ÉTUDES, DÉVELOPPEMENT & INTÉGRATION

CHOISIR L'AGILITÉ

Du développement logiciel à la gouvernance

Mathieu Boisvert
Sylvie Trudel

DUNOD

Table des matières

3

Démarrage d'un projet agile

Objectif

Dans ce chapitre, vous découvrirez pourquoi la phase de démarrage est essentielle au succès d'un projet en mode agile, comment se découpe cette phase en différents ateliers, quels sont les livrables attendus, qui devrait y prendre part et quelles sont les conditions d'arrêt de façon à pouvoir commencer la première itération d'un projet.

Mise en situation : François, Vincent et le démarrage d'un projet

Vincent est le développeur senior de son équipe de développement. Avec François, un chargé de projet, il participe à une formation d'introduction au rôle de Scrum Master. Tous les deux sont employés dans une compagnie d'assurance et ils sont affectés à un prochain projet pilote ayant pour objectif de vérifier les bienfaits des méthodes agiles dans l'organisation et de vérifier si ces méthodes sont compatibles avec les processus de développement de la compagnie.

Le projet consiste à développer un outil pour permettre aux clients corporatifs de faire eux-mêmes les demandes des réclamations et d'inscrire leurs employés à l'assurance collective. La compagnie a terminé son plan de mise en marché et elle est prête à démarrer dans les deux prochaines semaines. La compagnie a investi dans une équipe de projet composée d'un responsable de produit, un Scrum Master, un analyste d'affaires et quatre développeurs qui travailleront au développement du système pendant neuf mois. Un mois supplémentaire a été prévu pour la période de stabilisation et la livraison du système.

En tant que chargé de projet, François, avec l'aide de Vincent, doit préparer le démarrage du projet : réunir l'équipe pour les ateliers de démarrage, prendre connaissance de l'énoncé de projet et planifier la réalisation du cahier des charges.

Tous deux ont compris que le carnet de produit est un aspect important de la planification dans la méthode Scrum. Mais comment faire pour transformer le cahier

des charges en carnet de produit ? Comment construire un plan de livraison à partir d'un carnet de produit ? Et à quel moment les spécialistes devront-ils être invités ?

François : On devrait peut-être se construire des ateliers à partir des préalables de la méthode Scrum : un carnet de produit priorisé, un responsable de produit, une équipe de projet, un Scrum Master, puis s'entendre sur une définition de terminé et la durée des itérations.

Vincent : Je suis d'accord. Scrum est une méthode incrémentale et il faut éviter de faire l'analyse détaillée complète du cahier des charges pour commencer. Je propose qu'on limite nos ateliers de démarrage à l'objectif « d'être prêt à planifier notre première itération ».

François : Limitons nos efforts et démarrons une première itération Mais mon expérience de chargé de projet me recommande d'en faire un peu plus. Je pense que l'organisation serait plus confortable si notre projet diffusait un plan de livraison avant le démarrage.

Vincent : Ça me va, mais comment faire un plan de livraison sans faire une analyse détaillée ?

3.1 LA PORTÉE DES ACTIVITÉS DE DÉMARRAGE

Les méthodes agiles préconisent une approche incrémentale, soit le développement de logiciel par incréments. Pour être en mesure de livrer une solution logicielle de manière incrémentale, il faut que les disciplines du développement logiciel soient exécutées en parallèle plutôt qu'en séquence. Chaque itération comporte donc un certain lot d'activités de planification, d'analyse, de conception, d'écriture de code, de tests et de livraison.

Si avec une approche agile les activités sont distribuées à l'intérieur du projet, il peut être déduit que les activités de démarrage requerront moins d'efforts. Nous ne disons pas que la période de démarrage est inutile sous une approche agile. Nous affirmons cependant qu'il est possible de revoir les livrables préalables et limiter les efforts requis dans le but de commencer le développement le plus tôt possible.

3.1.1 Cycle de projet agile

La période de démarrage est la première étape d'un projet de développement. Selon la méthode *Agile Unified Process*[1] (Agile UP), le cycle de développement d'un projet se divise en quatre phases :

- l'évaluation[2];
- l'élaboration ;
- la construction ;
- la transition.

