

UML est-il soluble dans les méthodes agiles ?

octobre 07

Résumé

On entend beaucoup parler actuellement de deux approches ayant l'air fondamentalement opposées :

- l'approche Model-Driven, préconisée par l'OMG, s'appuyant sur une modélisation UML très poussée visant à une génération automatique de code quasi-complète,
- les méthodes agiles mettant plus l'accent sur la production rapide de code opérationnel que sur la documentation et minimisant donc la modélisation en amont.

Qu'en est-il vraiment ? UML est-il utilisable dans un contexte agile ou réservé au Model-Driven ? La modélisation agile peut-elle exister ? Si oui, quels sont ses principes et ses meilleures pratiques ?

Table des matières

1. INTRODUCTION	4
2. UML ET LES METHODES AGILES	5
2.1 Les principes agiles.....	5
2.2 RUP™	6
2.3 OpenUP	6
2.4 XP	7
2.5 Scrum.....	8
3. LA MODELISATION AGILE	9
3.1 Valeurs de la modélisation agile	9
3.2 Principes de la modélisation agile.....	9
3.3 Pratiques de la modélisation agile	10
3.4 La modélisation agile selon Craig Larman	10
4. CONCLUSION	12
POUR ALLER PLUS LOIN	12

Liste des Figures

Figure 1 :	Model Driven Architecture	4
Figure 2 :	Les 13 types de diagramme UML2.....	5
Figure 3 :	Le RUP est « Use-Case Driven ».....	6
Figure 4 :	La modélisation visuelle élève le niveau d'abstraction.....	7
Figure 5 :	Modélisation participative sur les murs	8
Figure 6 :	Principes de la modélisation agile.....	9
Figure 7 :	Pratiques de la modélisation agile.....	10
Figure 8 :	La modélisation sur projet selon Craig Larman.....	11

1. Introduction

Depuis quelques années, la modélisation objet avec le langage UML est devenue une pratique courante sur de nombreux projets informatiques. UML a en particulier été porté par la vague du Processus Unifié, avec surtout la version RUP d'IBM/Rational. UML est également le pilier central de l'initiative MDA (Model Driven Architecture) de l'OMG.

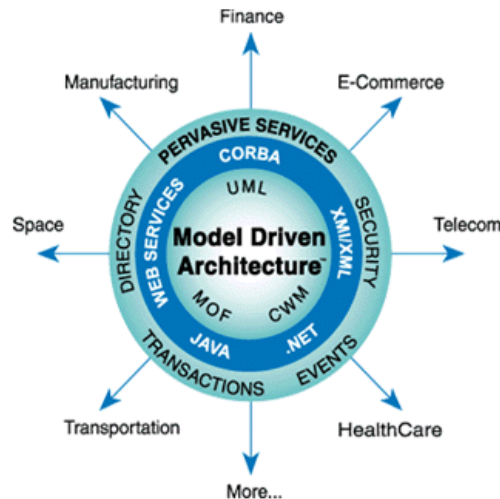


Figure 1 : Model Driven Architecture

Pourtant, on ne trouve pas la moindre allusion à UML ni à la modélisation dans le livre de référence sur Scrum¹... Alors, UML est-il devenu obsolète, la modélisation est-elle incompatible avec les méthodes agiles à la mode ? Nous allons essayer de démontrer le contraire dans cette présentation.

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage.

UML 2 s'articule autour de treize types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel.

¹ *Agile Project Management with Scrum*, Ken Schwaber, Microsoft Professional, 2004.

