

Techniques (d'Adaptation et) de Génie Logiciel

Introduction

Thomas Ropars

`thomas.ropars@imag.fr`

ERODS research team – LIG/IM2AG/UGA

2015

Organisation du cours

- ▶ 12 heures de cours / 21 heures de TP
- ▶ Toutes les ressources sur Moodle (pwd: TAGL2016)

Évaluation

- ▶ Note de TP
- ▶ Examen final

Est ce que vous êtes d'accord pour que le projet inclus:

- ▶ Des threads (problèmes de synchronisation)
- ▶ Des sockets (communications réseaux)
- ▶ Les deux
- ▶ Aucun des deux

- ▶ Systèmes informatiques
 - ▶ 80% de logiciel
 - ▶ 20% de matériel
- ▶ Le matériel est fourni pour un nombre restreint de fabricants
 - ▶ Relativement fiable
- ▶ La plupart des fonctionnalités dans les systèmes informatiques sont fournies par le logiciel

Analyse de plus de 50000 projets de développement logiciel à travers le monde.

Réussite des projets

- ▶ Succès: 29%
- ▶ Problématiques: 52%
- ▶ Échec: 19%

16% de succès en 1994.

La probabilité de succès décroît avec la taille du projet:

- ▶ Petit: 70% de succès
- ▶ Moyen: 22% de succès
- ▶ Grand: 11% de succès

La probabilité de succès décroît avec la taille du projet:

- ▶ Petit: 70% de succès
- ▶ Moyen: 22% de succès
- ▶ Grand: 11% de succès

Attention les chiffres de ce rapport sont parfois contestés. Ils donnent tout de même une idée des enjeux.

Causes des défaillances pour applications clouds

Analysis of Business Data Processing Cloud Apps (Di Martino et al (2012))

- ▶ 34% des défaillances sont dues à des entrées utilisateur non prévues
- ▶ 32% des défaillances sont des timeouts
- ▶ 95% des défaillances sont dues aux mêmes 5 modules (37% des LOCs)

Causes des défaillances pour applications clouds

Analysis of Business Data Processing Cloud Apps (Di Martino et al (2012))

- ▶ 34% des défaillances sont dues à des entrées utilisateur non prévues
- ▶ 32% des défaillances sont des timeouts
- ▶ 95% des défaillances sont dues aux mêmes 5 modules (37% des LOCs)

Enseignements:

- ▶ Il faut traiter correctement les exceptions
- ▶ Un petit nombre d'erreurs sont à l'origine de la majorité des défaillances → De meilleures procédures de tests sont requises

D'autres défaillances dans les nuages

- ▶ Amazon Web Services (2015)
 - ▶ Indisponibilité de 6 heures
 - ▶ Source: problème dans la gestion des méta-données dans DynamoDB (base de donnée NoSQL)
 - ▶ La défaillance s'est propagée à d'autres services
 - ▶ 2 des plus gros clients de AWS sont Amazon.com et Netflix
 - ▶ Netflix a annoncé ne pas avoir été impacté
 - ▶ Une analyse des plaintes des consommateurs montrent une très forte hausse pendant cette période

Coût d'une indisponibilité dans le cloud

- ▶ Amazon.com a subi une panne de 45 minutes en 2013
- ▶ Le coût en terme de ventes a été estimé à 4 millions de dollars

Coût du développement logiciel

Source E. Chenu

Ordres de grandeur

- ▶ 1 H/An = 1350 heures
- ▶ 1h \simeq 50 €
- ▶ Productivité \simeq 2 à 5 lignes/h

Dimension	Nb lignes	Heures	Coût	Hommes/An
Petit	30000	6000	300 K€	4
Gros	500000	100000	5 M€	74

Une Définition

Le terme génie logiciel désigne l'ensemble des méthodes, des techniques et outils concourant à la production d'un logiciel, au delà de la seule activité de programmation

Une Définition

Le terme génie logiciel désigne l'ensemble des méthodes, des techniques et **outils** concourant à la production d'un logiciel, au delà de la seule activité de programmation

- ▶ La taille des projets: pour certains, des millions de ligne de code (MLOC)

- ▶ La taille des projets: pour certains, des millions de ligne de code (MLOC)
- ▶ Contribution à des projets existants:
 - ▶ Peu de projets démarrés *from scratch*
 - ▶ Plus de valeur d'ajouter 100 LOCs à un grand projet largement utilisé que d'écrire 10000 LOCs dans son coin.

Les défis

- ▶ La taille des projets: pour certains, des millions de ligne de code (MLOC)
- ▶ Contribution à des projets existants:
 - ▶ Peu de projets démarrés *from scratch*
 - ▶ Plus de valeur d'ajouter 100 LOCs à un grand projet largement utilisé que d'écrire 10000 LOCs dans son coin.
- ▶ Réutilisation de code existant
 - ▶ Est ce que le concepteur d'une voiture commence par réinventer la roue?

- ▶ La taille des projets: pour certains, des millions de ligne de code (MLOC)
- ▶ Contribution à des projets existants:
 - ▶ Peu de projets démarrés *from scratch*
 - ▶ Plus de valeur d'ajouter 100 LOCs à un grand projet largement utilisé que d'écrire 10000 LOCs dans son coin.
- ▶ Réutilisation de code existant
 - ▶ Est ce que le concepteur d'une voiture commence par réinventer la roue?
- ▶ Collaboration avec d'autres développeurs
 - ▶ Comment interagir?
 - ▶ Comment fournir du code réutilisable?

Les défis

- ▶ La taille des projets: pour certains, des millions de ligne de code (MLOC)
- ▶ Contribution à des projets existants:
 - ▶ Peu de projets démarrés *from scratch*
 - ▶ Plus de valeur d'ajouter 100 LOCs à un grand projet largement utilisé que d'écrire 10000 LOCs dans son coin.
- ▶ Réutilisation de code existant
 - ▶ Est ce que le concepteur d'une voiture commence par réinventer la roue?
- ▶ Collaboration avec d'autres développeurs
 - ▶ Comment interagir?
 - ▶ Comment fournir du code réutilisable?
- ▶ Des utilisateurs/clients

L'industrie du logiciel, c'est 5% de projets "*from scratch*" et 95% de projets existants. Le travail consiste alors à:

- ▶ Réutiliser
- ▶ Faire évoluer
- ▶ Étendre
- ▶ Adapter
- ▶ Maintenir
- ▶ Réorganiser

- ▶ Gestionnaire de versions
- ▶ Versions
- ▶ Source code management
- ▶ Automatisation
- ▶ Refactoring
- ▶ Débugger
- ▶ Tests unitaires
- ▶ Couverture de code
- ▶ Intégration continue
- ▶ Livraison continue

- ▶ Notes de cours de D. Donsez
- ▶ Notes de cours de P. Gérard