1. http://www.ambysoft.com/unifiedprocess/agileUP.html consulté en mars 2011.
2. Traduction libre du terme anglais « *inception* »

La phase d'**évaluation** a pour objectif de produire les livrables relatifs à la portée du projet, comme la charte de projet, l'architecture de haut niveau, un cahier de spécifications et un plan de livraison initial. Au terme de cette phase, les parties prenantes devraient être en mesure d'évaluer si un projet doit se poursuivre ou s'arrêter. Lorsqu'un projet franchit l'acceptation de la phase d'évaluation, l'objectif de la phase de l'**élaboration** est de gérer les éléments de risque du projet : le chemin critique fonctionnel, l'architecture, l'environnement de développement. Avec des risques mitigés, la phase de **construction**, se consacrera à la réalisation productive du système. Lorsque le projet est prêt pour une livraison finale, le projet sera alors dans sa phase de **transition,** qui comprend la stabilisation et le déploiement.

La Figure 3.1 illustre comment, avec une approche agile, la proportion des efforts requis pour chacune des disciplines du développement logiciel n'est pas la même à chacune des phases du cycle du projet. Par exemple, les efforts de modélisation sont concentrés dans les phases d'évaluation et d'élaboration alors que la programmation est concentrée à la phase de construction. Cependant, aucune des disciplines n'est dédiée qu'à une seule phase et les efforts leur étant consacrés varient selon le déroulement du projet.

La représentation graphique de la distribution des efforts par discipline démontre que l'objectif de la phase de l'évaluation ne conclut pas les activités d'analyses. Il faut donc accepter, qu'au terme de la phase de l'évaluation, il restera plusieurs incertitudes liées au projet comme :

- Comment utiliser les technologies mal maîtrisées, parce qu'elles sont nouvelles ou désuètes ?
- Comment utiliser les nouveaux outils de développement ?
- Quels sont les inconnus et les points de litige du domaine d'affaires ?
- Comment collaborer avec les équipes externes et intégrer leurs travaux au reste du projet ?

Ces incertitudes nécessiteront de prévoir du temps à l'intérieur des différentes phases pour développer des preuves de concept qui viendront valider les hypothèses brutes de l'analyse, de la conception et de la programmation. Dans le cas où ces hypothèses s'avèrent invalides, un projet itératif permet d'introduire des alternatives sans nuire à la planification. Le principe est de repousser la plupart des décisions au « dernier moment responsable », évitant ainsi d'accumuler un inventaire d'hypothèses dont l'exactitude n'est pas prouvée au départ d'un projet.

L'accumulation d'un inventaire de travaux qui ne sont pas validés est la première cause de l'effet tunnel. Celui-ci survient lorsque la visibilité sur le déroulement du projet se perd pendant plusieurs semaines, voire des mois. Le résultat de l'effet tunnel est souvent le développement d'un produit inutile, mal ciblé, ne répondant pas aux besoins du client et demandant un échéancier et un budget plus grands qu'initialement prévu. Pour éviter ce risque, une synchronisation est effectuée au terme de chacune des itérations, permettant de démontrer l'avancement du produit et d'ajuster le plan, même pendant les phases d'évaluation et d'élaboration.

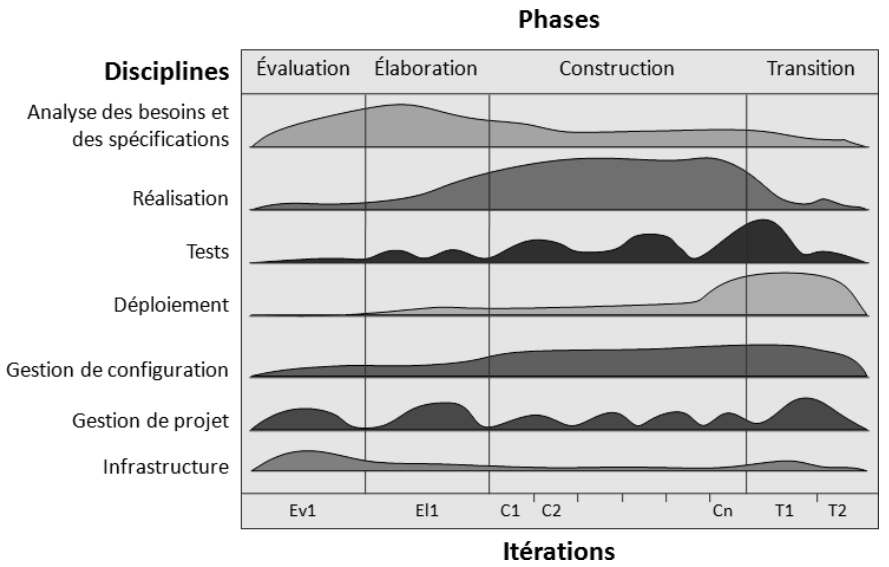


Figure 3.1 — Disciplines à travers les phases de l'*Agile Unified Process*.

