

Le processus de développement logiciel

Qu'est ce qu'un processus ?

- Un processus est une série de tâches qui fournissent un ou plusieurs délivrables.
- > Un processus définit qui fait quoi, quand et comment pour atteindre un objectif donné.
- Un processus de développement logiciel est un ensemble (structuré) d'activités que conduisent à la production d'un logiciel.

ou améliorées

Exigences nouvelles Processus d'ingénierie logicielle

Système nouveau ou amélioré

Le processus de développement logiciel

A quoi sert un processus logiciel?



➤ Un processus logiciel fournit une approche pour assigner des tâches et des responsabilités à l'intérieur d'une organisation.

➤ Un processus permet la production d'un logiciel de haute qualité avec un temps et un budget limité.

Les objectifs d'un processus de développement logiciel ?

L'objectif d'un processus est de produire un logiciel de haute qualité en respectant des contraintes de délai, de coûts et de performance

- Fournit les lignes directrices pour un développement efficace d'un logiciel de qualité;
- Réduit les risques et améliore les prévisions;
- > Décrit les meilleures méthodes de travail pour :
 - ☐ Apprendre des expériences précédentes;
 - ☐ L'amélioration du support de formation;
- > Établit une vision et une culture commune.

4

Les objectifs d'un processus de développement logiciel ?

Facilite la mise en œuvre : grâce aux six meilleures pratiques de Rational, le processus est facile à mettre en œuvre.

➤ Dicte au développeur comment implémenter en utilisant les outils standards de développement.

Maturité du processus de développement logiciel

- ➤ Le Modèle de Maturité (CMM) du SEI

 □ Initial
 □ Reproductible
 □ Défini
 □ Maitrisé
 □ Optimisé
- ➤ Pour maîtriser le processus de développement logiciel et assurer la qualité du logiciel, il faut :
 - Séparer le développement en plusieurs étapes
 - Organiser ces étapes et modéliser le processus de développement
 - Contrôler le processus de développement

Maturité du processus de développement logiciel

- > Niveau de maturité 1 : Initial
 - ☐ Chaotique : plans et contrôles inefficaces
 - Processus essentiellement non contrôlé, non défini;
 - Le succès dépend des individus.
- > Niveau de maturité 2 : Reproductible
 - ☐ Intuitif : dépend encore des individus
 - Procédures de gestion utilisées, gestion des configurations et assurance qualité;
 - Pas de modèle formel de processus.

Maturité du processus de développement logiciel

- Niveau de maturité 3 : Défini
 - ☐ Qualitatif: institutionnalise
 - Procédures formelles pour vérifier que le processus est utilise
- Niveau de maturité 4 : Maîtrisé
 - ☐ Quantitatif : Processus de mesures
 - Gestion quantitative de la qualité
- ➤ Niveau de maturité 5 : Optimisé
 - ☐ Améliorations retournées dans le processus
 - ☐ Stratégies d'amélioration du processus

Méthodes de développement logiciels

Il existe différents méthodes de développement logiciels qui organisent de façon différentes les activités de cycle de vie d'un logiciel.

Définition du processus unifié

Un processus de développement logiciel, c'est-à-dire qu'il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel (Jacobson, Booch, Rumbaugh 1999). UP est itératif, centré sur l'architecture, Piloté par des cas d'utilisation et orienté vers la diminution des risques.

Une méthode de développement pour les logiciels orientés objets vient pour compléter la systémique des modèles UML.

C'est un patron de processus pouvant être adaptée à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise. 10

Z.LAAREDJ 3 LMD 2020-2021

Les caractéristiques du processus unifié

1. UP est itératif et incrémental

Le projet est découpé en itérations ou étapes de courte durée qui permettent de mieux suivre l'avancement globale.

A la fin de chaque itération une partie exécutable du système finale est produite, de façon incrémentale (par ajout).

Les caractéristiques du processus unifié

2. UP est centré sur l'architecture

Tout système complexe doit être décomposé en partie modulaire afin d'en faciliter la maintenance et l'évolution. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modéliser en UML.

