UPEDU : Unified Process for Education

PARTIE 2 :
Identification des besoins
Analyse et conception
Activités d'architecture
Vues et disciplines
Tests
CCM

 Learning software process with UPEDU Traduction et adaptation par BTD 1

Identification des besoins

- Présenter l'étendue des besoins
  - Les besoins capturent la compréhension basique de toutes les parties prenantes. Ils doivent être exprimés de façon à faciliter la communication, permettre les évolutions et supporter les changements.
- Définir des constituants des besoins
- Expliciter des besoins
- Supporter l'évolution des besoins

 Learning software process with UPEDU Traduction et adaptation par BTD 2

Objectifs de l'identification des besoin

- Constituer une base de communication entre toutes les parties
- Engagement contractuel entre parties
- Destination
  - Equipe de conception
  - Tests du logiciel
  - Assurance qualité
  - Documentation utilisateur
- Référence pour le chef de projet
- Outil pour le contrôle des évolutions du système

Adapted from Alan Davis

 Learning software process with UPEDU Traduction et adaptation par BTD 3

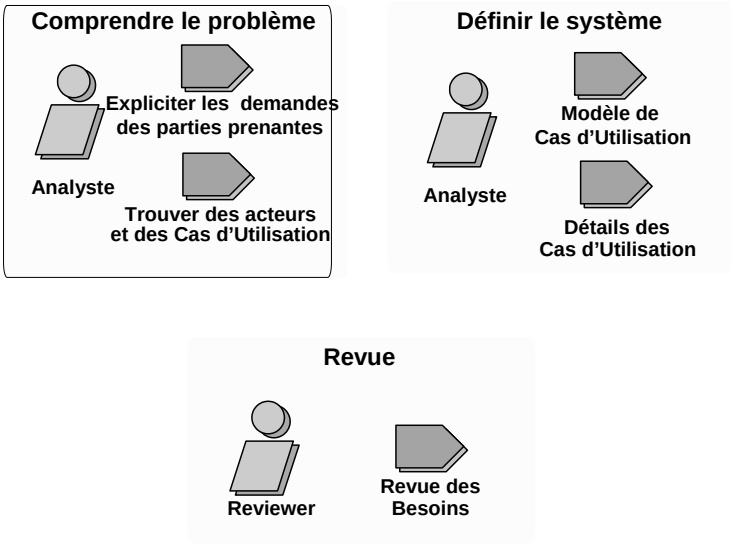
Parties prenantes

Une partie prenante est un individu ou une organisation qu'est matériellement affecté par les retombées du système.

partenaires
utilisateurs
clients
experts du domaine
analystes industriels
développeurs

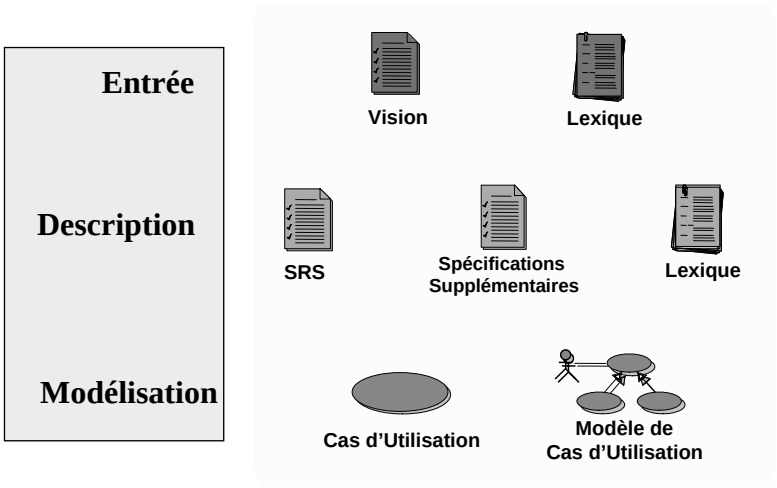
 Learning software process with UPEDU Traduction et adaptation par BTD 4

Identification des besoins est composée de 5 activités



 Learning software process with UPEDU Traduction et adaptation par BTD 5

Trois groupes de documents (artéfacts)



 Learning software process with UPEDU Traduction et adaptation par BTD 6

Identification des besoins

- Présenter l'étendue des besoins
- Définir des constituants des besoins
  - Document de vision
  - Document lexique
  - Besoins fonctionnels
  - Besoins non-fonctionnels
- Expliciter des besoins
- Supporter l'évolution des besoins

 Learning software process with UPEDU Traduction et adaptation par BTD 7

Besoins de haut niveau

- La documentation du niveau "système" qui décrit les QUOI et COMMENT du produit ou de l'application
- On se focalise sur:
  - Des besoins des utilisateurs
  - Des buts et des objectifs
  - Des marchés cibles
  - Des environnements des utilisateur et des plates-formes
  - Des caractéristiques du produit

Document Vision

Un document qui permet à toutes les parties de travailler à partir du même document

 Learning software process with UPEDU Traduction et adaptation par BTD 8

Caractéristiques de spécifications (SRS) bien construites

- Correctes
  - Chaque besoin contribue à la satisfaction des exigences identifiées
- Complètes
  - Contient des besoins significatifs comme réponses aux demandes clairement étiquetées et référencées
- Consistantes
  - Pas de sous-ensembles de besoins en conflit.
- Non-ambiguës
  - Chaque besoin a une seule interprétation
- Rangées par importance et stabilité
  - Les identifieurs indiquent clairement l'importance et la stabilité vérifiable.
- Modifiables
  - Les changements peuvent être fait facilement, complètement et de façon consistante.
- Traçables
  - Facilite les références en avant et en arrière (Backward and Forward referencing)

ref - IEEE 1993

 Learning software process with UPEDU Traduction et adaptation par BTD 9

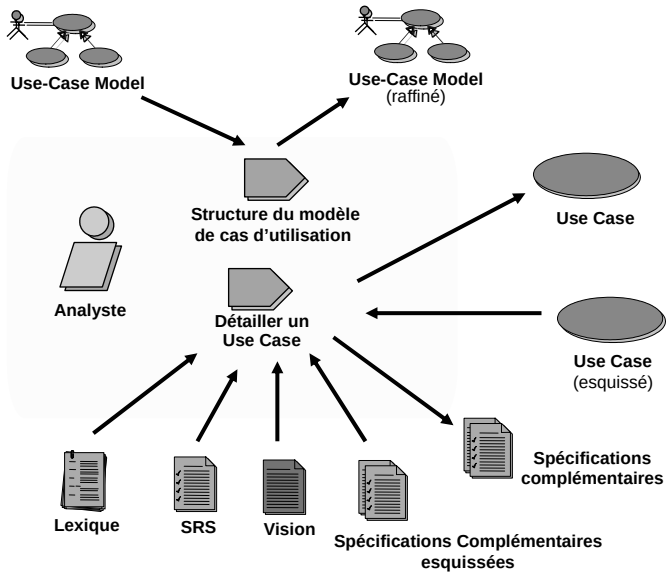
Les attributs de spécifications supplémentaires

- *Utilisabilité :*
  - La facilité avec laquelle le logiciel peut être appris et utilisé
- *Fiabilité :*
  - La capacité du logiciel d'être consistant
- *Performance :*
  - La mesure de de rapidité et de l'efficacité du logiciel exécuté
- *Evolutivité:*
  - La capacité du logiciel à être facilement modifiable et réparable
- *Contraintes de Conception*
  - Le besoin ne devrait pas impliquer des options de conception

Voir :
UPEDU Concept: Requirements

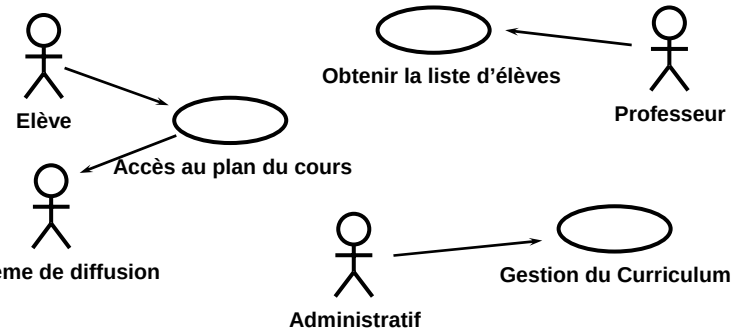
 Learning software process with UPEDU Traduction et adaptation par BTD 10

Les cas d'utilisation explicitent et clarifient des besoins



 Learning software process with UPEDU Traduction et adaptation par BTD 11

Notions liées aux cas d'utilisation



Voir :

UPEDU Guideline: Actors

UPEDU Guideline: Use-Case

UPEDU Guideline: Use-Case Model

UPEDU Guideline: Use-Case Diagram

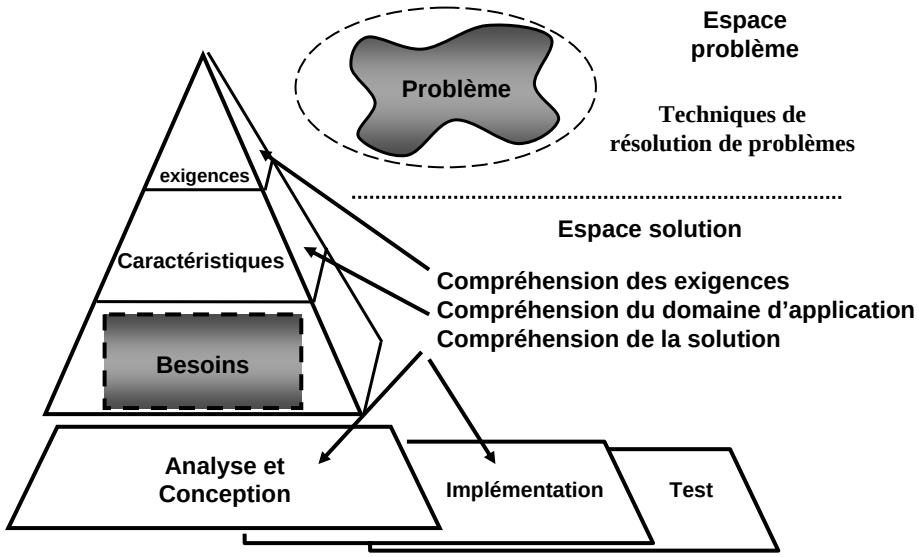
 Learning software process with UPEDU Traduction et adaptation par BTD 12

Identification des besoins - Résumé

- Présenter l'étendue des besoins
- Définir des constituants des besoins
- Expliciter des besoins
  - Délimiter les frontières
  - Interviews et questionnaires
  - Séminaire d'identification des besoins
  - Prototypage
- Supporter l'évolution des besoins

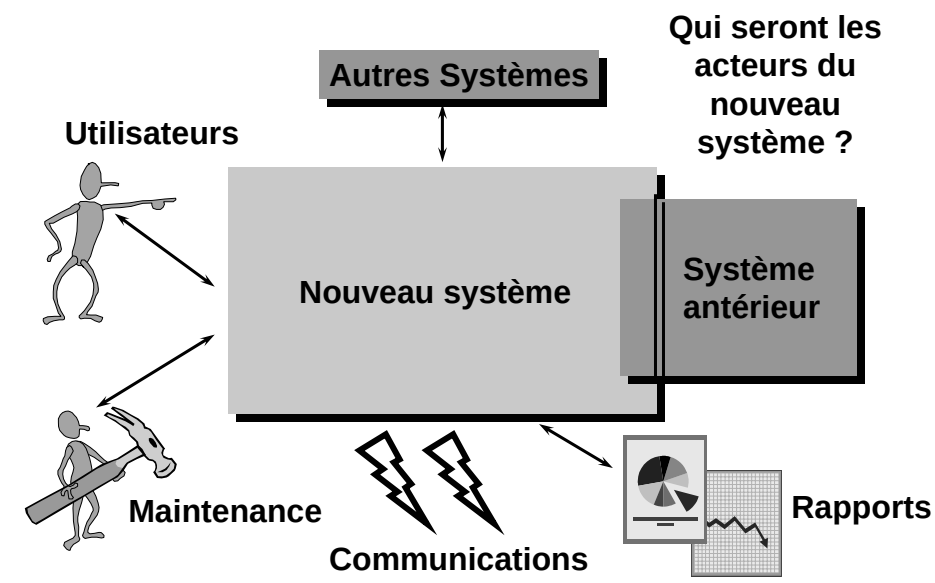
 Learning software process with UPEDU Traduction et adaptation par BTD 13

