

IFT3912 – Développement et maintenance de logiciels

Introduction

Bruno Dufour
Université de Montréal
dufour@iro.umontreal.ca

Horaire

- Cours magistraux :
 - Lundi de 14h30 à 15h30, Pav. André-Aisenstadt salle 1207
 - Mercredi de 9h30 à 11h30, Pav. André-Aisenstadt salle 1207
- Démonstrations :
 - Lundi 15h30 à 17h30, Pav. André-Aisenstadt salle 1207

<http://www.iro.umontreal.ca/~dufour/cours/ift3912/>

Préalables

- IFT2255 (Génie Logiciel)
- Programmation en Java

3

Plan de cours

- **Gestion**
 - Notion de projet
 - Personnel et équipes
 - Développement agile
 - Gestion des risques
 - Estimation
 - Tâches et ordonnancement
- **Développement et maintenance**
 - Patrons de conception
 - Antipatrons et réusinage (*refactoring*)
 - Tests
 - Réutilisation et systèmes hérités
 - Systèmes de contrôle de révision
 - Analyse de programmes et outils

4

Évaluation

- Démonstrations : 10%
 - 10 x 1%
- Examens: 45 %
 - Intra : 20 %
 - Final : 25 %
- Projet : 45 %
 - En équipes de 4-5 personnes
 - Java, HTML, base de données, etc.

5

Retards

- Aucun retard ne sera toléré pour la remise de travaux.
- La note attribuée à un travail en retard sera zéro.

6

Plagiat

- Les discussions avec les autres étudiants du cours sont encouragées, mais le plagiat sous toute forme sera sévèrement puni, conformément aux règlements de l'Université de Montréal
 - échec du cours
 - sanctions

7

Bibliographie suggérée

- Robert K. Wysocki. *Effective Project Management* (3ième édition), Wiley. ISBN: 0471432210.
- Ian Sommerville. *Software Engineering* (9ième édition), Addison-Wesley. ISBN: 0137035152.

8

Introduction

Les logiciels autour de nous

- Logiciels de système
 - Système d'exploitation, compilateur, chargeur (*loader*), etc.
- Logiciels d'affaires
 - Service de la paye, comptabilité, etc.
- Logiciels scientifiques et de génie
 - Design assisté par ordinateur (*CAD*), simulation, prédictions météorologiques, etc.
- Logiciels personnels
 - Tableurs, éditeurs de texte, jeux, ...
- Logiciels de systèmes embarqués (*embedded*)
 - Automobiles, électroménagers, modems, routeurs, boîtes de câblodistribution, etc.

Les logiciels autour de nous

- Logiciels web
 - *B2C: business-to-customer* (ex. amazon.ca)
 - *B2B: business-to-business*
 - Cible principale des services web (*Web Services*)
 - *B2E: business-to-employee*
 - Intranet
 - Pour employés sur place et à distance

Source: B. G. Ryder / A. Rountev

11

Qu'est-ce qu'un logiciel?

- Un logiciel est:
 - Du code exécutable
 - Les données associées au programme
 - Des documents: besoins des usagers, design, guide d'utilisateur, guide de programmation, etc.
- Les logiciels jouent un rôle clé:
 - Produisent, gèrent et présentent de l'**information**

Source: B. G. Ryder / A. Rountev

12

Problèmes communs

- Trop chers
- Trop longs à développer
- Qualité inadéquate
- Trop complexes à supporter et maintenir
- Se dégradent avec l'âge
- Trop peu de personnel qualifié pour développer les logiciels

Source: B. G. Ryder / A. Rountev

13

Ariane 5

- Fusée Ariane 5
- Construite par l'Agence Spatiale Européenne
- Première mise à feu en juin 1996
- A explosé 40 secondes après le lancement
 - Heureusement, pas de passagers à bord
- Coûts:
 - Fusée et contenu : 500 millions \$
 - Développement : 7 milliards \$
- Cause de l'explosion: faute du logiciel

<http://www.ima.umn.edu/~arnold/disasters/ariane.html>

<http://www.around.com/ariane.html>

Source: B. G. Ryder / A. Rountev

14

Que s'est-il passé?

- Dépassement de capacité durant la conversion de la vélocité d'un entier de 64 bits à un entier de 16 bits
- L'exception n'a pas été captée
 - Le système de référence d'inertie a fait défaut (ainsi que le système de secours)
 - La fusée a dévié de sa trajectoire
 - Le module d'autodestruction s'est (correctement) activé
- Le code provenait d'Ariane 4
 - Même logiciel, mais des environnements d'exécution différents
 - Le code était demeuré actif après la phase de vol auquel il était dédié

Source: B. G. Ryder / A. Rountev

15

Un cas isolé?

