

Chapitre 2 - Modèles de cycle de Vie et Test



Responsable du cours :
Héla Hachicha

Année Universitaire : 2016 - 2017

2

Sommaire

- Modèles de processus de développement du logiciel
- Les activités de ces processus

3

Les facteurs clés en génie logiciel

- Trois facteurs clés en génie logiciel
 - **Personnel** : nombre, habiletés, moral
 - **Processus**: procédures d'accomplissement de travail
 - **Technologie**: plateforme et domaine
- Les bons **processus** aident le **personnel** à appliquer la **technologie**
 - De façon **efficace** : sans perte de temps, ni effort, ni ressources
 - De façon **efficace** : tout en obtenant le résultat désiré

4

Qu'est-ce qu'un processus?

- Un **processus** est une suite d'étapes impliquant des **activités**, des **acteurs**, des **ressources**, et des **contraintes** pour produire un résultat escompté

5

Le processus de développement de logiciel

- Un ensemble structuré d'activités nécessaires pour développer un logiciel
- Un modèle de développement de logiciel est une représentation abstraite d'un processus
- De nombreux modèles différents mais pour tous :
 - **Spécification** : on définit ce que le système devra faire
 - **Conception** et **implémentation** : on définit l'organisation du système et on l'implémente
 - **Validation** : on vérifie que le système fait bien ce que veut le client
 - **Evolution** : on modifie le système en réponse aux changements des besoins du client

6

Modèles de processus de développement logiciel

- Un modèle de processus de développement logiciels met en relief :
 - les **activités** de travail à accomplir pour produire le produit logiciel,
 - l'**ordre** dans lequel les **activités** de travail et les **tâches** doivent être effectuées,
 - les **façons** dont les activités de travail et les tâches peuvent être **superposées** et **réitérées**, et
 - les **produits** de travail résultants, et les flux entre diverses activités de travail.

7

Les modèles de processus

- Chaque élément d'un modèle de processus a
 - **Inputs** nécessaires
 - **Procédures** pour l'accomplissement du processus
 - **Produits** de travail à produire
 - **Critères** d'acceptation pour les produits des travaux de sortie

8

Quelques terminologies

- Un **processus** est une description de la **façon** d'accomplir une activité de travail
 - Une **procédure** est un ensemble d'étapes pour l'accomplissement des tâches d'un processus
 - Une **technique** est la manière dont un individu accomplit une procédure
- Un **processus** inclut les **procédures** pour conduire les activités de travail

9

Un Exemple

-- Procédures de réparation de défaut-

Correction d'un défaut signalés par les clients implique les **procédures** suivantes :

- 1. reproduire la défaillance
- 2. trouver le défaut
- 3. corriger l'erreur
- 4. modifier la suite de tests
- 5. accomplir le test de régression
- 6. documenter le correctif
- 7. mettre à jour d'autres produits de travail, si nécessaires
- 8. vérifier le code modifié et les documents
- 9. distribuer le code modifié
- 10. clôturer le rapport de problème

10

Problèmes techniques dans les projets logiciels

- Le développement de produits comprend :
 - L'ingénierie du système
 - L'ingénierie des exigences logicielles
 - Le design du logiciel
 - L'implantation du logiciel
 - La **vérification** et la **validation** du logiciel
 - L'intégration et la validation du système

11

Quelques terminologie

- La **vérification** du cycle de vie est le processus de détermination qu'un produit de travail **satisfait aux conditions imposées** par d'autres produits de travail et processus de travail
 - i.e., est-ce que le produit de travail est complété, correct et cohérent avec les autres produits de travail et processus de travail ?
- ☛ **Sommes-nous en train de faire le bon produit ?**
- La **validation** du cycle de vie est le processus de détermination qu'un produit de travail **satisfait aux besoins prévus de son utilisation** lorsqu'il est utilisé par ses utilisateurs dans l'environnement prévu
 - i.e., est-ce que le produit de travail est approprié pour son utilisation ?
- ☛ **Est-ce que nous faisons le produit correctement ?**

En pratique

- souvent confondus, ou pris l'un pour l'autre
- on parle de « V&V » (validation et vérification)

12

Techniques de vérification

- Les techniques de vérification incluent :
 - La traçabilité,
 - les révisions,
 - le prototypage,
 - l'analyse, et
 - les tests fonctionnels.

verification can, and should, be applied to all significant work products of a software project

13

Techniques de validation

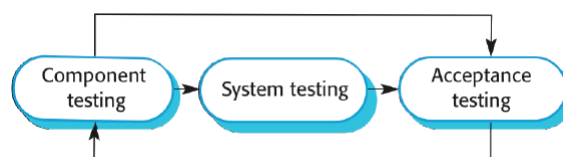
- Les techniques de validation incluent:
 - Les révisions,
 - le test du système,
 - les tests opérationnels, et
 - les démonstrations.

