TCH099 – CHAPITRE 1 MÉTHODOLOGIES DE DÉVELOPPEMENT

Anis Boubaker, Ph.D.
Maître d'enseignement
École de Technologie Supérieure

1

PLAN DU CHAPITRE

1. Génie logiciel : Analyse et modélisation
2. Méthodologies de développement
   1. Cascade
   2. Spirale
   3. Prototypage
   4. Processus unifié (UP)
   5. Méthodologies agiles

2

INTRODUCTION AU GÉNIE LOGICIEL

- Programmer ≠ Analyser et concevoir un système informatique
- La technique ? nécessaire, mais pas si importante que ça !
- Le VRAI problème difficile : l'organisation, la gestion
  - difficulté de formalisation
  - multitude de paramètres, facteurs
  - gestions des humains

3

GÉNIE LOGICIEL

- Ensemble de moyens (techniques, méthodes) mis en œuvre pour la construction de systèmes informatiques
- IEEE: approche systématique, disciplinée et quantifiable pour le développement, la maintenance et l'organisation des logiciels

ÉTS

4

ANALYSE

- Ensemble d'activités qui permettent de définir le *quoi et qui* dans le logiciel
- Objectif: Décrire et spécifier les besoins et les exigences des utilisateurs
- Livrable:
  - Cahier des charges

ÉTS

5

PLAN DU CHAPITRE

1. Génie logiciel : Analyse et modélisation
2. **Méthodologies de développement**
   1. Cascade
   2. Spirale
   3. Prototypage
   4. Processus unifié (UP)
   5. Méthodologies agiles

ÉTS

6

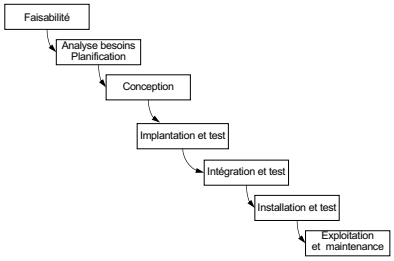
MÉTHODES DE DÉVELOPPEMENT LOGICIEL

- Plusieurs méthodes de développement:
  - Cascade
  - Spirale
  - Prototypage
  - Unified Process (UP)
  - Méthodes agiles

ÉTS

7

LE MODÈLE EN CASCADE



ÉTS

8

LE MODÈLE EN CASCADE

- Approche disciplinée mais très rigide
- Toutes les spécifications doivent être complétées avant de commencer la conception et l'implémentation
- Approprié pour le développement de systèmes caractérisé par un degré important de complexité et d'interdépendance
- Applicable avec succès pour des petits projets mais échec sur projets de grande envergure

ÉTS

9

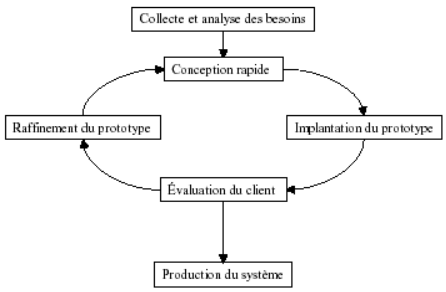
LE MODÈLE PAR PROTOTYPAGE

- Le modèle en cascade suppose que les besoins sont clairs, arrêtés et bien définis
- Le modèle par prototypage est intéressant
  - Besoins pas clairement définis
  - Besoins changeant au cours du temps
- Le prototypage permet le développement rapide d'une ébauche du futur logiciel
  - Prototype jetable / prototype évolutif
  - Feedback avec le client pour assurer la satisfaction des besoins

