

UPEDU : Unified Process for Education

PARTIE 3 :
Gestion du projet logiciel
Workflow de l'équipe
Evaluation du processus
Dimensionnement du processus

Gestion du projet logiciel

◆ Définition de rôle de gestionnaire de projet

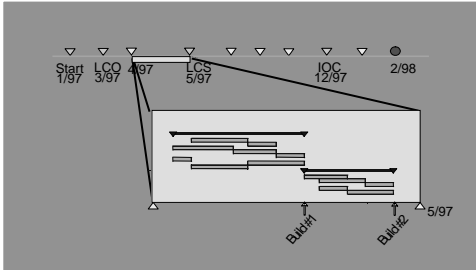
- Le gestionnaire de projet
- Les artefacts et jalons d'un projet

◆ Planning des Projets

◆ Gérer les itérations

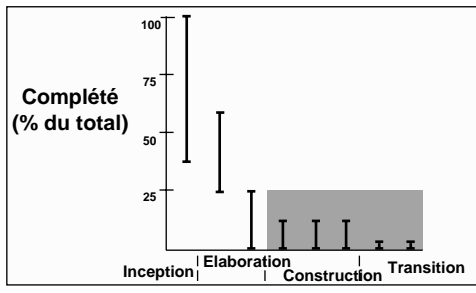
◆ Gérer les risques

Plannings et Contrôles



Planning

Contrôles

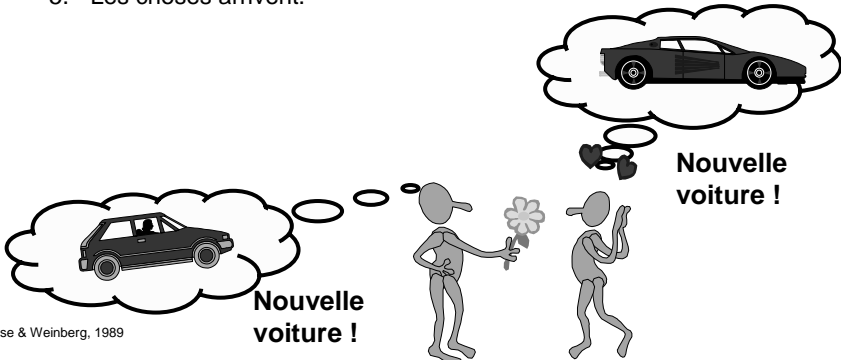


© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-3

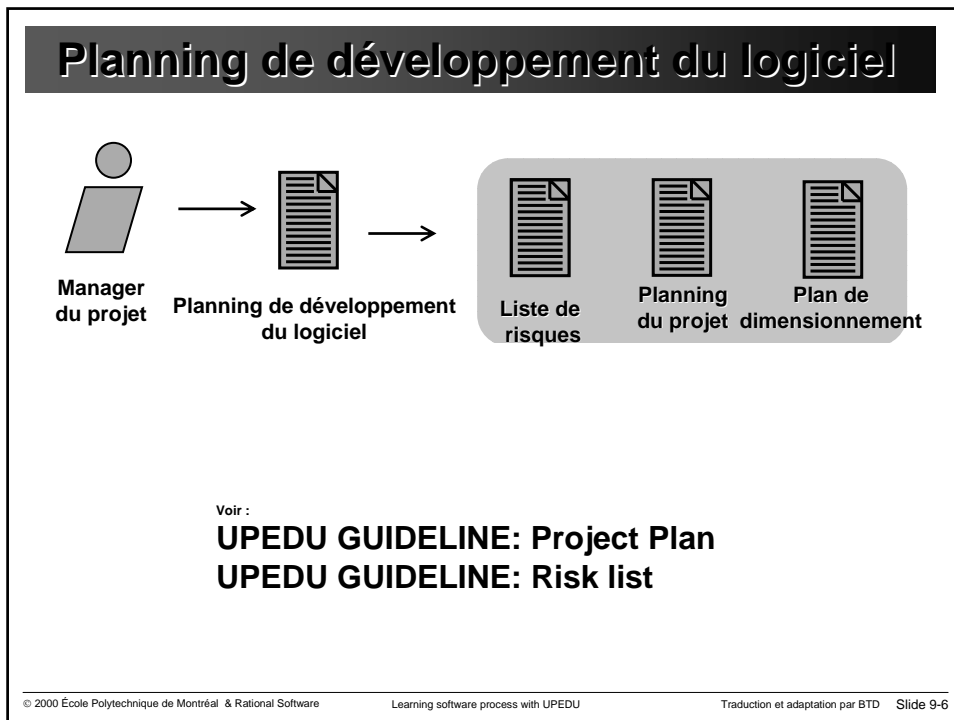
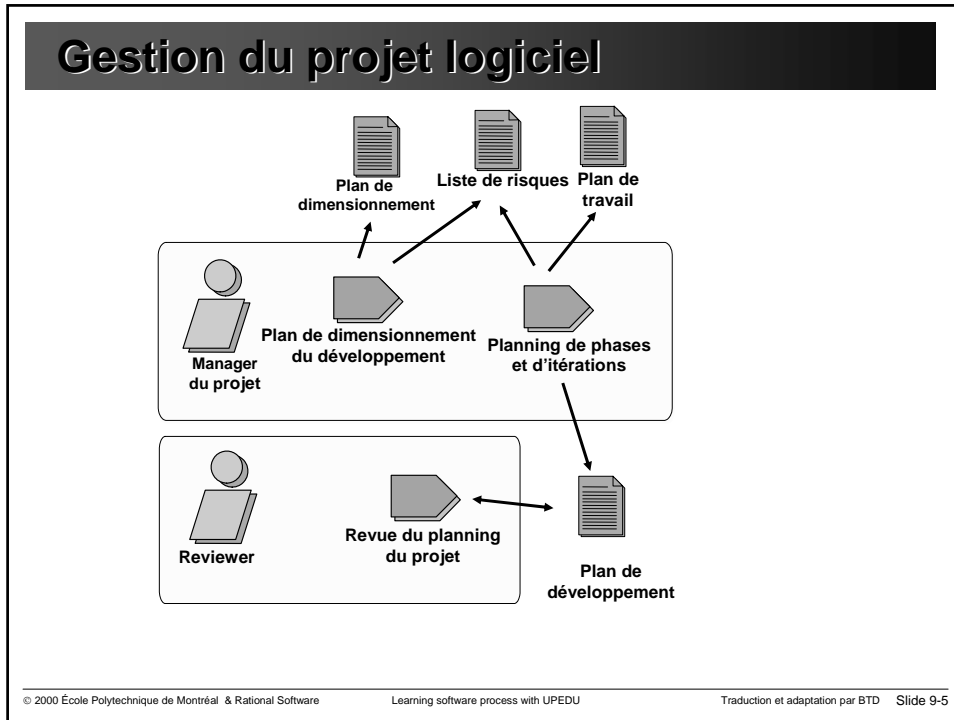
Gérer des attentes