validation can, and should, be applied to all significant work products of a software project

14

Les phases de test

- Tests unitaires
 - Les composants sont testés individuellement
- Tests d'intégration
 - Test du système global
- Tests de recette (test de validation)
 - Test avec des données clients pour vérifier que le système répond aux exigences du client



15

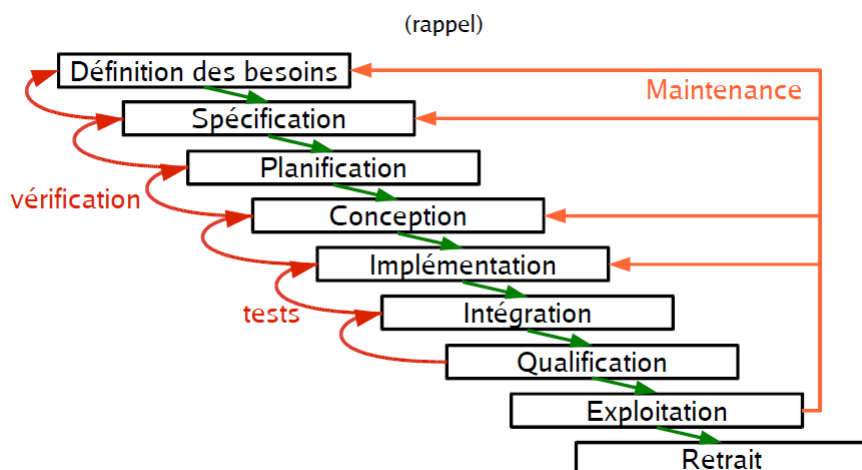
Cycle de vie et Tests

Différents modèles de développement logiciel :

- Le modèle en cascade
- Le modèle en V
- Le développement incrémental (prototypage)
- Le modèle évolutif
- Le modèle en spirale
- La méthode Agile : Extreme Programming (XP)
- Le modèle orienté réutilisation
- ...

16

Cycle de vie - Modèle en cascade



17

Caractéristiques du modèle en cascade

(date des années 70)
(mais reste pertinent)

- Séquentiel
- Importance du contrôle du processus
 - rétro-actions
 - validation, vérification, tests
- Vérification : Le système est conforme à la spécification
- Validation : Le système répond aux exigences du client
- ➔ Inspections et tests
- Tests
 - On exécute le système avec des cas de tests issus de la spécification de données réelles du système futur

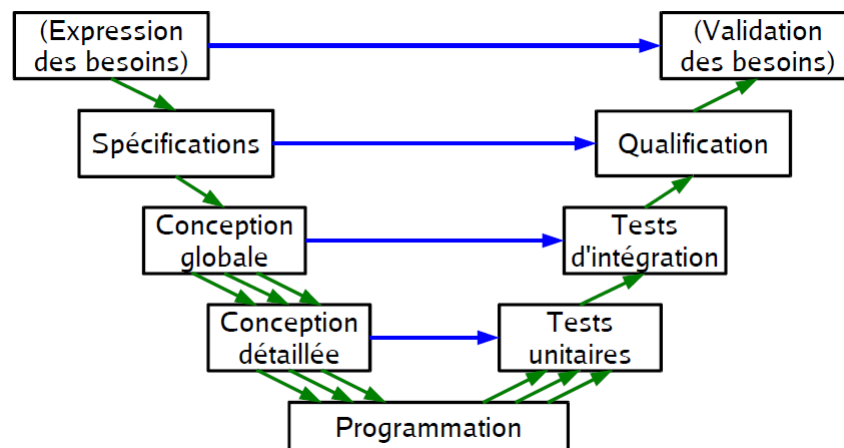
18

Critique du modèle en cascade que peut-on lui reprocher ?

- Modèle trop séquentiel
 - dure trop longtemps
- **Validation trop tardive** : les tests sont prévus tardivement et remise en question coûteuse des phases précédentes
- Sensibilité à l'arrivée de nouvelles exigences
 - refaire toutes les étapes

