

Méthodes

Licence professionnelle IDSE
2012-2013
<http://anubis.polytech.unice.fr/iut/>

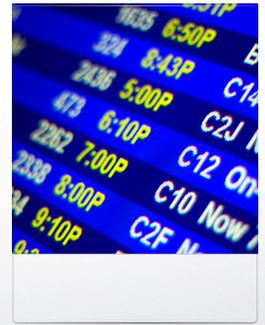
Mireille Blay-Fornarino
blay@unice.fr

1



Plan

- Cycle de vie
- Méthodes
- Agilité
- Scrum



2

Cycle de vie du logiciel

- 1) **Activités**
- 2) Différents types de cycles

3

Activités du cycle de vie du logiciel

- 1) **Définition des besoins (Requirements)**
 - Expression des besoins dans le langage du client
- 2) **Spécifications**
 - Traduction des besoins dans un langage plus informatique
 - Description du système d'un point de vue extérieur
 - Ce qu'il doit faire
 - Pas comment il le fait
 - Spécifications fonctionnelles et non-fonctionnelles
 - Doit rester accessible au client (contrat)

Jean-Paul Rigault 2005

4

Activités du cycle de vie du logiciel

- 2) **Spécifications (suite)**
 - **Spécifications fonctionnelles**
 - Les fonctions/services rendus par le système
 - **Spécifications non-fonctionnelles**
 - Utilisabilité
 - Fiabilité (reliability)
 - Performance
 - Supportabilité (maintenabilité)
 - Conditions d'implémentation, de déploiement...
 - Interface
 - Conditions d'exploitation
 - Conditionnement
 - Aspects juridiques
 - Aspects financiers...

5

Activités du cycle de vie du logiciel

- 3) **Conception**
 - Traduction des spécifications en termes d'artefacts logiciels
 - Peut être plus ou moins détaillée
- 4) **Codage**
 - Traduction de la conception en code
- 5) **Test unitaires**
 - Test de chaque module individuellement
 - Généralement de la responsabilité du développeur du module
- 6) **Tests d'intégration**
 - Test de la composition de plusieurs modules (sous-systèmes)

6

Activités du cycle de vie du logiciel

7) Validation

- Test du système final par rapport aux spécifications

8) Recette/Mise en exploitation

- Acceptation du système final
- Peut être l'objet d'une procédure formelle et parfois officielle

9) Gestion des changements, gestion de configuration

7

Cycle de vie du logiciel

- 1) Activités
- 2) Différents types de cycles

8

Modèles de cycle de vie d'un logiciel

● Modèles de cycle de vie

- organiser les différentes activités du cycle de vie pour l'obtention d'un logiciel fiable, adaptable et efficace
- guider le développeur dans ses activités techniques
- fournir des moyens pour gérer le développement et la maintenance
 - ressources, délais, avancement, etc.

• Modèles linéaires

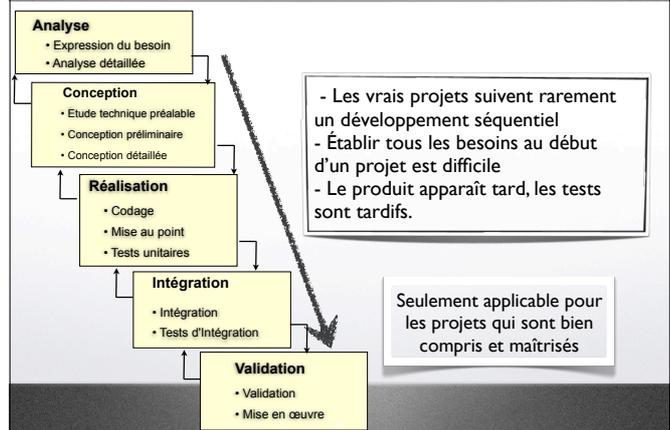
- en cascade et variantes

• Modèles non linéaires

- en spirale, incrémentaux, itératifs

9

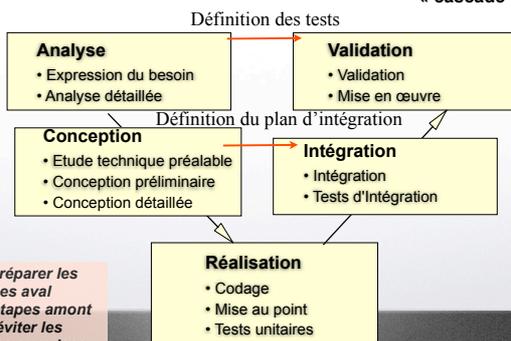
Modèle en cascade (70)



Modèle en «V»

- Cycle de vie normalisé AFNOR

Variante du cycle en « cascade »



Vise à préparer les étapes aval lors des étapes amont pour éviter les mauvaises surprises

11

Modèle en «V»

