ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE

République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي والبحث العلمي

INVENTEURS DU MONDE NUMÉRIQUE

Plateformes et techniques de rétro-ingénierie orientée objet

INTRODUCTION & CONTEXTE

Aujourd'hui, les industries dépendent fortement des technologies logicielles, et la majorité des entreprises, indépendamment de leur taille et du secteur d'activité, font désormais face au problème bien connu de devoir gérer, maintenir, faire évoluer ou remplacer leurs systèmes logiciels existants.

Afin de migrer/moderniser ces projet logiciels, on utilise les Plateformes et Les Techniques de Rétro-Ingénierie pour avoir une abstraction et mieux comprendre les systèmes logicielle à travers le principe de modélisation.

Le but principal de ce poster est de donner une vision complète sur les:
Les Techniques et Les Plateformes de Rétro-Ingénierie

OBJECTIVES

Nous résumons les différents objectifs de ce master à:

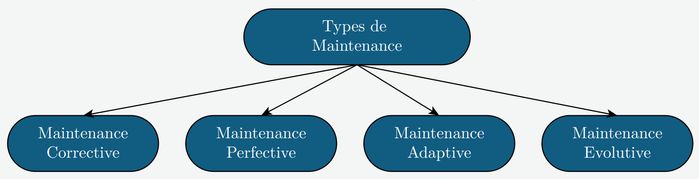
- Définir la maintenance et l'évolution et ainsi leur rôle dans le SDLC.
- Élaborer la modélisation dans le contexte de la rétro-ingénierie MDRE.
- Présenter les différentes techniques utilisées pour la rétro-ingénierie.
- Comparer et évaluer les solutions de rétro-ingénierie.

MAINTENANCE ET ÉVOLUTION LOGICIELLE

Les activités de maintenance et d'évolution représentent une grande partie des coûts logiciels dans une organisation.

1- Maintenance:

La maintenance logicielle consiste principalement à corriger les anomalies d'un système logicielle tout en préservant ses fonctionnalités. Il existe 4 types :



Types De Maintenance

2- Évolution:

L'évolution logicielle consiste à effectuer un changement continu d'un état inférieur, plus simple à un état supérieur ou meilleur.

3- Rôle de la maintenance et l'évolution:

La majeure partie du budget logiciel est consacrée à la modification de logiciels existants (maintenance évolutive) plutôt qu'au développement de nouveaux logiciels.

Si le logiciel ne continue pas à s'adapter aux changements des besoins et de l'environnement, il devient progressivement moins utile (1ère Loi de Lehman, évolution logicielle).
Les changements sont inévitables et se produisent pour plusieurs raisons:

- Nouvelles fonctionnalités/exigences apparaissent;
- L'environnement industriel change ;
- La croissance de nombre d'utilisateurs

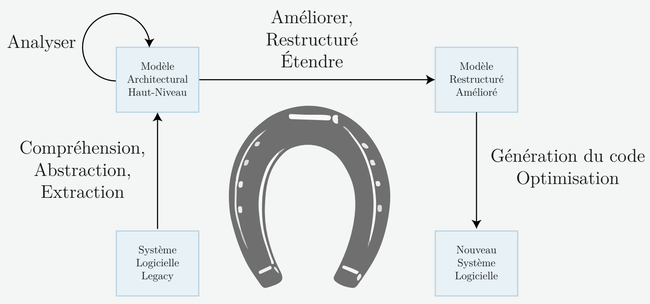
MODÉLISATION ET RÉTRO-INGÉNIERIE

1- La Modélisation:

La modélisation est un paradigme du génie logiciel reposant sur la création, la manipulation et l'utilisation intensive de modèles de nature diverses et variées.

2- Le MDRE (Model-Driven Reverse Engineering):

C'est l'application de la modélisation à la rétro-ingénierie, il vise à construire des modèles logiciels de plus haut niveau, plus abstraits, à partir du code source.