Trois dimensions pour l'explicitation des besoins



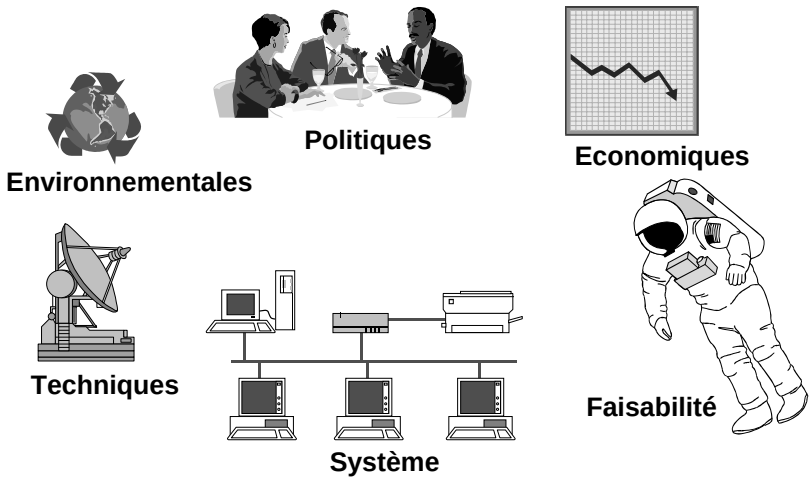
 Learning software process with UPEDU Traduction et adaptation par BTD 14

Définition des contours du système



 Learning software process with UPEDU Traduction et adaptation par BTD 15

Contraintes sur le système



 Learning software process with UPEDU Traduction et adaptation par BTD 16

Des interviews directes de personne à personne

- Ne pas demander aux gens de décrire des choses qu'ils ne décrivent pas usuellement.
- Poser des questions ouvertes
- Eviter des questions qui commencent par "Pourquoi ?"
- Ne pas s'attendre aux réponses simples
- Ne pas brusquer l'interviewé à répondre
- Ecouter, écouter et écouter !
- Les questionnaires ne se substituent pas aux interviews

 Learning software process with UPEDU Traduction et adaptation par BTD 17

Conduire un séminaire d'identification des besoins

- Accélérer le processus d'explicitation
- Mettre les parties prenantes ensemble pour une période intensive et focalisée
- Tout le monde peut s'exprimer
- Les résultats sont immédiatement disponibles
- Fournit un cadre pour mettre en œuvre d'autres techniques d'explicitation
  - Brainstorming
  - Storyboarding
  - Role-playing
  - Review existing requirements

 Learning software process with UPEDU Traduction et adaptation par BTD 18

Différentes variétés des prototypes

- Une démonstration précoce de certains ou tous les comportements externes observables du système.

Jetable

- Utilisé pour
  - Obtenir un retour sur la solution proposée
  - Démontrer les possibilités dans le domaine d'application
  - Valider des besoins connus
  - Découvrir les besoins non identifiés

Evolutif

- Outils de prototypage
  - Visual Basic, PowerSoft, Gupta, Access, Delphi
  - Toolbook
  - Programmes de démo
  - Simulations

Opérationnel

 Learning software process with UPEDU Traduction et adaptation par BTD 19

Identification des besoins - Résumé

- Présenter l'étendue des besoins
- Définir des constituants des besoins
- Expliciter des besoins
- Supporter l'évolution des besoins

 Learning software process with UPEDU Traduction et adaptation par BTD 20

Minimiser l'impact

- Analyser des problèmes réels et expliciter des besoins des utilisateurs
- Obtenir un agrément du client / utilisateur sur les besoins
- Modéliser l'interaction entre l'utilisateur et le système
- Mettre en place un processus le contrôle des fondements et des changements
- Maintenir une traçabilité des besoins dans les deux sens (forward and backward traceability)
- Utiliser un processus itératif
- Effectuer des revues
  - Traversé guidée (Walkthrough)
  - Inspection
  - Revue formelle

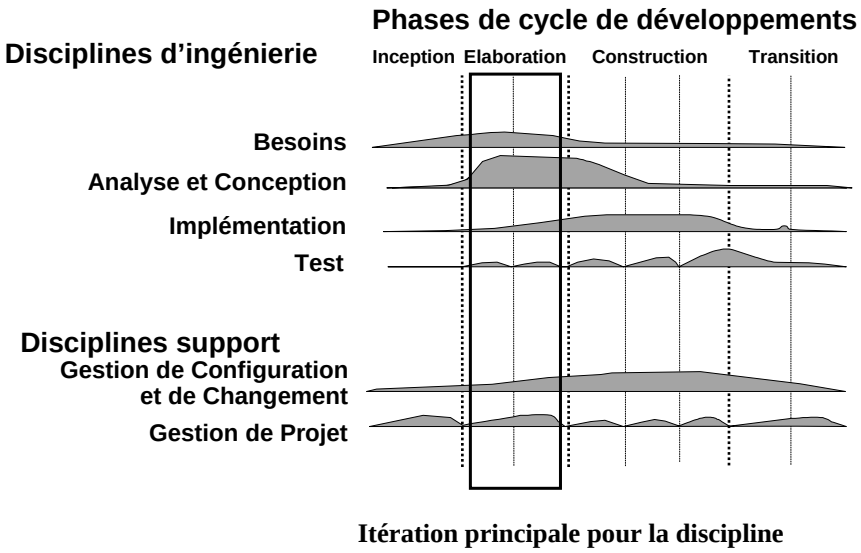
 Learning software process with UPEDU Traduction et adaptation par BTD 21

Analyse et Conception

- Introduire l'analogie avec la cristallisation
  - Treillis d'information
  - Etapes dans le processus de cristallisation
- Comprendre la discipline analyse et conception
- Définir les activités d'analyse et de conception
- Documenter la discipline d'analyse et de conception
- Modèle de visualisation

 Learning software process with UPEDU Traduction et adaptation par BTD 22

Point de vue processus et cycle de vie



 Learning software process with UPEDU Traduction et adaptation par BTD 23

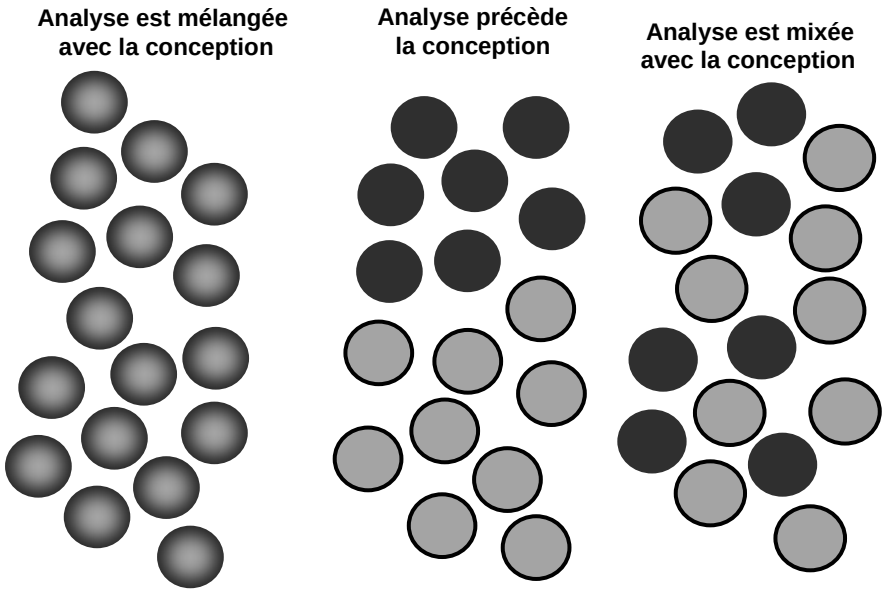
Analyse et Conception

- Introduire l'analogie avec la cristallisation
- Comprendre la discipline analyse et conception
  - Les concepts liés aux activitités
  - La qualité des activités
  - Le rôle du concepteur
- Définir les activités d'analyse et de conception
- Documenter la discipline d'analyse et de conception
- Modèle de visualisation

Voir :
UPEDU Concept: Analysis Mechanisms

 Learning software process with UPEDU Traduction et adaptation par BTD 24

Activités d'analyse et de conception



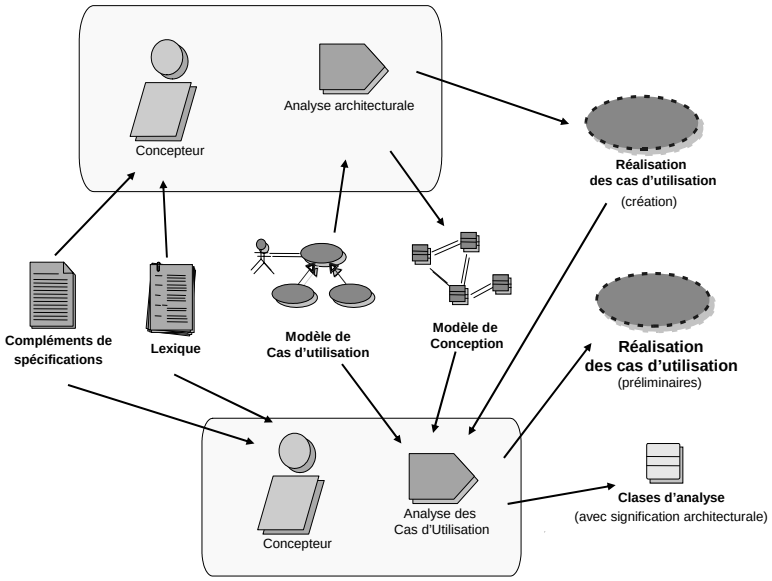
 Learning software process with UPEDU Traduction et adaptation par BTD 25

Analyse et Conception

- Introduire l'analogie avec la cristallisation
- Comprendre la discipline analyse et conception
- Définir les activités d'analyse et de conception
  - Définir l'architecture
  - Cultiver la conception
  - Valider l'architecture et la conception
- Documenter la discipline d'analyse et de conception
- Modèle de visualisation

 Learning software process with UPEDU Traduction et adaptation par BTD 26

Activités d'architecture



 Learning software process with UPEDU Traduction et adaptation par BTD 27

Les bénéfices d'une bonne architecture

- L'architecture permet aux développeurs d'obtenir et de conserver le contrôle intellectuel sur le projet, de gérer la complexité et de maintenir l'intégrité du système.
- L'architecture fournit une base effective pour la réutilisation en gros grain.
- L'architecture fournit la base pour la gestion de projet.
- L'architecture facilite de développement à base de composants :
  - Une composant component exprime une fonction dans le contexte d'une architecture clairement définie
  - Une composant est en conformité et fournit une réalisation concrète d'un ensemble d'interfaces
  - Les composants existent par rapport à une architecture donnée

Voir :