19

Cycle de vie - modèle en V



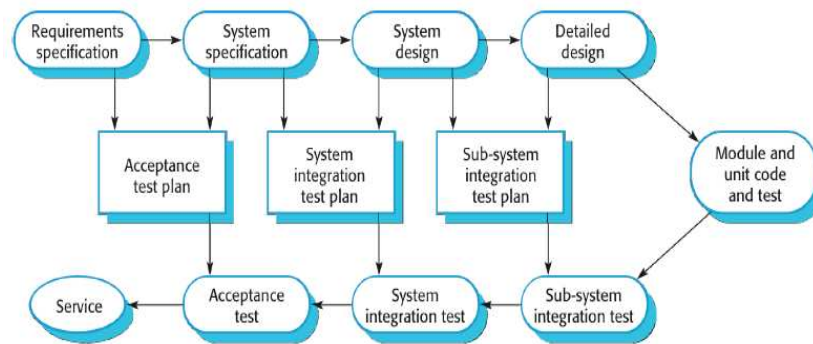
20

Caractéristiques du modèle en V

- Tâches effectuées en parallèle
 - horizontalement : préparation de la vérification
 - Ex. : dès que la spécification fonctionnelle est faite : (↑)
 - plan de tests de qualification
 - plan d'évaluation des performances
 - documentation utilisateur
 - verticalement : développement des modules
 - Ex. : dès que la conception globale est validée : (↑)
 - conception détaillée des modules
 - programmation et tests unitaires

21

Cycle de vie - modèle en V



22

Certaines réalités sur le développement logiciel

- 1. Les exigences changent toujours en raison de:
 - Changement des attentes des clients et des besoins des utilisateurs
 - Analyse initiale inadéquate des exigences
 - Compréhension et aperçu deviennent plus clairs par l'expérience
 - Évolution de la technologie
 - Évolution de la situation compétitive
 - Rotation du personnel : ingénierie, gestion, marketing, clientèle
- 2. Le design n'est jamais correct dès le premier coup
 - Le design est un processus créatif, de résolution de problèmes
- 3. Les démonstrations fréquentes de la progression et d'alerte précoce des problèmes sont souhaitables

Iterative development models are best

23

Développement itératif

- L'**itération** est le processus par lequel le résultat souhaité est développé par des cycles répétés
- En génie logiciel, une approche itérative permet la révision et l'ajout, étape par étape, des produits de travail
- Différents types de modèles itératifs supportent :
 - La **révision** et l'ajout des **exigences**
 - La **révision** et l'ajout de **design**
 - La **révision** et l'ajout du **code**
 - Le **test** d'une partie du système
 - et ainsi de suite

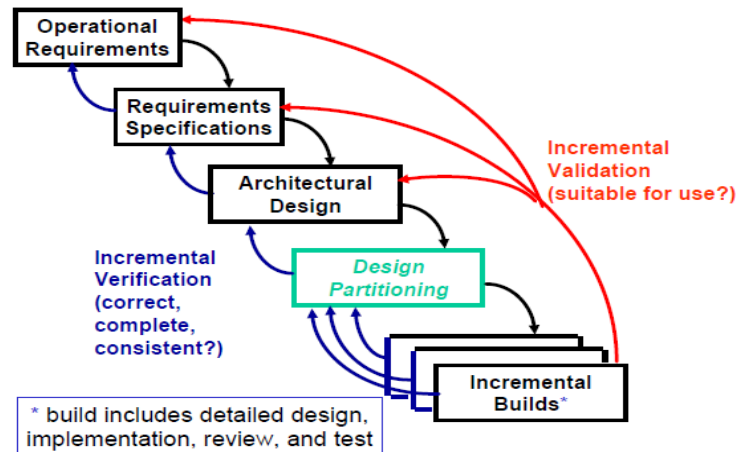
24

Développement itératif

- Les objectifs de développement **itératif** sont les suivants :
 - fréquentes démonstrations de la progression
 - alerte précoce des problèmes
 - capacité d'intégrer les changements de façon élégante
- Quatre types de modèles de développement itératif :
 - 1. **Construction-incrémentale** : code-test-demo itératif
 - 2. **agile** : satisfaire aux exigences opérationnelles itérativement
 - 3. **évolutif** : le développement exploratoire
 - 4. **spirale** : la gestion des risques

25

Le processus de construction-incrémentale



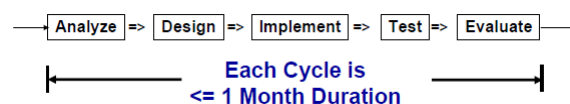
26

Le modèle évolutif

- Utilisé dans la cas où il est (presque) impossible de spécifier **à l'avance** une première version stable des exigences



- Détails de chaque cycle:



27

Directives pour le développement évolutif

- Utilisé lorsque les exigences ne peuvent pas être spécifiées à l'avance la plupart du temps
- Cycles évolutifs se termine lorsque
 - Le projet est converti en une approche incrémentale
 - ou, le projet est annulé parce qu'il est infaisable
 - ou, le produit est livré
- Utiliser une approche évolutive indique que le projet a **un risque très élevé**

