

CHOISIR L'AGILITÉ

**Du développement logiciel
à la gouvernance**

Mathieu Boisvert

*Conseiller en adoption des méthodes agiles,
Scrum Master*

Sylvie Trudel

*Spécialiste en développement logiciel
et amélioration des processus,
Scrum Master*

DUNOD

Toutes les marques citées dans cet ouvrage sont des marques déposées par leurs propriétaires respectifs.

Illustration de couverture :
Plage des seychelles © Pat on stock - Fotolia.com

Le pictogramme qui figure ci-contre mérite une explication. Son objet est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit, particulièrement dans le domaine de l'édition technique et universitaire, le développement massif du photocopillage.

Le Code de la propriété intellectuelle du 1^{er} juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée dans les établissements

d'enseignement supérieur, provoquant une baisse brutale des achats de livres et de revues, au point que la possibilité même pour

les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée. Nous rappelons donc que toute reproduction, partielle ou totale, de la présente publication est interdite sans autorisation de l'auteur, de son éditeur ou du Centre français d'exploitation du droit de copie (CFC, 20, rue des Grands-Augustins, 75006 Paris).



© Dunod, Paris, 2011
ISBN 978-2-10-056874-1

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2° et 3° a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

Table des matières

Avant-propos	XI
Chapitre 1 – Qu’est-ce que l’agilité ?	1
1.1 Le choix de l’agilité	2
1.2 Les quatre valeurs de l’agilité	2
1.3 Les douze principes de l’agilité	3
1.4 Avantages et bienfaits	5
1.4.1 Avantages	5
1.4.2 Bienfaits	6
Chapitre 2 – Les méthodes agiles	9
2.1 Les méthodes agiles et leurs particularités	10
2.1.1 Scrum	10
2.1.2 XP (eXtreme Programming)	10
2.1.3 Lean/Kanban	11
2.1.4 Agile UP	11
2.1.5 Crystal	12
2.2 Les pratiques et livrables agiles	12
2.2.1 Sprint 0 (démarrage de projet)	12
2.2.2 Pratiques de gestion	12
2.2.3 Pratiques de développement	13

2.2.4 Livrables agiles	14
2.3 L'adoption partielle des pratiques	14
2.4 La combinaison des méthodes	15
2.5 Les ententes de travail	17
2.6 Préparer l'adoption de l'agilité	19
2.6.1 Les critères pour choisir un projet candidat	20
2.6.2 Projet en cours ou nouveau projet ?	21
Chapitre 3 – Démarrage d'un projet agile	23
3.1 La portée des activités de démarrage	24
3.1.1 Cycle de projet agile	24
3.1.2 Les objectifs des activités de démarrage	26
3.2 Former une équipe de démarrage	27
3.2.1 Les dirigeants	29
3.2.2 Le client, le responsable de produit, les experts d'affaires	29
3.2.3 Les architectes, les analystes, les ergonomes.....	30
3.2.4 L'équipe de projet	30
3.2.5 Le coach, chargé de projet ou Scrum Master	30
3.2.6 Les autres intervenants	31
3.3 Limiter les efforts requis	31
3.4 Rédiger une charte de projet	32
3.5 Construire un carnet de produit	35
3.5.1 Story mapping	36
3.5.2 Ordonner le carnet de produit	36
3.5.3 Calculer la valeur d'affaires des items du carnet	38
3.5.4 Détailler les exigences	41
3.6 Fixer la définition de terminé	41
3.7 Bâtir un plan de livraison	42
Chapitre 4 – Architecture incrémentale	45
4.1 L'architecte logiciel	46
4.2 L'autorité de l'architecte logiciel	46
4.3 La conception incrémentale comparée à l'analyse préliminaire détaillée	47