12

Les caractéristiques du processus unifié

3. UP est guidé par les cas d'utilisation d'UML

Le but principal d'un système d'informatique est de satisfaire les besoin de client. Ce qui inclut l'utilisateur dans le processus de développement.

Le cas d'utilisation permettent d'illustrer ces besoins. Ils détectent puis décrivent les besoin fonctionnel et leur ensemble constitue le modèle de cas d'utilisation qui montre les fonctionnalités complètes du système.

Les caractéristiques du processus unifié

4. UP est piloté par les risques

Les risques majeurs du projet doivent être identifiés au plus tôt mais surtout levés le plus rapidement. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.

La méthode RUP

Qu'est ce que « RUP » ?

RUP

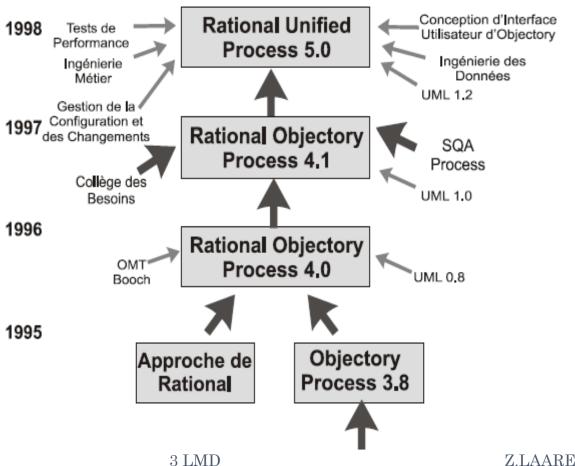
- ☐ Rational Unified Process
- ☐ Dérivée de UP (UP a été créée en 1996)
- ☐ Commercialisée par IBM en 1998

Méthode de développement logiciel

- ☐ Itérative
- ☐ Incrémentale
- ☐ Pilotée par les cas d'utilisation
- ☐ Centrée sur l'architecture et la réduction des risques
- ☐ Produit de qualité

Un bref historique du RUP

Le RUP est une instance spécifique et très détaillée d'un processus plus générique décrit par Ivar Jacobson, Grady Booch et James Rumbaugh dans le manuel pédagogique The Unified Software Development Process.



La méthode RUP (Concept et vocabulaire)

Les principaux éléments du processus

Rôles

- · Comportement et responsabilités d'un ensemble de personnes.
- · Décrit le Qui.

Activités

- · Opération exécutée au sein d'un état.
- · Une activité peut être interrompue.
- · Décrit le Comment

Artefacts

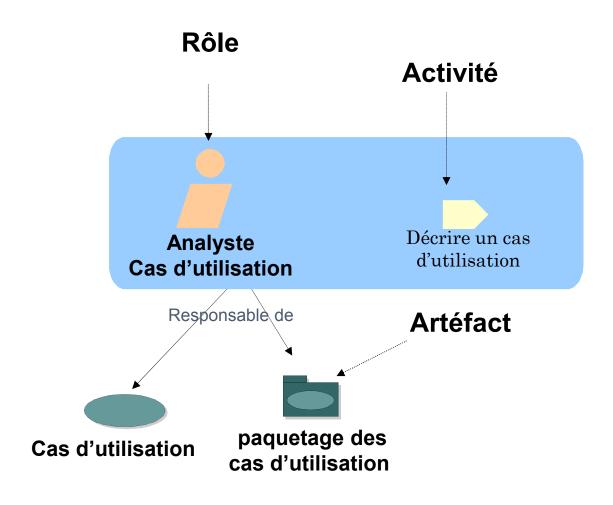
- Élément d'information, produit ou utilisé lors d'une activité de développement logiciel (modèle, source...)
- · Décrit le Quoi