Le modèle du Processus Horseshoe

TÉCHNIQUES DE RÉTRO-INGÉIERIE

1- Analyse lexicale :

C'est le processus de décomposition d'une séquence de caractères du code source en unités lexicales constitutives appelées tokens.

2- Analyse syntaxique :

La processus de déterminer les expressions, les instructions et les modules d'un programme par les compilateurs.

3- Analyse de flux de contrôle :

Après avoir déterminé la structure du programme, CFA est une analyse statique du code pour déterminer le CFG. Cette analyse aide par exemple à produire des programmes optimisés.

4- Analyse de flux de données :

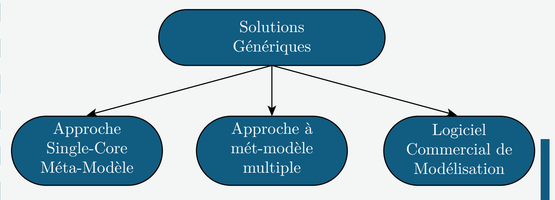
L'analyse des flux de données (DFA) s'intéresse à la façon dont les valeurs des variables définies circulent et sont utilisées dans un programme.

5- La Visualisation:

La visualisation logicielle est une stratégie utile pour permettre à un utilisateur de mieux comprendre les systèmes logiciels.

SOLUTIONS DE RÉTRO-INGÉNIERIE

Pour les solutions on distingue deux grandes familles : les solutions spécifiques et les solutions à usage général.



Classification des solution Génériques

1- Plateformes Générique de Rétro-Ingénierie

	Approche single-core méta-modèle	Approche à méta-modèle multiple	Logiciel de modélisation général
CORUM-II	✓		
Moose		✓	
Garces et al.		✓	
MoDisco		✓	
Rational software			✓
MagicDraw			✓
Enterprise Architect			✓
Modelio			✓
Visual Paradigm			✓

2- Solutions Spécifiques de Rétro-Ingénierie

	Découverte	Migration	Intégration	Anaylse et Compréhension
JaMoPP	✓			
Spoon	✓			
Briand et al.	✓			
Sneed	✓	✓		
RE Analyzer	✓	✓		
Barbier et al.	✓	✓		
Fleurey et al.	✓	✓		
Clavreul et al.	✓		✓	
Alnusair et al.	✓		✓	
Ramon et al.	✓		✓	
ConQAT	✓			✓
GUPRO	✓			✓
SWAG Kit	✓			✓
CodeCrawler	✓			✓
Columbus	✓			✓
SolidFx	✓			✓
Pacione et al.	✓			✓
Olsson et al.	✓			✓
Fradet et al.				✓

CONCLUSION

En conclusion, nous avons une vision complète sur la rétro-ingénierie, ses techniques et plateformes. En effet, nous avons défini le contexte de la rétro-ingénierie logicielle (MDRE) tout en étudiant les différentes techniques et solutions de rétro-ingénierie qui aident à la compréhension des systèmes. Cette compréhension est établie grâce à la modélisation qui est un paradigme de génie logiciel basé sur les modèles.

RÉFÉRENCE

Hugo Bruneliere. "Generic model-based approaches for software reverse engineering and comprehension". Thèse de doct. Nantes, 2018.

Kelly Garcés et al. "White-box modernization of legacy applications : The oracle forms case study". In : Computer Standards & Interfaces 57 2018)), p. 122-110

Nicolas Anquetil et al. "Modular Moose : A new generation software reverse engineering environment" (2020).

Lowell Jay Arthur. Software evolution : the software maintenance challenge. Wiley-Interscience, 1988.

Keith Bennett. "Legacy systems : Coping with success". In : IEEE software 1995) 12.1), p. 23-19.

Ecrit par :
- **Ahmed Zaki BENNECER**

Encadré par :
- **Lamia YESSAD**

Promoteurs:
- **Anne ETIEN**
- **Nicolas ANQUETIL**

Master MSL **Mars 2022**