4.4	Objectifs d'une architecture agile	48
4.4.1	<i>Livrer une solution fonctionnelle</i>	48
4.4.2	<i>Maximiser la valeur d'affaires</i>	50
4.4.3	<i>Prévoir les prochains travaux</i>	51
4.4.4	<i>Répondre à des besoins d'affaires réels</i>	52
4.4.5	<i>Gérer le changement</i>	52
4.5	Documentation	53
Chapitre 5 – Équipes et livraison incrémentale		57
5.1	La planification	59
5.1.1	<i>La planification itérative</i>	59
5.1.2	<i>La longueur des itérations</i>	59
5.1.3	<i>Le carnet d'itération</i>	60
5.1.4	<i>La planification des travaux techniques</i>	61
5.1.5	<i>L'entretien du carnet de produit</i>	62
5.2	Les itérations	63
5.2.1	<i>La revue d'itération</i>	63
5.2.2	<i>La définition de terminé</i>	64
5.2.3	<i>La dette technique</i>	64
5.2.4	<i>La stratégie de tests</i>	65
5.3	L'introspection et l'amélioration continue	66
5.4	La mise en place des équipes matures et autogérées	68
Chapitre 6 – Gestion de projet		77
6.1	Les approches de gestion	78
6.1.1	<i>L'approche déterministe</i>	78
6.1.2	<i>L'approche empirique</i>	79
6.1.3	<i>Passer d'une approche déterministe à une approche empirique</i>	80
6.2	Mesurer l'avancement	82
6.2.1	<i>Les coûts et les délais</i>	83
6.2.2	<i>La qualité : levier ou contrainte ?</i>	83
6.2.3	<i>Les fonctionnalités et caractéristiques</i>	84
6.2.4	<i>L'atteinte des objectifs d'affaires</i>	84
6.2.5	<i>La valeur acquise</i>	85

6.3	Les répercussions sur les intervenants	86
6.3.1	<i>Transfert de responsabilité vers le responsable de produit</i>	86
6.3.2	<i>Transformation du rôle de chargé de projet</i>	87
6.3.3	<i>L'adaptation du bureau de projet</i>	90
6.4	La capitalisation	90
6.4.1	<i>La période de stabilisation</i>	93
6.5	La gestion des risques	94
6.5.1	<i>Budgéter l'atténuation et la contingence des risques</i>	94
6.6	Les livrables	95
6.6.1	<i>Le bilan d'itération</i>	95
6.6.2	<i>Le plan de livraison</i>	97
Chapitre 7	– Déploiement	107
7.1	La portée des activités de déploiement et de livraison	108
7.2	Les enjeux et les contraintes du déploiement	109
7.3	Les approches agiles au service des enjeux du déploiement	110
7.3.1	<i>L'intégration continue et la livraison automatisée</i>	110
7.3.2	<i>Les livraisons fréquentes</i>	111
7.4	Les répercussions sur la gestion du changement et la formation	112
Chapitre 8	– Infrastructure technologique	115
8.1	Le développement versus l'infrastructure	115
8.1.1	<i>Dans le coin gauche : les développeurs</i>	115
8.1.2	<i>Dans le coin droit : les responsables de l'infrastructure</i>	116
8.1.3	<i>Le bras de fer</i>	116
8.2	S'adapter aux impacts de l'agilité	116
8.2.1	<i>Les environnements de développement</i>	116
8.2.2	<i>L'interaction</i>	117
8.2.3	<i>Les livraisons fréquentes</i>	118
8.2.4	<i>La qualité production</i>	119
8.3	Les administrateurs de système : partie prenante de la solution d'affaires	120
8.4	Principes de l'infrastructure agile	120
8.4.1	<i>Simplicité</i>	120