Disciplines

- · Décrire les enchainements d'activités
- · Décrit le Quand

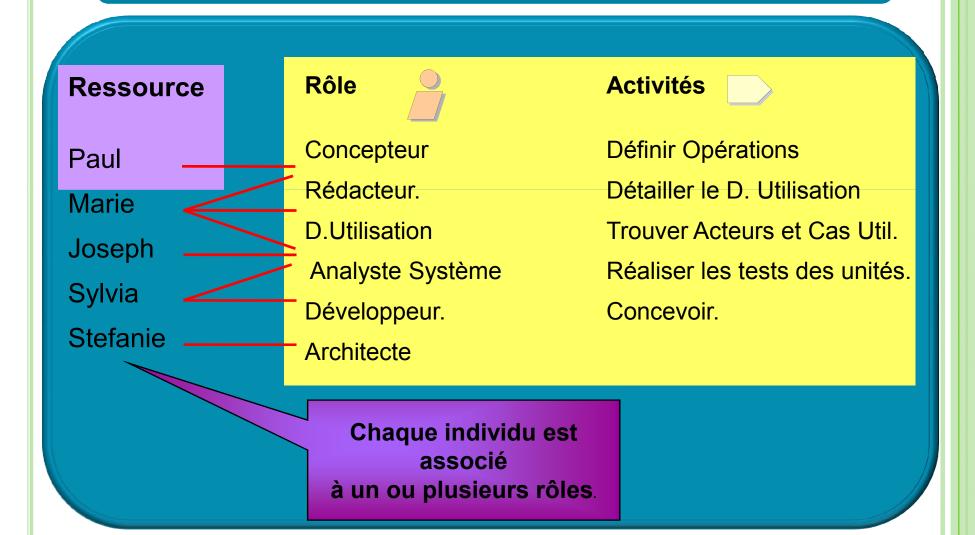
17

La méthode RUP (Concept et vocabulaire)



La méthode RUP (Concept et vocabulaire)

Les Rôles dans La Planification Des Ressources

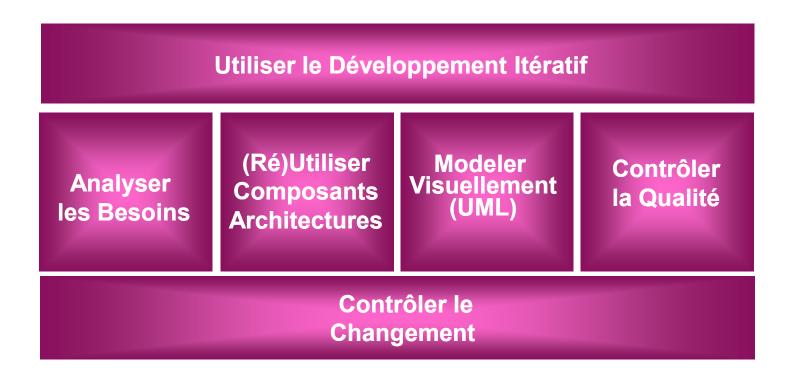


Principes de RUP

- ➤ Pilotée par les cas d'utilisation
- > Construction d'un système à base de composants
- ➤ Adaptable aux changements
- > Gestion des risques
- Livraison de qualité
- > Concentrée sur le code exécutable
- > Travail en équipe

Principes de RUP

Le processus Unifié Rational décrit comment appliquer les six directives de l'ingénierie logicielle



21

Principes de RUP

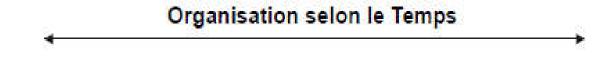
- Les six meilleures pratiques fournissent les bases pour le Processus Unifié de Rational.
- ➤ Cependant, cette application nécessite des instructions étapes par étapes.
- ➤ Ces instructions sont fournies dans le Processus Unifié de Rational, qui comprend toutes les activités devant être appliquées pour construire un logiciel.