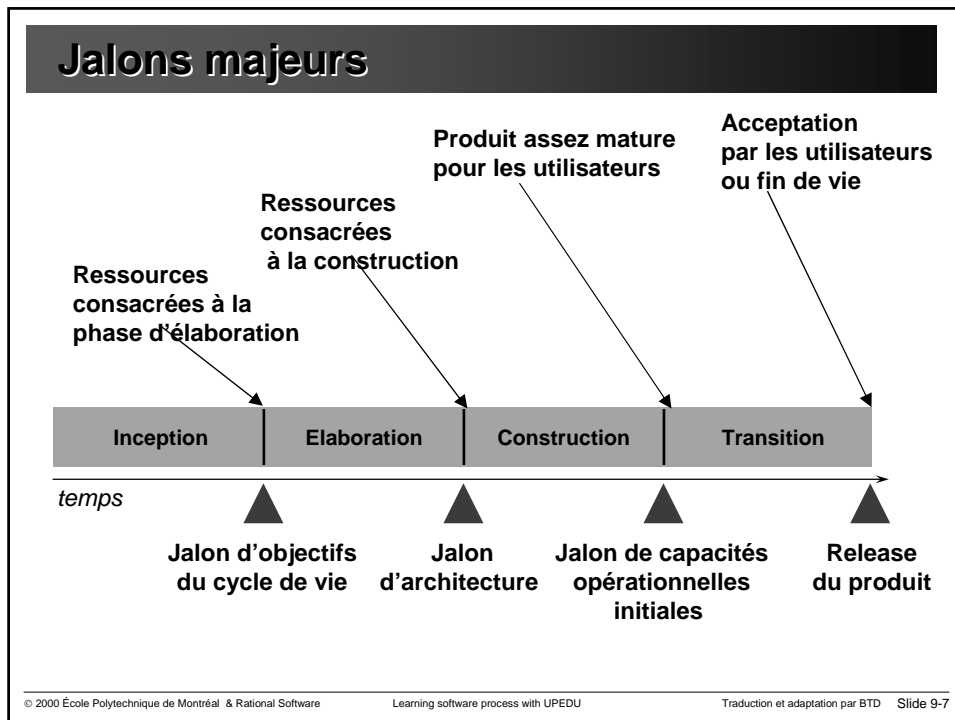
Pourquoi gérer les attentes ?

1. No one is perfect.
2. Les humains ne sont pas logiques.
3. Les humains perçoivent des choses différemment .
4. Ingénieurs logiciels sont également des humains.
5. Les choses arrivent.



© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-4





Gestion du projet logiciel

- ◆ Définition de rôle de gestionnaire de projet

- ◆ **Planning des Projets**
 - Organisation du projet
 - Stratégies du planning
 - Les artéfacts du planning du projet

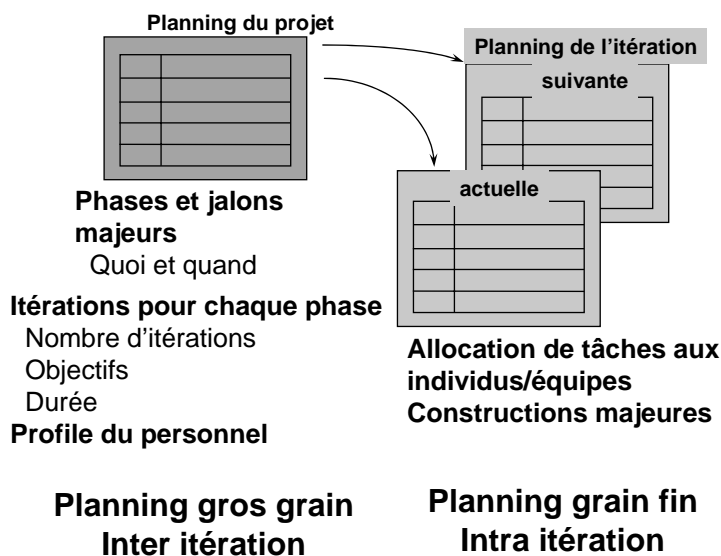
- ◆ Gérer les itérations
- ◆ Gérer les risques

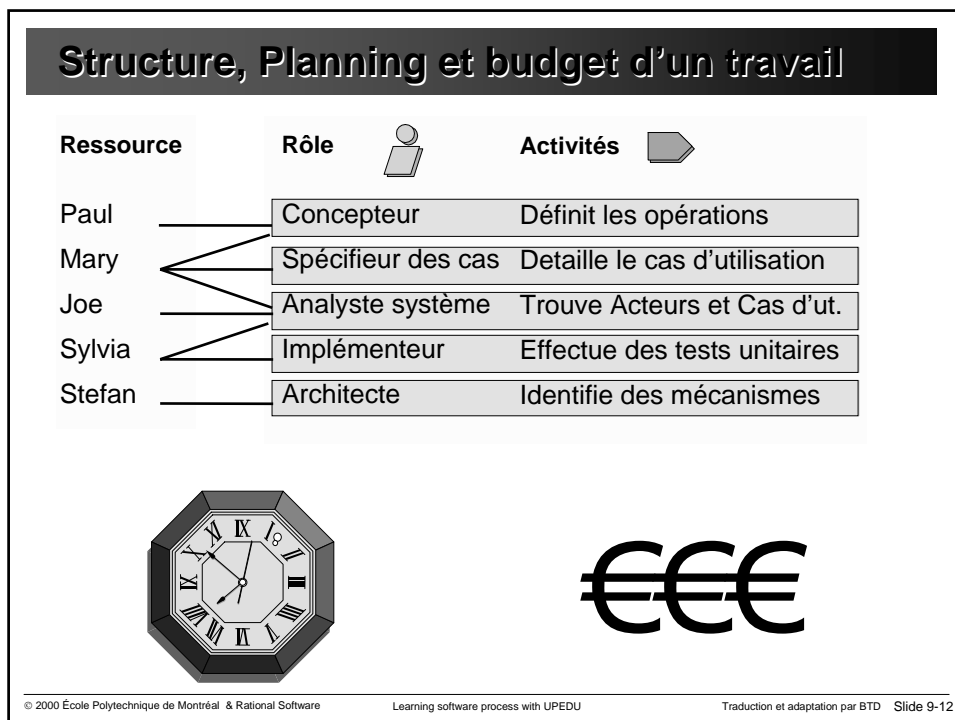
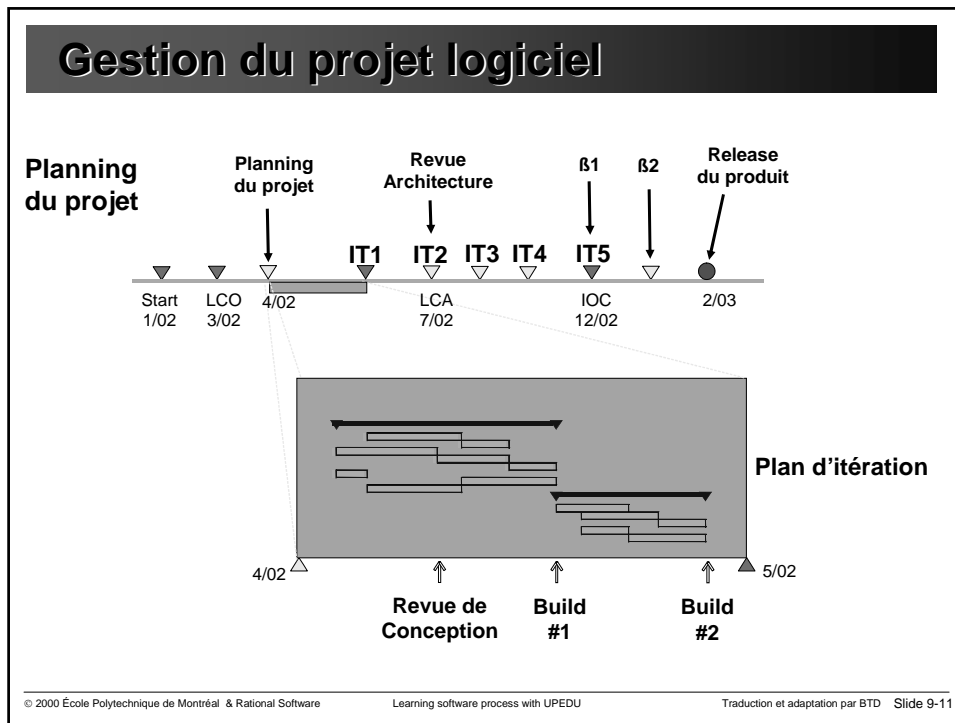
© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-8

