - Analyses of Team Collaborative Work
  - Analyses of Activities that Compose a Task
- Measuring Cognitive Activities

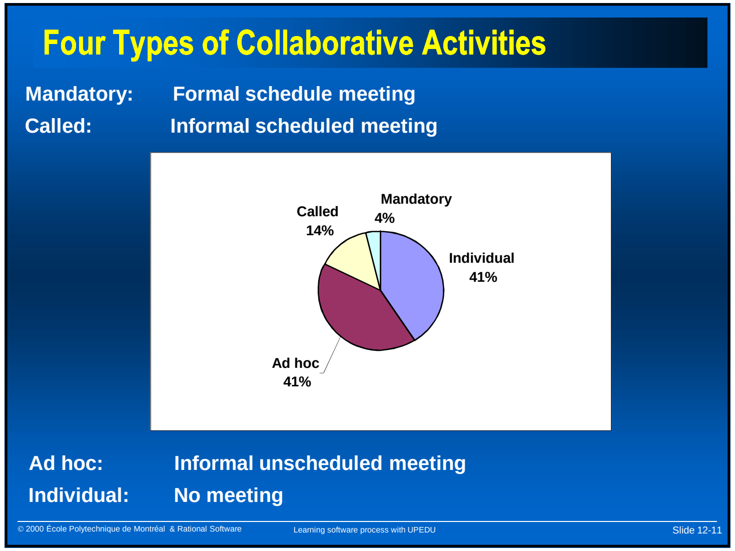
© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 12-9



Validity of the Effort Measurement Data

- Invalidity of effort measurements
  - Limit the expressiveness of descriptors
- Non-uniformity of time intervals
  - Limits granular9ty of intervals
- Non-uniformity of token use
  - Disambiguate the descriptors
- Token ambiguity
  - Validate the descriptors on input