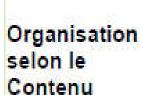
Les deux visions du RUP

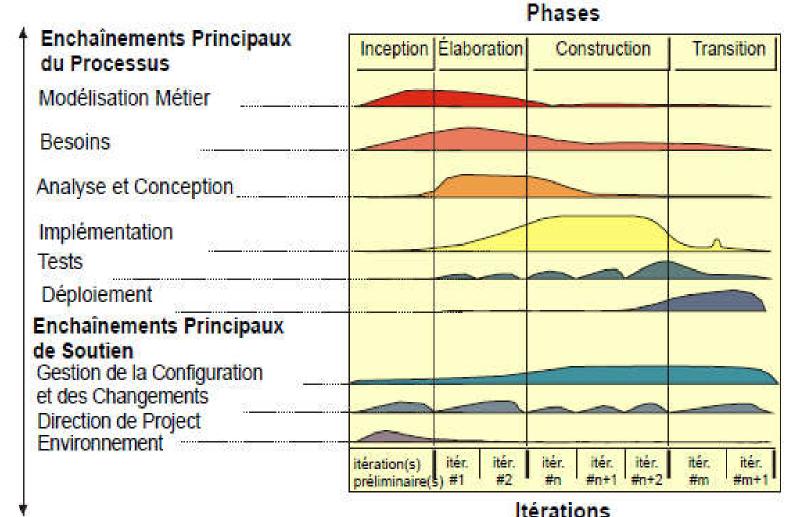


- L'axe vertical : représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature. Cette dimension exprime l'aspect statique du processus termes de composants, de processus, d'activités, d'enchaînements, d'artefacts et de travailleurs.
- L'axe horizontal: représente le temps et montre le déroulement du cycle de vie du processus; cette dimension exprime l'aspect dynamique du processus en terme de cycles, de phases, d'itérations et de jalons.

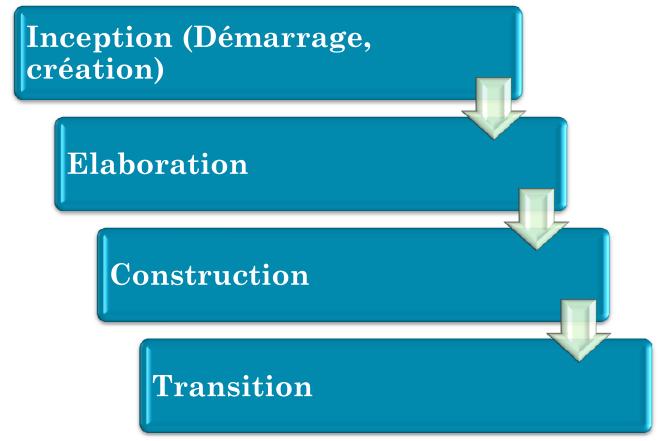
Les deux visions du RUP







L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques. Cela est fait en appliquant 4 phases.



25

Inception

- > Définit le champ d'action du projet
- > Durant l'étude d'opportunité (démarrage), les objectifs du projet sont définis.
 - ☐ En identifiant tous les acteurs et les cas d'utilisation, et en dessinant les cas d'utilisation essentiels (20% du modèle).
 - ☐ Un plan de gestion de projet est fait pour déterminer les ressources nécessaires pour le projet.

Elaboration

- > Durant l'élaboration, il faut concentrer sur deux choses :
 - ☐ Avoir une bonne connaissance des besoins (exigences très avancées (90%)) et ;
 - ☐ Etablir une base de l'architecture.
- > Possibilité d'éliminer beaucoup de risques;
- > Avoir une bonne idée de ce qui doit être fait, et
- Une bonne estimation des ressources et des coûts;
- Programmation largement engagée.

Construction

➤ Durant la Construction, on développe le produit en plusieurs itérations pour une version bêta (développement de toutes les fonctionnalités, doc incluse).

Transition

➤ Durant la Transition, on prépare le produit pour l'utilisateur final et la formation, l'installation, le support, autrement dit déployer dans l'environnement opérationnel.

RUP: Processus agile ou lourd?