ÉTS

10

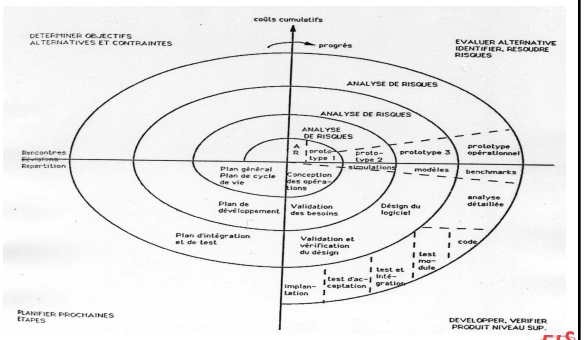
LE MODÈLE PAR PROTOTYPAGE



ÉTS

11

LE MODÈLE EN SPIRALE



Boehm, 1988

ÉTS

12

LE MODÈLE EN SPIRALE

- Apport majeur : considération du risque
- Permet de réduire les impacts négatifs du modèle en cascade en ajoutant des étapes de prototypage
- Développement par itérations assez longues (6 mois à 2 ans)
- Évoluer des prototypes vers le produit final avec la participation du client

ÉTS

13

UNIFIED PROCESS (UP)

- Processus de développement logiciel mature et ouvert initié par les auteurs d'UML (Jacobson, Booch, Rumbaugh 1999)
- Approche itérative et incrémentale
- Rational Unified Process (RUP) dont IBM fait la promotion
- Traité dans le cours. Cf. Chapitre 2 : Processus unifié

ÉTS

14

MÉTHODES AGILES

- C'est en février 2001 que d'éminents membres de la communauté du développement logiciel, mécontents par la complexité des méthodes de développement, ont écrit le manifeste du développement agile
- Les objectifs
  - Mettre en œuvre des individualités et des interactions, plutôt que des procédés
  - Produire un logiciel entièrement testé et qui fonctionne, plutôt qu'une documentation claire

ÉTS

15

MÉTHODES AGILES :

- Courtes itérations (quelques jours à 1-2 semaines) incluant toutes les étapes du développement logiciel
- Favorise le travail en équipe en mettant l'accent sur la collaboration entre les membres de l'équipe plutôt que sur la documentation
- Favorise la participation du client au développement du logiciel
- Sorte d'UP accéléré

ÉTS

16

MÉTHODES AGILES : 12 PRINCIPES

1. Satisfaire le client en livrant tôt et régulièrement.
2. Le changement est bienvenu, même tardivement dans le développement.
3. Livrer toutes les deux semaines à deux mois, avec une tendance pour la période la plus courte.
4. Les concepteurs et les développeurs doivent collaborer quotidiennement au projet.
5. Bâtissez le projet autour de personnes motivées. Donnez leur l'environnement et le soutien dont elles ont besoin, et croyez en leur capacité à faire le travail.
6. La méthode la plus efficace de transmettre l'information est une conversation en face à face ➔(user story)

ÉTS

17

MÉTHODES AGILES

7. Un logiciel fonctionnel est la meilleure unité de mesure de la progression du projet.
8. Les processus agiles visent à supporter un rythme de développement soutenable. Commanditaires, développeurs et utilisateurs devraient pouvoir maintenir le rythme indéfiniment.
9. Une attention continue à l'excellence technique et à la qualité de la conception améliore l'agilité.
10. La simplicité - l'art de maximiser la quantité de travail à ne pas faire - est essentielle.
11. Les meilleures architectures, spécifications et conceptions sont issues d'équipes qui s'auto-organisent.
12. À intervalle régulier, l'équipe réfléchit aux moyens de devenir plus efficace, puis accorde et ajuste son comportement dans ce sens.

