Cours Génie Logiciel Avancé
Chapitre 1 : Introduction

INSTITUT SUPERIEUR INFORMATIQUE ISI
المعهد العالي للإعلامية

Responsables du cours :
Héla Hachicha
Hatem Ben Sta

Année Universitaire : 2014 - 2015

Plan du cours

- Chapitre 1 : Introduction au génie logiciel
- Chapitre 2 : Gestion de projet informatique
- Chapitre 3 : Processus Unifié de Modélisation
- Chapitre 4 : Activités dans le Processus Unifié
- Chapitre 5 : Les Design Pattern
- Chapitre 6 : MDA : Model-Driven Architecture
- Chapitre 7 : Le langage de contraintes OCL
- Chapitre 8 : Méta-modélisation MOF
- Chapitre 9 : Les profils UML
- Chapitre 10 : MDE: Model Driven Engineering

Sommaire

Chapitre 1 : Introduction au génie logiciel

- Définitions
- Crise du logiciel
- Objectif du génie logiciel
- Problèmes actuels du génie logiciel

Qu'est-ce que le génie logiciel ?

Le **génie logiciel** est un domaine des sciences de l'ingénieur dont l'objet d'étude est la conception, la fabrication et la maintenance des systèmes informatiques complexes.

Génie logiciel : ensembles d'activités conduisant à la production d'un logiciel

Cours Génie Logiciel Avancé
Yann Régis-Gianas, 2011

Qu'est-ce que le génie logiciel ?

- ***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 de qualité* en vue d'utiliser un ordinateur pour résoudre certains problèmes

Gaudel et al, 1996
Précis de génie logiciel, Edition Masson.

Qu'est-ce que le génie logiciel ?

- L'application des principes de l'ingénierie à la conception des logiciels : l'établissement et l'application des **approches méthodiques** au **développement**, à **l'opération** et à la **maintenance** *des logiciels* dans le but d'obtenir *des systèmes logiciels* ***économiques, fiables et efficaces*** dans un contexte de fonctionnement pratique

Tony Wong,
Ecole Technologique Supérieure,
Université du Québec

Qu’est-ce que le génie logiciel ?

- Ensembles des activités de **conception** et de **mise en œuvre** des *produits et des procédures* tendant à **rationaliser** la production du logiciel et de son suivi.

Dictionnaire Encyclopédique du Génie Logiciel, Masson, 1997

Quelques exemples de spécifications plus complexes

- Un traducteur automatique : est-ce qu'un texte anglais « bien écrit » ?
- Un logiciel « boursicoteur » (effectuant des achats et des ventes en bourse) : Comment établir une spécification sans y inclure un modèle du système financier ?
- Un jeu vidéo : Comment spécifier ce qui est amusant ?

Crise du logiciel

Historiquement, il y a eu une prise de conscience dans les années 70, appelée la **crise du logiciel**, dû à un tournant décisif : c'est à cette époque que le coût de construction du logiciel est devenu plus important que celui de la construction du matériel.

- Deux constatations :
  - *Le logiciel n'était pas fiable*
  - *Il était incroyablement difficile de réaliser dans des délais prévus des logiciels satisfaisant leurs cahiers des charges*

Crise du logiciel

- *Le logiciel n'était pas fiable :*
  - La première sonde Mariner vers Venus qui s'est perdue dans l'espace à cause d'une erreur dans un programme Fortran
  - En 1971, lors d'une expérience météorologique en France, 72 ballons contenant des instruments de mesure furent détruits tout d'un coup à cause d'un défaut dans le logiciel
  - Dans la nuit du 15 au 16 décembre 1990, les abonnés de ATT de la côte Est des Etats-Unis furent privés de tout appel longue distance à cause d'une réaction en chaîne dans le logiciel de réseau due à un changement de version de ce logiciel
  - En juin 1996, 37 sec après le décollage, la fusée Ariane 501 (prototype de la version Ariane 5) fut détruite par une explosion

Crise du logiciel

- *Il était incroyablement difficile de réaliser dans des délais prévus des logiciels satisfaisant leurs cahiers des charges*
  - Certains projets n'aboutissent jamais
    - Compilateur PL1 chez Control Data dans les années 70
  - D'autres aboutissent avec des retards importants et des remises en cause dramatiques
    - SNCF a rencontré des difficultés importantes à la mise en service du système Socrate
    - Dans les années 60, OS-360 d'IBM fut livré en retard, il nécessitait plus de mémoire que prévu, son prix de revient dépassait de beaucoup les estimations, et ses premières versions comportaient des erreurs