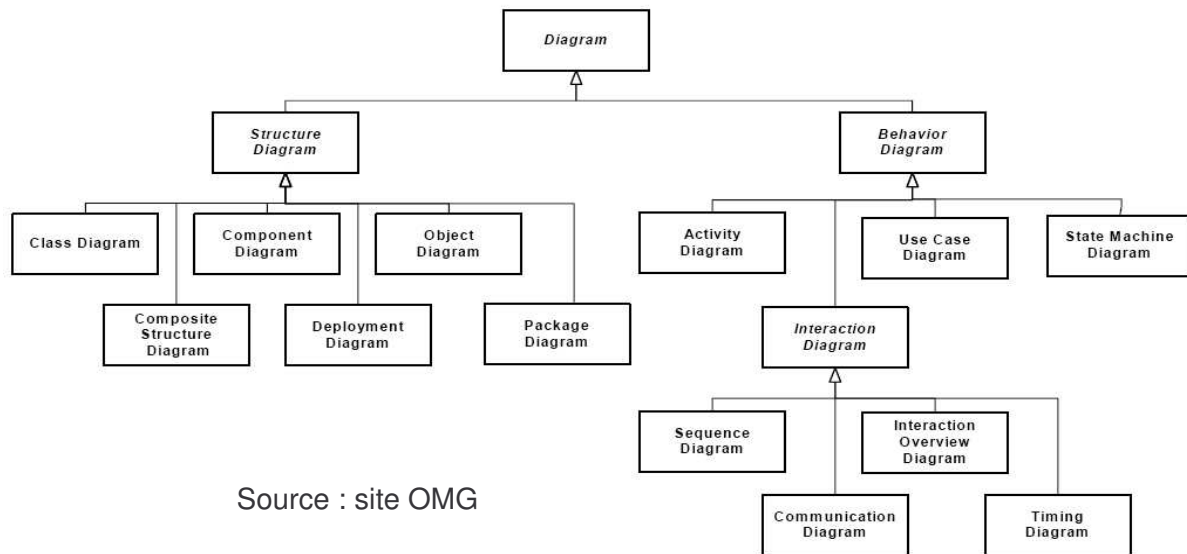


Figure 2 : Les 13 types de diagramme UML2

2. UML et les méthodes agiles

2.1 Les principes agiles

Le manifeste agile énonce quatre principes fondateurs :

- **Priorité aux personnes et aux interactions ...**

...Plutôt que sur les processus et les outils

- L'accent est mis sur les individus, leur expertise, l'esprit d'équipe plutôt que sur les processus et les outils

- **Des applications fonctionnelles opérationnelles**

...Plutôt qu'une documentation pléthorique

- On privilégie le code testé

- **Collaboration avec le client**

...Plutôt que négocier un contrat

- Le client devient un partenaire qui participe au projet pour donner régulièrement son feedback

- **Réactivité au changement**

...Plutôt que suivre un plan

- Le planning est flexible pour accepter les modifications nécessaires

D'après le second principe agile, l'utilisation de modèles UML à des fins de documentation doit être restreinte au strict nécessaire ... Regardons cependant plus en détail ce que disent vraiment les principales méthodes agiles, et leurs promoteurs, de la modélisation.

2.2 RUP™

Parmi les principales « best practices » promues par le RUP, on en trouve deux qui correspondent à notre sujet.

- Modéliser visuellement

RUP préconise d'enregistrer les pensées et de communiquer en utilisant des langages visuels et schématiques, comme UML, parce que les langages visuels sont naturels et faciles à appréhender pour le cerveau humain.

- Développer itérativement

Le développement itératif et incrémental est une valeur commune à toutes les méthodes agiles. RUP préconise de planifier le contenu technique des itérations en fonction de la priorisation des cas d'utilisation. Le cas d'utilisation (*use case*) est un concept UML qui représente un ensemble de séquences d'actions réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.

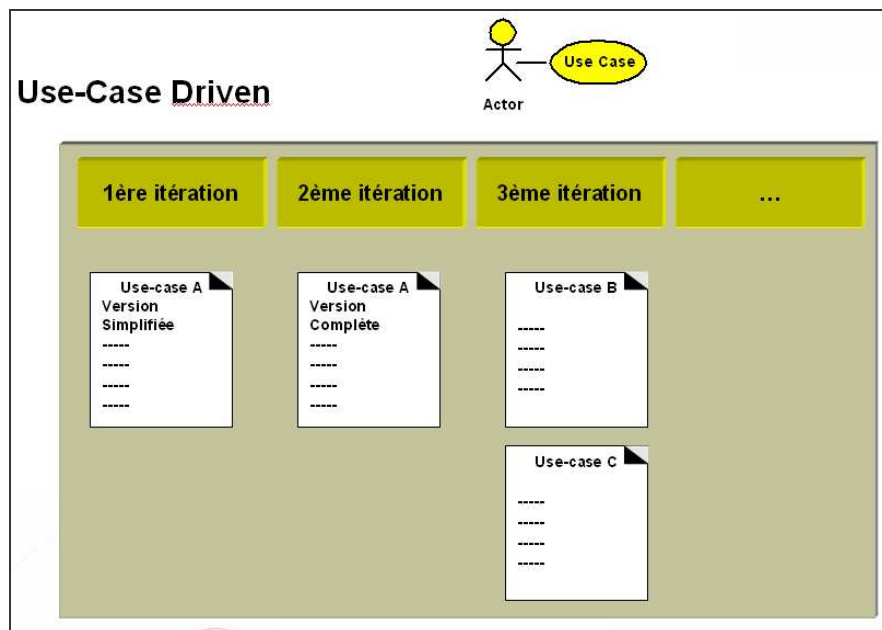


Figure 3 : Le RUP est « Use-Case Driven »

