

UE ED4INFGM

Projet logiciel

Sous-UE ED4INFG1

Initiation au projet

MASSIE Henri, Maître de conférences (UPS-IRIT)

Responsable du parcours ISI du L3 Informatique
(Ingénierie des Systèmes Informatiques)

Responsable du parcours LC du futur M2 informatique-DL
(Développement Logiciel - Logiciels Critiques)

IRIT1 / 413 / 05 61 55 63 01 / Henri.Massie@irit.fr

Objectifs de la sous-UE

- Présenter une **approche de développement modulaire**, permettant :
 - de mener à bien le développement d'un projet en équipe (petite)
 - d'harmoniser les développements des divers membres de l'équipe
- **Mettre en œuvre ce processus sur un projet, en équipe (petite)**
- *Pré requis : Connaissances en programmation impérative*

Cours

- Succinct : 10h
 - Introduction au génie logiciel : compléments pour la réalisation du projet
- Plan
 - Introduction
 - Spécifier le travail à réaliser
 - Concevoir une architecture modulaire
 - Coder les modules
 - Tester
 - Faire le bilan

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

4

Projet tutoré

- But : mettre en pratique les connaissances et savoir-faire de l'étudiant dans la réalisation d'un programme "conséquent", à plusieurs
- 120h de travail personnel, **en binômes issus du même groupe de TDM**
- **Le même sujet pour tous les binômes**
- Encadré par un tuteur : réunions hebdomadaires
- Documents intermédiaires, notés
- Recette finale, avec remise d'un rapport et soutenance orale, notés

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

5

Cours

Introduction

- Qu'est-ce que le génie logiciel ?
- Qu'est-ce qu'un projet logiciel ?
- Qu'est-ce qu'un processus de développement ?
- Qu'est-ce que spécifier ?
- Qu'est-ce que concevoir ?
- Qu'est-ce que coder ?
- Qu'est-ce que tester ?

Génie Logiciel (software engineering)

- Discipline née en Europe, à Garmisch-Partenkirchen, en 1968, en réponse à :
 - Crise du logiciel, énorme constat d'échec :
 - Le logiciel n'était **pas fiable**
 - Il était **difficile de réaliser, dans les délais (et coûts) prévus, des logiciels satisfaisant leurs cahiers des charges**
 - ⇒ besoin de fonder le développement sur :
 - des bases théoriques
 - des méthodes
 - des modèles
 - des outils
- En constante évolution depuis

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

8

Génie logiciel

- *Définition* : c'est l'art de **spécifier**, de **concevoir**, de **réaliser**, et de **faire évoluer**, avec des **moyens** et dans des **délais raisonnables**, des **programmes**, des **documentations** et des **procédures qualité**, en vue d'utiliser un ordinateur pour résoudre certains problèmes [Gaudel 96]

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

9

Constat d'échec, insatisfactions

- dépassement des délais de réalisation prévus
- dépassement des coûts
- logiciels ne correspondent pas aux besoins des utilisateurs (réels ou exprimés)
- manque de fiabilité, erreurs résiduelles persistantes
- sensibilité aux erreurs humaines, aux pannes matérielles
- difficultés de mise en œuvre, de transition ancien/nouveau
- difficultés d'évolution : maintenance complexe et coûteuse
- faible portabilité

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

10

Erreurs célèbres

- arrêt du réseau TRANSPAC pour 7000 entreprises et 1 000 000 d'abonnés (~1987)
- trains fantômes à la SNCF (~1993)
- erreur de 10 km lors d'une visée laser sur satellite (1995)
- destruction d'Ariane V (vol 501, 1996)
- crash de sondes sur Mars (~2000, etc.)