UPEDU Concept: Software Architecture

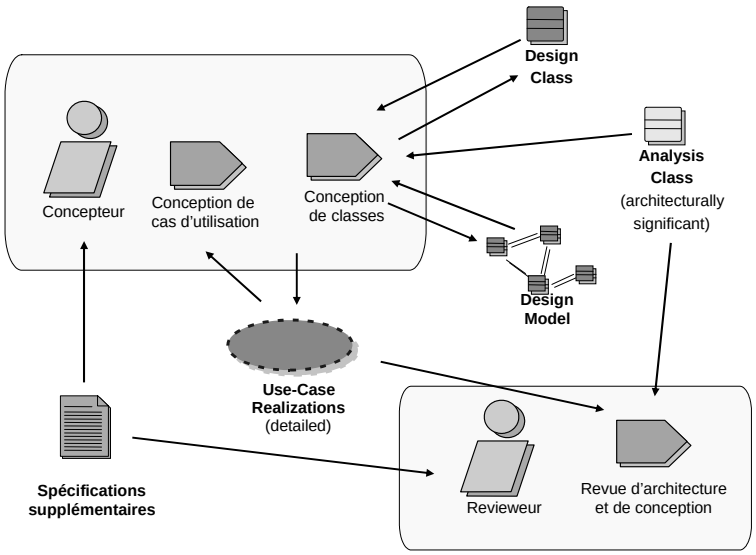
 Learning software process with UPEDU Traduction et adaptation par BTD 28

Réutilisation de systèmes et élasticité

- Les bonnes architectures satisfont aux besoins, sont élastiques (*resilient*) et basées sur les composants
- une architecture élastique permet :
  - la maintenabilité et l'extensibilité perfectionnées
  - la réutilisation économiquement significative
  - un partage clair du travail entre les équipes de développeurs
  - l'encapsulation des dépendances matériels et système
- une architecture a base de composants permet :
  - la réutilisation où l'adaptation des composants existants
  - le choix entre les milliers de composants disponibles commercialement
  - L'évolution incrémentale du logiciel existant

 Learning software process with UPEDU Traduction et adaptation par BTD 29

Conception d'architecture



 Learning software process with UPEDU Traduction et adaptation par BTD 30

Analyse et Conception

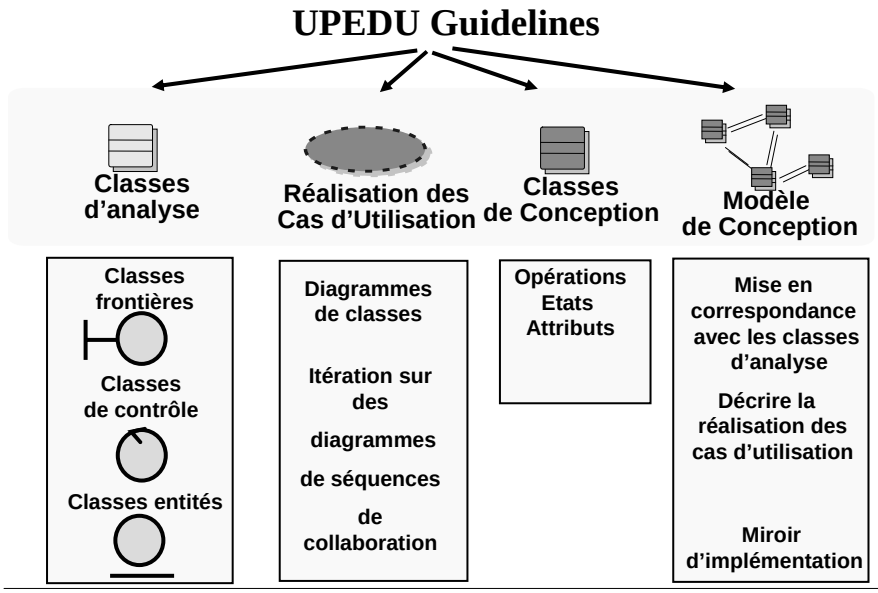
- Introduire l'analogie avec la cristallisation
- Comprendre la discipline analyse et conception
- Définir les activités d'analyse et de conception

- Documenter la discipline analyse et conception
  - Artéfacts
  - Les classes d'analyse
  - Réalisation des cas d'utilisation
  - Conception de classes
  - Modèle de conception

- Modèle de visualisation

 Learning software process with UPEDU Traduction et adaptation par BTD 31

Quatre artéfacts de base



 Learning software process with UPEDU Traduction et adaptation par BTD 32

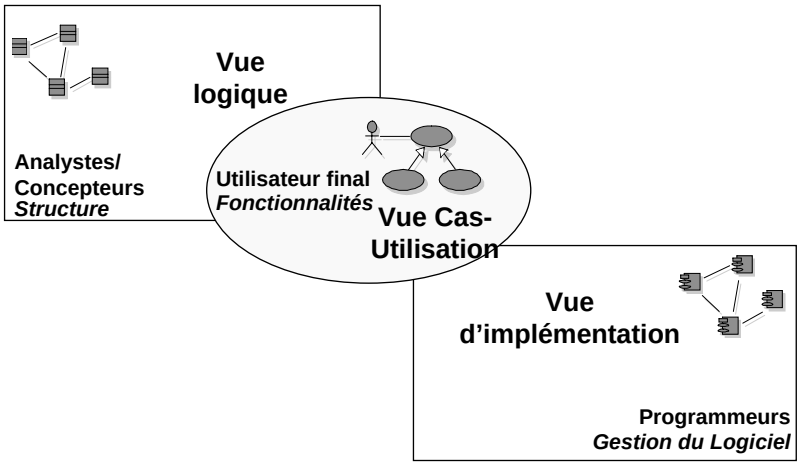
Analyse et Conception

- Introduire l'analogie avec la cristallisation
- Comprendre la discipline analyse et conception
- Définir les activités d'analyse et de conception
- Documenter la discipline analyse et conception

- Modèle de visualisation
  - Les vues
  - La vue Cas d'utilisation
  - La vue logique
  - La vue Implémentation

 Learning software process with UPEDU Traduction et adaptation par BTD 33

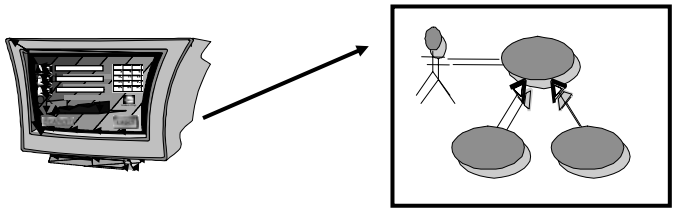
2+1 Vues complémentaires



 Learning software process with UPEDU Traduction et adaptation par BTD 34

La Vue - Cas d'Utilisation

- Décrit :
  - un sous-ensemble architecturalement significatif du modèle de cas d'utilisation.
- Concerne :
  - fonctionnalités, fonctions critiques, performance
- Représente :
  - graphiquement, sur les diagrammes des cas d'utilisation



- **Acteurs**
- **Cas d'Utilisation**

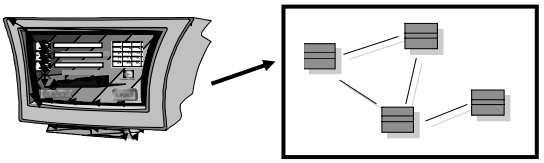
Voir :

UPEDU Concept USE-CASE View

 Learning software process with UPEDU Traduction et adaptation par BTD 35

La vue logique

- Décrit :
  - Un sous-ensemble architectural du modèle de conception (un sous-ensemble de classes et de réalisations de cas d'utilisations).
- Concerne :
  - Les fonctionnalités, le comportement, l'utilisation des frameworks et des patterns
- Représente :
  - graphiquement, par Diagrammes de classes, diagrammes de séquences et diagrammes d'états



- **Structure de paquetages**
- **Classes intéressantes**
- **Réalisation des cas d'utilisation**

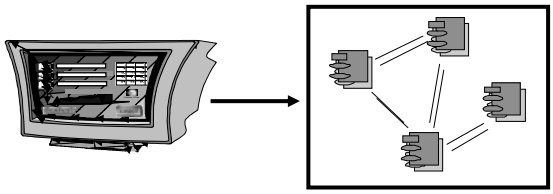
Voir :

UPEDU Concept: Logical View

 Learning software process with UPEDU Traduction et adaptation par BTD 36

La vue - Implémentation

- Décrit :
  - Organisation des composants logiciels dans l'environnement de développement
- Concerne :
  - Facilité de développement, organisation des équipes, prise en compte des systèmes composants existants, Gestion de configuration et du logiciel
- Représente :
  - Graphiquement sur les diagrammes de composants



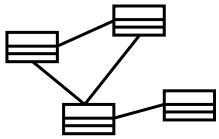
- **Structure de la bibliothèque de codes**
- **Les sources des livrables (.exe , .DLL, Data files)**

Voir :

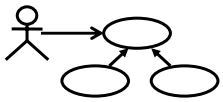
UPEDU Concept: Implementation View

 Learning software process with UPEDU Traduction et adaptation par BTD 37

Taches d'implémentation



Modèle de Conception

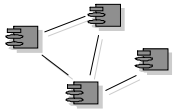


Modèle de cas d'utilisation

Plan de développement du logiciel

Spécifications supplémentaires

- **Définit le modèle de Conception / implémentation**
- **Plan d'intégration**
- **Composants d'implémentation**
- **Exécute des tests unitaires**
- **Tests d'évaluation**



Modèle d'implémentation

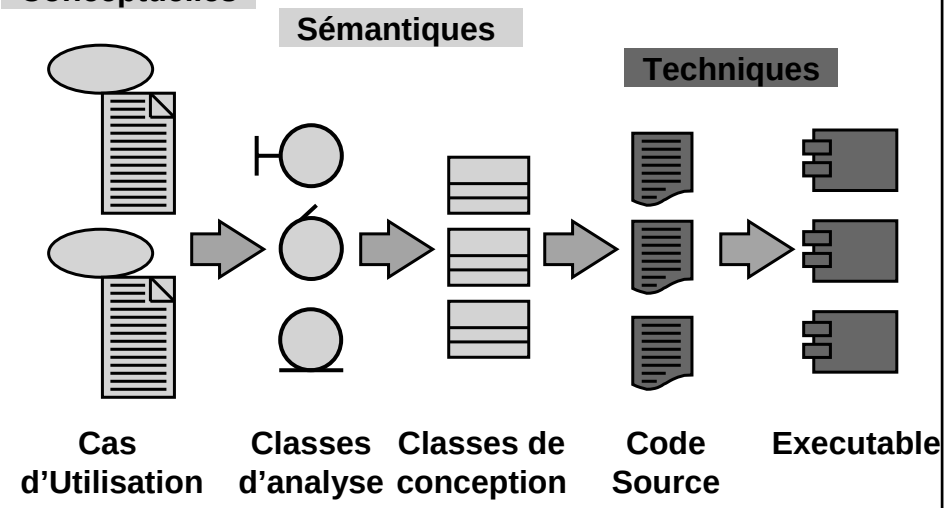
Cas de tests

Plan de tests

Résultats de tests

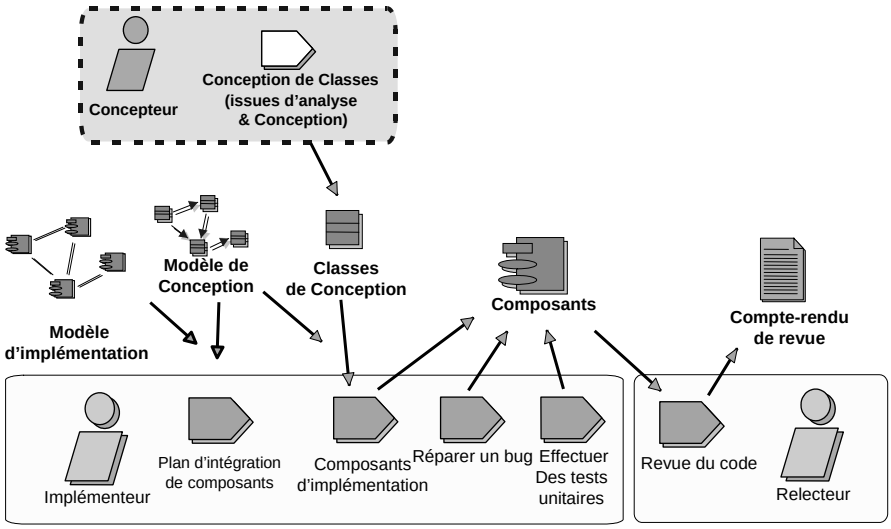
 Learning software process with UPEDU Traduction et adaptation par BTD 38