Certains affirment qu'il est difficile de distribuer l'analyse et la conception à l'intérieur des itérations, parce qu'elles sont trop courtes pour démontrer des résultats. À notre avis, les équipes de projets ont souvent cette perception lorsqu'ils traitent des problèmes très complexes. Ils ont alors le réflexe d'étirer les ateliers de démarrage pour répondre à cette complexité. Plus souvent qu'autrement, la complexité n'est pas la racine du problème. Ce serait plutôt le manque d'expérience à aborder les projets par petits morceaux et par ordre de priorité.

3.1.2 Les objectifs des activités de démarrage

Les objectifs des ateliers de démarrage sont donc les mêmes que ceux de la phase de l'évaluation, soit :

- fournir suffisamment d'information aux parties prenantes responsables d'autoriser la poursuite du projet ;
- produire les livrables requis pour pouvoir démarrer la phase de l'élaboration.
- C'est également une période pour :
- **communiquer la vision**, les objectifs, les enjeux, les contraintes, les risques et les conditions de succès du projet ;
- **dispenser de la formation** aux équipes sur les processus et les méthodes utilisés dans le projet, même à ceux connaissant déjà le sujet. L'objectif est que tous les contributeurs partagent la même connaissance ;
- **poursuivre l'analyse et la conception** : prévoir des moments individuels pour que les intervenants recueillent des informations manquantes et fassent quelques

preuves de concepts pour réduire les incertitudes inhérentes aux premières itérations ;

- **construire une équipe de projet** : les méthodes agiles prônent la collaboration et l'engagement. Un bon objectif serait que tous les contributeurs connaissent leur rôle et leurs responsabilités au commencement du projet.

Selon notre expérience, ces objectifs de démarrage sont atteints à travers quatre grandes activités, soit :

- la formation d'une équipe de démarrage ;
- la rédaction d'une charte de projet ;
- l'élaboration d'un carnet de produit ;
- la préparation d'un plan de livraison.

3.2 FORMER UNE ÉQUIPE DE DÉMARRAGE

Qui devrait participer aux activités de démarrage ? Le plus de gens possible ! Les architectes, les analystes, le marketing, les experts d'affaires, l'ergonome, les développeurs, le vice-président, etc. Un projet de développement est un investissement et il est important que tous les intervenants sachent de quoi il en retourne. Pour augmenter l'engagement de tous les collaborateurs autour du succès du projet, il est important qu'ils interagissent tôt ensemble et qu'ils s'informent, avec le moins d'intermédiaires possibles, des enjeux et des objectifs du projet. Une grande participation augmente le sentiment d'appartenance général envers le projet et encourage les initiatives au quotidien qui ne dépendront plus d'un seul responsable.

Chaque intervenant ayant un rôle différent, certains participants pourront quitter avant la fin des ateliers. Seule l'équipe de réalisation, le coach d'équipe et le client[1] sont tenus de rester jusqu'à la toute fin de la période de démarrage. L'idée n'est pas d'inviter n'importe quel curieux ou de faire perdre du temps à l'organisation, mais plutôt de réunir les intérêts et les préoccupations de toutes les personnes concernées par le projet.

Si vous êtes préoccupés que le nombre d'individus réduise l'efficacité des rencontres, nous vous invitons à sélectionner les meilleurs intervenants pour chaque activité et à diviser les participants en sous-groupes selon le sujet des ateliers.

1. Pour un souci de simplification, nous utiliserons le terme « *client* » dans ce chapitre lorsque nous voulons décrire la personne experte du domaine d'affaires responsable de la spécification et de la planification du projet. Nous utiliserons le terme « *coach* » lorsque nous voulons décrire la personne qui accompagne et aide l'équipe à devenir responsable de la gestion de projet de manière efficace et collaborative.

Retour d'expérience : Identifier la contribution des participants

En tant que coach agile, j'ai régulièrement l'occasion d'animer les ateliers de démarrage des équipes qui débutent avec les approches agiles. Quand je me retrouve devant dix personnes et plus, j'aime bien commencer avec l'activité du « diagramme des rôles ».

Je demande à tous les participants d'inscrire leur nom sur un Post-it™ et de le placer sur un diagramme des rôles comme celui de la Figure 3.2.