Crise du logiciel

- **Les symptômes les plus caractéristiques de cette crise sont :**
  - Les logiciels réalisés ne correspondent souvent pas aux besoins des utilisateurs
  - Les logiciels contiennent trop d'erreurs (qualité du logiciel insuffisante)
  - Les coûts de développement sont rarement prévisibles et sont généralement prohibitifs
  - La maintenance des logiciels est une tâche complexe et coûteuse
  - Les délais de réalisation sont généralement dépassés
  - Les logiciels sont rarement portables

Crise du logiciel

- **Quelques sources de la crise :**
  - Une idée grossière du logiciel à réaliser est suffisante pour commencer à écrire un programme (il est assez tôt de se préoccuper des détails plus tard).

    => Faux : une idée imprécise du logiciel à réaliser est la cause principale d'échecs

  - Une fois que le programme est écrit et fonctionne, le travail est terminé.

    => Faux : la maintenance du logiciel représente un travail important : le coût de la maintenance représente plus du 50 % du coût total d'un logiciel

Crise du logiciel

- **Quelques sources de la crise :**
  - Si les spécifications du logiciel à réaliser changent continuellement, cela ne pose pas de problème, puisque le logiciel est un produit souple

    => Faux : des changements de spécifications peuvent se révéler coûteux et prolonger les délais

  - Si la réalisation du logiciel prend du retard par rapport aux délais prévus, il suffit d'ajouter plus de programmeurs afin de finir dans les délais.

    => Faux : si l'on ajoute des gens à un projet, il faut compter une période de familiarisation. Le temps passé à communiquer à l'intérieur du groupe augmente également, lorsque la taille du groupe augmente.

Crise du logiciel

- **Pourcentages :**
  - Plus de 30 % de tous les projets logiciels sont abandonnés avant la fin.
    - Equipe de MEP du projet non compétente
    - Dépassement du budget et/ou du temps
    - Ne cadre pas les besoins des utilisateurs, …
  - Plus de 70 % du reste échouent à la livraison des caractéristiques attendues.
    - Le logiciel ne répond pas aux besoins des utilisateurs, …
  - Le projet moyen dépasse par un facteur de 189 % :
    - son budget et
    - son échéancier.

Comment concevoir un logiciel de qualité ?

En plus du respect (essentiel) de sa spécification, la qualité d'un logiciel dépend des 4 critères suivants :

- Maintenabilité : Peut-on faire évoluer le logiciel ?
- Robustesse (Fiable) : Le logiciel est-il sujet à des dysfonctionnements ?
- Efficacité : Le logiciel fait-il d'une manière optimisée ce qu'on lui demande ?
- Utilisabilité : Est-il facile à utiliser ?

Objectif du génie logiciel

Le **génie logiciel** vise à garantir que :

1. la spécification répond aux besoins réels de ses clients ;
2. le logiciel respecte sa spécification ;
3. les coûts alloués pour sa réalisation sont respectés ;
4. les délais de réalisation sont respectés.

Problèmes actuels du génie logiciel

- La taille et la complexité du logiciel :
  - La complexité fonctionnelle
    - On demande de plus en plus de fonctionnalités
    - Les utilisateurs sont de plus en plus exigeants
  - Les technologies en perpétuelle mutation
    - Évolution continue
      - Matériel
      - logiciels
      - Technologie (web, …)
  - Une complexité architecturale
    - Architecture répartie en réseaux (client/serveur, …)
    - Sites distants, …

Problèmes actuels du génie logiciel

- La taille croissante des équipes :
  - Des compétences de plus en plus variées
    - Les fonctionnalités des logiciels sont de plus en plus complexes, une seule personne ne peut pas maîtriser toutes les exigences des utilisateurs (on ne peut pas être bon dans toutes les disciplines)
  - Des applications stratégiques au cœur du métier de l'entreprise
    - Aide à la décision, statistiques, planification, …
  - Des délais de plus en plus courts

Problèmes actuels du génie logiciel

- Des spécifications du logiciel sont peu précises :
  - Les spécifications des besoins :
    - sont les fondements d'un projet informatique ,
    - sont une interface difficile entre le domaine métier et l'informatique.

    Si elles sont mal expliquées, tout le projet peut échouer.