8.4.2	Communication	121
8.4.3	Collaboration avec les clients	121
8.4.4	Processus	122
8.5	La méthode Lean/Kanban	123
8.5.1	La méthode agile la moins restrictive	124
8.5.2	Le tableau Kanban	125
8.5.3	La planification et la mesure de productivité	126
8.5.4	Le travail en cours et les goulots d'étranglement	127
8.5.5	La livraison	129
Chapitre 9 – Entretien et évolution des applications		131
9.1	Les incidents, les problèmes et les requêtes de modification	133
9.2	Les types de maintenance du logiciel	135
9.2.1	Les fausses demandes d'amélioration	136
9.3	Les mécanismes de gestion des requêtes et des problèmes en mode agile	137
9.3.1	La gestion par carnet de produit	138
9.3.2	Le « SWAT team »	138
9.3.3	La création d'un projet de maintenance	140
9.3.4	Le Kanban	141
9.4	La prévention des défauts	141
9.4.1	Amélioration continue	142
Chapitre 10 – Culture organisationnelle		147
10.1	Les types de culture	147
10.2	Connaître son type de culture	150
10.3	La culture influence le succès de l'agilité	152
10.3.1	Causes des échecs de l'agilité	152
10.3.2	La culture de préférence de l'agilité	153
10.3.3	Le savoir-être agile	154
10.3.4	Caractéristiques favorables et défavorables des cultures face à l'agilité	154
10.4	Gestion de la transition agile	158
10.4.1	La gestion du changement	159
10.4.2	L'équipe et son nouveau contexte opérationnel	161
10.4.3	Élaboration d'une stratégie	162

Chapitre 11 – Gouvernance	165
11.1 Agilité et alignement stratégique	166
11.1.1 <i>Compréhension commune de l'agilité</i>	166
11.2 Identifier les répercussions sur l'organisation	168
11.2.1 <i>Répercussions sur les objets de gouvernance</i>	168
11.3 Le positionnement de l'imputabilité vers les affaires	169
11.4 Les répercussions sur le rôle de gestionnaire	171
11.5 Les répercussions sur les structures en place	172
11.5.1 <i>Répercussion sur la structure organisationnelle</i>	173
11.5.2 <i>Répercussion sur la structure de gouvernance</i>	175
11.5.3 <i>La culture : levier ou frein à l'adoption de l'agilité ?</i>	175
11.6 La mise en œuvre de l'agilité	176
11.6.1 <i>La gestion de changement</i>	176
11.6.2 <i>Gérer les risques de l'adoption de l'agilité</i>	177
11.6.3 <i>Stratégies de transition agile</i>	178
11.7 Évaluation périodique de l'application des pratiques agiles	180
11.7.1 <i>Le PATH : un outil de diagnostic de l'agilité</i>	181
Chapitre 12 – Évolution des processus	183
12.1 Documenter son processus	184
12.1.1 <i>Portée de la documentation</i>	185
12.1.2 <i>Utiliser la documentation existante</i>	187
12.1.3 <i>Recommandations sur la manière de documenter</i>	188
12.2 Gestion du processus	189
12.2.1 <i>Amélioration continue</i>	189
12.2.2 <i>Assurance qualité</i>	191
12.2.3 <i>Diffusion du processus</i>	195
Chapitre 13 – Soutien au développement	197
13.1 La mission du soutien au développement	198
13.1.1 <i>Conséquences d'un manque de soutien au développement</i>	199
13.1.2 <i>Critères de succès pour les activités de soutien au développement</i>	201

