

IFT2255 - Génie logiciel

Processus de développement

Bruno Dufour
dufour@iro.umontreal.ca

Cycle de vie du logiciel

Activités de développement

3

- Planification du projet
- Analyse et spécification
- Conception
- Implémentation
- Vérification
- Installation
- Maintenance
- En continu:
 - Documentation
 - Vérification et validation
 - Gestion

Planification (étude préliminaire)

4

- **Q: Est-ce possible?**
- Définition globale du problème
- Confirmer la faisabilité
 - évaluation des stratégies possibles
 - évaluation des ressources, coûts et délais
- Produire le calendrier du projet
- Trouver le personnel
- Lancer le projet
- Documents
 - rapport de planification

Analyse des besoins

5

- **Q: Quoi faire?**
- Cueillette d'informations
 - exigences fonctionnelles
 - qualités non-fonctionnelles (contraintes)
- Spécification du système
 - accord entre le développeur du système et le client / l'utilisateur
- Construction de prototypes (pour élaborer la spécification)
 - Prioriser les éléments de la spécification
 - Produire et évaluer des solutions alternatives
 - Examiner les recommandations avec le chef de projet et/ou le client...

Analyse des besoins

6

- Documents
 - cahier des charges / document de spécification (analyse)
 - prototype
 - plan de test

Conception

7

- **Q: Comment faire?**
- Conception architecturale
 - décomposition et organisation de l'application en modules plus simples définis par une interface.
 - ex: bases de données, environnement d'exploitation, interfaces
- Conception détaillée
 - Pour chaque module, description de la manière dont les services et fonctions sont réalisés
 - algorithmes essentiels
 - structures de données utilisées, etc.

Conception

8

- Documents
 - document de conception (spécification)
 - prototype
 - plan de test global
 - plan de test par module

Implémentation

9

- Traduction de la conception dans un langage de programmation ou mise en œuvre en utilisant des outils de développement
- Construire les composantes logicielles
- Documents
 - dossiers de programmation
 - code source commenté
 - prototype

Vérification

10

- **Q: Est-ce bien fait?**
- Évaluation de la solution en fonction de la spécification
- Différents niveaux de tests
 - Tests unitaires: par module
 - Tests d'intégration: composition de modules
 - Tests de système: logiciel entier
 - Tests d'acceptation: définis par le client
- Documents
 - rapport de vérification par test

Installation / déploiement

11

- Mise en fonctionnement opérationnel chez les utilisateurs
- Conversion des données
- Parfois restreint à des utilisateurs sélectionnés
 - alpha / beta testing

Maintenance

12

- Maintenance corrective: corriger les erreurs
- Maintenance adaptative: s'adapter à des changements d'environnement
- Maintenance perfective: améliorations
- Maintenance préventive: pour faciliter les opérations de maintenance à venir

Activités en continu

13

- Gestion
 - Du processus de développement (suivi de projet, révision, etc.)
 - De la configuration: politique de gestion des versions, des documents, politique de réutilisation
 - Des ressources humaines
 - Du risque

Activités en continu

14

- Vérification: «Construit-on le produit comme il faut?»
 - Le produit est-t-il correct (par rapport à la spécification)?
 - S'assurer de la qualité du produit (révisions et inspections)
 - S'assurer de satisfaire la spécification
- Validation: «Construit-on le bon produit?»
 - Le produit répond-il aux besoins du client?
- Documentation
 - Traçabilité

L'équipe de développement

Équipe de développement

16

- Analyste
- Concepteur
- Programmeur
- Testeur
- Formateur (*training specialist*)

Rôles des membres

17

- Planification Analyste
- Analyse Analyste
- Conception
 - architecturale Analyste, concepteur
 - détaillée Concepteur, programmeur
- Implémentation Programmeur
- Tests
 - unitaires Programmeur, testeur
 - intégration Testeur
 - système Testeur
- Installation Formateur

Processus de développement

Processus de développement

19

- Description abstraite et idéalisée de l'organisation des activités du développement d'un logiciel
- Décrit un ensemble de tâches ordonnées
- Doit être «personnalisé» pour l'entreprise de façon à définir l'ordonnancement idéal des activités
 - spécifier les artefacts à produire (types de documents, format, échéancier)
 - attribuer les activités & les artefacts aux acteurs
 - proposer des critères pour superviser l'évolution du projet, ses résultats et prévoir plans futurs (vérification, validation, documentation, etc.)
 - proposer une méthodologie pour gérer les changements tant dans le processus et que le logiciel

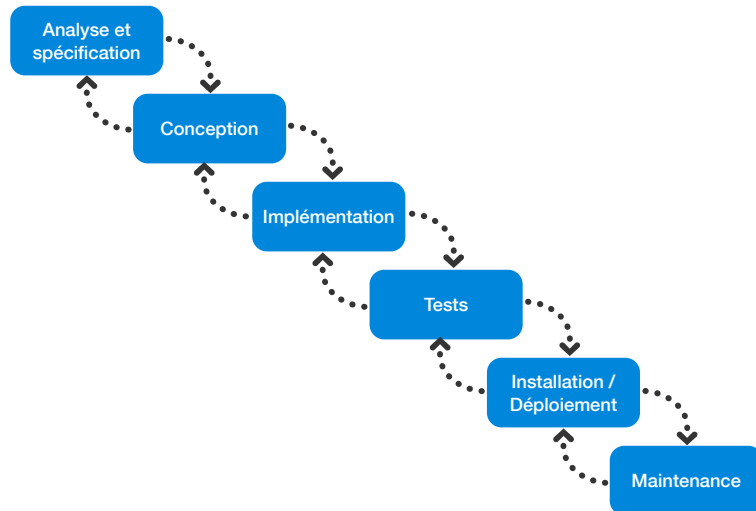
Quelques modèles existants...