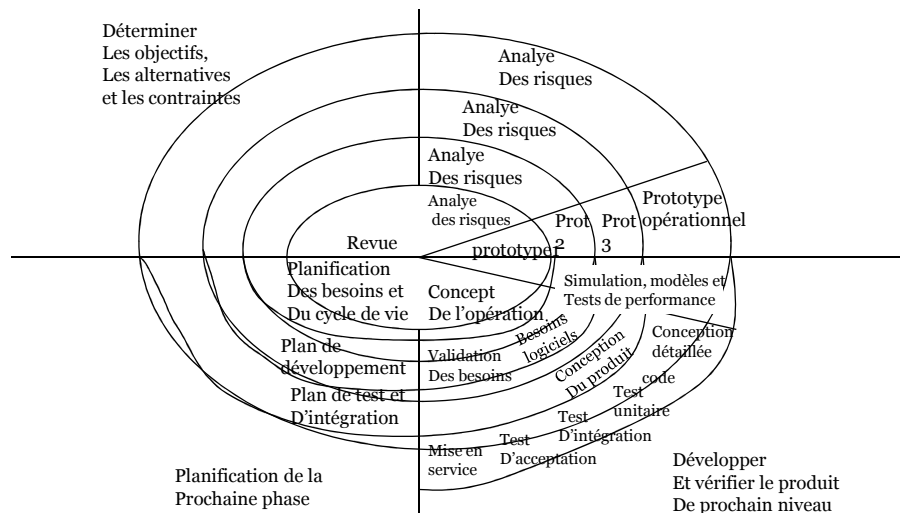
28

L'approche spirale

- Le processus de développement en spirale est un modèle de méta-niveau pour les modèles de développement itératif
 - Des activités antérieures sont revisitées, révisées, et raffinées à chaque passage de la spirale
- Chaque cycle d'un modèle en spirale comporte quatre étapes:
 - Étape 1 - déterminer les objectifs, les alternatives, et des contraintes
 - Étape 2 - **identifier les risques pour chaque alternative** et choisir l'une des alternatives
 - Étape 3 - mettre en œuvre la solution (l'alternative) choisie
 - Étape 4 - évaluer les résultats et le plan pour le prochain cycle de la spirale
- Les cycles continuent jusqu'à ce que les objectifs souhaités soient atteints (ou jusqu'à ce que le temps et les ressources sont utilisées)

29

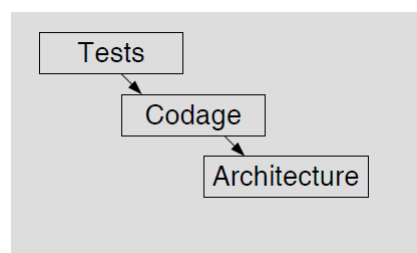
Modèle en spirale



30

Développement Agile

- Création des tests selon les spécifications
- Codage en conformité aux tests
- Emergence de l'architecture
 - À partir du code



31

Leçons apprises

- Un cadre du processus de développement est un modèle de processus générique qui peut être ajusté et adapté pour répondre aux besoins des différents projets.
- Le processus de développement pour chaque projet logiciel doivent être conçus avec le même soin utilisé pour la conception du produit.
- Le design du processus se fait mieux en ajustant et en adaptant des modèles de processus de développement et des cadres de processus bien-connus, tout comme le design des produits qui se fait mieux en ajustant et en adaptant des styles architecturaux et des cadres architecturaux bien connus.

32

Leçons apprises

- Il ya plusieurs modèles de processus de développement de logiciels bien connus et largement utilisés, incluant le modèle en cascade, incrémental, évolutif, agile, et le modèle spiral.
- Il ya différentes façons d'obtenir les composants logiciels nécessaires; différentes façons pour obtenir les composants logiciels nécessitent un mécanisme différent de la planification, de mesure et de contrôle.
- Les phases de développement d'un projet logiciel peuvent être entrelacées et répétées de diverses manières.

33

Leçons apprises

- Les processus de développement itératif offrent les avantages suivants:
 - L'intégration continue,
 - La vérification et la validation itérative d'un produit évolutif,
 - Des démonstrations fréquentes de la progression,
 - La détection précoce des défauts,
 - L'alerte précoce des problèmes de processus,
 - L'incorporation systématique d'un travail inévitable qui se produit dans le développement du logiciel, et
 - La livraison anticipée des sous-ensembles de capacités (si désiré).
- Selon le processus de développement itératif utilisé, la durée de l'itération s'étend de 1 jour à 1 mois.
- Le prototypage est une technique pour acquérir des connaissances, ce n'est pas un processus de développement.

34

Les coûts relatifs à la correction d'un défaut logiciel

