

Rational Unified Process

CANESSA Marine
FINOCCHIARO Emmanuel
PILOT Guillaume
TREFOLONI Guillaume

Sommaire

- Historique
- Présentation
- Caractéristiques
- Déroulement
- Les activités
- Les phases
- Les risques
- Conclusion

1996: apparition de UP (Unified Process)

- ⊗ Harmoniser processus de développement
- ⊗ Capitalisation des connaissances

1998: apparition de RUP (Rational Unified Process)

- ⊗ Version commerciale de UP
- ⊗ Démarche d'organisation
- ⊗ Description et modélisation métier
- ⊗ Production de livrables documentaires



2003: Rachat de rational Corporation
et sortie de RUP

Processus de développement logiciel

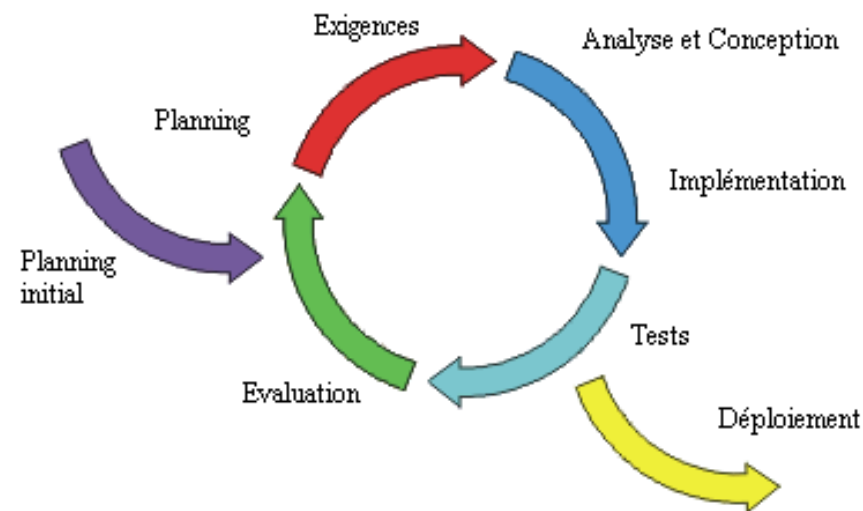
- Itératif
- Incrémental (à base de composants)
- Centré sur l'architecture
- Piloté par des cas d'utilisation (UML)
- Orienté vers la diminution des risques



Composants:

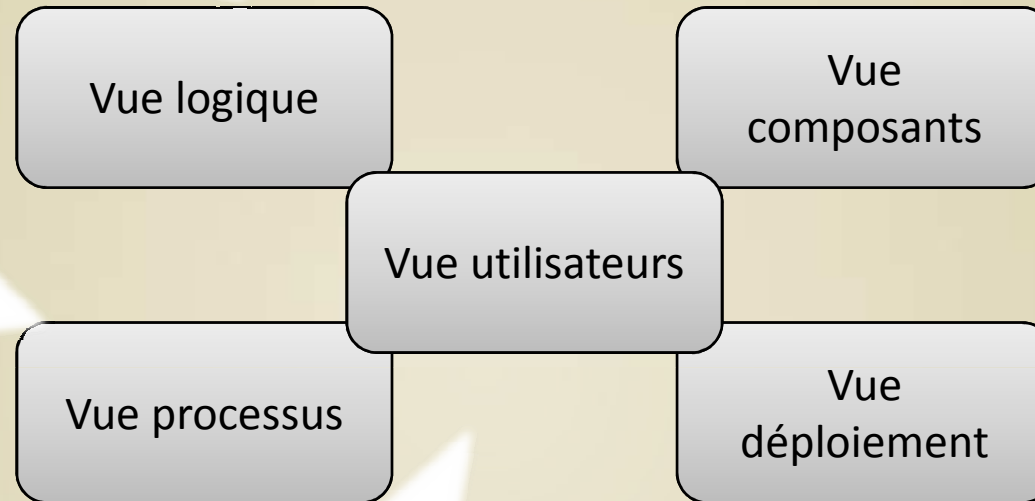
- Rôles
- Productions
- Tâches

- ❖ Itérations de courte durée (15 jours à 2 mois suivant les projets)
- ❖ Fonctionnalités principales développées dès le départ
- ❖ Re-cadrages dès le début
- ❖ Expression des besoins par prototypes
- ❖ Implication du client : versions exécutable



Représentation du SI sur les données:
Diag. de classes, séquence ...