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

11

Projets douloureux

- Certains projets n'aboutissent jamais !
 - Advanced Logistics System
abandonné après 217 millions \$ de dépenses !
- D'autres aboutissent, mais à quel prix !
 - 93-94 - SNCF, Socrate
1 an de retard
prévision initiale du projet x 2 :
coût total du projet = ~320 millions d'€
 - 95-97 - NASA / armée américaine, visée laser sur satellite
2 ans de retard (fiabilisation)
 - 96-97 - Ariane V
16 mois de retard
coût de correction / fiabilisation :
prévu : ~230 millions d'€
réel : ~320 millions d'€

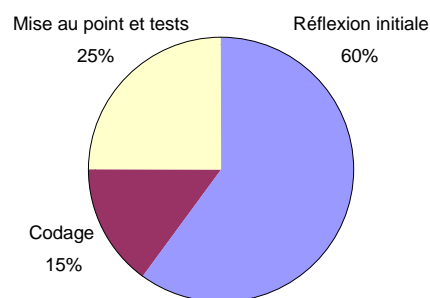
2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

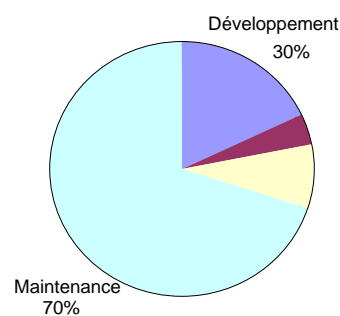
12

Coût de la non qualité

Ce qu'on voit d'abord



Ce qu'on aurait dû voir



Source : Lientz et Swenson, 1978

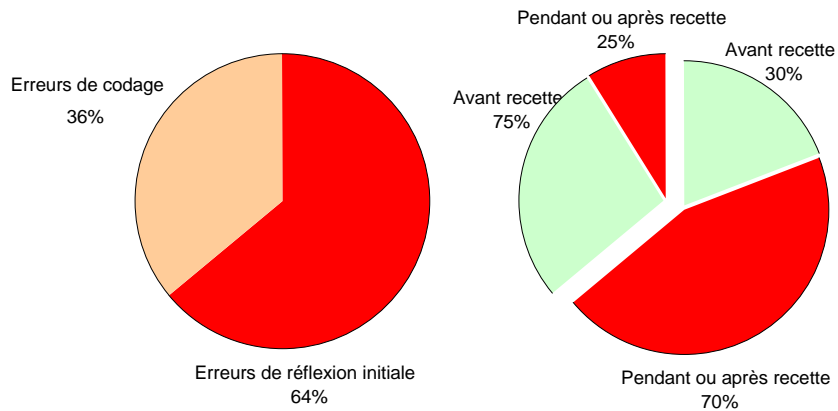
2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

13

Coût de la non qualité

% d'erreurs, par phase



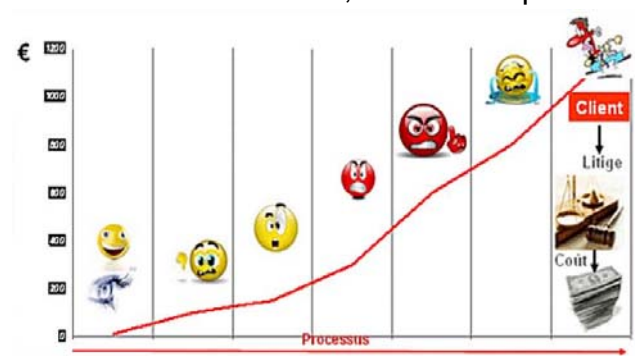
2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

14

Coût de la non qualité

Coût des erreurs, suivant les phases



Evolution du coût de la non-qualité suivant l'endroit de sa découverte
Source : Le blog de la qualité

Phases, temps

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

15

Catastrophisme ou réalité ?

- faut-il vraiment viser la qualité totale ?
Non, mais il faut avoir une estimation de la qualité obtenue
- ces problèmes sont-ils spécifiques au logiciel ?
Non, c'est le cas pour tout produit manufacturé complexe, mais les dépassements de coûts et délais sont très importants en logiciel

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

16

Spécificités du logiciel

- Invisible, sauf tardivement
- Pas d'usure
- Fabrication à l'unité (pas de série, pas de rebut)
- Complexe
 - complexité des problèmes
 - complexité des solutions
- Théorie de la fiabilité rarement appliquée
- Qualité dépendante de l'environnement matériel et logiciel
- "Facile" à modifier, difficile et coûteux à valider et vérifier

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

17

40 ans de Génie Logiciel

- bilan positif !
- logiciels toujours plus
 - critiques
 - complexes
 - grande tailleet avec de plus en plus d'exigences de
 - changement
 - fiabilité, disponibilité
- fabrication de tels produits pas encore complètement maîtrisée

Processus de développement

- procédé de production
- raffinements successifs, « cycle de vie »
- itératif
- importance de la documentation
- fin développement => mise en exploitation => maintenance => redéveloppement, d'où « cycle »

Processus de développement

- Choix en fonction :
 - des caractéristiques du produit
 - du contexte de développement
- Prendre en compte les aspects :
 - techniques
 - organisationnels
 - humains

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

22

Processus de développement

- Activités à mener :
 - Analyse des besoins
 - Spécification
 - Conception préliminaire
 - Conception détaillée
 - Programmation
 - Intégration
 - Test
 - Gestion de versions et de configurations
 - etc.

