

Cours Qualité et Tests

Chapitre 1 - Introduction



Responsable du cours :
Héla Hachicha

Année Universitaire : 2016 - 2017

2

Ce cours



s'intéresse à la sûreté de fonctionnement de logiciels

Plan du cours

- Chapitre 1 : Introduction
- Chapitre 2 : Modèles de cycle de vie et Test
- Chapitre 3 : Tests
 - Introduction au test de logiciels
 - Les concepts de test de logiciels
 - Les méthodes de design des cas de test
 - Les stratégies de tests
- Chapitre 4 : Métriques
- Chapitre 5 : Qualité

Sommaire

- Introduction : Rappel
 - Génie logiciel
 - Utilité de logiciels
 - Crise de logiciels
 - Comment résoudre les problèmes ?
 - Comment réussir ?

5

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.

6

Définition du « logiciel »

« Un logiciel est :

- (1) l'ensemble des **instructions** qui après leurs exécutions nous fournissent la fonction et la performance désirée,
- (2) les **structures de données** qui permettent aux programmes de manipuler adéquatement l'information,
- (3) des **documents** qui décrivent les opérations et l'utilisation des programmes. »

Pressman, R., Software Engineering - A Practitioners Approach, McGraw-Hill, Fifth Edition, 2001

7

Utilité du logiciel

- Le logiciel est omniprésent dans nos sociétés !
- Le logiciel est de plus en plus **complexe**
- Le logiciel est de plus en plus **critique**
- L'industrie du logiciel est un des moteurs de la nouvelle économie et offrira d'excellentes opportunités d'emplois pour de nombreuses années à venir

8

Utilité du logiciel

- En dépit des énormes succès, la réputation de l'industrie du logiciel n'est pas si glorieuse que cela:
 - Dépassement de budget
 - Dépassement d'échéance
 - Mauvaises fonctions livrées
 - Erreurs (bugs) et autres problèmes

9

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 ?

10

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*

11

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 : crash de la fusée
 - Therac-25: appareil canadien servant à traiter le cancer et ayant tué plusieurs patients d'une surdose de radiation

12

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

13

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

14

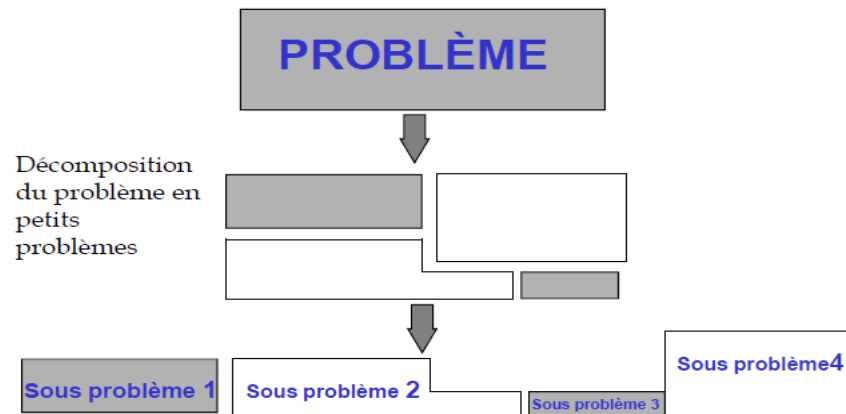
Comment résoudre les problèmes?

- Les produits logiciels sont nombreux et complexes
- Analyse et synthèse des exigences de développement
 - **Analyse:** décomposer un grand problème en petits problèmes plus compréhensibles
 - L'abstraction est la clé
 - **Synthèse:** construire (assembler) un logiciel à partir de blocs de construction plus petits
 - L'assemblage est le défi

15

Comment résoudre les problèmes?