Traitement d'un volume important d'informations



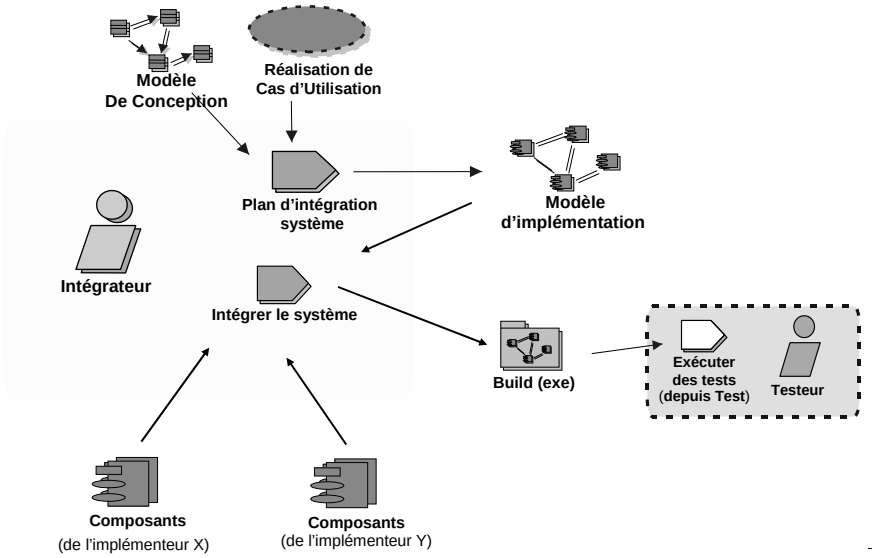
 Learning software process with UPEDU Traduction et adaptation par BTD 39

Cinq activités dans la discipline - Implémentation



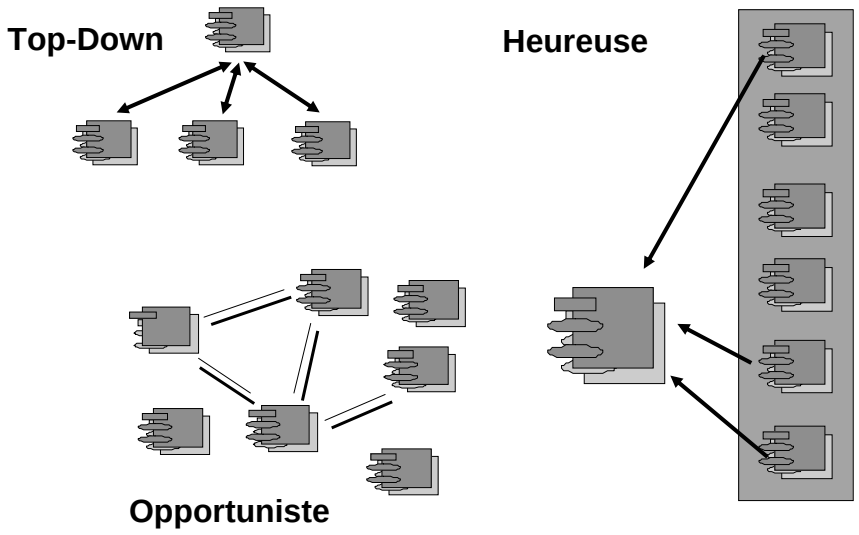
 Learning software process with UPEDU Traduction et adaptation par BTD 40

Activités de planning et d'intégration



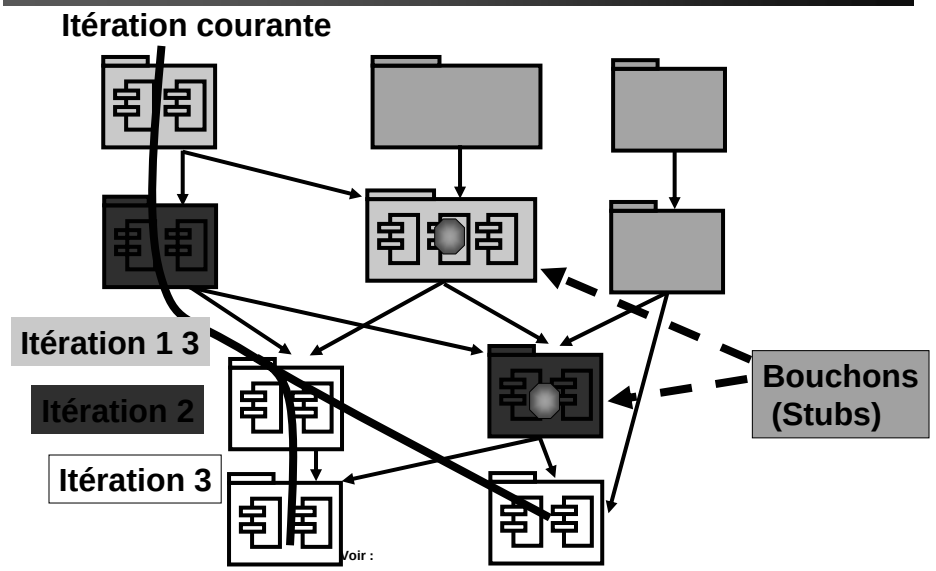
 Learning software process with UPEDU Traduction et adaptation par BTD 41

Stratégies de Planning



 Learning software process with UPEDU Traduction et adaptation par BTD 42

Componsants intégrés



Voir :

UPEDU Concepts: Software Integration

 Learning software process with UPEDU Traduction et adaptation par BTD 43

Programmation, Débogage et Test unitaire

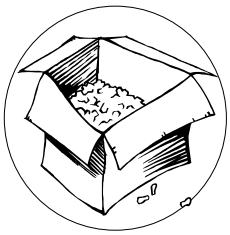
- Programmation
  - Traduction de la conception en code
  - En relation avec la maîtrise du langage de programmation

- Débogage
  - Identifier et faire disparaître des défauts de programmation
  - En relation avec le savoir-faire du programmeur

- Test unitaire
  - Enlever des défauts de traduction
  - En relation avec la compréhension de la conception

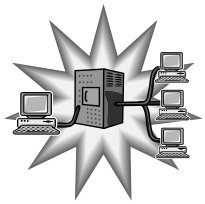
 Learning software process with UPEDU Traduction et adaptation par BTD 44

Approches de test unitaire

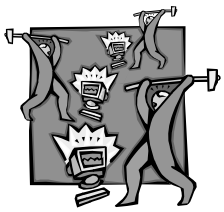


Boite blanche

Boite noire



Performance



Fiabilité

 Learning software process with UPEDU Traduction et adaptation par BTD 45

Plusieurs techniques de revue de code

- Inspection du code
  - Technique d'évaluation formelle
  - Recherche de consensus au sein de l'équipe sur la qualité du code
- Visite guidée (Walkthrough)
  - L'auteur conduit les relecteurs à travers le code
  - On procède ainsi à la synchronisation cognitive des membres de l'équipe
- Lecture du code
  - Lecture individuelle du code en relation avec les pratiques habituelles
  - Conformité du code au guide de programmation

Voir :

UPEDU: Work GUIDELINE: REVIEW

 Learning software process with UPEDU Traduction et adaptation par BTD 46

Deux types de composants :

Livrables

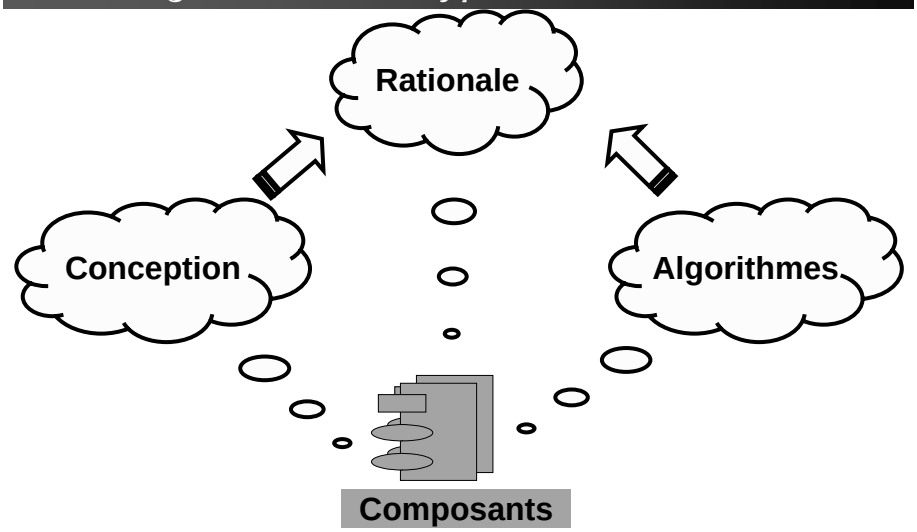
Exécutables fichiers .exe
Bibliothèques chargeables fichiers .dll
Applets .class pour Java
Pages Web fichiers .htm et .html
Tables des bases de données

Livrables produits par les composants

Fichiers de codes sources .h, .cpp et .hpp pour C++
Fichiers binaires fichiers .o et .a dans exec
Fichiers de construction makefiles

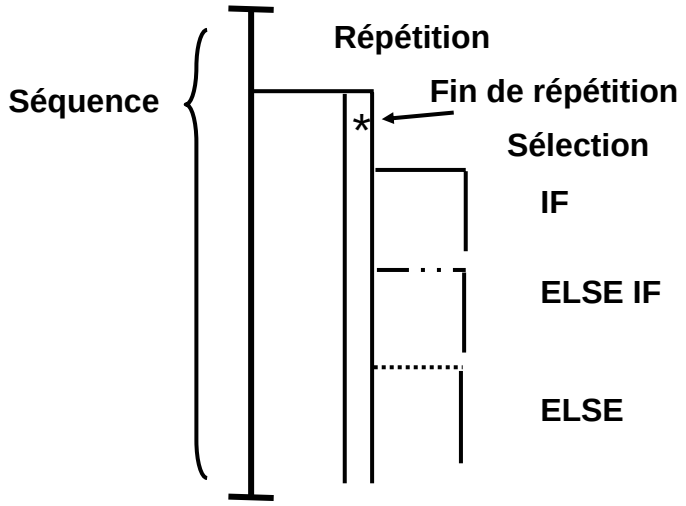
 Learning software process with UPEDU Traduction et adaptation par BTD 47

Convergence de trois types d'information



 Learning software process with UPEDU Traduction et adaptation par BTD 48

Pseudo-code schématique



 Learning software process with UPEDU Traduction et adaptation par BTD 49

Guide de Programmation

- Il est établi sous la responsabilité de la structure (entreprise, société de service, ...)
- Il est adapté aux besoins de l'organisation
- Il définit le niveau de documentation du code source
- Il spécifie les conventions de nommage des fichiers et des variables
- Il explique l'utilisation restrictive de certains concepts de programmation
- Il est créé itérativement en collaboration avec l'équipe de développement

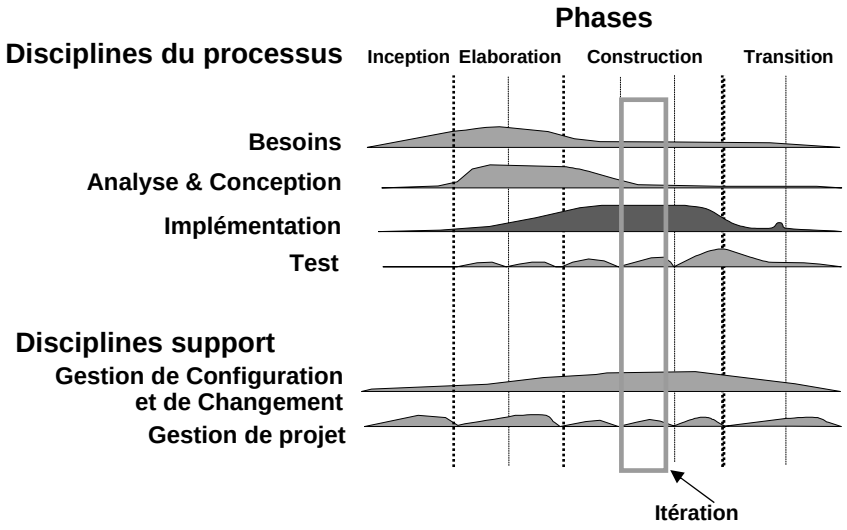
Voir :