Les participants choisissent de placer leur nom du côté affaires ou du côté informatique, selon leur expertise et la nature de leur rôle. Ceux qui s'engageront d'itération en itération placent leur nom au centre, car ils sont membres de l'équipe de projet.

Ce simple exercice me permet de discuter avec les intervenants sur leur contribution envers le projet. Souvent, l'exercice révèle des zones floues :

– certains ne savent pas où placer leur nom, parce qu'ils ne connaissent toujours pas leur rôle,

– certains placent leur nom sur une frontière, parce qu'ils voudraient s'engager, mais savent bien qu'ils en seront incapables avec toutes leurs affectations,

– parfois, l'équipe informatique est complètement absente, ou encore le client n'est pas présent.

Lorsque les rôles ne sont pas clairs, j'ajoute aux objectifs des ateliers de démarrage des discussions pour clarifier les rôles. S'il manque des membres de l'équipe de projet, j'ajourne les ateliers le temps de remédier à la situation.

Bien entendu, on peut utiliser n'importe quel autre type d'atelier pour clarifier la position de chaque rôle, mais je ne veux pas laisser planer le doute sur le rôle et les responsabilités des participants pendant les ateliers de démarrage.

Mathieu

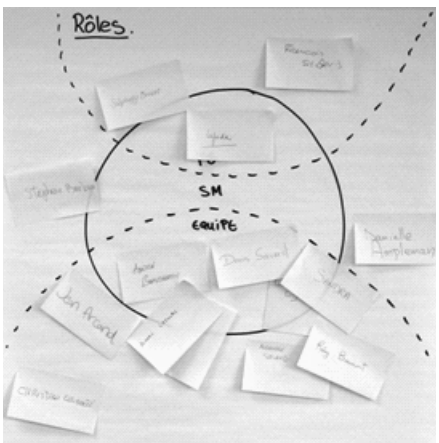


Figure 3.2 — Exemple de diagramme des rôles.

3.2.1 Les dirigeants

Les dirigeants sont un bon exemple d'intervenants ne participant pas toujours à la totalité des ateliers. Ils sont invités pour communiquer la vision et les enjeux au reste de l'équipe de projet. Il ne faut pas sous-estimer le message fort que la présence d'un dirigeant apporte aux ateliers de démarrage. Nous en avons rencontré plusieurs réticents à participer aux activités de démarrage, jugeant que leur présence serait interprétée comme de l'ingérence. Nous sommes d'avis que si un projet est stratégique, au point d'investir des sommes considérables dans son développement, les dirigeants se doivent de partager les attentes et l'intérêt suscités par le projet. Cette présence sera perçue comme du respect envers l'équipe et un signal fort de l'importance du projet.

Nous aimons commencer les ateliers en leur donnant la parole : leur discours n'est jamais bien long et il résume la raison d'être du projet. Parce que leur discours est souvent concis et ciblé, il n'est pas rare qu'on y entende LA phrase qui résume le projet dans la charte de projet. Si aucun dirigeant n'est disponible, le client se chargera d'introduire le projet au début des ateliers.

3.2.2 Le client, le responsable de produit, les experts d'affaires

> **Valeur agile** : Collaboration avec le client plutôt que négociation de contrat.

Le rôle du client dépasse celui de référent expert du domaine d'affaires. Son rôle est critique au succès du projet et ses responsabilités s'approchent de celles du gestionnaire de produit traditionnel. C'est lui qui doit planifier l'exécution du projet, collaborer étroitement et fréquemment avec l'équipe de projet et diffuser l'état d'avancement du produit au reste de l'organisation. Les ateliers de démarrage sont un moment propice pour valider que le client assigné est le bon candidat. Quelques symptômes de dysfonctionnement ne mentent pas :

- le client n'est pas disponible pour se présenter aux ateliers de démarrage ;
- il n'est pas en mesure d'expliquer la portée du projet ;
- il ne peut prendre de décision sans systématiquement aller vérifier avec ses pairs ;
- sa vision se limite à son expérience personnelle et ne tient pas compte des intérêts des autres parties prenantes du produit.

Un bon candidat est en mesure de s'affirmer lorsqu'il est interrogé au sujet des enjeux d'affaires du projet.

Même si le client se doit d'être un expert d'affaires le plus polyvalent et autonome possible, il n'est pas nécessaire qu'il soit un expert de tous les domaines d'affaires touchés par le projet. D'autres experts peuvent s'adjoindre à lui pour compléter ses compétences et ainsi faciliter la fluidité des ateliers de démarrage. Parmi les experts d'affaires, nous pensons notamment aux utilisateurs experts, aux analystes d'affaires ou métier, aux pilotes de systèmes et aux membres du service à la clientèle.

