



UNIVERSITÉ DE NANTES

UNIVERSITÉ DE NANTES

MASTER INFORMATIQUE

RAPPORT DE STAGE TER

---

## Faut-il tester le code ou les modèles de composants logiciels ?

---

RÉALISÉ PAR :

Daniel AHMED

Yamen ALNAJM

Soufyane BELHADJ KACEM

Promo 2018

*Tuteur :*

M. GILLES ARDOUREL

M. PASCAL ANDRE

M. JEAN-MARIE MOTTU

*Chef d'équipe :*

M. GILE

## REMERCIEMENTS

Nous tenons à remercier cordialement toutes les personnes qui nous ont permis de réaliser ce travail dans les meilleures conditions.

Un remerciement particulier à Mr. Gilles ARDOUREL et Pascal ANDRÉ, nos enseignants responsables, pour nous avoir encadrés tout au long de ce travail. Merci pour leurs conseils, leurs disponibilités et leurs implications.

Nous remercions les membres du jury qui ont bien voulu examiner et évaluer ce rapport.

## RÉSUMÉ

Ce sujet de rapport vise à étudier et appliquer l'approche d'ingénierie dirigée par les modèles pour le développement d'une application qui automatise les tests logiciels au niveau du modèle. Le Budget IT consacrée aux tests et à l'assurance qualité augmente de 9 points d'une année sur l'autre, pour atteindre 35% de la dépense IT selon Capgemini. Contre seulement 26% en 2014 ! Et 18% en 2012.

La SSII prévoit qu'en 2018, la part des tests et de la qualité passera à 40% du total. Ce budget colossal est divisé en parts égales entre maintenance d'applications existantes et nouveaux développement. Dans 35% des entreprises, cette inflation est considérée comme un problème [1].

Cette problématique pousse les entreprises à adopter une nouvelle stratégie et de miser sur la recherche d'un mécanisme d'automatisation des tests logiciels.

Cela nous ramène à penser à une solution pour automatiser les tests logiciel au niveau modèle, en se basant sur l'approche MDA, le but est d'offrir une interface qui donne la possibilité à l'utilisateur de choisir les composants ainsi que les services qui veulent tester, et notre application se charge du reste.

Toute la partie de développement de ce projet fut mis en place de manière itérative, basée sur la méthodologie Agile.

**MOTS-CLÉS : COMPOSANTS, SERVICES, MODÈLES,  
TEST, MDA**

## ABSTRACT

This report topic aims to investigate and apply the model-driven engineering approach for the development of an application that automates software testing at the model level. The IT budget dedicated to quality control and quality assurance increases by 9 points from one year to the next, reaching 35% of IT expenditure according to Capgemini. Against only 26% in 2014! And 18% in 2012.

The SSII predicts that in 2018, the share of tests and quality will increase to 40% of the total. This huge budget is divided equally between maintenance of existing applications and new development. In 35% of companies, this inflation is considered as a problem [1].

This problem pushes companies to adopt a new strategy and to bet on the search for a mechanism of automation of the software tests.

This brings us back to thinking about a solution for automating software-level tests at the model level, based on the MDA approach, the goal is to offer an interface that gives the user the possibility to choose the components and the services that want to test, and our application is getting rest. All the development part of this project was implemented in an iterative way, based on the Agile methodology.

## L'AVANT-PROPOS

L'une des grandes craintes des dirigeants et chefs de projet est de constater une série de bugs sur le produit fini, les conséquences peuvent être désastreuses, comme une grande perte financière ou voir pire humaine. La volonté de créer des applications de qualité à amener les entreprises à repenser leurs façon de faire, et à automatiser les tests logiciel. Entre 2014 et 2015 le pourcentage de test automatisé est passé de 28% à 45%. Dans le cadre d'un thème de projet de stage proposé par l'équipe AELOS du laboratoire LS2N à Nantes, une (// a voir selon l'avancement) a été développé et délivré. Ce rapport fournit toutes les informations concernant l'approche MDA et (// a voir selon l'avancement)

