

Mémoire de M2 : La connaissance du métier pour un logiciel de qualité

Khaoula Zitoun

5 novembre 2017

1 Introduction

Lors de mon stage de première année de Master MIAGE, j'ai fait parti d'une équipe de maîtrise d'ouvrage (MOA). J'ai donc souvent été en relation avec les équipes de maîtrise d'œuvre interne de la société. J'ai par ailleurs dû coder pour des besoins de data visualisations. Bien que j'avais en ma possession les schémas entité - association des bases de données, j'ai quelque peu eu du mal à trouver rapidement les champs dont j'avais besoin, et pour cause, l'équipe de maîtrise d'oeuvre (MOE) n'avait pas nommé ces champs tel que nous les nommions en MOA. Par exemple, ce que j'appelais « vendor » s'appellait « channel ».

Cet exemple illustre un réel problème : celui de la communication entre MOA et MOE. Il soulève également le problème de la compréhension du métier ; on peut se poser la question suivante "Sommes-nous sûrs et certains que la MOE comprend bien la différence entre ces deux termes dans notre domaine?".

Ce fossé entre équipe de développement et équipe fonctionnelle a de réelles conséquences. Parmi ces dernières, on peut citer le temps, la qualité et par extension le coût du logiciel. En effet, une mauvaise compréhension du métier ou une mauvaise communication entre les équipes entraine souvent des modifications du logiciel, qu'il s'agisse de sa conception ou de son implantation. Mais surtout, un logiciel qui ne répond pas au métier est un logiciel que ne satisfait pas le client, qu'il soit interne ou non.

Ce mémoire a pour but de répondre à la problématique suivante : comment réduire le fossé entre les équipes MOA et MOE dû à la compréhension du métier et la communication des équipes afin de satisfaire les clients ?

Dans ce mémoire, il s'agira dans un premier temps de présenter un ensemble d'approches, de techniques, de méthodes et d'outils autour des processus métier. Dans ces différentes approches nous analyserons notamment si le fait de rapprocher le métier au code, notamment avec des conceptions Domain Driven Design permet réellement une appréhension complète du métier de la part des développeurs.

Dans une seconde partie nous traiterons de l'implication de la gestion de projet dans ce processus de rapprochement entre MOA et MOE. En d'autres termes, nous

essayerons de démontrer en quoi la gestion de projet peut faire partie des clés de la communication entre les équipes et par conséquent à la compréhension du métier.

Puis, dans une troisième partie nous tacherons de déterminer si la combinaison d’approches méthodologiques et techniques avec des méthodes de gestions de projet peut drastiquement réduire le fossé entre le métier et le code.

[2] [1] [3] [4]

Références

- [1] Stephanie BAUCHE. *Sur la modélisation des processus métiers*. URL : http://bauche.stephanie.free.fr/documents/BPMN_memoireMaster2_StephanieBauche_vInternet.pdf.
- [2] Karim DAHMAN. “Bridging the Gap between the Business Processes and the Service Oriented Architecture”. In : *Inria Nancy - Grand Est, LORIA - NSS - Department of Networks, Systems and Services* (2011). DOI : <https://hal.archives-ouvertes.fr/inria-00608230/>.
- [3] Eric EVANS. *Domain-Driven Design - Tackling Complexity in the Heart of Software*. International series of monographs on physics. Addison Wesley, 2003. ISBN : 0-321-12521-5.
- [4] Juha-Pekka Tolvanen STEVEN KELLY. *Domain-Specific Modeling: Enabling Full Code Generation*. Wiley-IEEE Computer Society Press. 2008. ISBN : 978-0-470-03666-2.