2.3 OpenUP

OpenUP insiste beaucoup moins que le RUP sur l'utilisation de la modélisation UML pour les livrables du projet. Cependant, OpenUP met également l'accent sur l'intérêt de la modélisation visuelle qui « augmente le niveau d'abstraction ».

Visual modeling is the use of semantically rich, graphical and textual design notations to capture software designs. A notation, such as UML, allows the level of abstraction to be raised, while maintaining rigorous syntax and semantics. In this way, it improves communication in the team as the design is formed and reviewed, allowing the reader to reason about the design, and it provides an unambiguous basis for implementation.

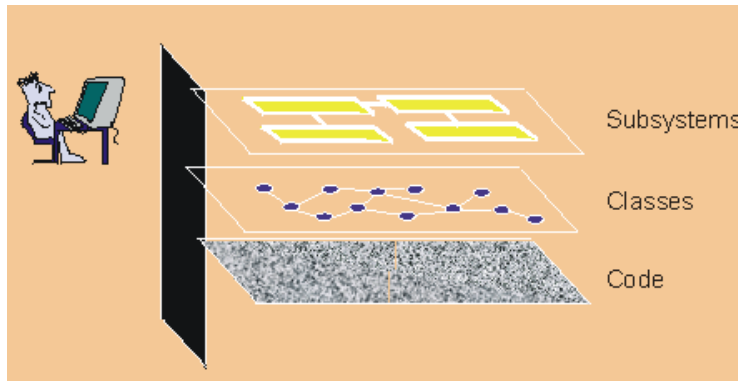


Figure 4 : La modélisation visuelle élève le niveau d'abstraction

Pour OpenUP, les modèles visuels aident à :

- Améliorer la compréhension des systèmes complexes
- Explorer et comparer des alternatives de conception à faible coût
- Former une fondation pour l'implémentation
- Capturer précisément les exigences
- Communiquer les décisions de façon non ambiguë

UML représente la convergence des meilleures pratiques en modélisation logicielle à travers l'industrie de la technologie objet.

2.4 XP

XP n'interdit pas les documents, mais pose dessus un regard critique ...

XP n'interdit pas la modélisation, mais rappelle que seul un code opérationnel est garant de qualité...

Si la création d'un modèle est une demande Client, alors il s'agit d'une « user story » à intégrer dans une ou plusieurs itérations, comme les autres ...

Alors, utiliser UML dans XP, pourquoi pas mais comment ?

- À la place des *CRC Cards* ?
 - Diagramme d'interaction
 - Diagramme de classes
- À la place des « *user stories* » ?
 - Cas d'utilisation
- Pour le client : modélisation métier
 - Diagrammes d'activité, de classes et d'états

Comme pour tout autre livrable, il faut se demander :

- Ce modèle est-il utile ? À qui ?
- Pour combien de temps ?
- Quelle est sa valeur ajoutée ?

2.5 Scrum

Scrum ne préconise aucune technique particulière d'ingénierie, mais :

- La modélisation agile est encouragée, au lieu d'une modélisation UML extensive
 - Les modèles aident à comprendre, ils ne sont pas considérés comme une documentation obligatoire
- Documentez le produit quand il est stable...
 - Si une documentation est requise, il vaut mieux la construire après que le code ait été testé (reverse engineering)

Certains experts préconisent :

- Un peu de modélisation au début de chaque itération
 - Pour mieux communiquer le but à tous les intervenants
- Un modèle métier lors de la réunion de planification en début de Sprint
 - Améliore la discussion sur les « user stories »

Tous encouragent :

- Une modélisation participative
 - Les modèles sont dessinés sur les murs (post-it géants, etc.), pendant des sessions de modélisation collaboratives



Figure 5 : Modélisation participative sur les murs ...