13.2 Les pratiques du soutien au développement	201
13.2.1 Prodiguer la formation continue sur les nouvelles pratiques	202
13.2.2 Coacher sur les pratiques agiles	203
13.2.3 Gérer le processus	205
13.2.4 Obtenir des ressources aux équipes	207
13.3 Mise en œuvre d'un groupe de soutien au développement	208
13.3.1 Les formes d'un groupe de soutien au développement	209
13.3.2 Insertion du soutien au développement dans une structure existante	210
13.3.3 Taille requise d'un groupe de soutien au développement	210
13.3.4 Pièges à éviter	211
Chapitre 14 – Agilité et documentation	213
14.1 Les besoins liés à la documentation	213
14.1.1 Comprendre ses besoins en information	213
14.1.2 La documentation : un outil de communication	214
14.1.3 Le syndrome de la « taille unique »	218
14.2 Les formes de documentation	218
14.3 Retrouver l'information efficacement	220
14.4 Exemples de documentation dans des contextes spécifiques	222
14.4.1 La modélisation	222
14.4.2 La conformité à des normes, lois ou règlements	224
Chapitre 15 – Mesures de performance	227
15.1 Concepts clés de la mesure	228
15.1.1 Processus de mesure	230
15.1.2 Principes essentiels	234
15.2 Mesurer la productivité du processus	234
15.2.1 Pourquoi mesurer la productivité du processus de développement ?	236
15.2.2 Comment mesurer la productivité et autres indicateurs similaires ?	237
15.2.3 La méthode COSMIC	240
15.2.4 Démarche de mise en œuvre	241
15.2.5 Estimation préliminaire des projets de développement	245
15.2.6 Étalonnage : comparer sa productivité objectivement	245
15.2.7 Autres bénéfices à mesurer la productivité	246

15.3 Mesurer la qualité relative des applications	246
15.4 Autres indicateurs utiles	247
15.5 L'aspect économique des mesures	248
15.6 La face cachée des mesures	249
Chapitre 16 – Agilité et CMMI	251
16.1 Le CMMI	252
16.1.1 Qu'est-ce que le CMMI ?	252
16.1.2 Ce que le CMMI n'est pas	252
16.1.3 D'où vient le CMMI ?	253
16.1.4 Pourquoi utiliser le CMMI ?	256
16.1.5 Les représentations du CMMI	256
16.2 Mythes liés à l'application du CMMI	256
16.3 Arrimage de l'agilité et du CMMI	261
16.3.1 Pourquoi vouloir arrimer CMMI et agilité ?	262
16.3.2 Arrimage des pratiques agiles avec celles du CMMI	262
16.4 Guide pour arrimer capacité du processus organisationnel et agilité	265
Chapitre 17 – Gestion des individus	267
17.1 Répercussions sur les individus	268
17.1.1 Les supporteurs de l'agilité	268
17.1.2 Les détracteurs de l'agilité	269
17.1.3 Les membres dysfonctionnels des équipes	272
17.1.4 Gestion des membres dysfonctionnels par l'équipe	275
17.2 Impacts sur les pratiques de la gestion des ressources humaines	275
17.2.1 Description de poste	276
17.2.2 Plan de carrière	280
17.2.3 Implication des pairs dans les processus de la GRH	280
Conclusion	283
Références bibliographiques	285
Index	293

Avant-propos

CE QUI NOUS A AMENÉS À ÉCRIRE CE LIVRE

Les méthodes agiles sont dites légères, car elles sont moins prescriptives que celles dites traditionnelles. Elles sont fondées sur un manifeste composé de quatre valeurs et douze principes et elles ne définissent que très peu de rôles, de livrables et de pratiques (Figure 1¹).

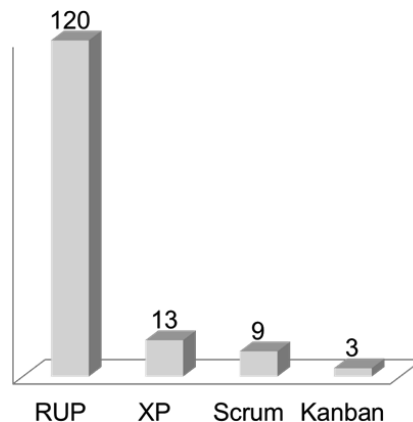


Figure 1 — Comparaison du niveau prescriptif des méthodes, basé sur le nombre de rôles, de livrables et pratiques définis par méthode