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

23

Processus de développement

- besoin de modèles de développement généraux :
 - enchaînements,
 - interactions entre activités
- instanciation sur un projet donné :
 - méthodes
 - outils supports
 - autres modalités associées à chaque activité

Méthodes

- Notation, modèles
 - ensemble de symboles, représentant
 - les concepts manipulés
 - les moyens de les combiner
- Démarche de résolution de problème
 - en utilisant cette notation
- Outils informatiques
 - aide à la mise en œuvre

Processus de développement

- Buts des modèles de développement
 - Obtenir des processus de développement
 - rationnels
 - reproductibles
 - contrôlables
 - Peut ne pas suffire
 - nature du logiciel : contrôles délicats
 - application du processus : compétence et implication des développeurs, volonté de l'encadrement

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

26

Processus de développement

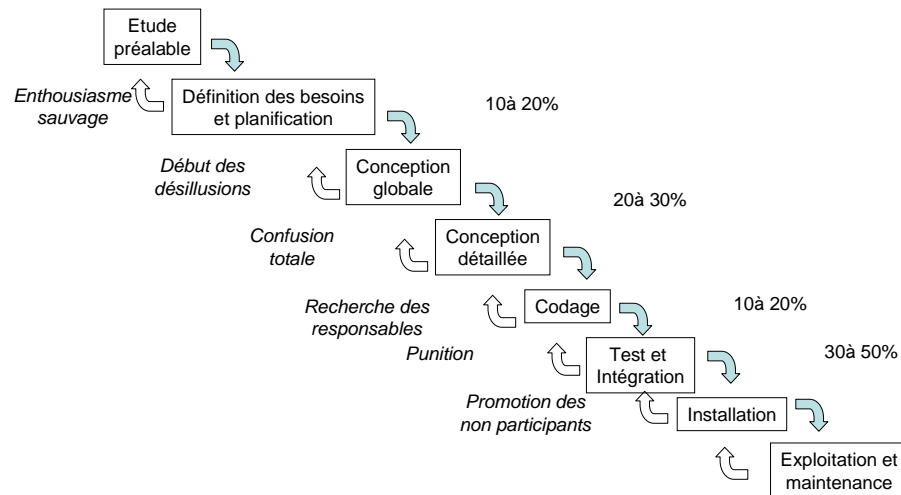
- Principaux modèles de développement :
 - Modèle de la cascade (linéaire),
 - Modèle en V (linéaire),
 - Modèle incrémental (itératif)
 - Modèle en spirale (itératif, prototypes),