Le cycle en V

- permet une meilleure anticipation
- présente des tests bien structurés

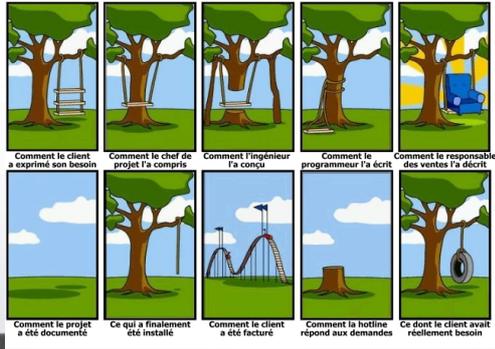
Mais

- le cadre de développement est rigide
- la durée est souvent trop longue
- le produit apparaît très tard (validation finale tardive très coûteuse s'il y a des erreurs)

Variante : W (validation d'une maquette avant conception)

12

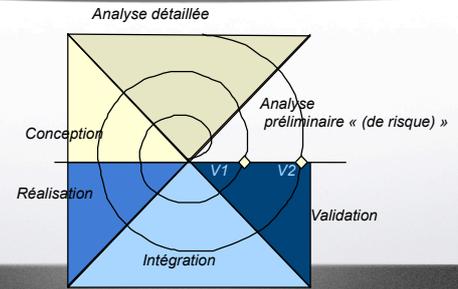
Problèmes avec le processus classique...



13

Modèle en « spirale » (Boehm 88)

Couplage de la nature itérative du prototypage avec les aspects systématiques et contrôlés du modèle en cascade.



14

Modèle en « spirale »

- ➔ Bien adapté aux développements innovants
 - les progrès sont tangibles : c'est du logiciel qui « tourne » et pas seulement des kilos de documents
 - réduit les risques si bien appliqué : possibilité de s'arrêter « à temps », i.e. avant que l'irréalisabilité du projet ait créé un gouffre financier
- ➔ Moins simple à manager
 - difficile à gérer en situation contractuelle
 - mal contrôlé => on retombe dans le *hacking*

À la base de tous les processus « agiles »

15

Stratégies d'itération

Large et peu profonde

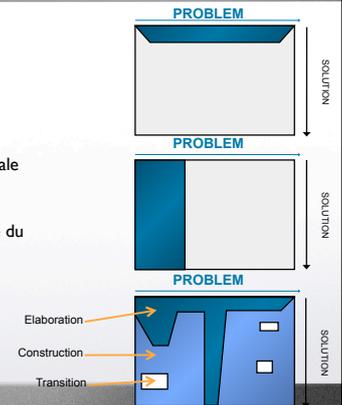
- ▶ Analyse de l'ensemble du domaine
 - Tous les cas d'utilisation sont étoffés
- ▶ Définition d'une architecture générale

Etroite et profonde

- ▶ Analyse en profondeur d'une partie du problème
- ▶ Développement de cette partie

Hybride

- ▶ Un mélange des stratégies



16

Méthodes

17

Méthode/Processus

- **Méthode :**
 - guide +/- formalisé, démarche reproductible pour obtenir des solutions fiables
 - capitalise l'expérience de projets antérieurs
- Une méthode définit
 - des concepts de modélisation / les **artefacts** à produire
 - activités et chronologie des activités
 - un ensemble de règles et de conseils pour tous les participants

notation + démarche (+ outils)
façon de modéliser et façon de travailler

18

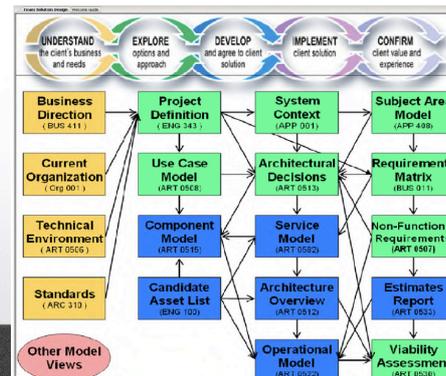
Notion d'artefact

- Un artefact est un élément produit ou modifié dans le cadre d'un processus de développement
- Il peut s'agir d'un document, un diagramme, un compte rendu de réunion...
- ...mais aussi un code source, un écran, une base de données...

19

Évolution des méthodes

IBM methodology : IBM Team Solution Design.



Extrait rapport de stage, M2 IFI AL, 2011, Mohamed Zouhaier Ramadhane, IBM La Gaude

Évolution des méthodes

- **Origine : fin des années 60**
 - problèmes de qualité et de productivité dans les grandes entreprises, mauvaise communication utilisateurs / informaticiens
 - méthodes = guides pour l'analyse et aide à la représentation du futur SI
 - conception par découpage en sous-problèmes, analytico-fonctionnelle
 - méthodes d'analyse structurée
- **Ensuite**
 - conception par modélisation : « construire le SI, c'est construire sa base de données »
 - méthodes globales qui séparent données et traitements

21

Évolution des méthodes

- **Maintenant**
 - conception pour et par réutilisation
 - Frameworks, Design Patterns, bibliothèques de classes
 - méthodes
 - exploitant un capital d'expériences
 - unifiées par une notation commune (par ex. UML)
 - procédant de manière incrémentale
 - validant par simulation effective

⇒ Méthodes agiles

22

Introduction aux méthodes agiles

Focus sur XP

Les grands principes uniquement

23