UPEDU Concept: Mapping from design to Code

UPEDU Guideline: Programming Guide

 Learning software process with UPEDU Traduction et adaptation par BTD 50

Plan d'itération de la phase de Construction



 Learning software process with UPEDU Traduction et adaptation par BTD 51

Artéfacts de la phase de construction

- Plan de développement de logiciel
  - (mise à jour pour l'itération)
- Modèle de cas d'utilisation
  - (mise à jour)
- Modèle de conception
- Modèles d'implémentation
- Modèle de test
- Demandes de Changement
- Pour chaque itération :
  - Plan de test
  - Plan d'itération
  - Matériel support pour l'utilisateur final (préliminaire)
  - Artéfacts d'installation
  - Documentation de la version (Release Notes)

 Learning software process with UPEDU Traduction et adaptation par BTD 52

Objectifs d'une itération

- Les besoins sont stables.
- L'architecture est complètement implémentée.
- Nombreuses fonctionnalités ont été implémentées et intégrées
- Un effort important a été consacré à l'implémentation et les tests
- Le projet se rapproche d'une première "beta" version

Voir :
UPEDU Concepts: Development and Integration Workspaces

 Learning software process with UPEDU Traduction et adaptation par BTD 53

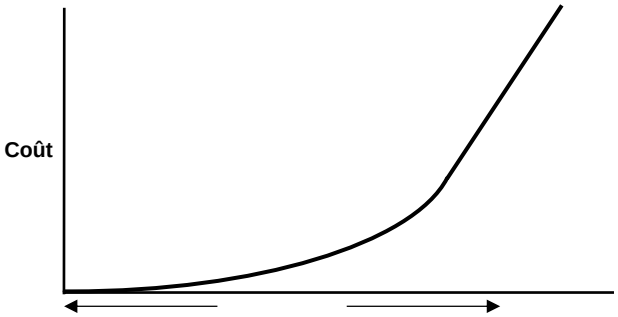
Discipline de tests

- Relier le test à la qualité
  - Opportunité de test
  - Indicateur des attributs de qualité par le test
  - Rôles
- Définir les activités de test
- Elaborer la discipline de test
- Evaluer les activités de test

 Learning software process with UPEDU Traduction et adaptation par BTD 54

Le test précoce réduit le coût de la qualité

Les problèmes de logiciel sont beaucoup plus coûteux à trouver et à réparer après le déploiement du logiciel



voir :

UPEDU Concept: Product Quality

 Learning software process with UPEDU Traduction et adaptation par BTD 55

Attributs de qualité

Type	Pourquoi ?	Comment ?
Fonctionnalités	L'application fait-elle ce qui a été demandé ?	Créer des cas de test pour chaque scénario implémenté
Fiabilité	L'application gère-t-elle correctement la mémoire ?	Outils d'analyse d'utilisation et instrumentation du code
Performance de l'application	L'application répond-elle acceptablement ?	Vérification de performance pour chaque cas d'utilisation /scénario implémenté
Performance de système	Le système est-il performant en charge ?	Test de performances de tous les cas d'utilisation sous une charge moyenne et la plus défavorable

 Learning software process with UPEDU Traduction et adaptation par BTD 56

Discipline de tests

- Relier le test à la qualité
- Définir les activités de test
  - Types de tests
  - Test Planning de tests et activités de conception
  - Activités d'implémentation de tests
  - Activités d'exécution de tests
- Elaborer la discipline de test
- Evaluer les activités de test

 Learning software process with UPEDU Traduction et adaptation par BTD 57

Niveaux de tests

Niveau	Cible visée	Activité testée
Débogage	Morceau du code source	Savoir-faire du programmeur
Unitaire	Unité du produit conçu	Réalisation de la conception par l'implémenteur
Intégration	Architecture des unités du produit	Réalisation du produit par l'implémenteur
Système	Environnement du produit	Les opérations du produit par l'implémenteur
Acceptation Alpha test	Fonctionnalité du produit	Compréhension du produit par le client
Beta test	Utilisabilité du produit	Utilisation du produit par l'utilisateur

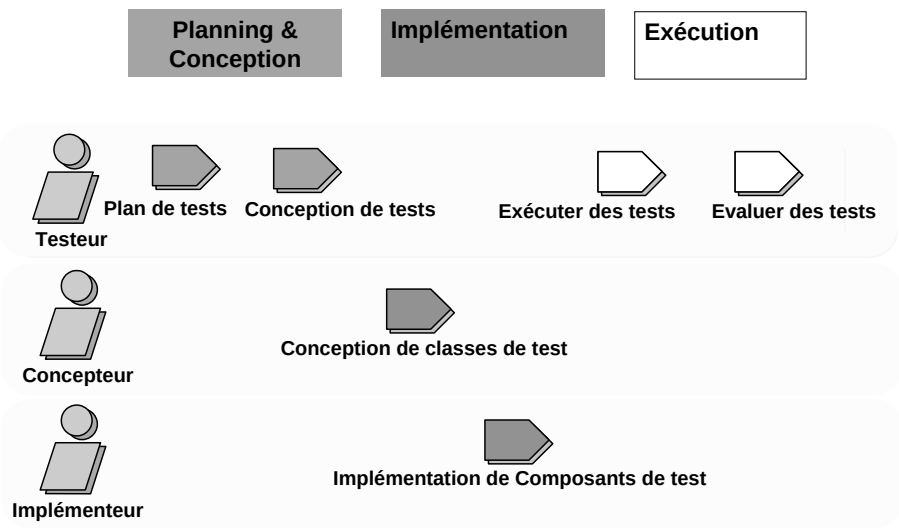
Voir :

UPEDU Concept: Stages of Test

UPEDU Concept: Acceptance Testing

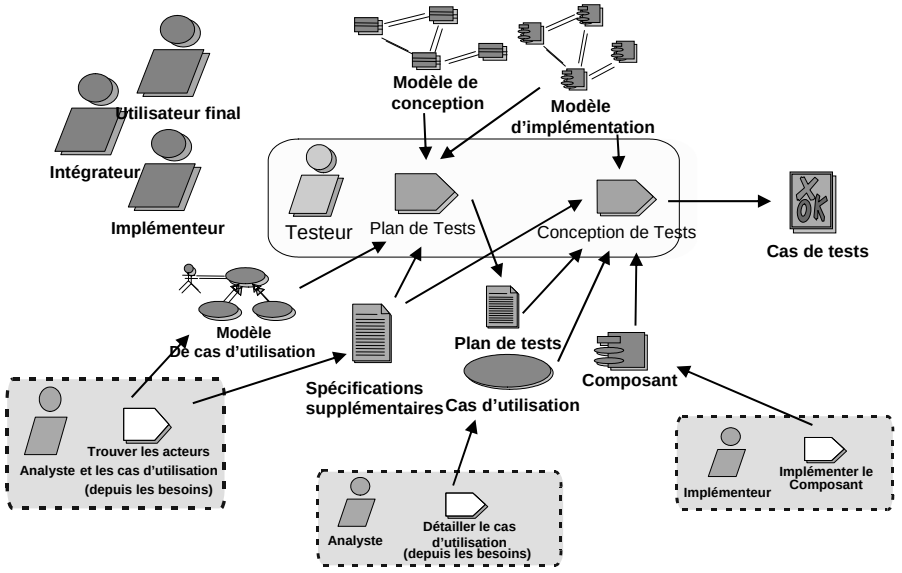
 Learning software process with UPEDU Traduction et adaptation par BTD 58

Nombreuses activités de la Discipline Test



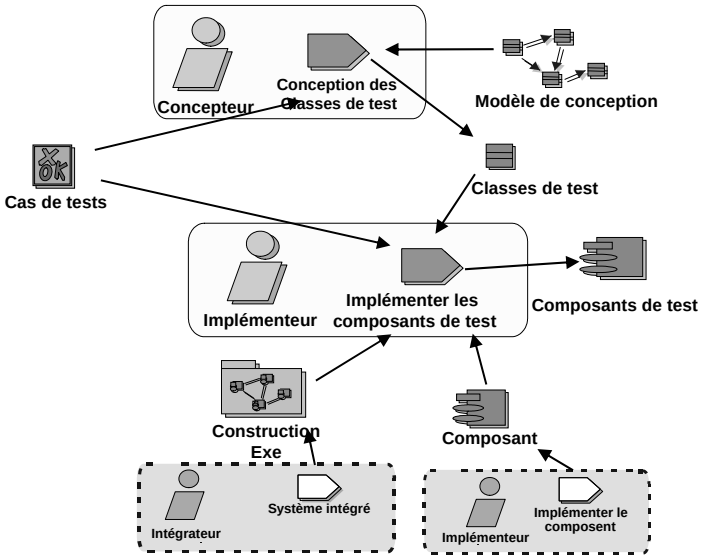
 Learning software process with UPEDU Traduction et adaptation par BTD 59

Activités de planning et de conception de tests



 Learning software process with UPEDU Traduction et adaptation par BTD 60

Activités d'implémentation de tests



 Learning software process with UPEDU Traduction et adaptation par BTD 61

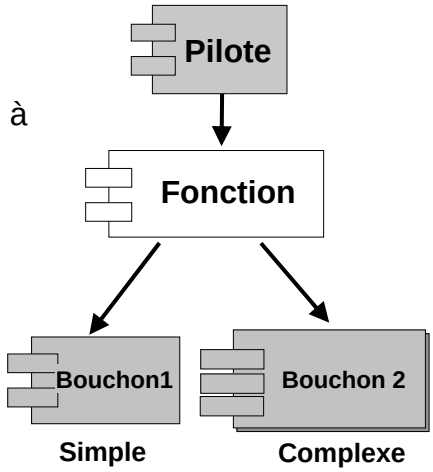
Fonctions de bouchon et de pilote (Stubs & Drivers)

Pilote (Driver) :

Le logiciel en amont ou interface qui fournit l'accès à la fonction

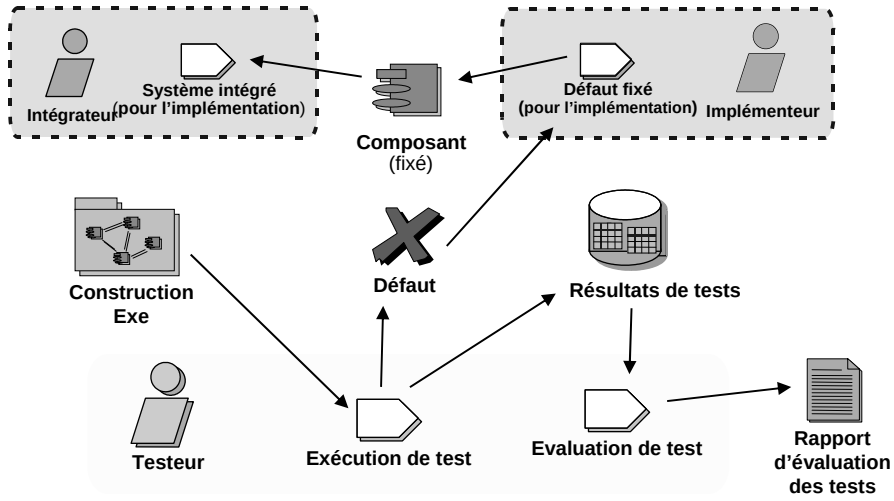
Bouchon (Stub) :