Plusieurs praticiens apprécient le caractère léger des méthodes agiles, car elles s'adaptent bien aux différents contextes des projets de développement logiciel. En 2004, après trois années comme développeur dans une institution bancaire française, Mathieu a découvert la méthode Scrum. Il a apprécié les méthodes agiles qui lui ont permis de se concentrer sur la réalisation de produits de qualité, avec une

1. Tiré de l'ouvrage *Kanban and Scrum - making the most of both* d'Henrik Kniberg et Mattias Skarin, <http://www.infoq.com/minibooks/kanban-scrum-minibook>, consulté en décembre 2009.

cadence de livraison de plus en plus prévisible, en collaboration avec son équipe de développement. D'équipes en équipes, il a cependant découvert les limites du travail au sein d'un noyau agile dans une entreprise qui ne l'est pas. Devenu Scrum Master, Mathieu s'est intéressé aux autres méthodes et à l'approche agile en général, ce qui lui a permis d'aider son équipe à franchir ces frontières et de sensibiliser les organisations aux bienfaits des méthodes agiles. Son expérience et son succès auprès de différentes organisations et plusieurs organismes l'amène aujourd'hui à conseiller au démarrage et à la transition aux méthodes agiles.

Le parcours de Sylvie diffère de celui de Mathieu, forte de ses 26 ans d'expérience en développement logiciel. Après avoir été programmeur-analyste, analyste fonctionnel et architecte fonctionnel pendant onze ans, elle a fait un virage vers les processus de développement et de maintenance logiciel au milieu des années quatre-vingt-dix, dans une industrie où le CMM (*Capability Maturity Model for Software*) omniprésent allait devenir un modèle à suivre *de facto*. Au début des années 2000, Sylvie a accompagné des organisations, petites et grandes, à améliorer leur processus de développement et à en mesurer la performance sous forme de coût unitaire, en heures par point de fonction COSMIC (*Common Software Measurement International Consortium*, <http://www.cosmicon.com>). En parallèle, elle a obtenu sa certification de Scrum Master qu'elle utilise en industrie pour transformer les façons de faire du logiciel. En appliquant l'agilité, plusieurs des organisations qu'elle a conseillées ont réussi à réduire de moitié leur coût unitaire moyen (mesuré objectivement), comparativement au coût unitaire moyen de leur méthodologie traditionnelle. Elle a été la première, en septembre 2006, à présenter le mariage des pratiques agiles et du CMMI lors d'un séminaire du SPIN de Montréal (*Software Process Improvement Network*), puis à la Fédération de l'Informatique du Québec (maintenant appelé le *Réseau Action TI*) le mois suivant, en collaboration avec François Beauregard, le premier instructeur Scrum francophone certifié.

Les avantages proposés par les méthodes dites « légères » plaisent aux gestionnaires et les clients que nous avons rencontrés se sont souvent montrés collaboratifs pour modifier leur cadre méthodologique de développement traditionnel.

La plupart des clients que nous avons conseillés ont été réceptifs aux méthodes légères. Ils ont accepté de modifier leur processus de développement pour profiter des avantages offerts par les méthodes agiles :

- qualité améliorée ;
- délais de mise en marché réduits ;
- engagements tenus par les équipes ;
- productivité améliorée ;
- et plusieurs autres avantages.

Aujourd'hui, le marché est friand de l'agilité et les mandats d'accompagnement sont plus ambitieux. Nous accompagnons maintenant des organisations à faire une transition agile complète : du coaching jusqu'à la rédaction du cadre méthodologique. Au moment d'écrire ce livre, Mathieu constate l'étendue des besoins dans une organisation d'envergure où les gestionnaires et les membres d'équipes doivent

développer, comprendre et communiquer de nouveaux processus. Sylvie le faisait depuis plus de quinze ans de façon générale et depuis cinq ans dans un contexte de transition agile. Jusqu'à maintenant, nos clients étaient bons joueurs et conservaient la responsabilité d'adapter leurs processus organisationnels.