Nombreux facteurs à prendre en compte

- ◆ **L'importance de l'effort de développement**
 - Formalité, standards, rigidité des processus.
- ◆ **Le degré de nouveauté**
 - Premier de ce type, Cycle d'évolution, Maintenance
- ◆ **Type d'application**
 - Mission critique, performance, contraintes mémoire,
- ◆ **Processus de développement courant**
 - Maturité du processus, expérience des développeurs et des managers
- ◆ **Facteurs d'organisation**
 - Attitudes de l'équipe envers de changements, enthousiasme de l'équipe par rapport au projet
- ◆ **Complexité technique et managérial**
 - Taille de l'équipe vs taille du projet

Planning Top-down et Bottom-up





Gestion du projet logiciel

◆ Définition de rôle de gestionnaire de projet

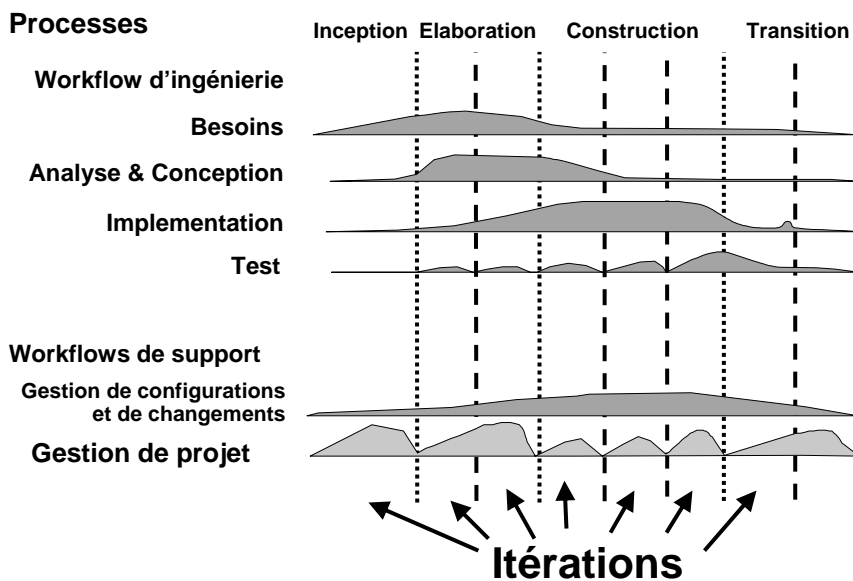
◆ Planning des Projets

◆ Gérer les itérations

- Définition d'une itération
- Etendu d'une itération
- Evaluation d'une itération

◆ Gérer les risques

Une itération



L'étendu de l'itération dépend des facteurs

- ♦ Combien d'itération on devrait trouver dans le planning du projet ?

	Total	I	E	C	T
Bas	3	0	1	1	1
Typique	6	1	2	2	1
Haut	9	1	3	3	2

- ♦ Quelle durée chaque itération devrait avoir ?
 - Ceci dépend de plusieurs facteurs :
 - La taille du système en construction : plus grand est le système, plus longue sera l'itération.
 - Nombre de personnes évoluant dans le projet : plus grand nombre implique une durée plus grande.

Voir :

UPEDU Concept: ITERATION

Basé sur des métriques

Métrique	Signification
Progrès	Taille et complexité
Stabilité	Taux de changement selon la complexité du projet et de sa taille
Modularité	Etendu de la charge
Qualité	Nombre d'erreurs
Maturité	Fréquence d'erreurs
Dépenses	Dépenses du projet par rapport au planning

Gestion du projet logiciel

- ◆ Définition de rôle de gestionnaire de projet
- ◆ Planning des Projets
- ◆ Gérer les itérations

◆ Gérer les risques

- Définition de risque
- Stratégies de gestion de risques
- Evaluation de risques

Les concepts associés à la définition du risque


- ◆ **Risque :**
 - Information manquante concernant un sujet donné
- ◆ **Risque indirect :**
 - Le projet a faible contrôle ou aucun
- ◆ **Risque direct :**
 - Projet a un contrôle direct
- ◆ **Attributs du risque :**
 - Probabilité que les circonstances se produisent
- ◆ **Indicateur de l'importance du risque :**
 - Haut, significatif, modéré, mineur, bas


Voir :


UPEDU Concept: RISK


Stratégies de gestion de risques


- Accepter le risque**
Vivre avec


- Eviter le risque**
Réorganiser pour éliminer le risque


- Atténuer le risque**
Réduire sa probabilité ou impact


- Eventualité du risque**
Choisir "Plan B"

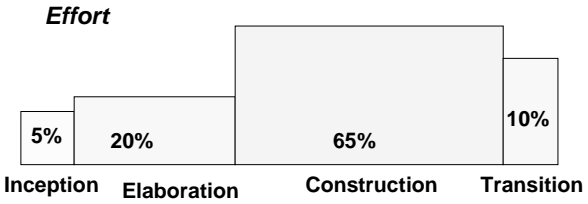

- Transfert de risque**
Sous-traiter le risque



© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-19

Dimensionnement et Métriques

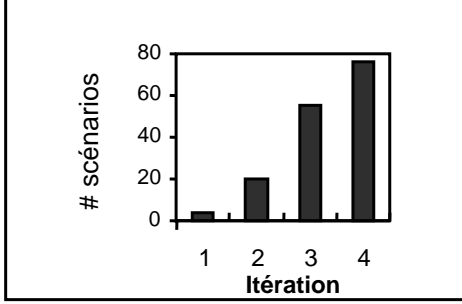
Effort



Phase	Effort (%)
Inception	5%
Elaboration	20%
Construction	65%
Transition	10%

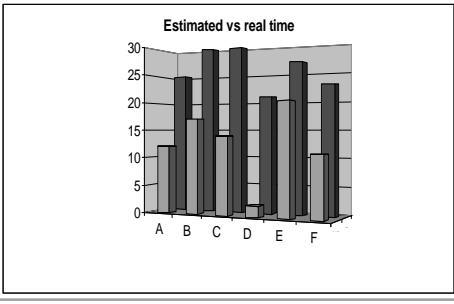
Temps

scénarios



Itération	# scénarios
1	5
2	20
3	55
4	75

Estimated vs real time



Catégorie	Estimated	Real Time
A	12	24
B	17	29
C	14	29
D	21	20
E	27	11
F	23	23

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-20

Workflow de l'équipe

- ◆ Comprendre les équipes
 - Aspects du travail en équipe
 - Organisation d'équipe

- ◆ Structuration des environnements d'équipes
- ◆ Définition d'équipes
- ◆ Bâtir une équipe
- ◆ Supporter les activités de l'équipe

Les challenges de développement de logiciels

Challenges

- Equipes importantes
- Spécialisation
- Distribution
- Changement rapide des technologies



Equipe des clients utilisateur



Equipe du chef programmeur



Equipe de testeurs



Equipe de secours



Equipe démocratique

Comment les personnes peuvent travailler ensemble

The diagram illustrates three ways people can work together:

- Développement personnel**: Five individual figures, each holding a document, representing personal development.
- Groupe partageant les outils**: A group of figures gathered around a computer monitor and keyboard, representing a team sharing tools.
- Equipe composite**: A group of figures of various sizes and colors (white, grey, black) standing together, representing a composite team.

