

HAI913I - Évolution et restructuration des logiciels

Cours: 1

Date: 13/09/2022

Professeur/intervenant: Seriai

Objectifs de ce cours :

- Présenter les aspects essentiels de la maintenance
 - Aspect techniques de la maintenance
 - Aspects liés à l'organisation et à la gestion des activités de maintenance
-

Notes :

- Évaluer par les TPs + quelques QCM
- Méthodologie requise pour créer et faire évoluer le logiciel
- Cycle de vie d'un logiciel : plusieurs types : cascade / en V ...
- durée de vie d'un logiciel : 20 ans en moyenne
- Lois de Lehman : Tout logiciel qui n'évolue pas devient obsolète
 - Modification continue : Un programme doit changer continuellement pour rester utile et efficace dans le monde réel
 - Complexité croissante : Le logiciel devient de plus en plus gros lors des maintenances successives
 - Autorégulation : Correction de bugs entre les mises à jour majeures
- Maintenance adaptative/corrective/préventive/perfective
- La maintenance est très coûteuse. Elle correspond en moyenne à presque 70% du coût global des logiciels.
- Trois tâches principales de la maintenance : Réparer, Adapter et Ajouter/Modifier
- Problèmes liés à la maintenance :
 - Compréhension limitée
 - Rapidité avec laquelle un individu n'ayant pas travaillé sur le développement d'un logiciel, peut comprendre et modifier ce logiciel.

- La compréhension est encore plus difficile si elle se fait à partir du code source et non de sa documentation
- Test de regression :
 - Permet de vérifier que les défauts ont été supprimés après la maintenance
- Analyse d'impact :
 - Savoir l'impact d'un changement sur la totalité du logiciel.
 - Nécessite l'analyse de la structure et du contenu du logiciel
- Voir les autres problèmes sur les slides
- Techniques de maintenance :
 - La rétro-ingénierie logicielle :
 1. Analyse statique
 - Avantage :
 - pas d'exécution, donc pas de dommages et pas de délais
 - Inconvénient :
 - il n'est pas possible d'être certain de certaines propriétés avec, à cause du problème de l'indécidabilité.
 2. Analyse dynamique
 - Avantages :
 - Permet d'obtenir des résultats plus précis, des informations de nature temporelle à propos de l'exécution, des informations sur la fréquence ou l'importance de certains événements.
 - Inconvénients :
 - Possibilités de dommages
 - Temps d'exécution
 - L'étendue de l'analyse dépend des scénarios d'exécutions
- Objectif de l'analyse :
 - Vérification
 - Compréhension
 - Transformation
 - Décompilation
 - Obfuscation : Cacher le code
 - Optimisation
 - etc...
- L'analyse permet de détecter des erreurs
- Bytecode : un code intermédiaire entre les instructions machines et le code source - Il n'est pas directement compilable

- Deux phases de l'analyse :
 1. Extraction d'un modèle de représentation du code source
 - Modèle exacte versus approximatif
 - Représentation du flot de contrôle versus flot de données versus graphe d'appels...
 2. Analyse des propriétés basées sur les modèles extraits
 - Exemple :
 - Analyse classique pour la vérification
 - Informations extraites à partir de l'analyse statique :
 - AST
 - Graphe de flot de contrôle

Une représentation sous forme de graphe de tous les chemins qui peuvent être suivis par le programme lors de son exécution

Les sommets du graphe sont :

Bloc de base : un bout de code sans saut

Les arcs :

les sauts dans le code

La plupart des graphe de contrôle contiennent un bloc d'entrée et de sortie
 - Graphe de dépendance de contrôle

Montre quelles instructions sont exécutées en fonction de la valeur d'une expression dans le programme

Les noeuds du graphe sont les mêmes que ceux du graphe de flot de contrôle
 - Flot de données

Avoir l'information sur l'utilisation des variables dans le temps
 - Graphe de dépendance de données

Est un graphe dont les noeuds sont les mêmes que celui du flot de contrôle

Définitions :

- Maintenance : La modification d'un logiciel, après sa livraison, afin de corriger les défaillances, d'améliorer sa performance ou d'autres attributs ou de l'adapter suite à des changements d'environnements.

- Les changements au logiciel et à sa documentation causés par un problème ou le besoin de l'améliorer
- La totalité des activités requises afin de procurer un support, au meilleur cout possible, d'un logiciel.
- Certaines activités peuvent avoir lieux avant la phase de production du logiciel.
- Analyse : Analyser les propriétés d'un programme de manière automatique
- Analyses Statiques :
 - Basées sur l'analyse du code source
 - Considèrent toutes les exécutions possibles
 - Calculs complexes mais sans impact sur l'exécution
- Analyses Dynamiques :
 - Basées sur l'analyse d'une ou plusieurs exécutions du code
 - Considèrent certaines exécutions
 - Impact sur l'exécution proportionnel à la quantité analysée
- Modèle : Représentation abstraite
- Abstraction : Le fait de ne pas se soucier de certains détails