Logiciel qui simule le processus en aval



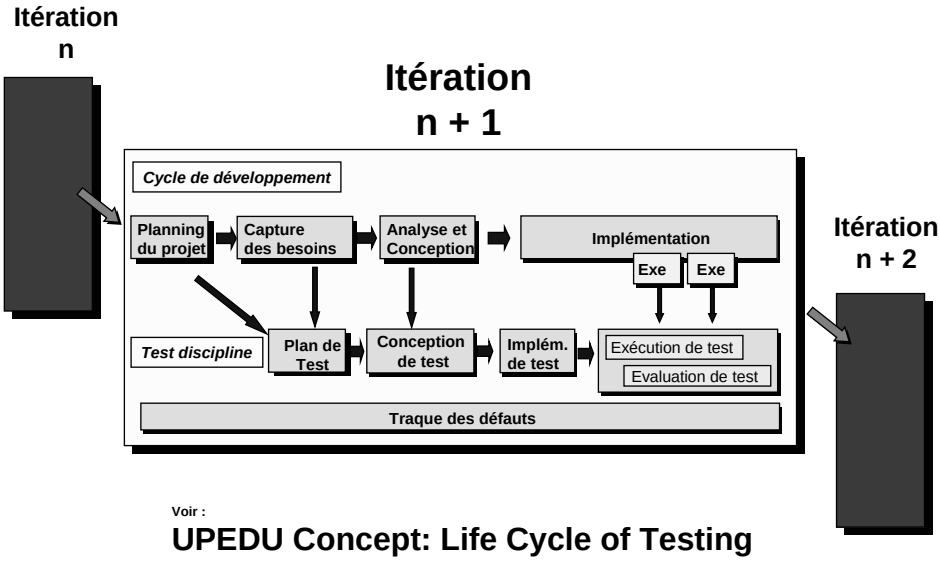
 Learning software process with UPEDU Traduction et adaptation par BTD 62

Activités d'exécution de tests



 Learning software process with UPEDU Traduction et adaptation par BTD 63

Les tests de régression sont accentués



Voir :

UPEDU Concept: Life Cycle of Testing

 Learning software process with UPEDU Traduction et adaptation par BTD 64

Discipline de tests

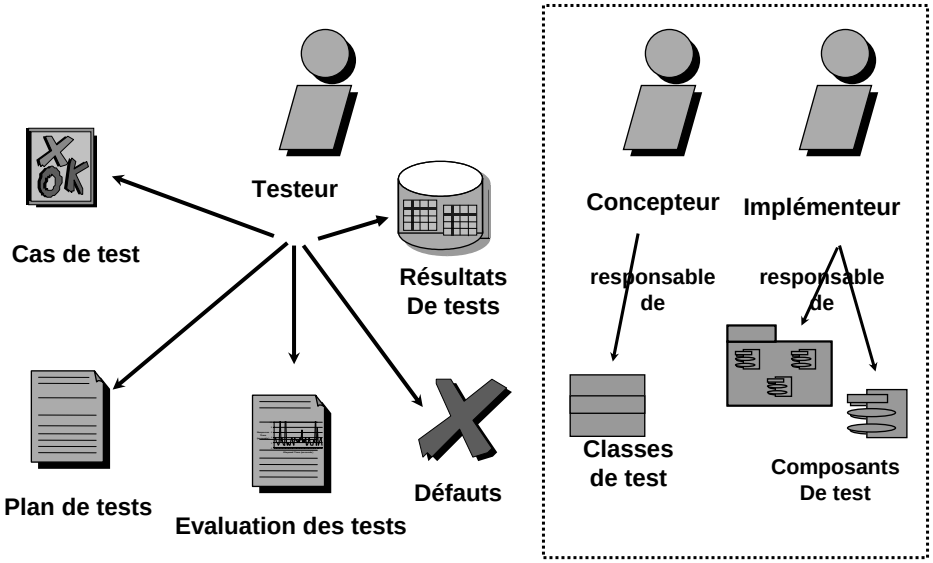
- Relier le test à la qualité
- Définir les activités de test

- Elaborer la discipline de test
  - Cas de tests
  - Planification de tests
  - Matrice d'évaluation des risques
  - Rapport de défauts et d'évaluation

- Evaluer les activités de test

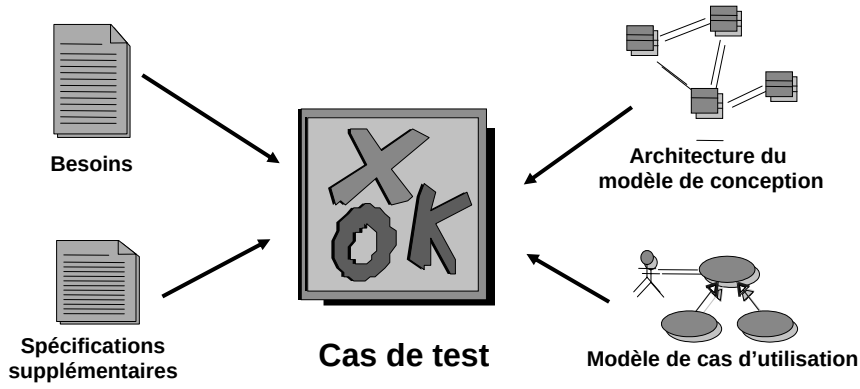
 Learning software process with UPEDU Traduction et adaptation par BTD 65

La diversité importante des Artéfacts



 Learning software process with UPEDU Traduction et adaptation par BTD 66

Les éléments majeurs que possèdent des cas de tests

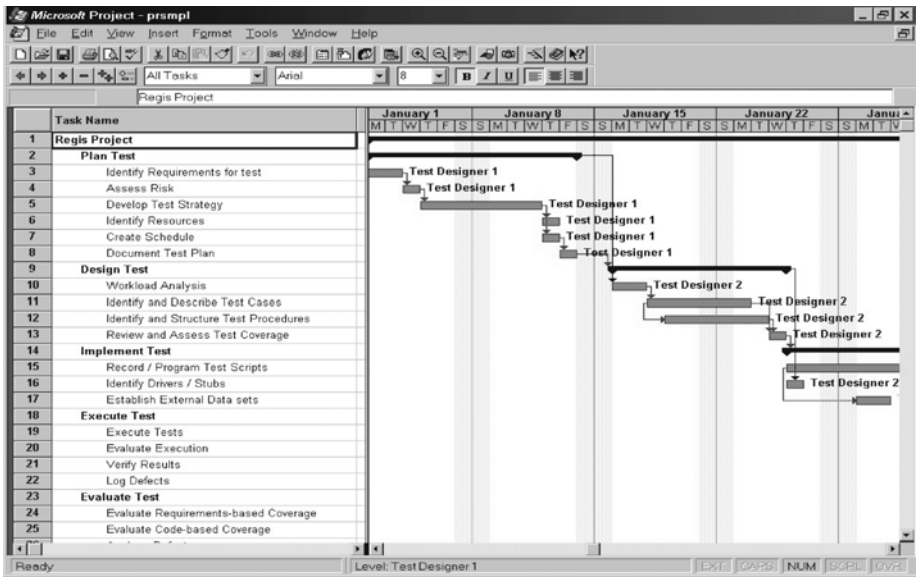


Voir :

UPEDU Guideline: Test Cases

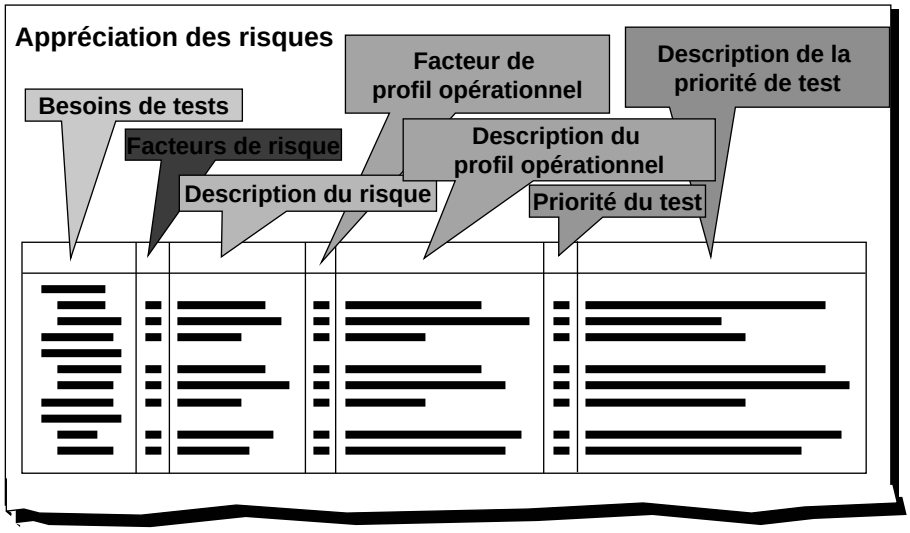
 Learning software process with UPEDU Traduction et adaptation par BTD 67

Résumé du calendrier du plan de test



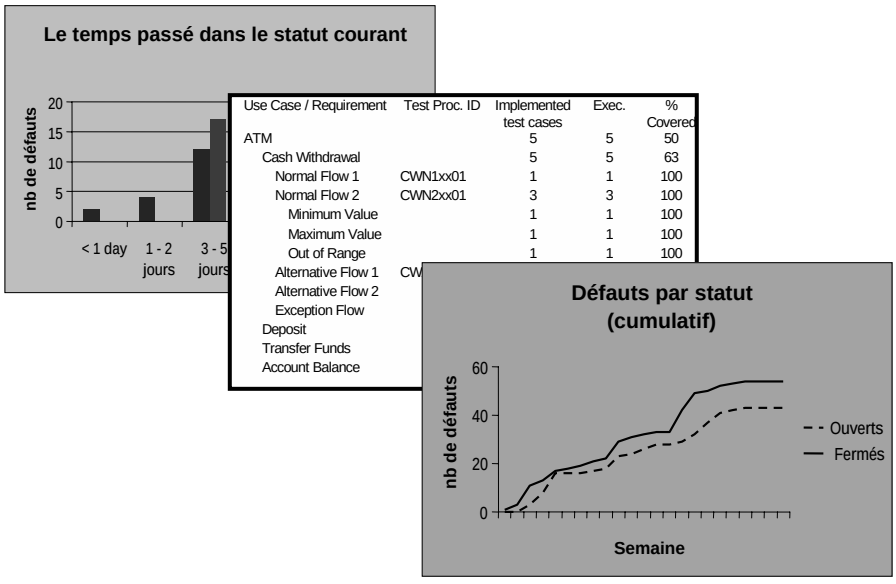
 Learning software process with UPEDU Traduction et adaptation par BTD 68

Matrice d'appréciation de risques



 Learning software process with UPEDU Traduction et adaptation par BTD 69

Un rapport d'évaluation des défauts



 Learning software process with UPEDU Traduction et adaptation par BTD 70

Discipline de tests

- Relier le test à la qualité
- Définir les activités de test
- Elaborer la discipline de test

- Evaluer les activités de test
  - Les défauts sont-ils significatifs ?
  - Les activités de tests sont-elles sérieux ?
  - La qualité du produite est-elle atteinte ?

 Learning software process with UPEDU Traduction et adaptation par BTD 71

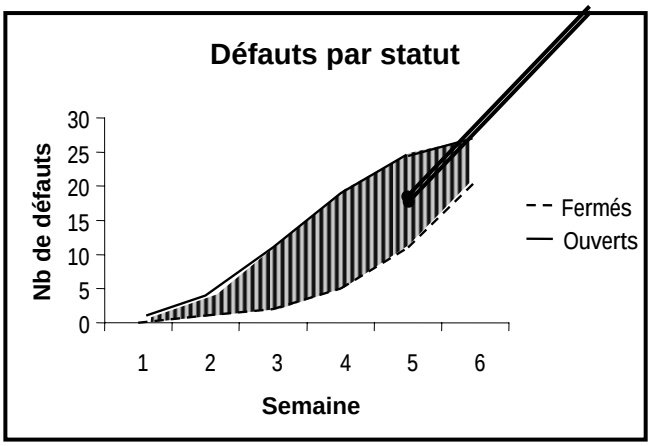
Quatre principaux paramètres de l'analyse des défauts

- Statut
  - Ouvert, solutionné, fermé, etc..
- Priorité
  - A solutionner immédiatement, priorité haute, traitement normal, priorité basse.
- Gravité
  - Erreur fatale, fonction majeure ne fonctionne pas, contrariété mineure.
- Origine
  - Besoin, architecture, module N, bibliothèque.