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-23

Workflow de l'équipe

- ◆ Comprendre les équipes
 - ◆ Structuration des environnements d'équipes
 - Structure de gestion d'équipe
 - Structure des personnes
 - Structure de l'espace partagé
 - ◆ Définition d'équipes
 - ◆ Bâtir une équipe
 - ◆ Supporter les activités de l'équipe

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-24

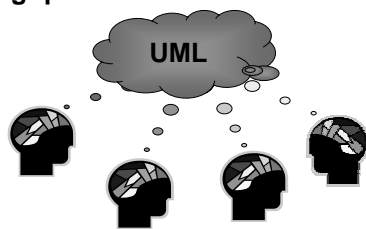
Récompenses et structures incitatives

- ◆ Développer une attitude d'équipe
- ◆ Eviter trop de vedettes
- ◆ Fournir un modèle d'équipe

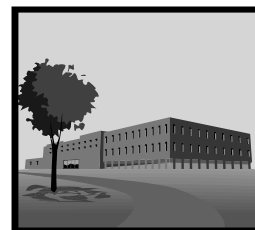


Le concept de l'espace partagé

Logique



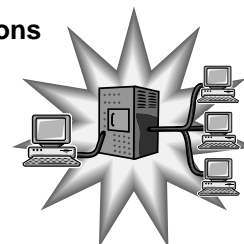
Physique



Réunion



Communications



Workflow de l'équipe

Comprendre les équipes

- ◆ Structuration des environnements d'équipes

- ◆ Définition d'équipes

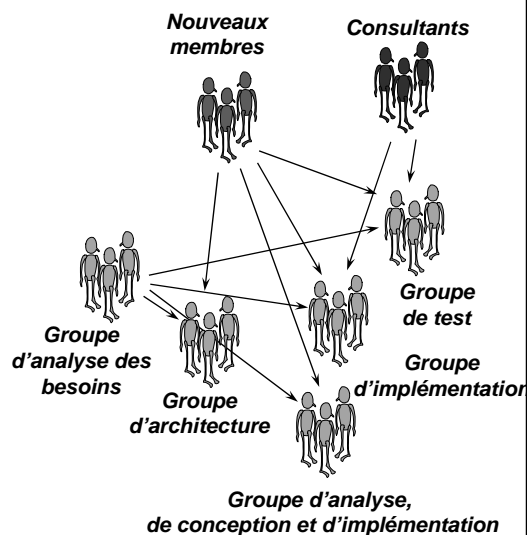
- Distinguer un groupe d'une équipe
- Des fondamentaux d'une équipe
- Organisation et comportement d'équipe

- ◆ Bâtir une équipe

- ◆ Supporter les activités de l'équipe

Un groupe typique est un ensemble d'individus

- ◆ Les mêmes personnes depuis l'analyse des besoins jusqu'à l'implémentation
 - Minimiser le renouvellement ou rotation du personnel
- ◆ Une équipe équilibrée
- ◆ Un général, des équipe devraient être composées de 5-7 personnes



Les pré-requis pour bâtir une équipe

- ♦ Adhérer aux objectifs de l'équipe
- ♦ Respecter le plan de travail
- ♦ Identifier les rôles des membres de l'équipe
- ♦ Dégager le processus de travail commun
- ♦ Souscrire à l'engagement mutuel au sein de l'équipe par rapport aux buts, rôles et plan de travail
- ♦ Disposer d'un environnement support pour chacun
- ♦ Permettre une communication libre entre les membres de l'équipe
- ♦ Avoir un respect mutuel et une volonté d'entraide entre les membres de l'équipe



Voir :

UPEDU GUIDELINE: USE-CASE Workshop

Comportement d'équipe au sein de la structure d'organisation

Double en Tennis



Chaque membre s'adapte à l'autre

Equipe de Hockey



Toute l'équipe agit ensemble

Equipe de Baseball



Chaque joueur a une position fixée

Workflow de l'équipe

- ◆ Comprendre les équipes
- ◆ Structuration des environnements d'équipes
- ◆ Définition d'équipes

- ◆ Bâtir une équipe
 - Modèle de développement de l'équipe en quatre stades
 - Les 5 phases pour construire un esprit d'équipe
 - Trois étapes pour créer une équipe composite (jelled team)

- ◆ Supporter les activités de l'équipe

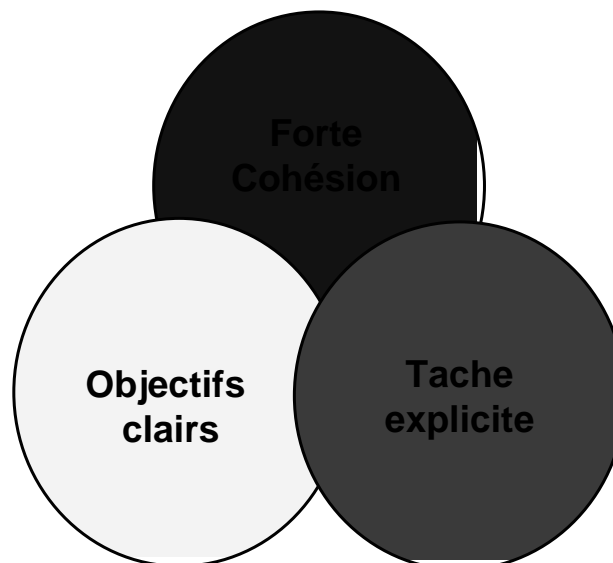
Modèle de développement d'équipe en 4 stades

- ◆ Former (Forming)
 - Le groupe cherche ses objectifs et sa pertinence. Les relations sociales s'établissent. Le groupe commence à travailler avec un chef d'équipe.
- ◆ Gérer les conflits (Storming)
 - Des conflits émergent entre membres dus aux différentes perceptions des objectifs, et la lutte pour le contrôle et les orientations.
- ◆ Normer (Norming)
 - Le sens commun et les intérêts partagés sont reconnus les processus de communication établis.
- ◆ Agir efficacement (Performing)
 - L'identité de l'équipe est dégagée et les membres s'y associent fortement.

Modèle en spirale en 5 phases pour l'adhésion à l'esprit d'équipe

- ◆ **Initier (Initiating)**
 - Partager les intérêts personnels
 - Organiser des activités sociales informelles
- ◆ **Partager (Visioning)**
 - Partager la vision et les objectifs comme un groupe
 - Revoir à plusieurs les travaux du groupe
- ◆ **Revendiquer (Claiming)**
 - S'identifier et se reconnaître dans les rôles et les objectifs de l'équipe
 - Dégager une structure de l'équipe
- ◆ **Célébrer (Celebrating)**
 - Reconnaître des capacités et talents de l'équipe et de ses individus
 - Célébrer par les événements sociaux les franchissements des jalons du projet
- ◆ **Agir (Letting go)**
 - Avoir un retour constructif vers les membres de l'équipe
 - Dégager une précieuse sensation de fidélité

Équipe cristallisée (Jelled Team)



Les membres de l'équipe ne sont pas interchangeables

- ◆ Définir les rôles
- ◆ Avoir des compétences appropriées
- ◆ Disposer d'outils appropriés

The diagram illustrates six team roles arranged in a circular path, each represented by a group of stylized human figures. The roles are: Analyste (top-left), Ingénieur de performance (top-right), Manager du projet (center), Développeur (right), Gestionnaire de versions (bottom-right), and Testeur (bottom-left). The path is a thick, black, wavy line that connects the roles in a clockwise direction.

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-37

Workflow de l'équipe

- ◆ Comprendre les équipes
- ◆ Structuration des environnements d'équipes
- ◆ Définition d'équipes
- ◆ Bâtir une équipe

- ◆ Supporter les activités de l'équipe
 - Equipes virtuelles
 - Outils d'équipes virtuelles pour les activités collaboratives

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-38

Approche par équipes virtuelles

Outil de « Brain Storming »

- Pour diverger des pensées coutumières



Catégoriseur

- Encourager la convergence des aspects clés



Outil de synthétisation

- Pour structurer des thèmes de manière arborescente.



Outil d'analyses alternatives

- Fournir un environnement pour des décisions multi-critère



Outil de vote

- Fournir des méthodes pour arriver à un consensus ou une décision.