# SOMMAIRE

o.1	Introduction . . . . .	6
<b>1</b>	<b>Chapitre 1 : vérification des systèmes à composant</b>	<b>7</b>
1.1	Introduction . . . . .	7
1.2	L'outil Kmelia . . . . .	7
1.3	L'architecture du système . . . . .	7
1.4	L'outil CostoTest . . . . .	8
1.5	Conclusion . . . . .	8
<b>2</b>	<b>Chapitre 2 : ingénierie dirigé par les model</b>	<b>9</b>
2.1	Introduction . . . . .	9
2.2	Les modèles MDA et ses transformations . . . . .	10
2.2.1	Les modèles MDA . . . . .	10
2.3	Les transformations de modèles . . . . .	10
2.4	Les objectifs du MDA . . . . .	11
2.5	Conclusion . . . . .	11
<b>3</b>	<b>Chapitre 3 : intention de test et assemblage</b>	<b>12</b>
3.1	Introduction . . . . .	12
3.2	L'intention de test . . . . .	12
3.3	Architecture . . . . .	12
3.4	L'assistance au harnais de test . . . . .	13
3.5	Conclusion . . . . .	13
<b>4</b>	<b>Chapitre 4 : transformation des models xtext</b>	<b>14</b>
4.1	Introduction . . . . .	14
4.2	Conclusion . . . . .	14
<b>5</b>	<b>Chapitre 5 : Application de la méthode proposée</b>	<b>15</b>
5.1	Introduction . . . . .	15
5.2	Processus de développement . . . . .	15
5.3	Evaluation . . . . .	15
5.4	Conclusion . . . . .	15
5.5	CONCLUSION GENERALE . . . . .	16
5.5.1	Conclusion . . . . .	16
5.5.2	Perspectives . . . . .	16

## O.1 INTRODUCTION

Dans le développement dirigé par les modèles (MDD), il est essentiel de s'assurer de la correction des modèles avant de commencer le processus de transformations et de génération de codes. On diminue ainsi le coût élevé de la détection tardive d'erreurs.

En effet, ce mode de développement permet de faire abstraction du système qu'on s'apprête à modéliser. C'est à dire elle permet de rendre compréhensible le système modélisé et d'apporter des réponses aux questions que l'on se pose sur le système. Aujourd'hui, il y a de nombreuses entreprises qui s'orientent vers ce modèle comme ils l'ont fait avant avec l'approche objet. Dans le MDD, au lieu de considérer que tout est objet comme dans la programmation objet, on va considérer que tout est modèle.

L'équipe que nous avons intégrée durant ce stage "AeLoS" travaille sur la vérification et la validation de logiciels aux niveaux modèles. Le test est un moyen très utilisé pour vérifier et valider les applications, mais ce dernier rencontre plusieurs problèmes comme la couverture des tests, l'hétérogénéité du code, séparation des points de vue...

En 2013, l'équipe AELOS a réalisé des travaux sur l'assistance à la construction du harnais de tests, en 2014 et 2015, le prototype a été améliorée par des groupes d'étudiants du Master ALMA en 2016. En 2018 nous reprenons le projet afin d'expérimenter une nouvelle approche.

En effet, suite à différentes approches qui ont été testées et jugées comme étant non abouties (Ecriture du Code Java du Harnais), laborieuses (Spécification manuelle du Harnais) ou simple et rapide (L'outil CostoTest), la nouvelle approche consiste à donner la possibilité de suivre à divers niveaux de raffinement du modèle le comportement de l'application de test, permettre l'enrichissement des tests et des modèles en parallèle mais aussi que les tests arrivent à résister aux diverses évolutions des modèles.

Dès lors, la question qui se pose, c'est comment s'y prendre pour arriver à terme de cette nouvelle approche? Parmi les réponses que l'on peut apporter, c'est de commencer par formaliser le test, par là nous voulons dire bien séparer les différentes parties qui vont constituer le test (domaine de test, intention de test, données de test...) et formaliser le processus de construction de harnais de test.

Dans la suite de ce rapport, nous verrons qu'elles sont les moyens logiciels qui ont été mis en place pour formaliser le test et la construction de harnais.

