



Master Méthodes Appliquées à la Gestion d'Entreprise - MIAGE

Mémoire Master 2

Implication du domaine et gestion des exigences

Auteur :

Mme. Khaoula ZITOUN

Encadrant :

Pr. Pascal POIZAT

Version 1.0 du
24 mai 2018

24 mai 2018

Table des matières

1	Chapitre 1 : Introduction	3
2	Chapitre 2 : Etat de l'Art	5
2.1	Introduction	5
2.2	Présentation de l'existant	5
2.3	Présentation d'éléments de pistes de solution	5
2.4	Conclusion	5
3	Chapitre 3 :	5

1 Chapitre 1 : Introduction

Depuis 1994, les résultats du rapport CHAOS du Standish Group sont mis à jour chaque année et montrent les principales raisons d'échec et de succès d'un projet au sein d'un panel représentatif d'entreprises américaines. Ce rapport classe les projets informatiques en trois catégories :

- **Type 1** : les projets qui se terminent dans le temps et le budget, avec un périmètre fonctionnel livré conforme au périmètre initialement défini.
- **Type 2** : les projets qui se terminent en ayant respecté le temps ou le budget avec un périmètre fonctionnel livré légèrement différent du périmètre initial.
- **Type 3** : les projets qui ont été abandonnés en cours de projet pour diverses raisons

Les différentes études montrent que les projets de type 1 représentent 20%, les projets de type 2 50% et les projets de type 3 représentent 30%. Il est donc à souligner que le taux d'échec des projets reste important.

Parmi les facteurs d'échec, nous retrouvons :

- **Exigences incomplètes (13,1%)**
- **Manque d'implication des utilisateurs (12,4%)**
- Manque des ressources
- **Attentes irréalistes (9,9%)**
- Manque de support de l'exécutif (9,3%)
- **Changement sur les exigences et les spécifications (8,7%)**
- etc

Les facteurs relatifs aux exigences représentent **44,1%**.

Par ailleurs, parmi les facteurs de réussite nous retrouvons :

- **Implication des utilisateurs (15,9%)**
- Support du management exécutif (13,9%)
- **Définition claire des exigences (13%)**
- Bonne planification
- **Attentes réalistes (8,2%)** etc

Les facteurs relatifs aux exigences représentent **37,1%**. [8]

Cette étude nous permet donc d'affirmer que l'implication des parties prenantes ainsi que la définition et la gestion des exigences jouent un rôle important dans l'avenir d'un projet.

L'implication des parties prenantes permet notamment de réduire l'écart qui existent entre celles-ci [2]. En effet, un écart de communication entre les experts du domaine et les ingénieurs en logiciel a été observé [9], comme cela a été rapporté par des chercheurs. [4]. Définir des exigences pour ensuite les implémenter peut donc devenir dangereux. Les experts domaines, n'ayant pas forcément des compétences techniques mais métier, et les ingénieurs en logiciel ayant des compétences techniques mais n'appréhendent pas entièrement tous les domaines, on peut se poser la question de **comment définir les exigences tout en s'assurant que les experts puissent les rédiger et les ingénieurs les comprendre ?**

Domain-Specific Requirement Elicitation

Une solution à cette première problématique est l'ingénierie des domaines. Cette dernière est un élément clé pour que l'ingénierie des exigences soit efficace. [3]. L'ingénierie des domaine

La spécification des exigences liés au domaine nécessite un langage d'exigences propre au domaine : les Domain Specific Requirement Language. De tels langages permettent de spécifier les exigences en terme d'abstractions de domaine d'application. [3] Une approche générique fournit une solution générale pour de nombreux problèmes dans un certain domaine, mais une telle solution peut être sous-optimale. Une approche spécifique fournit une bien meilleure solution pour un ensemble plus restreint de problèmes.[1] Les Domain Specific Model rendent la modélisation des exigences moins compliquée et réduit l'effort d'apprentissage pour les scientifiques et favorise le génie logiciel dans les projets.[2]

Le but de l'ingénierie de domaine est d'identifier, de modéliser, de construire, de cataloguer et de diffuser des artefacts qui représentent les points communs et les différences au sein d'un domaine.[6][1] La modélisation de ces artefacts peuvent se faire en empruntant différentes méthodes : des méthodes orientées objet, la notion UML, des méthodes formelles, les méthodes ViewPoint ou encore les DSL (Domain Specific Language) que nous présenterons dans la première partie de ce mémoire.