© 2000 École Polytechnique de Montréal & Rational Software

Learning software process with UPEDU

Traduction et adaptation par BTD Slide 9-39

Evaluation du processus logiciel

◆ Définir le modèle de processus et le modèle de référence

- Modèle de processus d'une organisation
- Modèle de référence
- Évaluer un processus logiciel

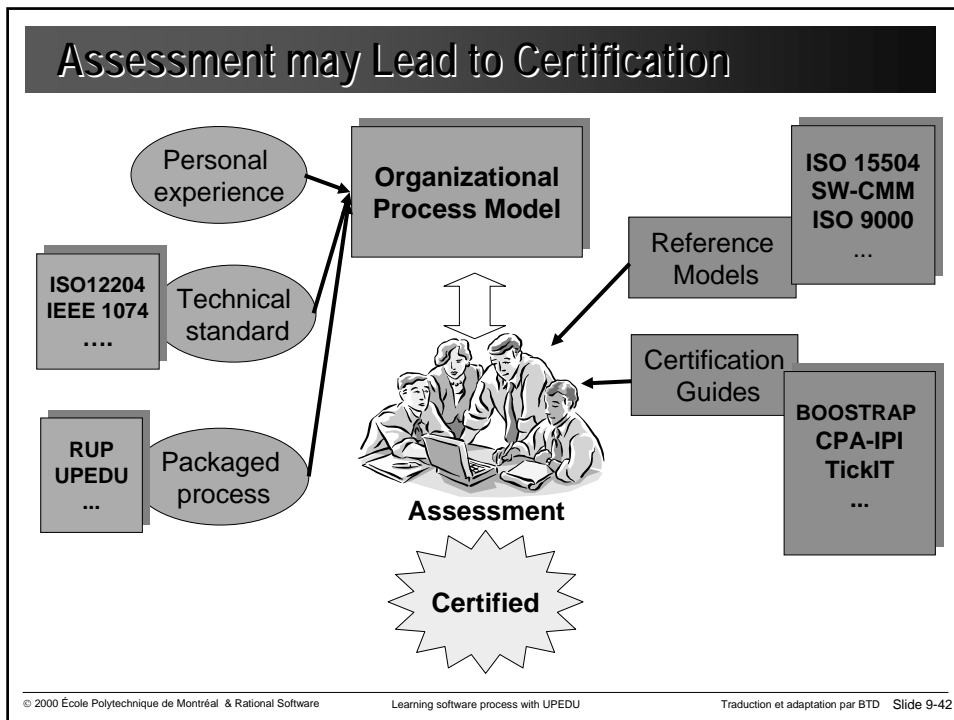
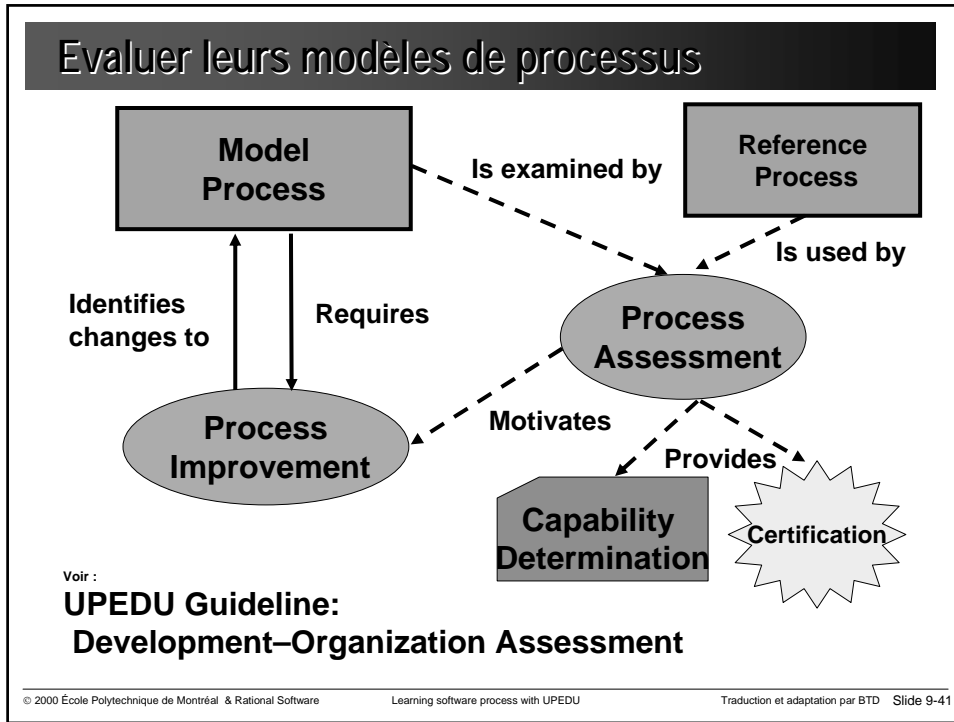
◆ Enregistrement et Certification

◆ Evaluation de UPEDU

© 2000 École Polytechnique de Montréal & Rational Software

Learning software process with UPEDU

Traduction et adaptation par BTD Slide 9-40



Evaluation du processus logiciel

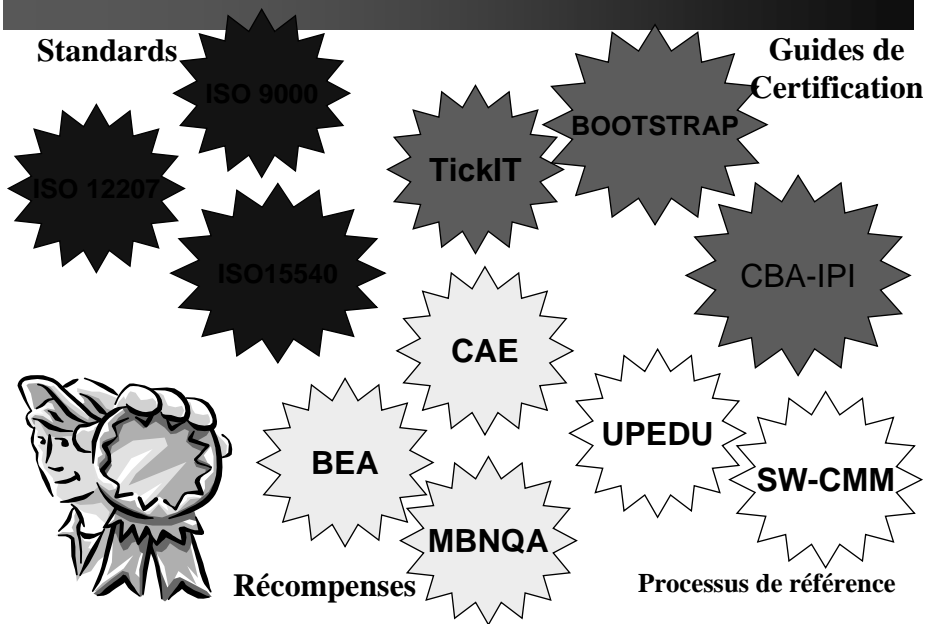
- ◆ Définir le modèle de processus et le modèle de référence

◆ Enregistrement et Certification

- ISO 9001: Quality Systems - Model for Quality Assurance in Design/Development, Production, Installation and, Servicing
- ISO/IEC 15504: Information Technology – Software Process Assessment
- The Software Capability Maturity Model (SW-CMM)

- ◆ Evaluation de UPEDU

Certification comme un mécanisme d'évaluation



ISO 9000-3 Guideline

<ul style="list-style-type: none"> ◆ Projects and maintenance <ul style="list-style-type: none"> ▪ General <ul style="list-style-type: none"> ▪ Contract Review ▪ Purchase's requirement ▪ Development planning ▪ Quality planning ▪ Design and implementation ▪ Testing and validation ▪ Acceptance ▪ Replication, delivery and installation ▪ Maintenance 	<ul style="list-style-type: none"> ◆ General management <ul style="list-style-type: none"> ▪ Management responsibility ▪ Quality system ▪ Internal quality system audits ▪ Corrective action
<ul style="list-style-type: none"> ◆ Supporting activities <ul style="list-style-type: none"> ▪ Configuration management ▪ Document control ▪ Quality records ▪ Measurements ▪ Rules, practices, conventions ▪ Tools and techniques ▪ Purchasing ▪ Included software product ▪ Training 	