Le cas d'études se portera sur le contrôle de voiture autonome.

# 1 | CHAPITRE 1 : VÉRIFICATION DES SYSTÈMES À COMPOSANT

---

1.1	Introduction . . . . .	7
1.2	L'outil Kmelia . . . . .	7
1.3	L'architecture du système . . . . .	7
1.4	L'outil CostoTest . . . . .	8
1.5	Conclusion . . . . .	8

---

## 1.1 INTRODUCTION

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

## 1.2 L'OUTIL KMELIA

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

## 1.3 L'ARCHITECTURE DU SYSTÈME

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula



sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

#### 1.4 L'OUTIL COSTOTEST

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

#### 1.5 CONCLUSION

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

# 2

## CHAPITRE 2 : INGÉNIERIE DIRIGÉ PAR LES MODEL

2.1	Introduction . . . . .	9
2.2	Les modèles MDA et ses transformations . . . . .	10
2.2.1	Les modèles MDA . . . . .	10
2.3	Les transformations de modèles . . . . .	10
2.4	Les objectifs du MDA . . . . .	11
2.5	Conclusion . . . . .	11

### 2.1 INTRODUCTION

Pour bien manipuler les composants nous utiliserons l'approche d'ingénierie dirigée par les modèles MDA, que nous allons présenter dans ce chapitre. L'ingénierie des systèmes informatiques se base sur les modèles. Ces modèles permettent de décrire les systèmes en cours de développement et leur environnement à différents niveaux d'abstractions. Ces abstractions permettent de concevoir des applications indépendantes des plate-formes cibles. L'approche de l'ingénierie dirigée par les Modèles (IDM) consiste à programmer au niveau des modèles, représentés ces derniers comme une instance d'un méta-modèle, et à les utiliser pour générer le code final des applications.



FIGURE 1: Pyramide de modélisation de l'OMG [2]

Le MDA (Model Driven Architecture) de l'OMG est une approche typique d'ingénierie dirigée par les modèles pour la conception d'application, elle est basée sur le standard UML pour la définition des modèles et sur l'environnement de méta-méta modélisation MOF (Meta Object Facility) pour la programmation au niveau modèle et la génération de codes. L'ingénierie dirigée par les modèles est donc un paradigme dans lequel la modélisation est considérée comme l'élément central d'un logiciel.

## 2.2 LES MODÈLES MDA ET SES TRANSFORMATIONS

### 2.2.1 Les modèles MDA

*CIM : Computation Independent Model*

Les modèles d'exigence CIM décrivent les besoins fonctionnels de l'application, aussi bien les services qu'elle offre et les entités avec lesquelles elle interagit. Les CIM peuvent servir de référence pour s'assurer que l'application finie correspond aux demandes des clients.

*PIM : Platform Independent Model*

Les modèles d'analyse et de conception de l'application PIM représentent, indépendamment de toute plate-forme technique, une spécification neutre d'un système (modèle de métier et de service) qui ignore tous les détails de mise en œuvre et d'utilisation du système décrit.

*PSM : Platform Specific Model*

Les modèles de code PSM sont des modèles de métier et de service liés à une ou plusieurs plate-formes d'exécution.

*Code source*

Le code source est le résultat final de tout le processus MDA, il est généré automatiquement à partir du modèle PSM.

## 2.3 LES TRANSFORMATIONS DE MODÈLES

Le passage de PIM à PSM fait intervenir des mécanismes de transformation de modèles. La transformation de modèles est le processus qui convertit un modèle en un autre modèle. L'approche MDA demande de créer d'abord un modèle d'analyse (PIM), ce dernier est raffiné en un ou plusieurs modèles spécifiques à une ou plusieurs (PSM). La transformation du PIM vers PSM peut être complétée par d'autres informations comme celles relative au mapping entre le PIM et le PSM ou à la description de la plate-forme (PDM).