Agile

Faible formalisme

Concentré sur le code exécutable

Choix idéal pour les petits projets

Lourde

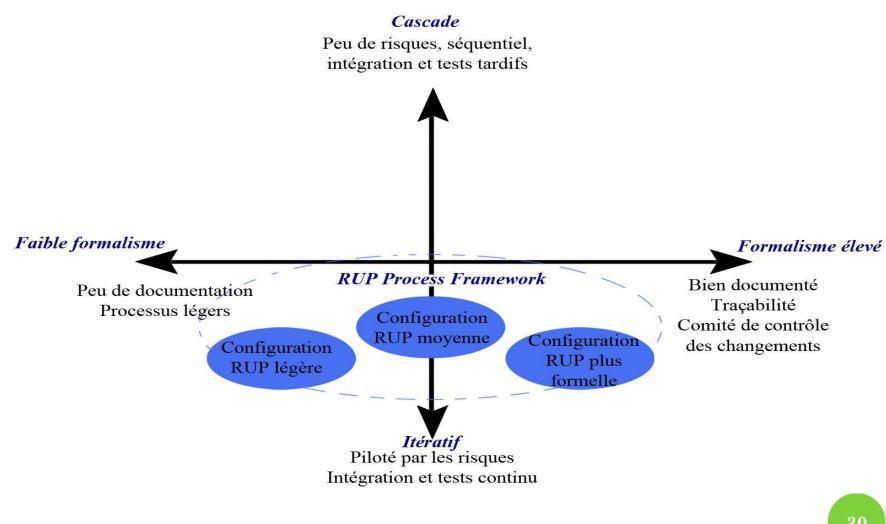
Extrêmement formaliste

Production de nombreux artefacts

Pour les projets complexes

29

Ou se situe RUP?



Les avantages de RUP

- Traçabilité à partir des Uses Cases jusqu'au déploiement grâce à l'utilisation de l'UML,
- > Approche basée sur l'architecture
- ➤ Gestion des risques dans les projets, autrement dit les risques sont atténués dés le début du projet car la planification est complète et détaillé,
- > Cadre propice à la réutilisation,
- ➤ Le changement est mieux géré, car il est possible de recadrer le projet après une itération.
- > Vérification constante de la qualité

Les inconvénients de RUP

- Coût de personnalisation souvent élevé
- > Très axé processus
- > Vision non évidente ni immédiate
- > Management lourd à mettre en place.

La méthode XP

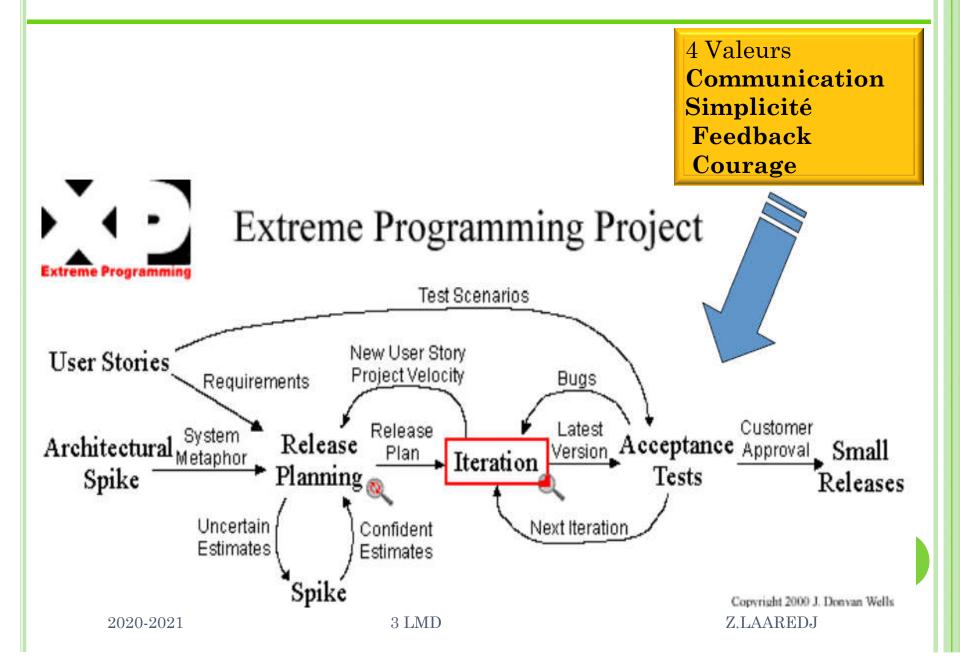
XP

- eXtreme Programming
- ☐ Une méthode de développement agile, orientée projet informatique et dont les ressources sont régulièrement actualisées.
- ☐ Destinée à accélérer la réalisation des projets de type flexible.