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 12-10



Four Types of Collaborative Activities

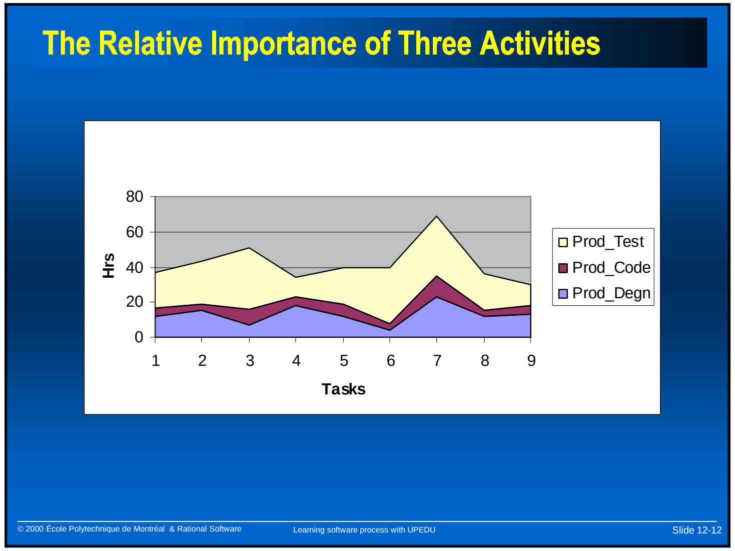
Mandatory: Formal schedule meeting

Called: Informal scheduled meeting

Ad hoc: Informal unscheduled meeting

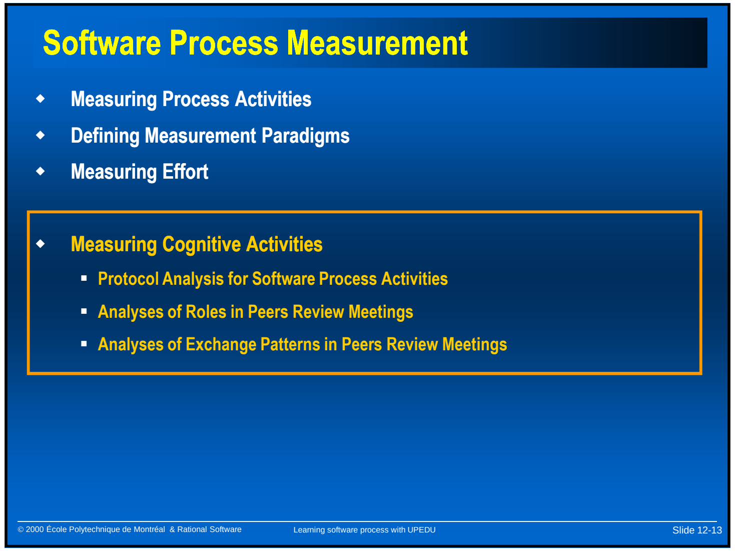
Individual: No meeting

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 12-11



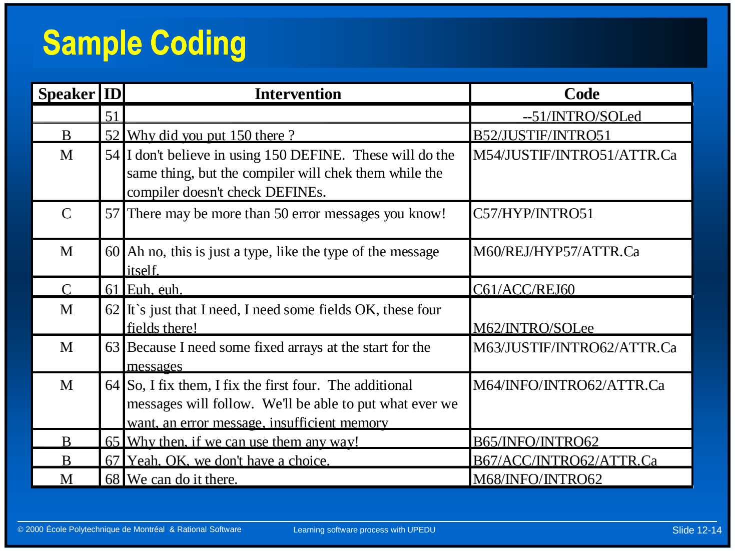
The Relative Importance of Three Activities

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Slide 12-12



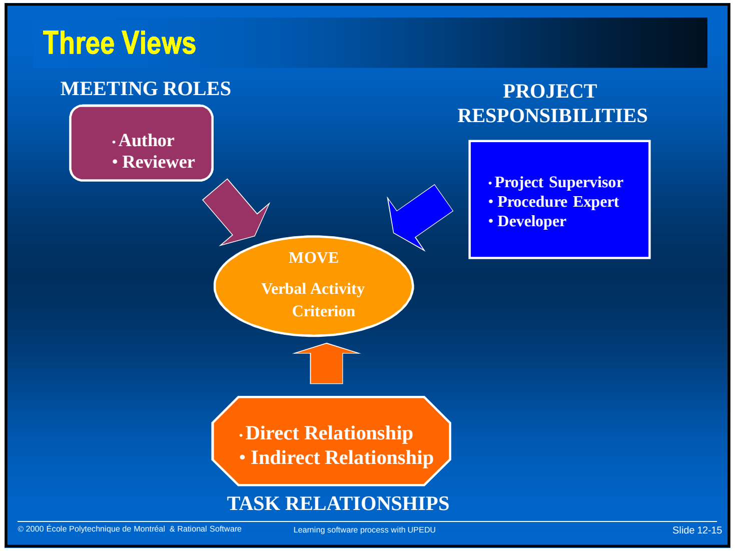
Software Process Measurement

- Measuring Process Activities
- Defining Measurement Paradigms
- Measuring Effort
- Measuring Cognitive Activities
  - Protocol Analysis for Software Process Activities
  - Analyses of Roles in Peers Review Meetings
  - Analyses of Exchange Patterns in Peers Review Meetings



Sample Coding

Speaker	ID	Intervention	Code
	51		--51/INTRO/SOLed
B	52	Why did you put 150 there ?	B52/JUSTIF/INTRO51
M	54	I don't believe in using 150 DEFINE. These will do the same thing, but the compiler will chek them while the compiler doesn't check DEFINEs.	M54/JUSTIF/INTRO51/ATTR.Ca
C	57	There may be more than 50 error messages you know!	C57/HYP/INTRO51
M	60	Ah no, this is just a type, like the type of the message itself.	M60/REJ/HYP57/ATTR.Ca
C	61	Euh, euh.	C61/ACC/REJ60
M	62	It`s just that I need, I need some fields OK, these four fields there!	M62/INTRO/SOLee
M	63	Because I need some fixed arrays at the start for the messages	M63/JUSTIF/INTRO62/ATTR.Ca
M	64	So, I fix them, I fix the first four. The additional messages will follow. We'll be able to put what ever we want, an error message, insufficient memory	M64/INFO/INTRO62/ATTR.Ca
B	65	Why then, if we can use them any way!	B65/INFO/INTRO62
B	67	Yeah, OK, we don't have a choice.	B67/ACC/INTRO62/ATTR.Ca
M	68	We can do it there.	M68/INFO/INTRO62