Non.

16

Autres bogues célèbres

- Mars Pathfinder (1997)
 - Concurrence
- Mars Spirit Rover (2003)
 - Manque de mémoire
- Inversion dans un simulateur de F16
 - L'avion se retournait à chaque fois qu'il traversait l'équateur
- Panne de courant en Amérique du nord (2003)
 - Problème de concurrence a paralysé le système d'alarme durant plus d'une heure
- Samy / MySpace XSS (2005)

17

Génie logiciel

- Une discipline qui comprend:
 - Le processus de développement de logiciels
 - La méthodologie pour l'analyse, la conception, le **développement**, la vérification et la **maintenance** de logiciels
 - Des outils qui supportent le processus et la méthodologie

Source: B. G. Ryder / A. Rountev

18

Processus, Méthodologie & Outils

- Plusieurs tâches sont requises pour développer des logiciels
 - ex. conception, tests, etc.
- **Processus GL**: organisation et gestion de ces tâches
- **Méthodologie GL**: manière dont les tâches sont accomplies
 - ex. conception de classes, méthodes de tests, etc.
- **Outils GL**: aident à accomplir les tâches
 - Développement : Eclipse, Jazz, etc
 - UML : ex. Rational Rose
 - Tests : JUnit, JCover, etc.
 - ...

Source: B. G. Ryder / A. Rountev

19

Bref historique du développement

L'importance de la perspective historique

“Ceux qui ne se rappellent pas du passé sont condamnés à le répéter”

- Oublier les échecs
 - Susceptible de les répéter
- Oublier les succès
 - Improbable qu'ils seront répétés

Source: Barry Boehm, ICSE06 Keynote

Source: B. G. Ryder / A. Rountev

21

Historique du développement de logiciels

- 1950: les logiciels sont développés comme du matériel
 - Applications: avions, circuits, etc.
 - Économie: le temps d'ordinateur coûte plus cher que le salaire des opérateurs

“We're paying \$600/hour for that computer, and \$2/hour for you, and I want you to act accordingly.”

-- *Superviseur de Barry Boehm, 1955*

Source: Barry Boehm, ICSE06 Keynote

Source: B. G. Ryder / A. Rountev

22

Historique du développement de logiciels

- 1960: les logiciels ne sont plus du matériel
 - Propriétés:
 - Invisibles, complexes, exécutés par des ordinateurs, difficiles à modifier
 - La demande pour les programmeurs excède l'offre
 - Culture de "hacker"
 - Processus: coder puis corriger
 - Meilleurs outils: compilateurs, systèmes d'exploitation
 - Départements d'informatique sont créés
 - Plusieurs succès (ex. Apollo)
 - Problèmes: échecs de la majorité de systèmes de grande taille, code impossible à maintenir

Source: Barry Boehm, ICSE06 Keynote

Source: B. G. Ryder / A. Rountev

23

Historique du développement de logiciels

- 1970: Approches formelles sont développées
 - Programmation structurée, élimination des "goto"
 - Modèle de développement en cascade (*Waterfall*)
 - Notions d'étude des besoins et de conception dans le processus de développement
 - Problèmes des méthodes formelles
 - Succès limités aux petits programmes critiques
 - Preuves démontrent la présence de fautes, pas leur absence
 - Pas de communauté d'experts du développement

Source: Barry Boehm, ICSE06 Keynote

Source: B. G. Ryder / A. Rountev

24

Historique du développement de logiciels

- 1980 : Productivité, réutilisation, objets
 - Augmentation de la productivité
 - Travailler plus rapidement: outils et environnements
 - Travailler plus intelligemment : processus et méthodologie
 - Éviter le travail inutile: réutilisation, simplicité, objets
 - Développement de processus permettant de produire des logiciels
 - Réutilisation: bibliothèques orientées objet (Smalltalk, Eiffel, C++)

Source: Barry Boehm, ICSE06 Keynote

Source: B. G. Ryder / A. Rountev

25

Historique du développement de logiciels

- 1990: Modélisation
 - Y2K
 - Logiciels libres (*open source*)
 - Systèmes hérités (*legacy systems*)

Source: Barry Boehm, ICSE06 Keynote

Source: B. G. Ryder / A. Rountev

26