

INF5151 – Génie logiciel : analyse et modélisation

Modèles de développement

Jacques Berger

Objectifs

Introduire les principaux modèles de développement de logiciels

Prérequis

Cycle de vie du logiciel

Modèles de développement

Plusieurs modèles de développement

Modèle en cascade

Modèle en spirale

Modèle en V

Développement itératif

Modèle en cascade

Chaque étape est faite l'une après l'autre

Lorsqu'une étape est complètement terminée, on passe à la suivante

Modèle en cascade

Étapes

1. Exigences logicielles
2. Conception
3. Implémentation
4. Vérification et validation
5. Maintenance

Modèle en cascade

Processus prévisible

Livrables bien définis à l'avance

Plus facile d'estimer l'effort d'un projet et de le projeter dans le temps

Modèle en cascade

Peu adaptatif

Réagit très mal aux changements de besoins et de fonctionnalités

Modèle en spirale

Modèle basé sur le prototypage et le raffinement successif du prototype

Itération cyclique de 4 étapes, mettant l'accent sur la gestion des risques

Modèle en spirale

Les 4 étapes :

1. Détermination des objectifs
2. Identification et réduction des risques
3. Développement et tests
4. Planification du cycle suivant

Modèle en spirale

Aucune possibilité de retour en arrière

Toutes les étapes doivent être faites dans l'ordre

Développement de plusieurs versions du prototype jusqu'à l'obtention d'un prototype opérationnel

Ensuite, le développement de la version finale est fait

Modèle en spirale

Voir le schéma

Modèle en V

Amélioration du modèle en cascade

Chaque étape produit un livrable nécessaire à l'étape suivante

On descend le V jusqu'à la programmation

Modèle en V

Ensuite, on remonte le V avec les tests

Chaque étape préalable à la programmation est jumelée à une étape de tests

Une étape d'analyse sert d'entrée à une étape de test

Modèle en V

Un défaut trouvé lors d'une étape de test permet de revenir sur une étape précise de l'analyse et de revoir le cycle à partir de cette étape

Modèle en V

Les étapes :

1. Analyse des besoins et faisabilité
2. Spécifications
3. Conception architecturale
4. Conception détaillée
5. Codage
6. Tests unitaires
7. Tests d'intégration
8. Tests de validation
9. Recette

Modèle en V

Voir le schéma

Développement itératif

Le développement est divisé en itérations

Les itérations sont de plus courtes durées

À la fin de chaque itération, nous avons un produit potentiellement livrable

Développement itératif

Itératif : un logiciel exécutable est prêt à la fin de chaque itération

Incrémental : des fonctionnalités sont ajoutées à chaque itération

Évolutif : le logiciel évolue constamment (qualité, performance, etc.)

Développement itératif

Bonne réactivité aux changements de besoins

Une phase d'analyse peut souvent précéder le début du développement

On commence les éléments les plus risqués lors des premières itérations

Développement itératif

On fait un cycle complet de développement à chaque itération

Développement itératif

Plus difficile à estimer et à planifier (calendrier)

Rend la négociation avec le client plus complexe

Développement itératif

Recommandé par les méthodes Agile
eXtreme Programming
Scrum