 Learning software process with UPEDU Traduction et adaptation par BTD 72

Rapport d'évolution du ratio de défauts identifiés

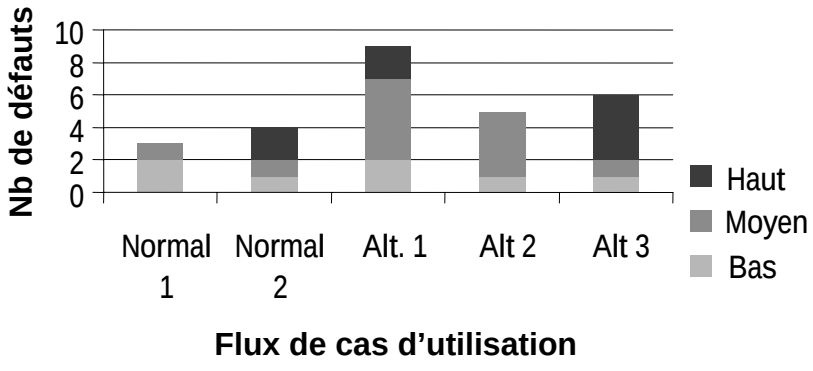
Ecart de qualité



 Learning software process with UPEDU Traduction et adaptation par BTD 73

Rapport de densité de défauts

Densité de défauts par le flux de cas d'utilisation et leur importance



 Learning software process with UPEDU Traduction et adaptation par BTD 74

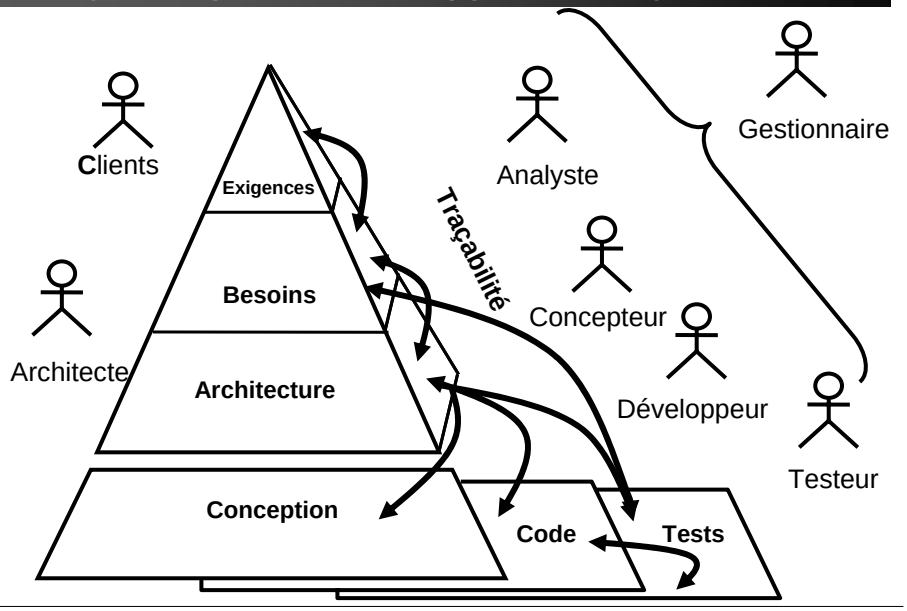
Gestion de configuration et des changements
CCM - Configuration and Change Management

- Définition de la problématique
  - Le monde changeant du développement de logiciels
  - Les activités de SCCM
  - Espaces de travail

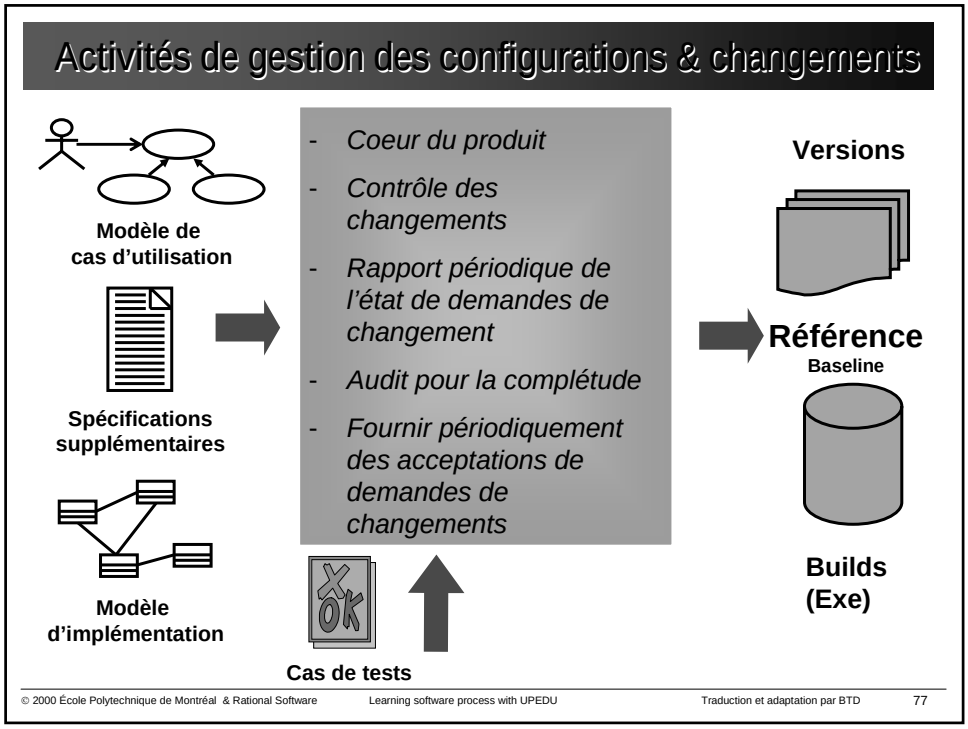
- Mise en évidence des aspects opérationnels de l'approche
- Gérer les configurations et les changements du logiciel
- Mettre en place une gestion de configurations et des changements

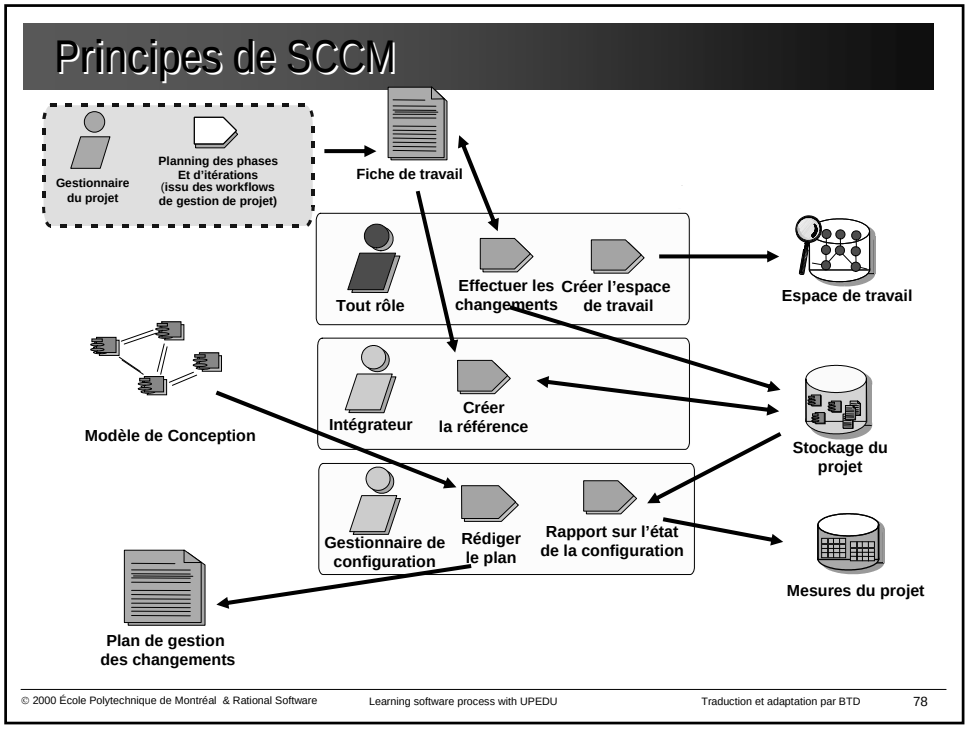
 Learning software process with UPEDU Traduction et adaptation par BTD 75

La dynamique du développement logiciel

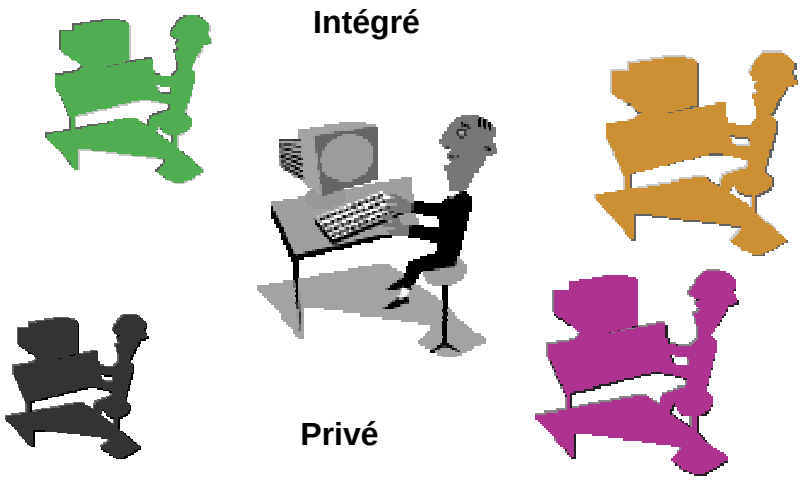


 Learning software process with UPEDU Traduction et adaptation par BTD 76


Activités de gestion des configurations & changements
Modèle de cas d'utilisation
Spécifications supplémentaires
Modèle d'implémentation
- Coeur du produit
- Contrôle des changements
- Rapport périodique de l'état de demandes de changement
- Audit pour la complétude
- Fournir périodiquement des acceptations de demandes de changements
Cas de tests
Versions
Référence
Baseline
Builds (Exe)
© 2000 École Polytechnique de Montréal & Rational Software
Learning software process with UPEDU
Traduction et adaptation par BTD
77


Principes de SCCM
Gestionnaire du projet
Planning des phases Et d'itérations (issu des workflows de gestion de projet)
Fiche de travail
Tout rôle
Effectuer les changements
Créer l'espace de travail
Espace de travail
Modèle de Conception
Intégrateur
Créer la référence
Stockage du projet
Gestionnaire de configuration
Rédiger le plan
Rapport sur l'état de la configuration
Mesures du projet
Plan de gestion des changements
© 2000 École Polytechnique de Montréal & Rational Software
Learning software process with UPEDU
Traduction et adaptation par BTD
78

Espaces de travail (Workspace)



Voir :

UPEDU Concept: Workspaces

 Learning software process with UPEDU Traduction et adaptation par BTD 79

CCM - résumé

- Définition de la problématique
- Mise en évidence des aspects opérationnels de l'approche
  - Identification des éléments de configuration du logiciel
  - Contrôle de la référence et des changements
  - Etat comptabilisé des composants et des changements
  - Audit fonctionnel et physique
- Gérer les configurations et les changements du logiciel
- Mettre en place une gestion de configurations et des changements

Voir :