Principe de base

☐ Participation du client au développement

La méthode XP



Valeurs XP

Communication

- > Développement = Effort collectif de création
 - ☐ Avoir une vision commune et pouvoir se synchroniser
 - ☐ Qualité de la communication
- > Communication directe et le contact humain
 - ☐ Faiblesse pour la traçabilité et la structuration
 - ☐ Augmentation de la réactivité
- > Communication écrite présente, en général par du code

Valeurs XP

Simplicité

- « La chose la plus simple qui puisse marcher »
- > Eviter la complexité inutile dans le code
- > Toute duplication doit être éliminée

Valeurs XP

Retour d'information (Feedback)

- > Boucles de feedback pour réduire les risques
 - ☐ Connaître l'état du projet
 - ☐ Rectifier le tir si nécessaire
- > Facteur de qualité
 - ☐ Acquisition d'expérience
 - ☐ Amélioration constante du travail

Valeurs XP

Courage

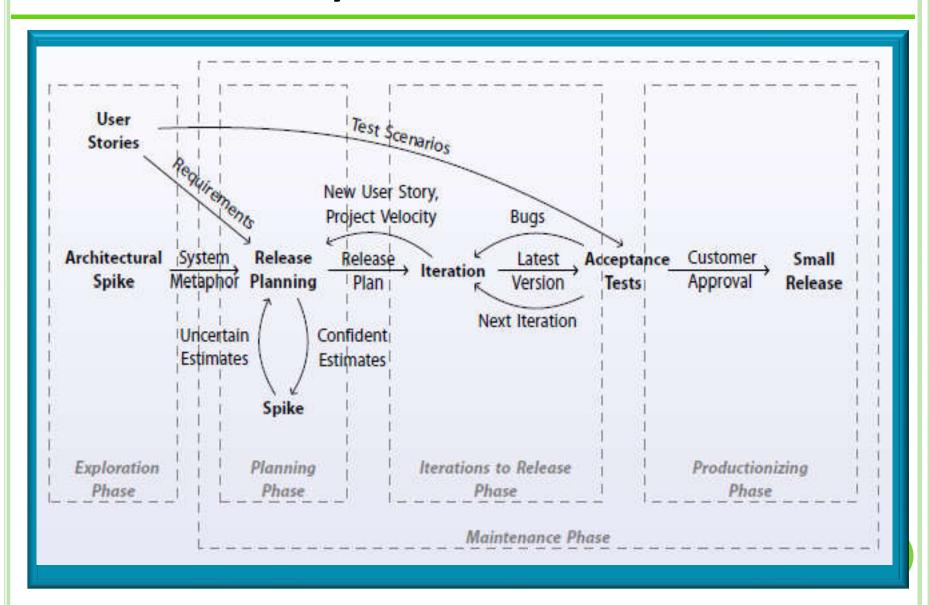
- > Se lancer dans un projet non entièrement spécifié;
- > Accepter de remanier une portion de code devenue complexe;
- > Appliquer les valeurs de feedback et de communication;
 - ☐ Accepter de montrer ses propres limites
 - ☐ Maintenir une transparence complète

La méthode XP

> Caractéristiques

- ☐ Rapidité de développement par cycle
 - Déterminer les besoins clients;
 - ◆ Transformer les besoins en fonctions à réaliser et en tests fonctionnels;
 - Chaque développeur s'attribue avec un binôme des tâches;
 - Lorsque les tests sont concluants, l'application est livrée,
 - ◆ Le cycle se répète tant que le client peut fournir des besoins.
- ☐ Réactivité aux changements
- ☐ Qualité du code et des tests
- ☐ Travail en équipe