## 2.4 LES OBJECTIFS DU MDA

L'approche MDA de l'OMG a été définie pour répondre aux problèmes liés à l'évaluation continue des technologies, parmi les objectifs de cette approche :

### *Les modèles pérennes :*

L'objectif majeur du MDA est l'élaboration des modèles pérennes (PIM), ces derniers doivent survivre le changement de plate-forme et être réutilisable pour plusieurs plate-formes. Cette pérennité est garantie grâce à l'utilisation d'UML qui est un standard stable mais aussi grâce aux modèles qui sont par nature des entités pérennes.

### *Les gains de productivité :*

Les modèles sont la base du processus MDA, ils facilitent la communication entre experts du domaine et développeurs. Dans cette approche les modèles sont considérés comme un outil de production à l'aide de l'automatisation des transformations de modèles. La génération de la totalité de modèles de code (PSM) devient alors automatique, à cause des modèles qui sont considérés comme des éléments qui véhiculent une information précise est nécessaire pour cette transformation. Le mappage automatique de la PSM permet d'enlever le risque d'erreur, il produit ainsi un gain de coût à la transformation de la PIM, c'est le gain de productivité.

### *La prise en compte des plate-forme d'exécution :*

La MDA permet d'intégrer les spécifications des différentes plate-formes d'exécution pendant la transformation du modèle PIM vers PSM. Et de cette manière le développement des applications mobiles multi-plateformes est devenu facile, puisque tout est basé sur un modèle PIM qui est indépendant des détails techniques des plate-formes d'exécution (J2EE, .Net, PHP, etc.) Il faut noter que toutes modifications dans le modèle PIM, conduit à un changement dans les fonctionnalités du système livré.

## 2.5 CONCLUSION

Dans ce chapitre nous avons détaillé l'approche d'ingénierie dirigée par les modèles MDA, mais ce n'est pas la seule approche de génie logiciel qui se base sur les modèles dans son processus de développement.

# 3

## CHAPITRE 3 : INTENTION DE TEST ET ASSEMBLAGE

---

3.1	Introduction . . . . .	12
3.2	L'intention de test . . . . .	12
3.3	Architecture . . . . .	12
3.4	L'assistance au harnais de test . . . . .	13
3.5	Conclusion . . . . .	13

---

### 3.1 INTRODUCTION

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

### 3.2 L'INTENTION DE TEST

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

### 3.3 ARCHITECTURE

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo

lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

### 3.4 L'ASSISTANCE AU HARNAIS DE TEST

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

### 3.5 CONCLUSION

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

# 4

## CHAPITRE 4 : TRANSFORMATION DES MODELS XTEXT

---

4.1	Introduction . . . . .	14
4.2	Conclusion . . . . .	14

---

### 4.1 INTRODUCTION

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

### 4.2 CONCLUSION

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

# 5

## CHAPITRE 5 : APPLICATION DE LA MÉTHODE PROPOSÉE

---

5.1	Introduction . . . . .	15
5.2	Processus de développement . . . . .	15
5.3	Evaluation . . . . .	15
5.4	Conclusion . . . . .	15
5.5	CONCLUSION GENERALE . . . . .	16
5.5.1	Conclusion . . . . .	16
5.5.2	Perspectives . . . . .	16

---

### 5.1 INTRODUCTION

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

### 5.2 PROCESSUS DE DÉVELOPPEMENT

Derouler un exemple.....

### 5.3 EVALUATION

Ce que nous avons réussi, a faire et a ne pas faire

### 5.4 CONCLUSION

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.



## 5.5 CONCLUSION GENERALE

### 5.5.1 Conclusion

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

### 5.5.2 Perspectives

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

## TABLE DES FIGURES

FIGURE 1	Pyramide de modélisation de l'OMG [2] . . . . .	9
----------	---	---

## LISTE DES TABLEAUX

## BIBLIOGRAPHIE

- [1] Reynald Fléchaux. Transformation digitale : le coût des tests et de la qualité logicielle explose, septembre 2015. <https://www.silicon.fr/>.
- [2] Denisse Yessica Muñante Arzapalo. Une approche basée sur l'ingénierie dirigée par les modèles pour identifier, concevoir et évaluer des aspects sécurité. *UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR ÉCOLE DOCTORALE DES SCIENCES EXACTES ET DE LEURS APPLICATIONS*, page 25, décembre 2014.