Nous avons compris depuis longtemps que les méthodes agiles n'étaient pas vraiment des méthodes, mais qu'elles sont plutôt des philosophies de travail demandant énormément de rigueur pour être efficaces. Cependant, pour changer la culture d'une entreprise de quelques centaines d'employés, la rigueur ne suffit pas. Les personnes impliquées ont besoin d'un cadre pour évoluer dans une organisation où chacun se demandera : Quel est mon rôle ? Quelles sont mes responsabilités ? Qui consomme mes livrables ? Qui sont mes parties prenantes ?

Les individus sont prêts à être agiles lorsqu'ils sont informés. C'est une erreur que de répondre à ces questions en utilisant seulement la littérature agile. Lors d'une d'intervention en clientèle, nous, Sylvie, Mathieu et Elsa Larreur, une collègue et Scrum Master d'expérience, avons eu l'opportunité d'associer le langage traditionnel ainsi que les principes des processus organisationnels avec les concepts des approches agiles. Nous avons traduit les responsabilités des rôles de Scrum et des membres d'une organisation avec un RACI (Responsable, Approbateur, Contributeur, tenu Informé). Nous avons expliqué le processus de l'amélioration continue agile à l'aide d'un modèle de gestion de la qualité PDCA (*Plan-Do-Check-Act*). Nous avons démontré que les points d'effort ne sont pas les seules façons d'estimer l'ampleur d'un projet et que le CMMI (*Capability Maturity Model Integration*) n'est pas l'ennemi des méthodes agiles.

La littérature traitant des méthodes agiles donne à tort l'impression qu'elles ne sont pas compatibles avec les processus déjà présents dans les organisations. Au-delà d'être persuadé des bienfaits de l'agilité, vous devez savoir qu'il y aura des impacts certains dans TOUS les niveaux de l'organisation. Cela aura pour effet d'amener votre organisation à revoir son cadre méthodologique (les processus organisationnels).

TERMINOLOGIE : COMMENT S'Y RETROUVER ?

Exemple de conversation entendue lors de nos interventions :

Alain : T'es au courant du nouveau truc qu'on va mettre en place ? C'est la méthode « agile ».

Marie-Jeanne : « agile », ce n'est pas une méthode, c'est une approche. Veux-tu parler des méthodes Scrum et XP ?

Alain : Oui c'est ça, Scrum et XP. Et en plus, il paraît qu'on poursuit avec la démarche « CMMI ». Je me demande comment on va arrimer toutes ces méthodes. Le CMMI, c'est une méthode, non ?

MJ : Non, c'est un modèle.

Alain : Ah bon ! ? Est-ce la même chose qu'une norme ?

MJ : Non, ce n'est pas une norme. Il contient des objectifs et des pratiques.

Alain : Est-ce un ensemble de techniques ?

MJ : Non, c'est plutôt un ensemble de pratiques reconnues comme étant les « meilleures ».

Alain : Ah, mais c'est un processus !

MJ : Non plus. Un processus est caractérisé par des flux d'opérations, un système d'activités qui utilise des ressources pour transformer des éléments d'entrée en éléments de sortie¹.

Alain : Mais n'est-ce pas ce qu'on appelle une « procédure » ?

MJ : Non, car une procédure est une description pas à pas d'une opération ou d'une partie d'opération.

Alain : Dans ce cas, parle-t-on de méthodologie ?

MJ : Pas du tout. Une méthodologie est une démarche suivie, une manière de procéder².

Alain : Qu'est-ce qu'un modèle alors ?

Avant de vous lancer dans votre lecture, nous aimerions vous inviter à revoir la définition de certains termes utilisés dans le présent ouvrage. La confusion des termes et les questionnements du précédent dialogue sont fréquents dans les organisations qui démarrent une démarche d'amélioration de processus. C'est pourquoi il importe de préciser chacun des termes pour éviter toute confusion, non pas à partir d'une simple définition trouvée dans un dictionnaire mais expliqué de manière concrète et étoffé d'exemples. Pour en faciliter la lecture, nous les avons décrits ci-après en ordre alphabétique.