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

27

Modèle de la cascade



2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

31

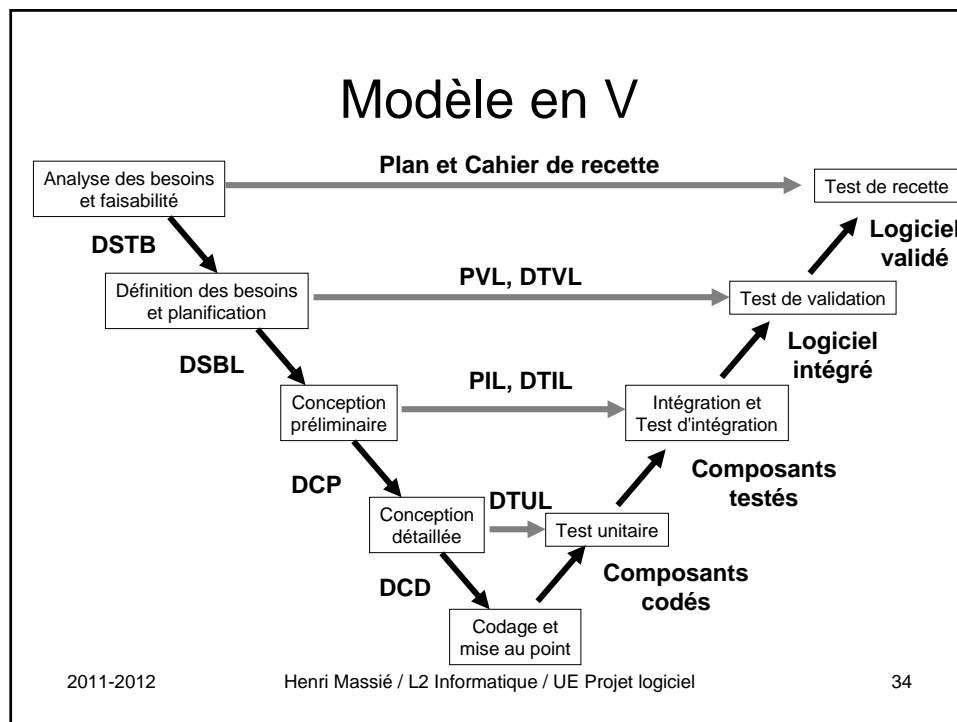
Modèle en V

- années 80
- met en évidence les dépendances entre étapes (séquentielles ou pas)
- met l'accent sur les activités de validation et vérification
- le plus utilisé, mais attention au contexte de développement

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

33



Mais ces modèles supposent que

- Les besoins sont connus avant la conception
- Les besoins changent rarement
- Les utilisateurs savent ce qu'ils veulent et n'ont pas besoin de le visualiser
- La conception peut être menée dans un environnement totalement abstrait
- Le moment venu, la technologie suit
- Le système n'est pas trop complexe

Limitations

- Livraison en tout ou rien
- La preuve d'un concept est repoussée en fin de cycle
- Jusqu'à l'intégration finale, seule de la documentation a été produite
- Le déploiement tardif cache de nombreux risques :
 - technologique, conceptuel, personnel
 - l'utilisateur ne voit rien avant la fin du cycle
 - le test du système ne peut commencer que très tard

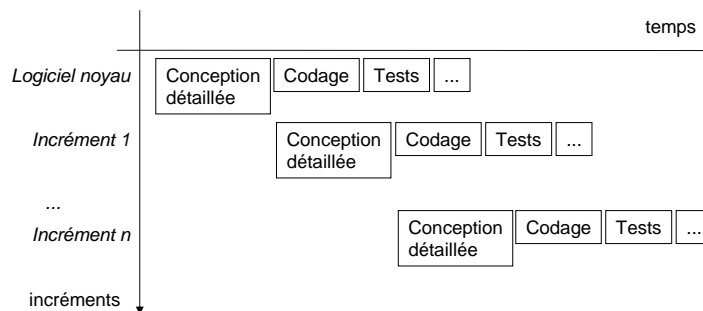
2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

37

Modèle incrémental

- Après conception globale :
 - développement d'un logiciel noyau
 - développement et intégration des incréments successifs