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-45

ISO 9000-3 guidance

ISO 9001:1994 says: 4.2.3 Quality planning

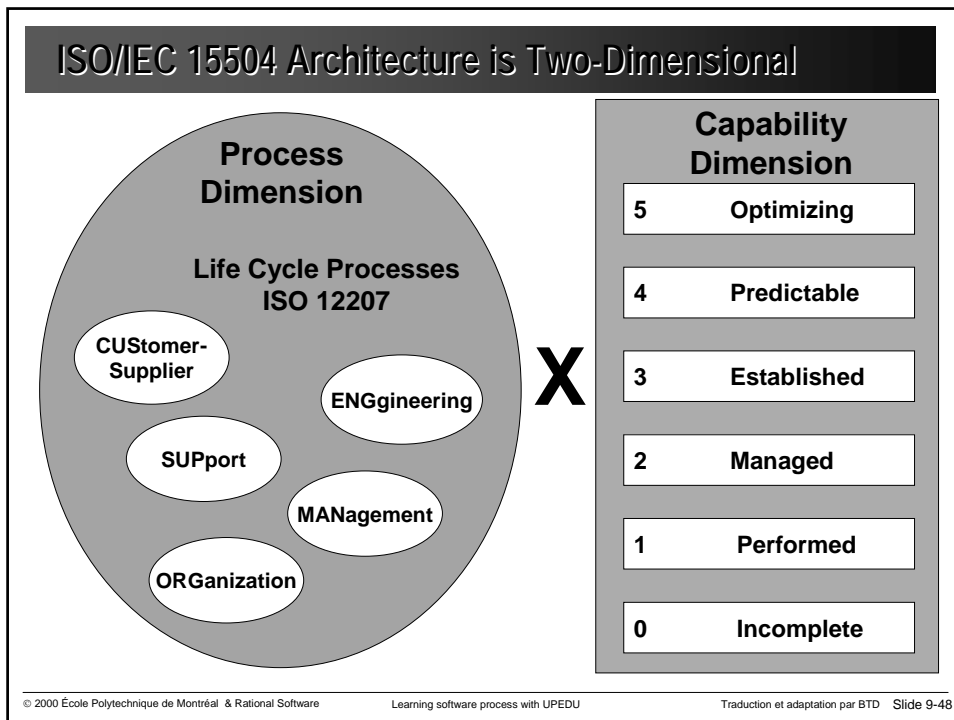
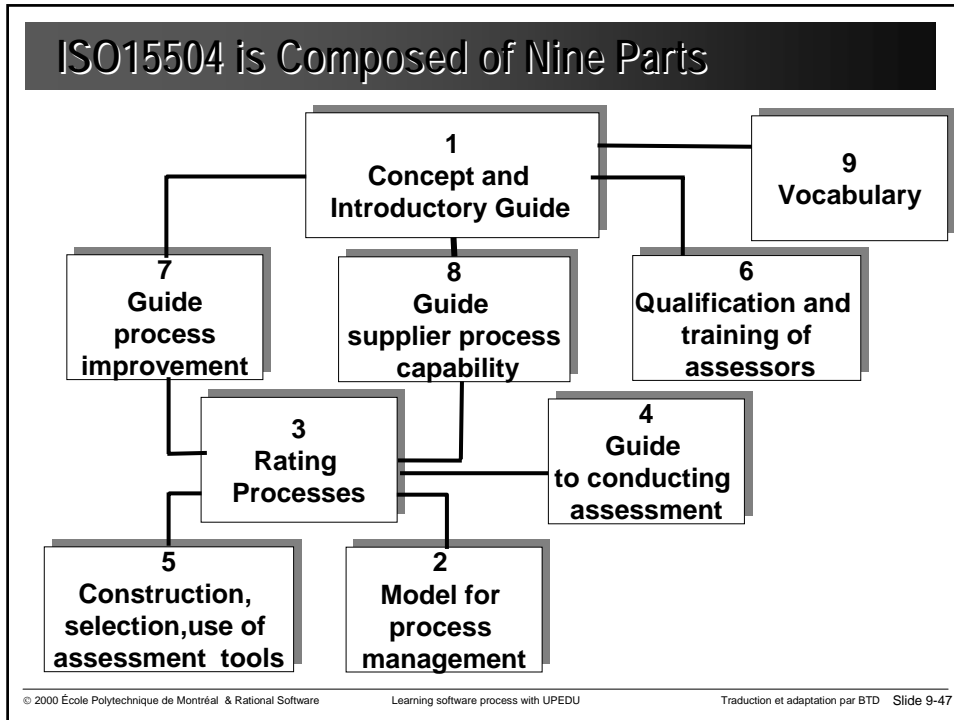
The supplier shall define and document how the requirements for quality will be met. Quality planning shall be consistent with all other requirements of a supplier's quality system and shall be documented in a format to suit the supplier's method of operation.

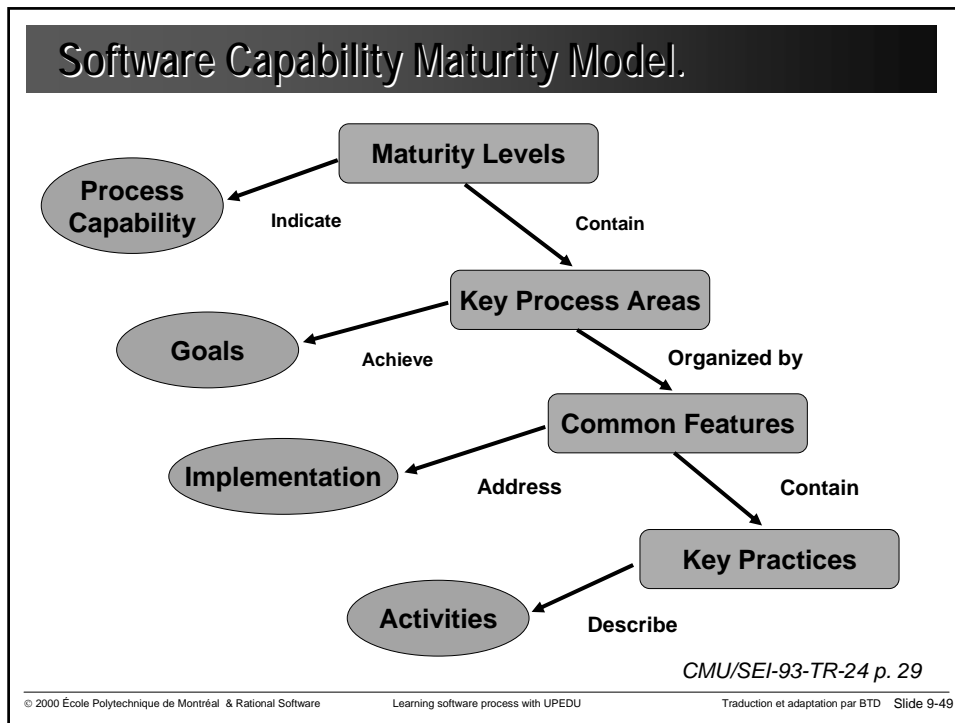
ISO 9000-3:1997 guidance includes:

Quality planning should address the following items, as appropriate:

- a) quality requirements, expressed in measurable terms;
- b) the life cycle model to be used for software development;
- c) defined criteria for starting and ending each project phase;
- d) identification of types of reviews, tests and other V&V to be carried out;
- e) identification of configuration management procedures to be carried out;

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-46





- ### Niveaux de maturité
- ◆ 5. **Optimizing :**
 - Continuous process improvement: quantitative feedback from the process and from testing new ideas and technologies.
 - ◆ 4. **Managed :**
 - Detailed measures are collected: software and product quality.
 - ◆ 3. **Defined :**
 - Activities are documented, standardized and integrated into an organization-wide software process.
 - ◆ 2. **Repeatable :**
 - Basic management processes: cost, schedule and functionality
 - ◆ 1. **Initial :**
 - *ad hoc* process. Success depends on individual effort.
- © 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-50

Evaluation du processus logiciel

- ◆ Définir le modèle de processus et le modèle de référence
- ◆ Enregistrement et Certification

- ◆ **Evaluation de UPEDU**

- Level 2, Key Process Area and Goals
- Level 3, Key Process Area and Goals

Voir :

UPEDU CONCEPTS: Process Quality

Level 2 KPAs and Goals

- ◆ **Requirements Management**
 - G-1 Requirements are controlled to establish baseline
 - G-2 Artifacts are kept consistent with requirements
- ◆ **Software Project Planning**
 - G-1 Estimates are documented for planning and tracking
 - G-2 Project activities are planned and documented
- ◆ **Software Project Tracking and Oversight**
 - G-1 Results and performances are tracked against plans
 - G-2 Changes to commitments are agreed to by affected group
- ◆ **Software Subcontract Management**
 - These four goals fall beyond the scope of the UPEDU.