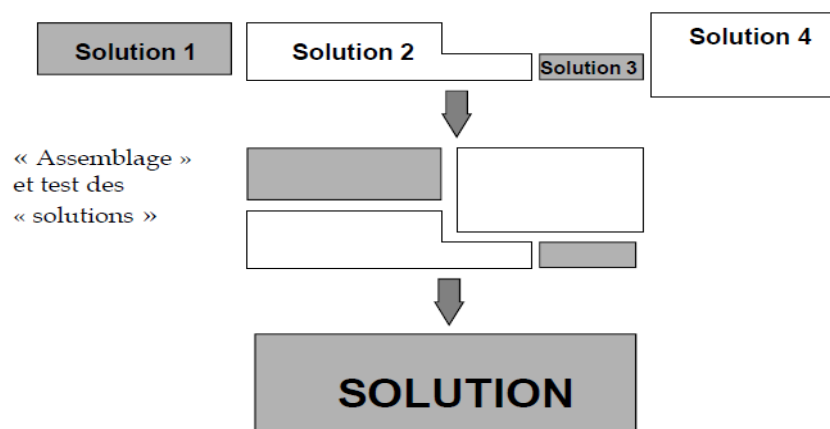
- **Processus d'analyse**



16

Comment résoudre les problèmes?

- **Processus de synthèse**



17

Comment résoudre les problèmes?

Pour aider à résoudre un problème, on utilise :

- **Méthode** : se réfère à une procédure formelle
- **Outil** : un instrument ou un système automatisé pour bien accomplir quelque chose
- **Procédure**: une combinaison d'outils et techniques pour produire un produit
- **Paradigme**: philosophie ou approche pour construire un produit

18

Comment réussir?

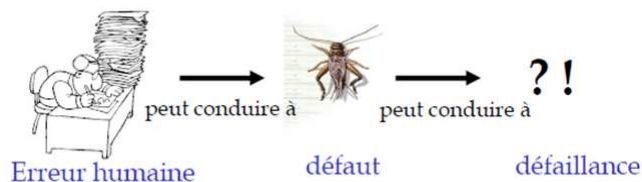
- Accomplir les tâches plus rapidement et efficacement
 - Traitement de texte, tableurs , e-mail
- Supporter les avances en médecine, agriculture, transport, éducation en multimédia, et la plupart des industries
- Cependant, le logiciel présente des problèmes !

19

Comment réussir?

Terminologies pour décrire les « Bogues »

- **Un défaut** : survient lorsque l'être-humain se trompe par mégarde, appelée une erreur, en accomplissant certaines activités logicielles (IEEE 729)
- **Une défaillance** : est la modification d'un comportement requis d'un système (IEEE 729)



20

Rapport de l'Institut National des Standards et de la Technologie au USA

- En l'an 2000, la recette totale des ventes de logiciels avait approximativement atteint les 180 billions \$. (billion = 10⁹!)
- Ce chiffre était atteint grâce à une large force de travail composée de:
 - 697,000 ingénieurs logiciels
 - 585,000 programmeurs
- Les erreurs logicielles se sont avérées assez coûteuses devant entraîner des pertes financières s'élevant à 59.5 billions \$ annuellement.



21

Rapport de l'Institut National des Standards et de la Technologie au USA

- Même si toutes les erreurs ne peuvent pas être supprimées, plus qu'un tiers des ces coûts (~ 22.2 billions \$) pourrait être éliminé par une infrastructure de **test** améliorée permettant une identification et une suppression précoces et efficaces des erreurs logicielles.
- Plus que la moitié des erreurs logicielles ne sont détectées qu'à des phases avancées du **cycle de vie** des produits commercialisés.



22

Une idée sur les salaires des testeurs

(D'après « Software Testing Institute Salary Survey », 2002)

	Without ASQ Certification	With ASQ Certification
Reliability Engineer	\$68,684.00	\$76,769.00
Software Quality Engineer	\$72,130.00	\$74,297.00
Quality Manager	\$66,058.00	\$71,163.00
Quality Engineer	\$56,180.00	\$61,544.00
Quality Auditor	\$48,800.00	\$53,830.00
Quality Technician	\$37,291.00	\$38,400.00

ASQ = American Society for Quality