3. La modélisation agile

Scott Ambler est le grand promoteur de la « modélisation agile »².

Il a repris les valeurs et principes de XP en les adaptant à la modélisation.

3.1 Valeurs de la modélisation agile

- *Communication*
- *Simplicity*
- *Feedback*
- *Courage*
- *Humility*

Les clés d'une modélisation réussie consistent à promouvoir une communication efficace entre tous les participants du projet, de se forcer à développer la solution la plus simple possible qui satisfasse tous les besoins, à obtenir des retours rapides et fréquents, à avoir le courage de prendre des décisions et à s'y tenir, et à avoir l'humilité de reconnaître qu'on ne sait pas tout et que les autres ont une valeur à apporter à ses propres efforts.

3.2 Principes de la modélisation agile

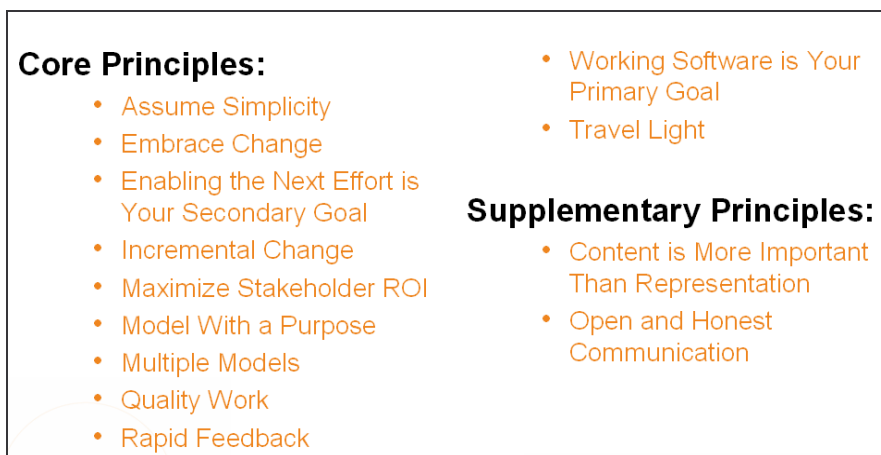


Figure 6 : Principes de la modélisation agile

Quelques précisions sur les principes : ils sont directement issus des principes XP, mais adaptés à la modélisation. Par exemple, l'importance de chercher la simplicité, de ne pas refuser le changement, mais au contraire de l'encourager, etc.

Il faut modéliser avec un but : si vous ne savez pas à quoi votre modèle va servir, ni ce qu'attend vraiment son auditoire, il ne sert à rien d'y travailler !

² Il a détaillé ses idées dans un livre : *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*, Scott Ambler, Wiley, 2002.

Il faut voyager léger : les modèles ne sont pas forcément à conserver et maintenir. Une fois qu'ils ont atteint leur but, jetez-en la plupart !

3.3 Pratiques de la modélisation agile

Core Practices	Supplementary Practices
<ul style="list-style-type: none"> • Active Stakeholder Participation • Apply the Right Artifact(s) • Collective Ownership • Create Several Models in Parallel • Create Simple Content • Depict Models Simply • Display Models Publicly • Iterate to Another Artifact • Model in Small Increments • Model With Others • Prove it With Code • Single Source Information • Use the Simplest Tools 	<ul style="list-style-type: none"> • Apply Modeling Standards • Apply Patterns Gently • Discard Temporary Models • Formalize Contract Models • Update Only When It Hurts

Figure 7 : Pratiques de la modélisation agile

Là encore, certaines pratiques sont directement issues de XP.

Commentons celles qui sont plus spécifiques à la modélisation :

- Il est important de créer plusieurs modèles en parallèle, d'utiliser le bon « artifact » : le diagramme le plus adapté à la situation, et de se tourner vers un autre diagramme dès qu'on se trouve bloqué pour continuer à avancer à une allure stable.
- Modélisez en petits incréments et avec les autres, au lieu d'essayer de créer le modèle « magique qui englobe tout » depuis votre tour d'ivoire ...
- Prouvez vos modèles avec du code pour montrer que vos idées marchent vraiment en pratique et pas seulement en théorie.
- Utilisez des notations simples afin que le contenu de vos modèles soit facile à valider, avec les outils les plus simples. Focalisez vous uniquement sur les aspects que vous devez modéliser et n'essayez pas de créer à tout prix un modèle très détaillé.
- Modélisez à plusieurs, montrez vos modèles publiquement sur les murs ou un site web, appliquez le principe XP de propriété collective.
- Ne vous embarrassez pas de modèles temporaires, et ne mettez à jour vos modèles que « quand ça fait mal » ...