Approche

Une approche est « *une manière d'aborder une question, un problème* »³, ce qui est généralement guidé par des principes et des valeurs. Une approche ne précise pas « quoi » faire, ni « comment faire ». Donc, quand on parle de « l'agilité » ou « d'agile », on parle alors d'approche de l'agilité ou d'approche agile. Une approche pourra être mise en œuvre par des méthodes qui respectent les principes et les valeurs qui définissent cette approche.

Méthode

Une méthode est une façon de faire. Elle décrit « *comment* » on fait les choses. En général, une méthode traite d'un aspect précis faisant partie d'un processus plus large ou d'une méthodologie. Voici quelques exemples :

- une méthode de gestion de projet et d'exigences (ex. Scrum) ;
- une méthode de définition d'exigences via des tests unitaires (ex. TDD) ;

1. Source : Norme ISO 9000:2000, Systèmes de management de la qualité – Principes essentiels et vocabulaire, Organisation Internationale de Normalisation, 2000.

2. Source : <http://www.linternaute.com/dictionnaire/fr/definition/methodologie/>, consulté en juillet 2010.

3. Source : <http://www.linternaute.com/dictionnaire/fr/definition/approche/>, consulté en juillet 2010.

- une méthode d'estimation de projet par consensus (ex. Poker planning) ;
- une méthode de revue de code en continu (ex. programmation par paire, appelée aussi binôme) ;
- une méthode de restructuration de code.

Ces exemples de méthodes traitent donc d'une opération précise ou d'un ensemble d'opérations pouvant faire partie d'un processus de développement et d'évolution du logiciel. Synonyme de « *technique* ».

Méthodologie

On trouve plusieurs définitions du mot « *méthodologie* » sur Internet et dans les dictionnaires, mais celle qui ressemble le plus à l'utilisation qu'on en fait dans le contexte du développement logiciel est un « *ensemble des méthodes appliquées à un domaine particulier* »¹.

Donc, un ensemble de méthodes que l'on peut agencer selon nos besoins organisationnels ou de projet pour former un processus. Une méthodologie pourrait s'apparenter à un menu de restaurant : nous utiliserons plusieurs items de son contenu mais pas forcément l'entièreté. Par exemple, une méthodologie pourrait fournir diverses méthodes pour documenter les exigences : cas d'utilisation, scénarios utilisateurs (*user stories*), méthodes formelles, ou langage naturel. Cependant, au démarrage d'un projet, il faudra décider laquelle de ces méthodes sera celle adoptée par le projet puisqu'il est souvent peu probable qu'un projet en utilise plusieurs. C'est souvent le cas à l'échelle organisationnelle, alors qu'on préfère limiter le nombre de méthodes à utiliser de façon à pouvoir réduire la courbe d'apprentissage des intervenants participant à divers projets.

Certaines méthodologies ont été ou sont populaires : Merise (à une certaine époque), Macroscopie (surtout au Québec), *Rational Unified Process* (RUP) qui, contrairement à ce que son nom laisse penser, n'est pas un processus mais une méthodologie.

Modèle

Un modèle est une « *représentation théorique d'un système d'éléments et de relations plus ou moins complexes* »². En développement logiciel, on parle de modèle de « meilleures pratiques » tel que le CMMI, mais aussi de modèle de données, modèle objets, et modèle de traitements qui sont des représentations d'un aspect du logiciel qu'ils décrivent.

« *Tous les modèles sont faux, quelques-uns sont utiles.* »

Georges Box³

1. Source : <http://dictionnaire.reverso.net/francais-definition/methodologie> , consulté en juillet 2010.

2. Source : <http://www.mediadico.com/dictionnaire/definition/Modele/1> , consulté en juillet 2010.

3. Traduction libre de : http://en.wikiquote.org/wiki/George_Box, consulté en février 2011.