Level 2 KPAs and Goals continue

- ◆ **Software Quality Assurance**
 - G-1 SQA activities are planned
 - G-2 Adherence of products and activities to standard is verified
 - G-3 Affected groups are informed of SQA activities and results
 - G-4 Noncompliance issues are addressed

- ◆ **Software Configuration Management**
 - G-1 Configuration management activities are planned
 - G-2 Selected work products are identified, controlled and available
 - G-3 Changes to work products are controlled
 - G-4 Affected groups are informed to the status and content of baselines

Level-3 KPAs and Goals

- ◆ **Organization Process Focus**
 - G-1 Software process activities are coordinated
 - G-2 Strengths and weaknesses of software process are identified
 - G-3 Organizational level process activities are planned

- ◆ **Organization Process Definition**
 - G-1 A standard software process for the organization is developed
 - G-2 Information related to the use of software process is collected and reviewed

- ◆ **Training Program**
 - Organizational responsibilities

- ◆ **Integrated Software Management**
 - G-1 Project software process is customized from standard process

Level-3 KPAs and Goals *continued*

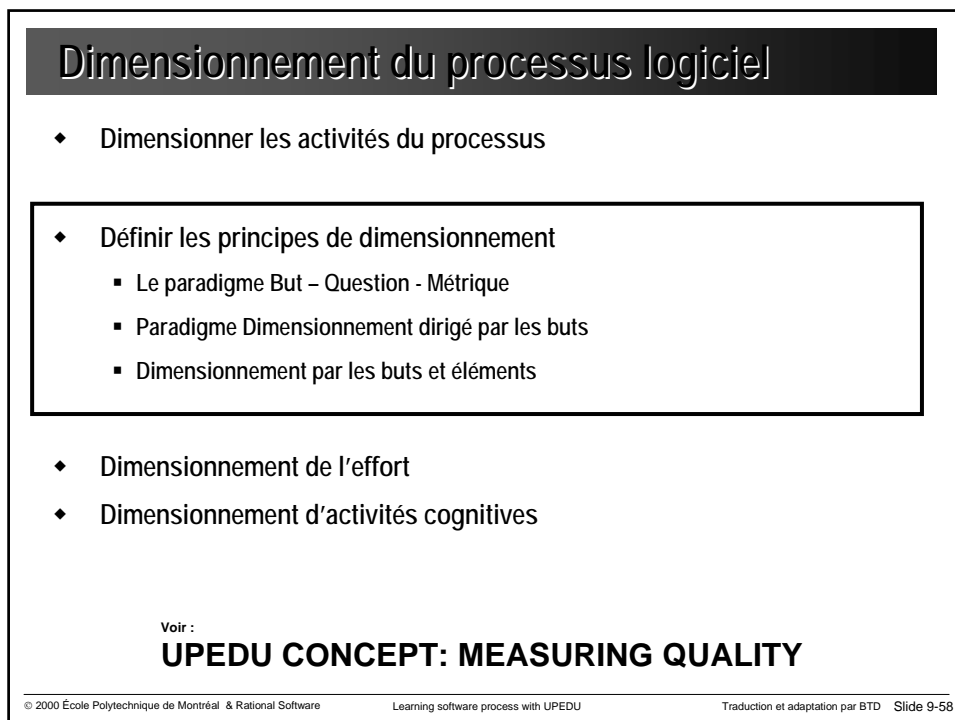
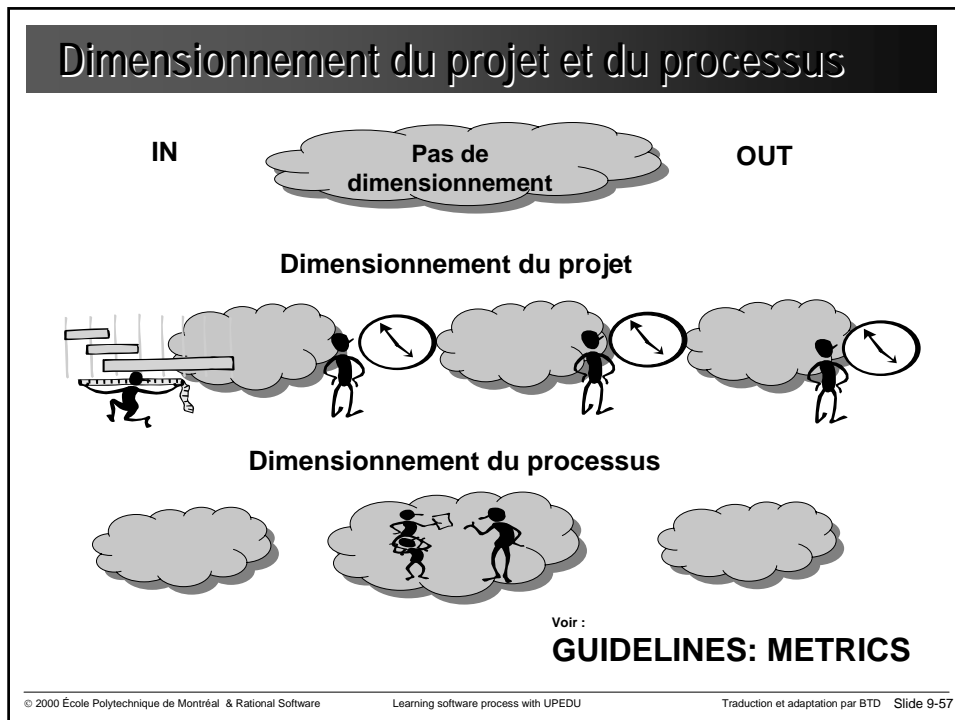
- ◆ **Integrated Software Management (continued)**
 - G-2 Project planned and managed according to software process
- ◆ **Software Product Engineering**
 - G-1 Tasks are defined, integrated and consistently performed
 - G-2 Software products are kept consistent with each other
- ◆ **Intergroup Coordination**
 - G-1 Requirements are agreed to by all groups
 - G-2 Commitments between various groups are agreed to by all
- ◆ **Peer Reviews**
 - G-1 Peer reviews are planned
 - G-2 Defects are identified and removed

Dimensionnement du processus logiciel

- ◆ **Dimensionner les activités du processus**
 - Processus contrôlés
 - Dimensionnement vs Métriques
 - Un programme de dimensionnement réussi
- ◆ Définir les principes de dimensionnement
- ◆ Dimensionnement de l'effort
- ◆ Dimensionnement d'activités cognitives

Voir :