Three Views

MEETING ROLES

- Author
- Reviewer

PROJECT RESPONSIBILITIES

- Project Supervisor
- Procedure Expert
- Developer

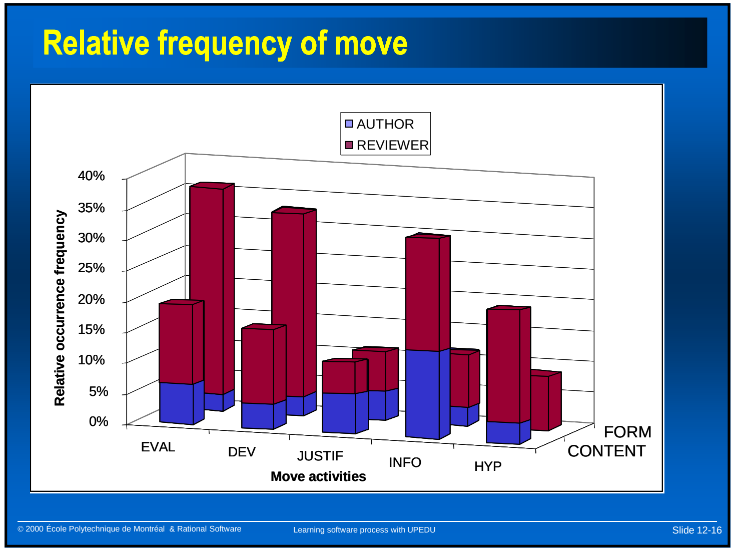
MOVE

Verbal Activity

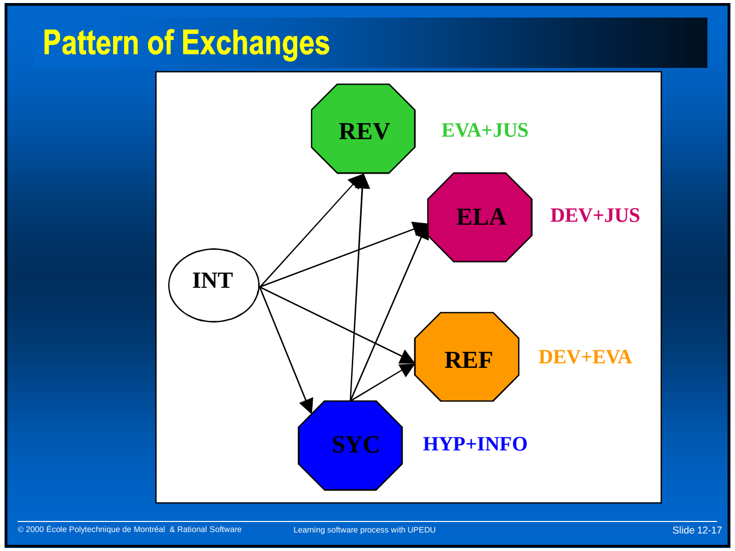
Criterion

- Direct Relationship
- Indirect Relationship

TASK RELATIONSHIPS



Relative frequency of move



Pattern of Exchanges

REV EVA+JUS

ELA DEV+JUS

INT

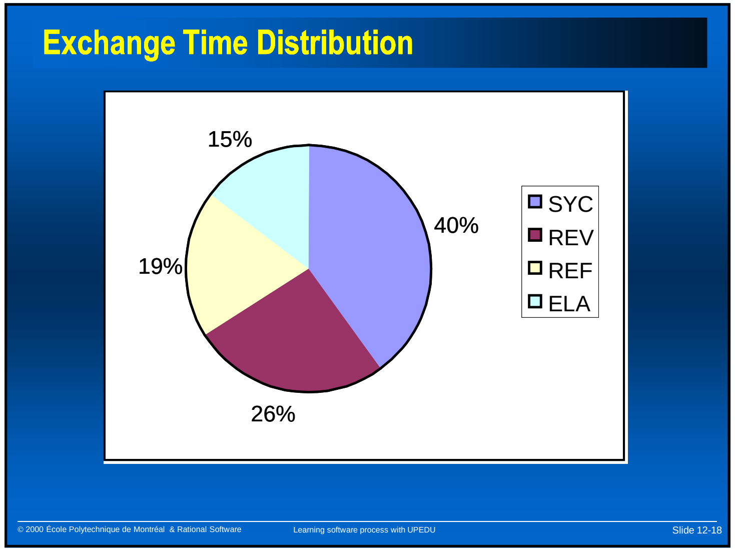
REF DEV+EVA

SYC HYP+INFO

© 2000 École Polytechnique de Montréal & Rational Software

Learning software process with UPEDU

Slide 12-17



Exchange Time Distribution

© 2000 École Polytechnique de Montréal & Rational Software

Learning software process with UPEDU

Slide 12-18