ÉTS

18

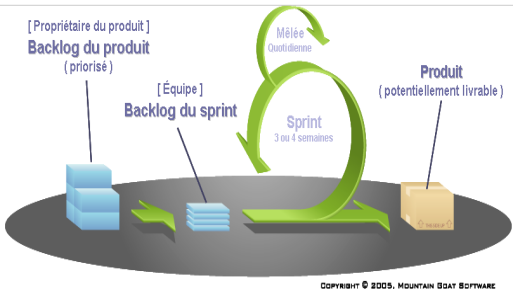
MÉTHODES AGILES

- Les principales méthodes
  - Scrum
  - Agile Modeling
  - Agile Unified Process (AUP) (version simplifiée de RUP)
  - Crystal Clear
  - Extreme Programming (XP)
  - Lean software development

ÉTS

19

MÉTHODES AGILES : SCRUM



ÉTS

20

MÉTHODES AGILES : AGILE UP

- Adaptation de la méthode RUP (Rational Unified Process) à l'agilité. Méthode dédiée à l'amélioration continue de l'efficacité des processus de développement
- La plus prescriptive des méthodes agiles → convient mieux aux grandes organisations
- 4 grandes périodes itératives et incrémentales (Inception, Élaboration, Construction, Transition)
- À considérer lorsque la culture organisationnelle nécessite une méthode plus structurante que les méthodes agiles moins prescriptives

ÉTS

21

MÉTHODES EXTRÊMES

- Méthode agile adaptée aux équipes réduites
- Approche plus flexible et ouvert au changement
- Itération courtes (quelques jours) avec tests intensifs (méthode agile accélérée)
- Pousse à l'extrême des principes simples

ÉTS

22

MÉTHODES AGILES VS AUTRES MÉTHODES

Un produit opérationnel	Plutôt que	Documentation exhaustive
Collaboration avec le client	Plutôt que	Négociation d'un contrat
Adaptation au changement	Plutôt que	Suivi d'un plan

23 ÉTS

23

MÉTHODES AGILES

Pourquoi de courtes itérations?

24 ÉTS

24

MÉTHODES AGILES ET LA MODÉLISATION

- Utilisation d'UML
- Modéliser les parties difficiles ou délicates
- Rester à un niveau de modélisation minimalement suffisant
- Outils simples et adaptés aux groupes

25 ÉTS

25

MÉTHODES AGILES:

Exemple de Scrum

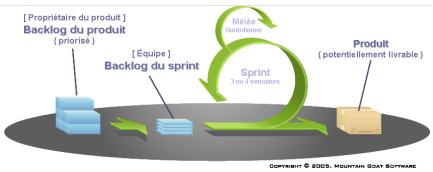
ÉTS

2
6

26

MÉTHODES AGILES : SCRUM

- Besoins capturés dans un carnet *de produit* (*backlog*)
- Le *backlog* est priorisé par une personne
- Les **Cycles** de développement sont des ***Sprints***
- Un *Sprint* dure *de* 2 à 4 semaines
- Mêlée quotidienne



27 ÉTS

27

SCRUM : LES PRINCIPES :

- **LA TRANSPARENCE**
  - s'assurer que les aspects du processus sont visibles
    - activité est « complétée » ➔ complétée
- **L'INSPECTION :**
  - Les différents aspects du processus doivent être inspectés régulièrement
- **L'ADAPTATION**
  - Il faut apporter les ajustements le plus rapidement possible afin d'éviter tout autre écart
    - *Si une dérive est constatée pendant l'inspection, le processus doit être adapté*
  - Comment ?
    - mêlée quotidienne, planification du sprint , revue du sprint et la rétrospective du sprint

28 ÉTS

28

SCRUM : LES ACTEURS

- Scrum Master
  - Veille au bon fonctionnement de l'équipe
  - Leader au service de l'équipe (*Servant Leader)*
  - Gardien des pratiques de Scrum
  - Serviteur de l'équipe – Facilitateur
- Attention :
  - Le Scrum Master peut être un membre de l'équipe; par exemple, un développeur.
    - Toutefois, cela amène souvent des conflits lorsque le Scrum Master doit choisir entre enlever des obstacles et réaliser ses propres tâches
  - Le Scrum Master n'est pas un chef de projet !