20

- Modèle en cascade
- Modèle en V
- Modèle par prototypage
- Processus unifié (*Rational Unified Process - RUP*)

Modèle en cascade

21



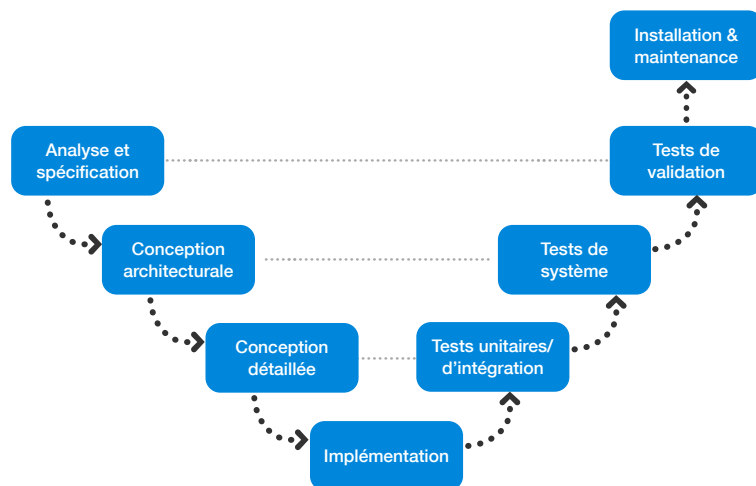
Modèle en cascade

22

- Axé sur la documentation
- Avantages
 - Simple et facile à comprendre
 - Permet une conception de grande qualité ("well thought-out")
 - A permis la normalisation des cadres conceptuels et terminologiques des différentes activités
 - Pertinent dans le cas des anciens systèmes
- Inconvénients
 - Rigide: fait l'hypothèse que les besoins ne changent pas au cours du développement
 - Pas de «feedback» avant la livraison au client
 - Ne reflète pas toujours les pratiques réelles

Modèle en V

23



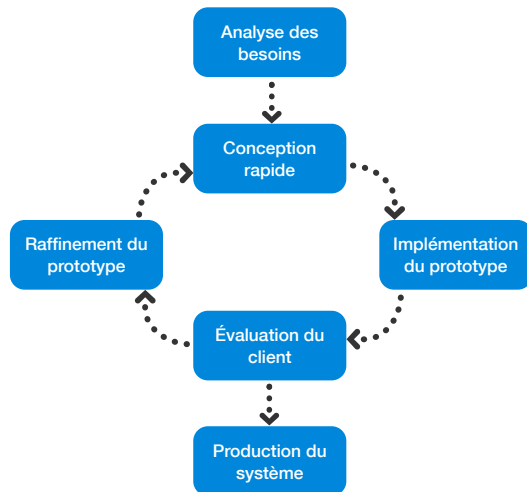
Modèle en V

24

- Axé sur la vérification / validation
- Variation du modèle en cascade
 - Chaque étape de développement a une phase de tests qui lui est associée

Modèle par prototypage

25



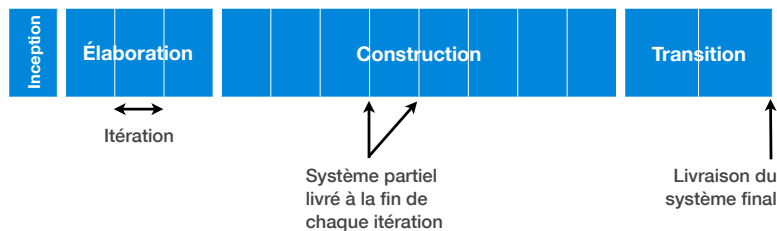
Modèle par prototypage

26

- Le modèle de prototypage est souhaitable pour les projets où les besoins
 - ne sont pas clairement définis
 - sont susceptibles de changer avec le temps
- Types de prototypes
 - jetables: aident à mieux comprendre les besoins du client, évaluer différentes solutions, etc.
 - évolutifs / réutilisables: complétés au cours des itérations subséquentes jusqu'à l'obtention du produit final
 - développement itératif: système complet à chaque itération
 - développement incrémental: système partiel complété progressivement à chaque itération

Processus unifié

27



Processus unifié

28

- Création (Inception)
 - Vision approximative, définition de l'étendue du projet, estimés vagues
 - Le projet est-il réalisable?
 - Création \neq analyse
- Élaboration
 - Vision raffinée
 - Développement itératif de l'architecture de base
 - Résolution des risques les plus élevés
 - Identification de la plupart des besoins
 - Estimés plus réalistes

- Construction
 - Implémentation itérative des éléments plus simples / à plus faible risque
 - Préparation pour le déploiement
- Transition
 - Implantation du système dans un environnement de production

- Process flexible
 - La plupart des artefacts sont optionnels
 - Documents peuvent être produits dans une itération et raffinés dans des itérations subséquentes
- Utilise souvent UML
- Basé sur des itérations de courte durée
- Chaque phase peut comprendre différents types d'activités de développement (analyse, conception, implémentation, tests, etc.)