Traduction du Si en modules:
Diag. composants



Aspect fonctionnel du SI:
Diag. d'états transition, Diag. d'activités

Guide l'analyse des besoins,
cimente les vues:
cas d'utilisation, scénarios

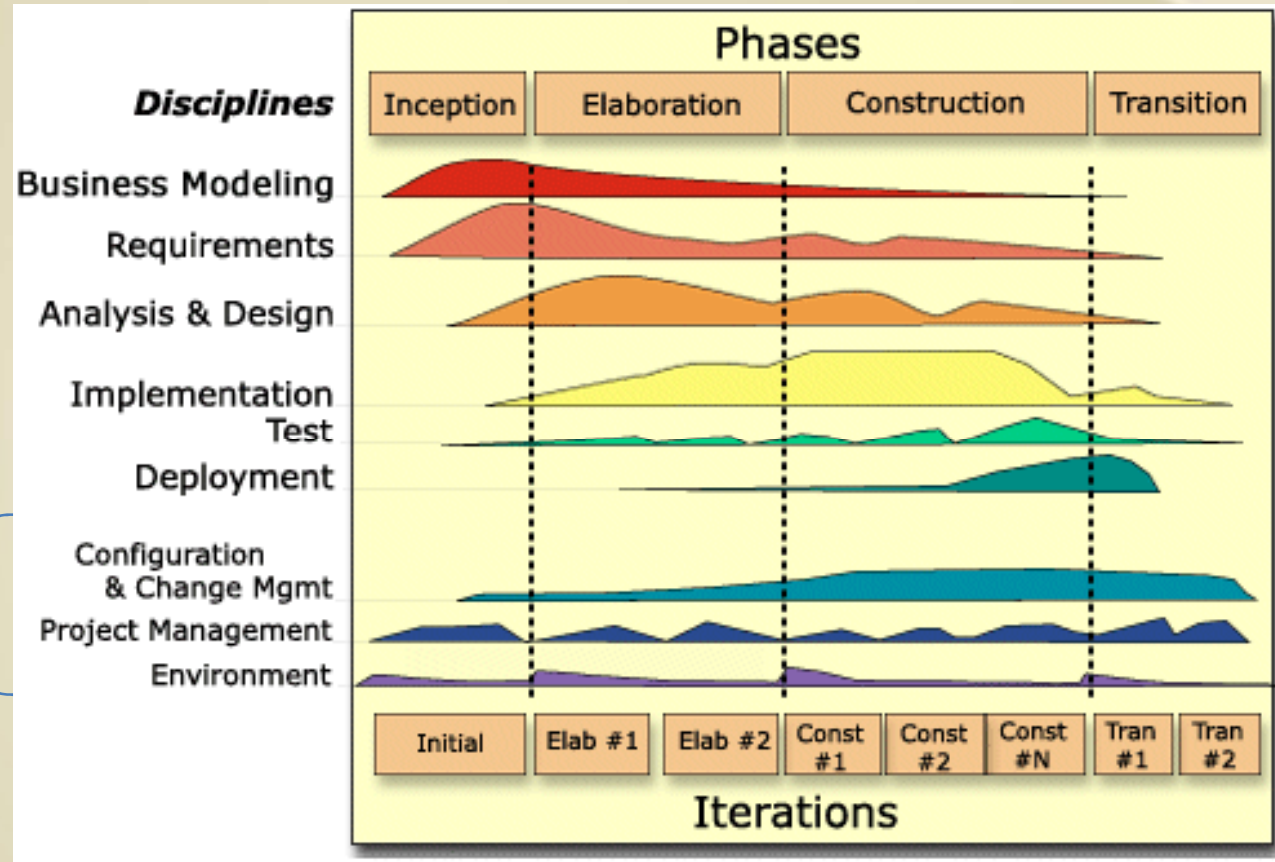
Projection des composants sur le matériel:
Diag. de déploiement



Chaque phase est itérative

Engineering

Support



Disciplines d'engineering (1/2)

- ◉ Expression des besoins
 - Communication avec le client
 - Cas d'utilisation
- ◉ Pré-requis
 - Définition des pré-requis du système
- ◉ Analyse et design
 - Compréhension des besoins
 - Spécifications
 - Scénarios, définition de l'architecture ...
 - Première ébauche du modèle de conception



- ◉ Conception
 - Prend en compte les contraintes (langages, composants, SE)
 - Détermine les principales interfaces
 - Décomposition du travail d'implémentation

Disciplines d'engineering (2/2)

◆ Implémentation

Production de codes sources, scripts, exécutables
Sous forme d'unité indépendante

◆ Test

Tester la construction, la cohésion des objets
Tester que les objectifs sont atteints
À planifier pour chaque itération
Créer des cas de tests



◆ Déploiement

Produire , packager, distribuer,
installer le produit
Assurer le support

Disciplines de support

- ◆ Environnement
 - Activités nécessaires aux processus du projet
 - Sert à la « customisation » du projet
- ◆ Configuration et changement
 - Réponses aux changements du client lors du déroulement du projet
- ◆ Projet
 - Gestion d'un projet itératif
 - Gestion des itérations
 - Décrit l'ensemble du projet



Que va faire le système ?

Quelle va être l'architecture générale?

Les délais? Les coûts? Les ressources? Les moyens?