Problèmes actuels du génie logiciel

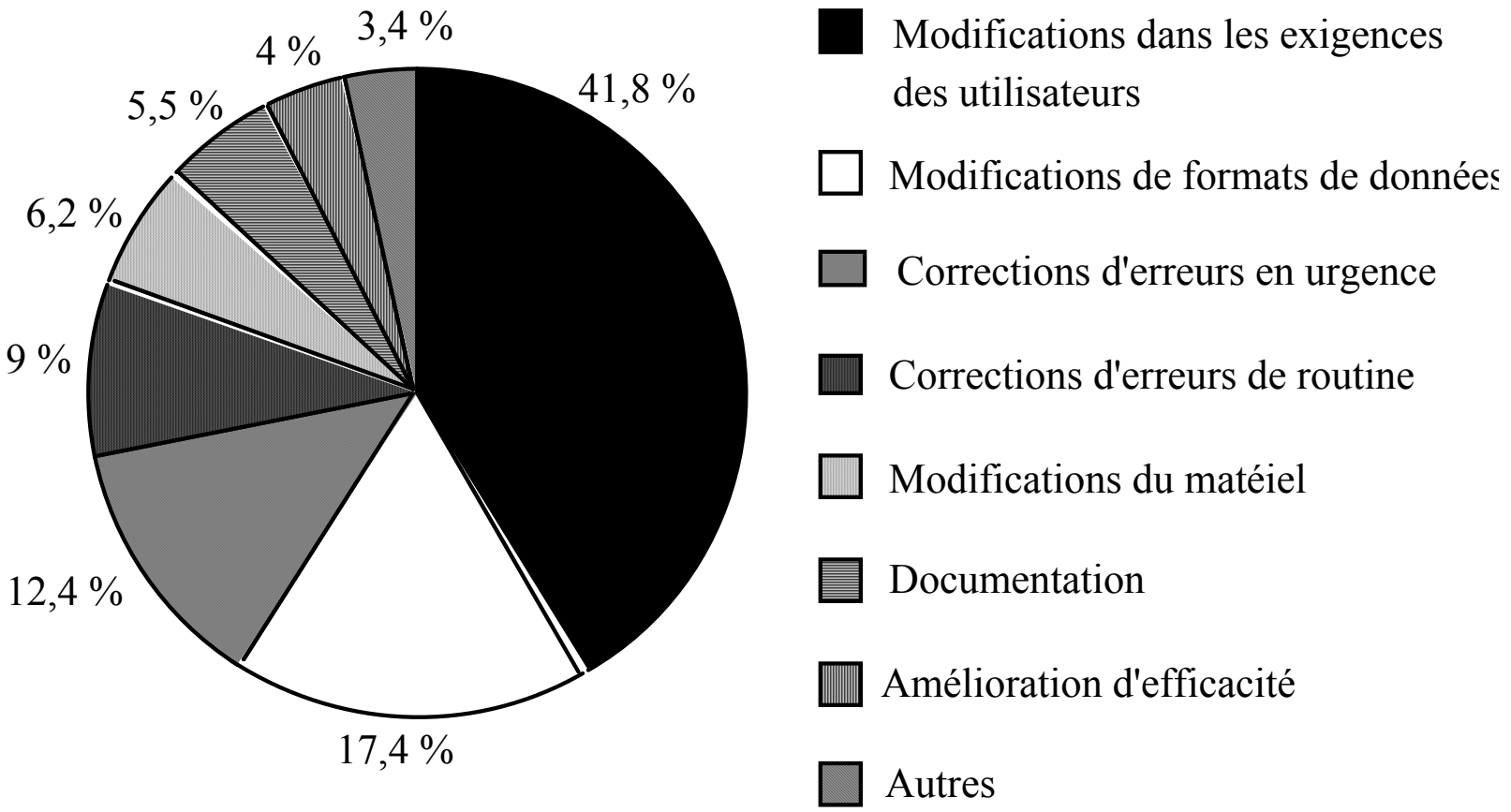
- L'évolution rapide des applications
  - Evolution fonctionnelle et technique
    - Modification des besoins du client
    - Modification de l'activité du client
    - Modification de l'environnement technique

La maintenance

- Plusieurs études ont été consacrées à l'évaluation des coûts de production des logiciels répartis sur tout le cycle de développement (i.e., analyse, conception, implémentation et maintenance).

- Avec les méthodes classiques de développement, toutes ces études s'accordent pour attribuer à l'étape de maintenance la moyenne de 70% du coût global de développement.

La maintenance



Répartition des coûts de maintenance ([Lientz 1979] in [Meyer 1988])