29 ÉTS

29

SCRUM : LES ACTEURS

- Propriétaire du produit (*Product Owner*)
  - Responsable de maximiser la valeur du travail accompli par l'équipe Scrum
  - Gère le *Backlog*
  - Définit les priorités
  - Accepte ou rejette les livrables
- Attention :
  - Le propriétaire du produit est une seule et unique personne, et non pas un comité

30 ÉTS

30

SCRUM : LES ACTEURS

- L'équipe
  - Doit transformer le contenu du carnet du produit en produit livrable à la fin de chaque sprint
  - Doit avoir toutes les compétences pour terminer le *sprint*
  - Un membre de l'équipe ne peut pas refuser de programmer parce qu'il est un architecte
    → Il ne serait pas utile dans l'équipe
  - Il n'y a pas de titres dans l'équipe SCRUM
  - L'équipe s'organise elle-même
  - La taille optimale d'une équipe est de 7 plus ou moins 2

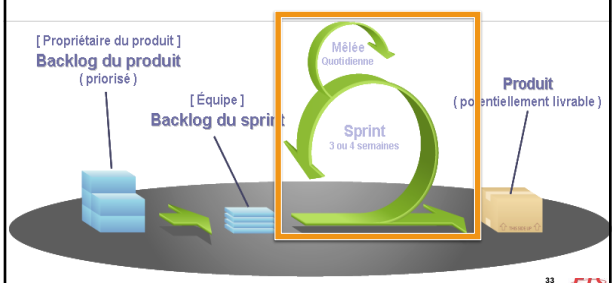
31

SCRUM : LES ACTEURS

- L'équipe
  - Les membres de l'équipe possèdent souvent des compétences spécialisées
    - e.g. la programmation, le contrôle de la qualité, l'analyse d'affaires, l'architecture, la conception d'interfaces, ou bien la conception de BD
  - Ne change pas pendant un Sprint
  - Peut changer à la fin d'un sprint
  - Personne ne doit dire à l'équipe de travailler sur des priorités différentes que celles établies par le propriétaire

32

SCRUM: LE SPRINT



33

SCRUM: LE SPRINT

- Le sprint est une itération
- Les sprints sont limités dans le temps (bloc de temps) (*Time-Boxes*).
- le ScrumMaster s'assure qu'aucun changement qui viendrait modifier l'objectif du sprint n'est apporté
- Les activités d'un sprint sont :
  - la réunion de planification de sprint,
  - le développement,
  - la revue
- Les sprints se succèdent dans le temps. Il n'y a aucun temps mort entre chacun des sprints

34

SCRUM : RÉUNION DE PLANIFICATION DE SPRINT (1/2)

- Permet de planifier l'itération (sprint)
- Sa durée est limitée à huit heures pour un sprint d'un mois (quatre heures pour un sprint de deux semaines)
- Le propriétaire présente à l'équipe les éléments du carnet de produit ayant la plus haute priorité
- Le carnet du produit est priorisé

35

SCRUM : RÉUNION DE PLANIFICATION DE SPRINT (2/2)

- Seule l'équipe est en mesure de savoir ce qu'elle peut accomplir à l'intérieur d'un sprint
- La réunion se divise en deux parties :
  1. l'équipe décide de ce qui va être fait « Quoi? »
  2. l'équipe détermine comment elle va développer les fonctionnalités sélectionnées « Comment? »
- Si l'équipe détermine qu'elle a trop à faire ou pas assez, elle peut renégocier le carnet de sprint avec le propriétaire du produit

36

SCRUM: MÊLÉE QUOTIDIENNE

- 15 minutes, tous les jours
- Elle se déroule au même endroit et à la même heure
- Le ScrumMaster enseigne à l'équipe comment garder la réunion la plus courte possible
- N'est pas une réunion sur l'état d'avancement
- Trois questions pour chacun
  - Qu'avez-vous fait hier?
  - Qu'allez-vous faire aujourd'hui?
  - Quels sont vos problèmes?

ÉTS

37