- Opportunité-faisabilité,
- Cahier des charges,
- Liste de fonctionnalités,
- Dictionnaire de données,

Évènement final : on poursuit ou non le projet. Maquette



⊗ Inception = Analyse des besoins

Spécification plus détaillée

Planification

Créer une architecture de référence

Identifier les risques, le coût et le calendrier

Définir les niveaux de qualité à atteindre

Formuler les cas d'utilisation

Planifier la phase de construction

- Élaborer une offre (calendrier, personne, budget)

Évènement final : architecture du système, prototype de l'architecture



Elaboration

Programmation incrémentale
Produit complet (tous les UC)

Évènement final : version bêta utilisable



Construction

Installation

Formation des utilisateurs clients

Elaboration des manuels

Mise en œuvre d'un service d'assistance

Correction des anomalies constatées

Préparation de la maintenance

Évènement final : première version finale utilisée



Transition

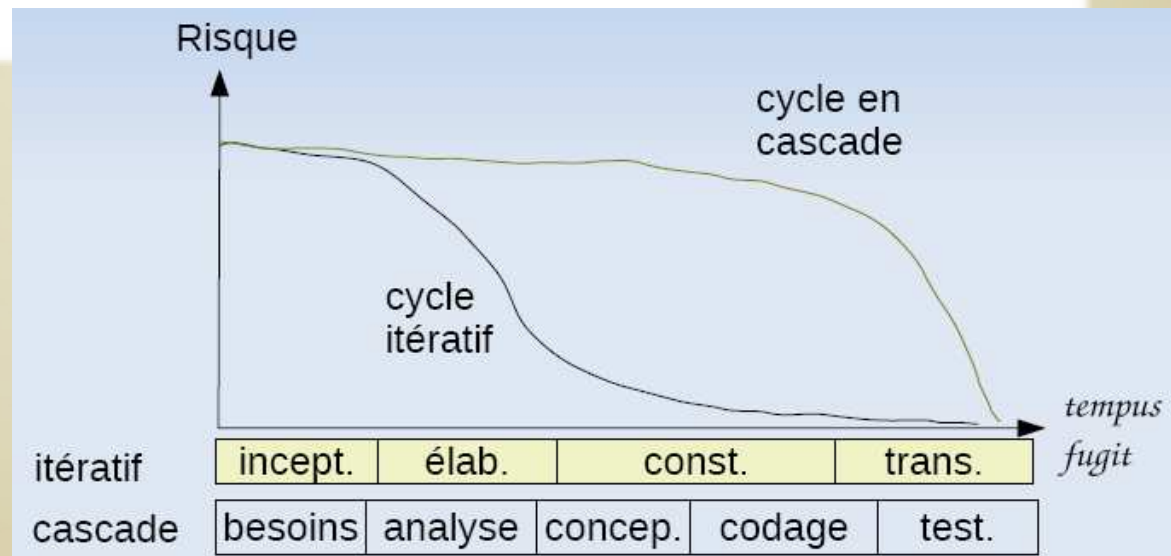
- Au final : ne pas répondre aux besoins
- Risques commerciaux
- Concurrence ? Occuper le terrain avec solution minimale ?
- Risques financiers
- Capacités de financement non surpassées
- Risques techniques
- Choix technique éprouvé ?
- Risques de développement
- Equipe productive immédiatement, formation nécessaire ?



⊗ Limitations des risques

Réduction des risques (1/2)

- Réduction possible en ciblant directement les besoins
- Illustrer concrètement les besoins par :
 - Maquette
 - Prototypes
- Régulièrement présentés au client
- Résultat tangible = mesure facilitée de l'avancée du projet
- Plus forte motivation de l'équipe



Réductions des risques (2/2)

- ❖ *Étude d'opportunité* : limitation du risque du projet
- ❖ *Maquette* : Maquette retouchable de 50 à 100 %
- ❖ *Élaboration* :
 - Réduction des risques d'incompréhension avec les usagers.
 - Appréhension des risques d'architecture.
 - Prototype d'architecture retouchable à 25% environ
- ❖ *Construction* : intégration progressive des besoins, du plus au moins prioritaires. Version bêta retouchable à moins de 10%



- ❖ *Transition* : risque de prise en main réduit par un déploiement progressif et par l'implication de l'utilisateur dans les phases précédentes.
Versions retouchables de 4 à 1%

- Cadre générique
- Gestion des risques dans les projets
- Cadre propice à la réutilisation
- Approche basée sur l'architecture
- Traçabilité à partir des cas d'utilisation jusqu'au déploiement



✦ FORCES

- Concepts difficiles à appréhender et à implanter dans une gestion de projet
- Assimilation difficile par les développeurs et utilisateurs
- Très axé processus
 - Peu de place pour le code et la technologie
- Vision non évidente ni immédiate
- Coût de personnalisation souvent élevé
 - Autres logiciels propriétaires (Rational) indispensables



✦ FAIBLESSES