Bien qu'il existe des méthodes pour modéliser le domaine métier que l'on pourrait donc étendre à la définition des exigences, il n'en reste pas moins que des tests d'acceptation sont nécessaires pour assurer la satisfaction du client. Ces tests sont souvent créés en fonction d'une spécification des exigences et servent à vérifier que les obligations contractuelles sont respectées.[7] Une nouvelle question que nous nous posons est donc : **comment valider des exigences orientées domaine en passant par les tests ?**

Il existe beaucoup de recherches et d'implémentations d'outils sur des tests basés sur des modèles, la dérivation de tests à partir d'un modèle.[5]

Les tests d'acceptation peuvent être spécifiés de plusieurs façons, depuis les user stories basées sur la compréhension de textes suivis jusqu'aux langages formels. Parce que l'exécution des tests d'acceptation est longue et coûteuse, il est hautement souhaitable d'automatiser ce processus. L'automatisation des tests d'acceptation donne une réponse objective lorsque les exigences fonctionnelles sont remplies.[7]

On sait donc qu'il est possible de modéliser un domaine et donc par extension les exigences de celui-ci, de générer des tests à partir d'un modèle et de valider une exigence à partir d'un test. Nous souhaitons à travers ce mémoire proposer des éléments de réponses à la problématique suivante : **comment à partir d'une spécification d'exigences, générer des tests d'acceptation en mettant le domaine métier au coeur de la démarche ?**

Dans ce document nous présenterons dans une première section comment définir des exigences, quelques outils de suivi, les différentes méthodes de modélisation du domaine ainsi que les techniques de génération de tests.

Dans la seconde partie de ce document nous proposons de créer un DSL d'exigence à partir duquel générer des tests d'acceptation. A partir de ces derniers nous proposons de "tagguer" les exigences pour lesquels les tests sont passés ou non afin

d'assurer le suivi des exigences et la qualité du logiciel.

Tests d'acceptance

2 Chapitre 2 : Etat de l'Art

2.1 Introduction

2.2 Présentation de l'existant

2.3 Présentation d'éléments de pistes de solution

2.4 Conclusion

3 Chapitre 3 :

Références

- [1] Paul Klint ARIE VAN DEURSEN et Joost VISSER. “Domain-Specific Languages : An Annotated Bibliography”. In : *Newsletter ACM SIGPLAN Notices Homepage archive Volume 35 Issue 6, Pages 26 - 36* (2000).
- [2] Jeff Gray BARRET R.BRYANT et Marjan MERNIK. “A domain specific Requirements Model for Scientific Computing : NIER Track”. In : *IEEE* (2011). DOI : 10.1145/1985793.1985922.
- [3] Jeff Gray BARRET R.BRYANT et Marjan MERNIK. “Domain-specific software engineering”. In : *Conference Paper* (2010).
- [4] D. Zowghi C. COULIN et “A Lightweight Workshop A. E. K. SAHRAOUI. “Centric Situational Approach for the Early Stages of Requirements Elicitation in Software Systems Development”. In : *Proceedings of the International Workshop on Situational Requirements Engineering Processes (SREP 2005), France* (2005).
- [5] Michael FELDERER et al. “A Tool-Based Methodology for System Testing of Service-Oriented Systems”. In : 0 (août 2010), p. 108–113.
- [6] Yair Wand IRIS REINHARTZ-BERGER Arnon Sturm. *Domain Engineering*. Springer, 2010. ISBN : 978-3-642-36654-3. DOI : <https://doi.org/10.1007/978-3-642-36654-3>.
- [7] Grigori MelnikKris ReadFrank MAURER. “Suitability of FIT User Acceptance Tests for Specifying Functional Requirements: Developer Perspective”. In : *Extreme Programming and Agile Methods - XP/Agile Universe 2004* (2004).
- [8] Jean-Louis Boulanger STÉPHANE BADREAU. *Ingénierie des exigences: Méthodes et bonnes pratiques pour construire et maintenir un référentiel*. DUNOD. 2014. ISBN : 978-2-10-071340-0.
- [9] Olga De TROYER et Erik JANSSENS. “A feature modeling approach for domain-specific requirement elicitation”. In : *Dept. Computer Science Vrije Universiteit Brussel* (2014). DOI : https://www.researchgate.net/publication/286138717_A_feature_modeling_approach_for_domain-specific_requirement_elicitation.