Cycle de vie de XP



Pratiques XP



Programmation

- Développement piloté par les tests
- Conception simple
- Remaniement



Collaboration

- Programmation en binôme
- Responsabilité collective du code
- Règles de codage
- Intégration continue



Gestion de projet

- Client sur site
- Rythme durable
- Livraisons fréquentes
- Planification itérative

Principaux facteurs de succès de XP

Sur le plan Humain

- ☐ Quelques développeurs expérimentés... et ouverts
- ☐ Une équipe capable de travailler... en équipe

Principaux facteurs de succès de XP

Sur le plan Organisationnel

- ☐ Environnement de travail adapté (« War Room »)
- ☐ Hiérarchie consentante
- ☐ Culture d'entreprise adaptée
 - Pas de mérite basé sur les heures supplémentaires
 - Pas de jeu politique (gagnant-gagnant)
 - Pas d'attachement aux méthodes linéaires

Principaux facteurs de succès de XP

Sur le plan Technique

- ☐ Possibilité de travailler sur des portions réduites de l'application
 - Valable dans le cas d'une reprise d'un gros projet
 - Redécouper le logiciel dans un tel cas
- ☐ Langage de programmation permettant des mécanismes d'abstraction et de factorisation
 - ◆ La plupart des langages objets entrent dans cette catégorie
- ☐ Disposer d'un outil de gestion de versions efficace

Les points forts de la méthode XP

- > Itératif
- > Simple à mettre en œuvre
- Fait une large place aux aspects techniques : prototypes, règles de développement, tests...
- > Innovant: programmation en duo...

Les points faibles de la méthode XP

- ➤ Ne couvre pas les phases en amont et en aval au développement : capture des besoins, support, maintenance, tests d'intégration...;
- Élude la phase d'analyse, si bien qu'on peut dépenser son énergie à faire et défaire;
- Assez flou dans sa mise en œuvre: quels intervenants, quels livrables ?

2TUP

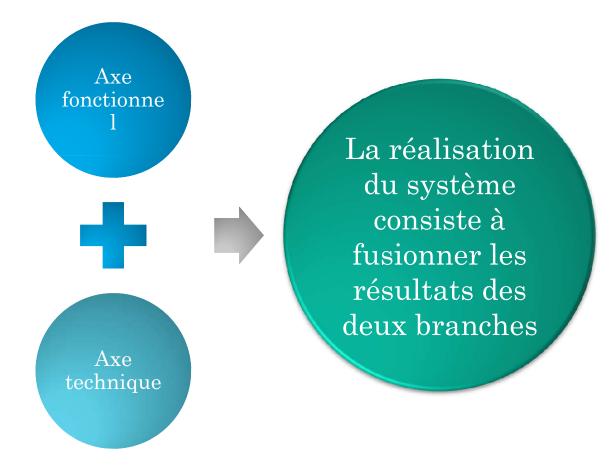
- □ 2 Track Unified Process
- ☐ C'est un processus qui répond aux caractéristiques du Processus Unifié.
- ☐ Apporte une réponse aux contraintes de changement continuel imposées aux systèmes d'information de l'entreprise.
- ☐ Renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes.