UPEDU CONCEPT: METRICS



Des approches de dimensionnement

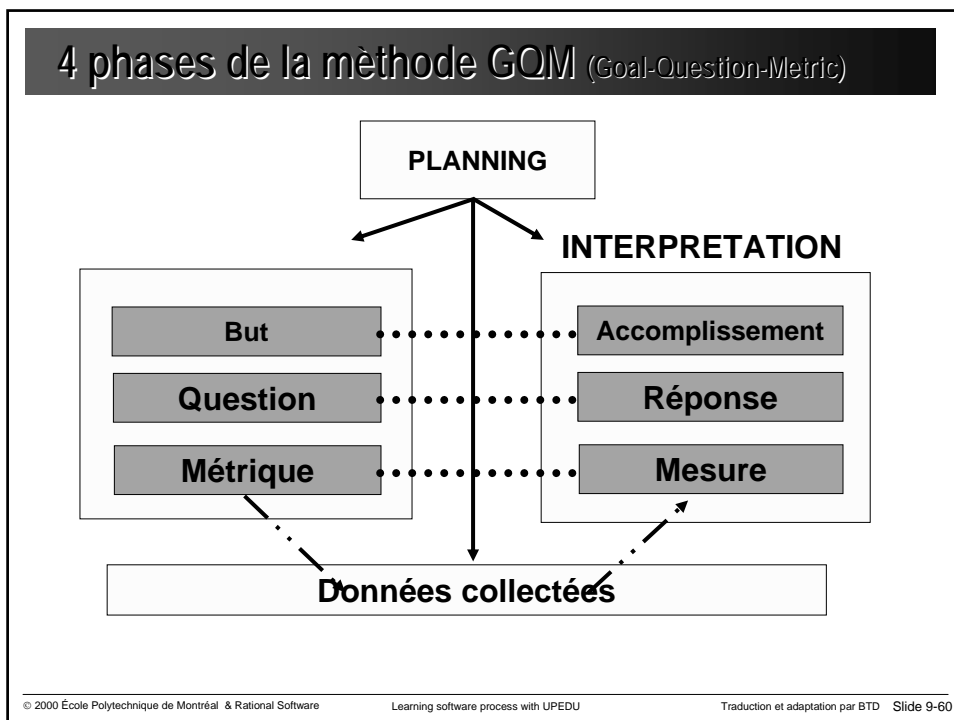
♦ **PDCA (Plan, Do, Check, Act):**

- **PLAN:** Identifier les aspects et déterminer les mesures à collecter.
- **DO:** Collecter les données et le représenter graphiquement.
- **CHECK:** Analyser les graphes pour comprendre les aspects.
- **ACT:** Publier les résultats, formuler les recommandations et identifier des nouveaux aspects.

♦ **ETVX (Entry, Task, Validation, Exit):**

- **ENTRY:** Définir les critères de démarrage d'activité et les conditions de début.
- **TASK:** Décrire et énumérer ce qui doit être effectué.
- **VALIDATION:** Vérifier la qualité des résultats produits par les tâches.
- **EXIT:** Définir les critères de complétude des activités et des conditions de fin.

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-59



Dimensionnement dirigé par les buts

1. Identifier des buts de l'application
2. Identifier ce qui doit être connu ou appris
3. Identifier des sous-buts
4. Identifier les entités et attributs liés aux sous-buts
5. Formaliser le dimensionnement des buts
6. Identifier les questions et indicateurs
7. Identifier les données élémentaires
8. Définir les mesures à utiliser
9. Identifier les actions qui mettent en œuvre des mesures
10. Préparer le plan pour l'implémentation des mesures

Dimensionner les buts dérivés des buts applicatifs

Des modèles mentaux fournissent le contexte et le focus

Traduire des buts en structures exécutables et mesurables

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTS Slide 9-61

Définition de buts mesurables

Questions	Significations
Quoi ?	L'objet mesuré
Pourquoi ?	Pour comprendre, contrôler et perfectionner l'objet
Quel aspect ?	La qualité du focus sur l'objet
Qui ?	Les personnes qui mesurent l'objet
Où ?	L'environnement dans lequel la mesure prend place

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTS Slide 9-62

Le dimensionnement requière quatre éléments

- **Objet de l'intérêt**
Phase, workflow, activité, tâche, ressource, ou environnement à mesurer.
- **Raison**
comprendre, prédire, planifier, contrôler, comparer, évaluer, ou approfondir certains aspects de l'objet.
- **Perspective**
développer, maintenir, manager, ou clarifier les objectifs du client.
- **Environnement et contraintes**
Projet, équipe, processus, outils, ou cycle de vie qui fournit le contexte pour l'interprétation des mesures et résultats.

Dimensionnement du processus logiciel

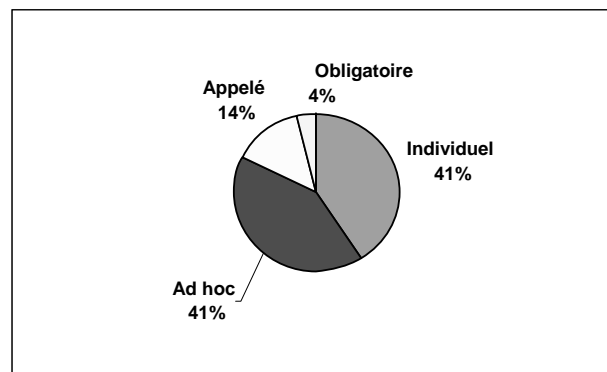
- ◆ Dimensionner les activités du processus
 - ◆ Définir les principes de dimensionnement
- ◆ Dimensionnement de l'effort
 - Les aspects importants de dimensionnement d'effort
 - Analyses du travail collaboratif de l'équipe
 - Analyses des activités qui composent une tâche
- ◆ Dimensionnement d'activités cognitives

Validité des données de mesure d'effort

- ◆ Invalidité de mesures d'effort
 - Les limites de l'expressivité des descripteurs
- ◆ Non uniformité des intervalles de temps
 - Limites de granularité des intervalles
- ◆ Non uniformité of marquages utilisés
 - Disambiguer les descripteurs
- ◆ ambiguïté des marquages
 - Valider les descripteurs d'entrée

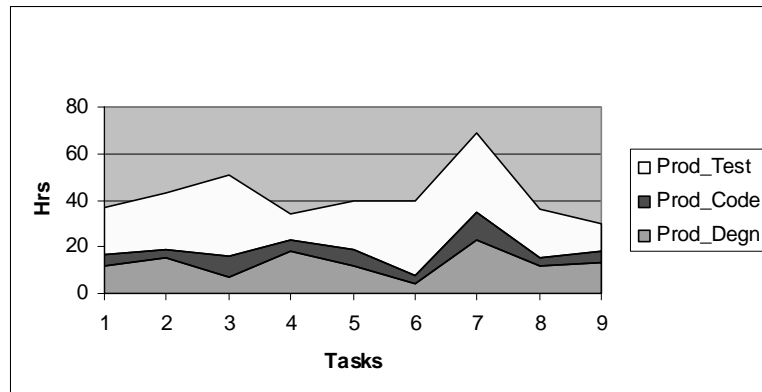
Quatre types d'activités collaboratives

Obligatoire : Réunion programmée formelle
Appelé : Réunion programmée informelle



Ad hoc : Réunion non programmée informelle
Individuel : Pas de réunion

Importance relative de trois activités : test, codage, conception



© 2000 École Polytechnique de Montréal & Rational Software

Learning software process with UPEDU

Traduction et adaptation par BTD Slide 9-67

Dimensionnement du processus logiciel

- ◆ Dimensionner les activités du processus
 - ◆ Définir les principes de dimensionnement
 - ◆ Dimensionnement de l'effort
- ◆ Dimensionnement d'activités cognitives
 - Protocol Analysis for Software Process Activities
 - Analyses of Roles in Peers Review Meetings
 - Analyses of Exchange Patterns in Peers Review Meetings

© 2000 École Polytechnique de Montréal & Rational Software

Learning software process with UPEDU

Traduction et adaptation par BTD Slide 9-68

Echantillon du code

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

© 2000 École Polytechnique de Montréal & Rational Software Learning software process with UPEDU Traduction et adaptation par BTD Slide 9-69