3.4 La modélisation agile selon Craig Larman

L'un des meilleurs experts des méthodes agiles et de Scrum en particulier, notre collègue américain

Craig Larman³, résume ainsi comment il voit maintenant l'utilisation de la modélisation et donc d'UML :

Ne modélisez pas seul !

La modélisation doit être collaborative et participative

Des outils simples qui encouragent la créativité

Larman encourage l'utilisation de feutres et paperboards, plutôt que de progiciels coûteux et lourds

Des modèles multiples et en parallèle

Par exemple diagrammes de classes et de séquence pour faire une esquisse de conception objet

Les modèles ne sont pas de la documentation!

Nous modélisons pour avoir une conversation ensemble et pour développer une compréhension commune

Tout modèle est faux ! Et c'est OK

Par cet aphorisme provocant, Larman veut mettre l'accent sur l'aspect provisoire des modèles, le seul modèle qui fait foi à la fin du projet, c'est le code !

The code is the model!

L'organisation de la modélisation agile selon Larman peut être représentée ainsi :

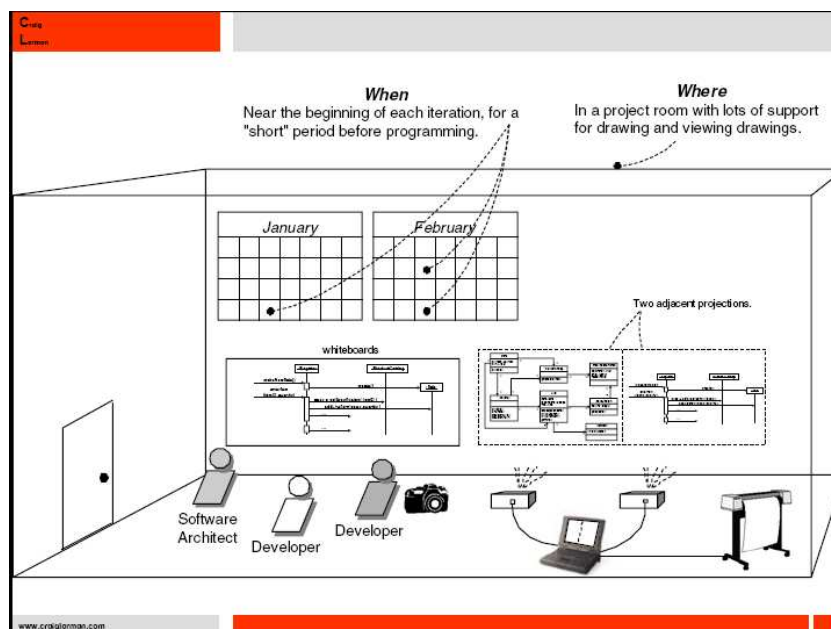


Figure 8 : La modélisation sur projet selon Craig Larman

³ Agile and Iterative Development: A Manager's Guide, Craig Larman, Addison-Wesley, 2003.

4. Conclusion

Finalement, la question de l'utilité d'UML au sein des méthodes agiles revient à poser celle de l'utilité de la modélisation !

Qu'est-ce qu'un modèle agile ?

- Il remplit son objectif et reste compréhensible
- Il est suffisamment précis, cohérent et détaillé
- Il est aussi simple que possible mais procure une valeur claire
- Exemples :
 - Use cases pour la vision du projet
 - Modèle du domaine pour la compréhension commune
 - Dépendances entre composants ou packages
 - Classes et interactions pour illustrer des *frameworks*
 - ...

Agile models are just barely enough! (S. Ambler)

A est un bon modèle de B si A permet de répondre de façon satisfaisante à des questions prédéfinies sur B (D.T. Ross)

Pour aller plus loin

Sites :

<http://www.agilemodeling.com/>

<http://www.agilealliance.com>

<http://www.craiglarman.com>

Blogs :

<http://blog.valtech.fr/>

<http://xp.thierrycros.net/>

<http://scrum.aubryconseil.com/>

<http://www.dotnetguru2.org/proques/>