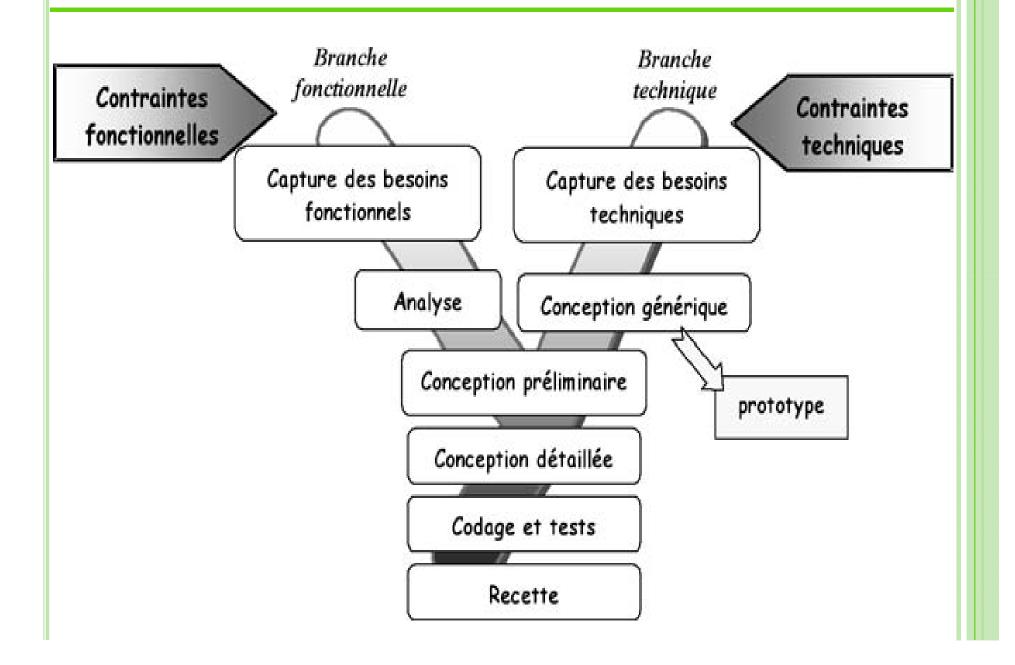
2TUP

« 2 Track» signifie littéralement que le processus suit deux chemins. Il s'agit des «chemins fonctionnels » et « d'architecture technique », qui correspondent aux deux axes de changement imposés au système d'information.



Principe de base





Un processus piloté par les risques

4 principaux risques

Les solutions apportées par ce processus

L'inadéquati on aux besoins des utilisateurs L'incapacité de l'architecture technique à répondre aux contraintes opérationnelles

Le non respect des couts et délais

Le manque de qualité

Gestion prioritaire des deux premiers risques

Politique d'incréments

Un processus piloté par les exigences des utilisateurs

Deux types d'acteurs

L'utilisateur consommateur des fonctions du système

L'utilisateur exploitant le système

La branche gauche est chargée de capturer les besoins fonctionnels auprès des utilisateurs consommateurs

La branche droite est chargée de capturer les besoins techniques auprès des utilisateurs exploitants

La méthode 2TUP dans la pratique

Capture des besoins



Cahier des charges

Acteurs

Messages

Modélisation du contexte

Besoins fonctionnels

Cas d'utilisations

Classes candidates

Validation et consolidation

Besoins techniques

Spécifications techniques

Spécifications de l'architecture

Cas d'utilisation techniques

La méthode 2TUP dans la pratique

Analyse

Découpage en catégorie

Découpage en catégorie

Dépendances

Modèle statique

Classes

Associations

Opération

Optimisation

Modèle dynamique

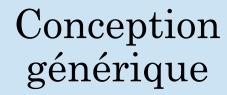
Scénarios

Diagrammes états transitions

Diagrammes d'interaction

Validation

Conception d'architecture



Framworks techniques

Modèle logique

Développement de prototype

Conception préliminaire

Modèle de déploiement/ exploitation

Interfaces utilisateurs

Interface catégories

Conception IHM

Conception détaillée

Tout

Les apports de la méthode 2TUP

Capitalisation de la connaissance de l'entreprise



investissement pour le moyen et long terme

Capitalisation d'un savoir-faire technique



investissement pour le court et moyen terme



2020-2021 3 LMD Z.LAAREDJ

Les points forts de la méthode 2TUP

- > Itératif
- > Fait une large place à la technologie et à la gestion du risque
- > Définit les profils des intervenants, les livrables, les plannings,

les prototypes

Les points faibles de la méthode 2TUP

- ➤ Plutôt superficiel sur les phases situées en amont et en aval du développement : capture des besoins, support, maintenance, gestion du changement...
- ➤ Ne propose pas de documents types