2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

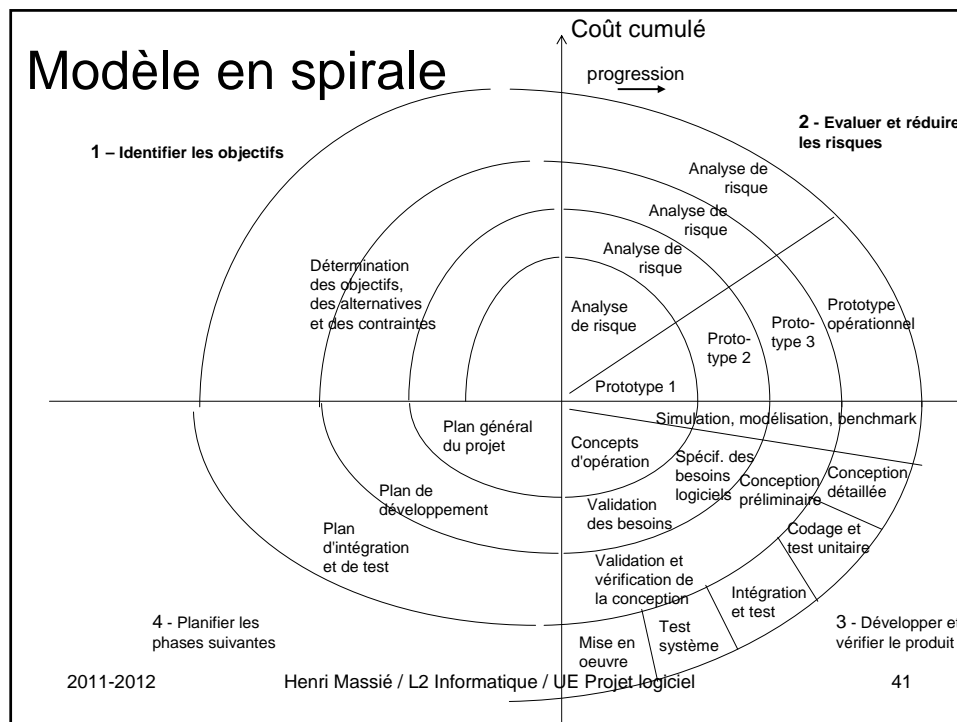
38

Modèle incrémental

- Avantages
 - développements moins complexes
 - intégrations progressives
 - livraisons échelonnées
 - lissage effort et effectifs dans le temps
- Inconvénients
 - risques
 - remise en cause incréments précédents
 - impossibilité d'intégrer un incrément

Modèle en spirale

- années 90
- plus général, plus complet, mais plus complexe
- introduit :
 - analyse de risques
 - maquettes



Risques et solutions : exemples

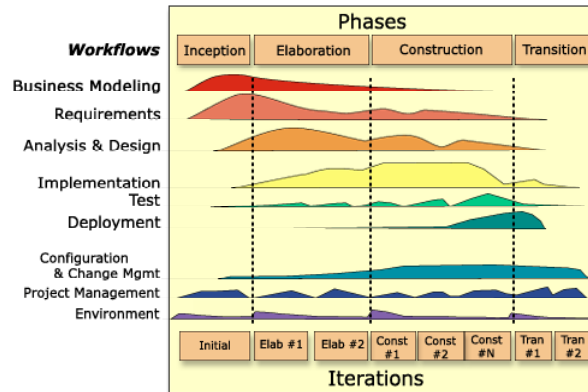
Volatilité des besoins	Seuil élevé de modification Masquage d'information Développement incrémental (les plus changeants en dernier)
Composantes externes manquantes	Inspections Essais et mesures Analyse de compatibilité
Tâches externes défallantes	Audit avant attribution de sous-traitance Contrats avec bonus Revue
Problèmes de performances	Simulations Modélisations Essais et mesures Maquettage
Exigences démesurées par rapport à la technologie	Analyses techniques de faisabilité Maquettage

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

43

RUP, OpenUP : Cycle de vie itératif et incrémental



Source : Rational Unified Process (RUP®)

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

47

Rational Unified Process, OpenUP

- Méthode de gestion de développement de logiciel OO
- Cadre de travail extensible
- Caractéristiques :
 - Développement itératif et incrémental
 - Gestion des besoins
 - Vision architecturale basée composants
 - Modélisation visuelle des systèmes
 - Gestion de la qualité
 - Gestion du changement

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

48

Caractéristiques du RUP et d'OpenUP

- Processus décrit par un ensemble de pages web
- Modèles, pour la plupart des produits
- Manuels et guides, pour la plupart des activités

Processus itératif

- Reconnaît que le changement des besoins est une réalité
- Encourage la réduction précoce des risques
- Permet le « planifier un peu, concevoir un peu, coder un peu, tester un peu »
- Encourage tous les participants à intervenir très tôt
- Permet d'améliorer le processus à chaque itération
- Axé sur des architectures de composants

Buts et caractéristiques de chaque itération

- Dirigée par les risques
- Elimination progressive des risques : performance, intégration, conception
- Itération = mini projet, développé suivant un cycle classique et menant à la fourniture d'un code exécutable, pour validation
- Résultat = incrément, dans une approche évolutive

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

51

Processus incrémental

- Permet l'évolution
- Permet l'amélioration
- Oblige à s'assurer de la stabilité
- Permet d'obtenir plus rapidement un système qui tourne
- Permet la gestion des risques

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

52