UPEDU GUIDELINES:
Important decisions in Configuration and Change Management

 Learning software process with UPEDU Traduction et adaptation par BTD 80

Les références sont les versions stables du produit

- Les composants de SCCM pour gérer le contrôle.
- Identifier les collections d'éléments de la configuration qui se réfèrent à une version unique de l'artéfact
- Instantané (Snapshot) des artéfacts de développement dans le modèle d'implémentation
- Les versions stables du produit avec les rapports d'erreurs et les rapports de demandes de changements sont stockées
- Référence officielle sur laquelle le travail à venir sera basé
- Référence formelle ou informelle

Voir :

UPEDU Concept: Product Directory Structure
UPEDU Concept: Baselining

 Learning software process with UPEDU Traduction et adaptation par BTD 81

Les contrôles pour éviter la confusion

- Mises à jours simultanées
- Développeurs multiples
- Equipes multiples
- Sites multiples
- Variantes et versions multiples
- Itérations multiples
- Releases multiples
- Projets multiples
- Plates-formes multiples
- Notifications limitées
  - Certains développeurs ne sont pas informés de l'évolution des artéfacts partagés

Sans le contrôle explicite, les développements parallèles conduisent à un chaos

 Learning software process with UPEDU Traduction et adaptation par BTD 82

Trois étapes dans le statut comptable

Enregistrement : **Défauts**
Notification : **Statut des composants et des changements**
Recueil : **Statistiques**

- Soumis
- Accepté
- Analysé
- Assigné
- Conçu
- Implémenté

- En test - vérification
- Intégré
- En test système
- Terminé
- Annulé
- En attente

 Learning software process with UPEDU Traduction et adaptation par BTD 83

Audit fonctionnel et physique

- Audit fonctionnel
  - Vérifier que les performances actuelles des éléments de la configuration actuelle du logiciel satisfont les besoins exprimés
    - **Préparer** la matrice de vérification
    - **Vérifier** que tous les changements demandés ont été implémentés
    - **Documenter** les divergences et définir les actions correctives
- Audit physique
  - Vérifier que les artefacts du cœur ont des bonnes versions.
    - **Créer** la liste des éléments sous la Gestion des Changements
    - **Inspecter** des éléments maintenus
    - **Pointer** tous les problèmes en attente
    - **Vérifier** que tous les artéfacts sont compatibles
    - **Créer** une liste de divergences

 Learning software process with UPEDU Traduction et adaptation par BTD 84

Gestion de configuration et des changements
CCM - Configuration and Change Management

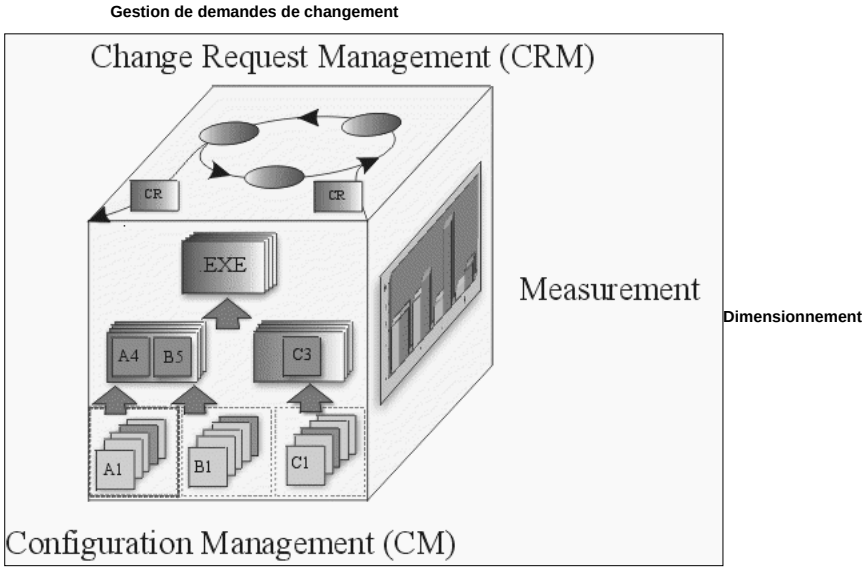
- Définition de la problématique
- Mise en évidence des aspects opérationnels de l'approche

- Gérer les configurations et les changements du logiciel
  - La vue gestion
  - Un scénario de gestions de changement de configuration du logiciel
  - Les étapes dans la gestion des changements du logiciel
  - L'évolution de la configuration du logiciel

- Mettre en place une gestion de configurations et des changements

 Learning software process with UPEDU Traduction et adaptation par BTD 85

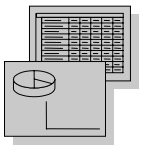
Le cube de gestion de configuration



 Learning software process with UPEDU Traduction et adaptation par BTD 86

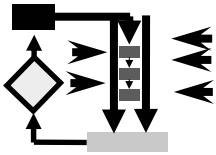
Scénario opérationnel de SCCM

Gestionnaire du projet



Rapport sur le statut du projet logiciel

Gestionnaire de configuration



Mise en place de mécanismes de contrôle de changements

Ingénieur logiciel

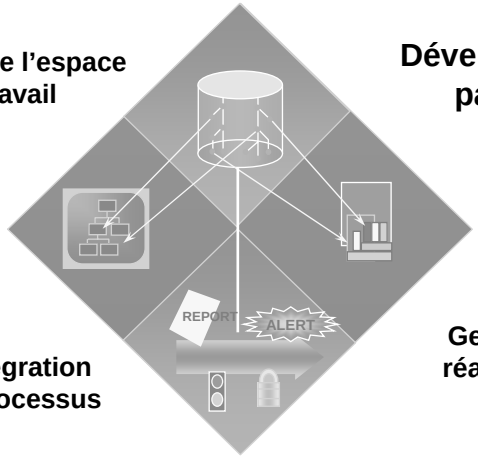
Communiquer des coordonner des tâches

 Learning software process with UPEDU Traduction et adaptation par BTD 87

Rôles du système de gestion de configurations et de changements

Gestion de l'espace de travail

Développements parallèles

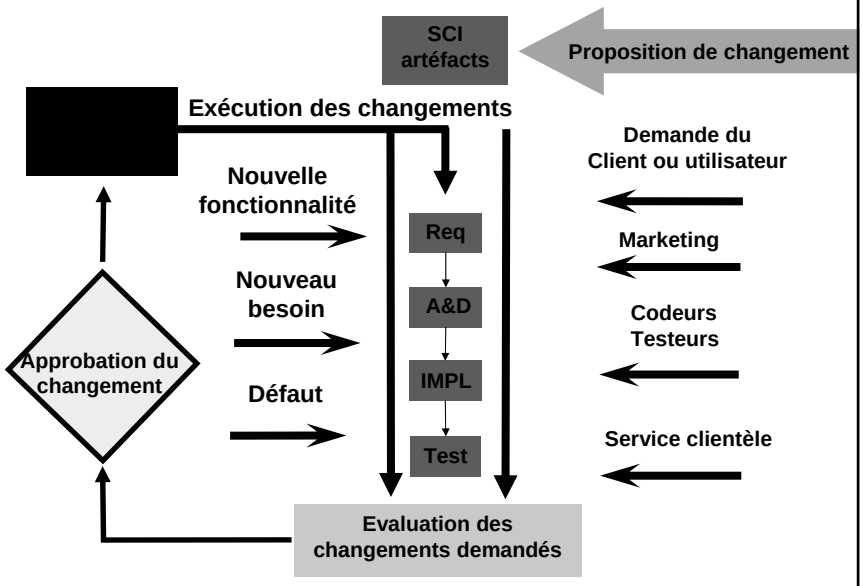


Intégration de processus

Gestion de réalisations

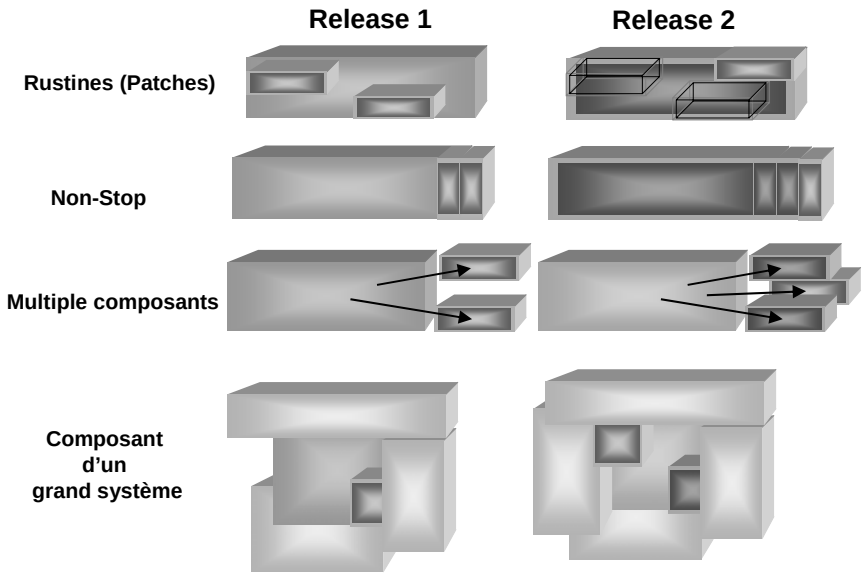
 Learning software process with UPEDU Traduction et adaptation par BTD 88

Contrôles des changes du produit



 Learning software process with UPEDU Traduction et adaptation par BTD 89

Quatre voies d'évolution du système



 Learning software process with UPEDU Traduction et adaptation par BTD 90

CCM – Résumé

- Définition de la problématique
- Mise en évidence des aspects opérationnels de l'approche
- Gérer les configurations et les changements du logiciel

- Mettre en place une gestion de configurations et des changements
  - Les aspects managériaux
  - Les aspects technologiques
  - Les aspects liés au processus

 Learning software process with UPEDU Traduction et adaptation par BTD 91

Les aspects managériaux

- Evaluer les systèmes de Gestion de changements
  - Prendre en compte les changements de technologies
- Prendre la bonne décision entre acheter et bâtir
  - Comprendre les coûts associés
- Personnes
  - Comprendre l'influence du système sur les activités
- Niveau de contrôle
  - Définir le degré de contrôle
- Niveau d'automatisation
  - Connecter les personnes et les contrôles

 Learning software process with UPEDU Traduction et adaptation par BTD 92

Concepts de CCM

- Construire une référence en concordance avec l'architecture du système
- Créer des espaces de travail sécurisés pour chaque développeur
  - Assurer l'isolation par rapport aux changements effectués dans d'autres espaces de travail
  - Contrôler tous les artéfacts logiciels - modèles, code, documentations, etc.
- Créer un espace de travail pour l'intégration
- Mettre en place un mécanisme de contrôle de changements renforcé
- Savoir quels changements interviennent dans chaque release
- Release est une référence établie et testée lors de l'accomplissement de chaque itération

 Learning software process with UPEDU Traduction et adaptation par BTD 93

Les aspects technologiques

- Adéquation des technologies
  - No silver bullet
- Ajuster les capacités de gestion
  - Autoriser des adaptations faciles
- Interopérabilité entre différents systèmes de gestion de configurations
  - Supporter différents systèmes
- Intégration et Base de données
  - Utiliser un stockage centralisé ou distribué
- Compatibilité ascendante
  - Maintenir l'utilisabilité pendant toute la durée de vie du produit

 Learning software process with UPEDU Traduction et adaptation par BTD 94

Les aspects liés au processus

- Adéquation de la définition et de la correspondance de la gestion de configuration (SCCM)
  - Correspondance entre les phases du cycle de vie et les itérations du projet
- Structure de l'organisation
  - Décider le nombre de groupes de gestion de configuration
- La gestion de configuration (CM) au niveau de l'entreprise, du projet et du programmeur
  - Choisir le niveau de gestion de configuration
- Rôles
  - Définir le degré d'implication de chaque rôle
- Complexité des applications
  - Prendre en compte les facteurs qui rendent les applications complexes

 Learning software process with UPEDU Traduction et adaptation par BTD 95