3.2.3 Les architectes, les analystes, les ergonomes...

L'analyse et la conception occupent une grande place des activités de démarrage et c'est pourquoi nous invitons les spécialistes à participer et à communiquer au reste de l'équipe l'architecture et la modélisation envisagées pour le projet. Cependant, les spécialistes doivent être vigilants à ne pas faire une analyse préliminaire détaillée du système à ce stade-ci. Au terme des ateliers de démarrage, les dossiers de conception et d'architecture doivent être juste suffisants[1] pour être en mesure d'estimer les exigences brutes du produit : un diagramme de contexte, les processus d'affaires documentés à haut niveau et le schéma des données d'affaires peuvent suffire. Si les besoins d'analyse sont plus grands, rappelons qu'il est envisagé d'étaler les travaux au-delà de la phase d'élaboration du projet.

> **Principe agile** : Les meilleures architectures, spécifications et conceptions sont issues d'équipes qui s'auto-organisent.

Les approches agiles préconisent que la conception et l'analyse d'un système relèvent de la responsabilité de l'équipe de projet. Même si l'objectif est de former des équipes multidisciplinaires, cela ne veut pas dire que les spécialistes sont inutiles et exclus. Les ateliers de démarrage vont permettre de clarifier le rôle des spécialistes au sein de l'équipe de projet et mesurer l'impact de l'adoption de l'agilité sur leur méthode de travail[2].

3.2.4 L'équipe de projet

Pendant la période du démarrage, l'équipe de projet est amenée à comprendre le projet, découvrir le matériel des spécialistes, estimer les exigences du produit et participer à la rédaction de la définition de terminé.

Le démarrage d'un projet coïncide souvent avec la constitution d'une nouvelle équipe de projet. La participation active des membres de l'équipe aux ateliers catalyse la construction d'une équipe performante. Les ateliers sont également une occasion pour vérifier que tous les membres de l'équipe ont une motivation à collaborer. Si un développeur pense que sa contribution n'est pas utile, il faut lui faire comprendre qu'au terme des ateliers, lui et ses coéquipiers doivent se sentir suffisamment en confiance pour s'engager auprès du client et réaliser une première itération.

3.2.5 Le coach, chargé de projet ou Scrum Master

L'agilité fait la promotion des équipes autonomes et des individus engagés. Une équipe de projet travaillant étroitement avec son client est un groupe autonome et pluridisciplinaire pouvant mener son projet au succès.

---

1. En référence au terme « *just enough* » de la littérature anglaise sur le Lean.
2. Le chapitre *4 Architecture incrémentale* décrit plus précisément l'impact de l'agilité sur le rôle de l'architecte.

Le coach d'équipe a les mêmes préoccupations que les chargés de projet classiques. Mais plutôt que de contrôler les activités, comme le ferait un chargé de projet, il fait réfléchir son équipe pour qu'elle ne perde pas de vue les valeurs de l'agilité et les objectifs du projet. Au commencement du projet, il anime son équipe et s'assure qu'elle ne s'éparpille pas dans le temps et qu'elle se concentre sur les objectifs des activités de démarrage.

3.2.6 Les autres intervenants

L'impact qu'aura un projet sur le reste de l'organisation est souvent négligé. Plus l'organisation est grande, plus l'impact d'un projet sur les individus est grand. Les personnes impactées auxquelles nous pensons comprennent notamment les utilisateurs finaux, les pilotes de systèmes, les formateurs, le service à la clientèle, l'équipe marketing, les opérateurs ou encore les administrateurs des applications.

Une bonne façon de réduire la résistance au changement causé par le développement et la livraison d'un nouveau produit est de les impliquer tôt dans le processus et de discuter avec eux du moment le plus propice pour qu'ils démarrent leurs interventions avec l'équipe de projet. S'il est contre-productif d'inviter ces personnes pendant les ateliers, il serait au moins bon de les identifier et de prendre la peine d'aller les rencontrer pour avoir cette discussion le plus tôt possible.

3.3 LIMITER LES EFFORTS REQUIS

> Réaction d'un détracteur :
> « *Notre projet est évalué à 2 000 jours-personnes, avec une équipe de 15 développeurs. Allons-nous vraiment faire participer toute l'équipe aux ateliers de démarrage* ? »

Nous reconnaissons qu'une quinzaine de personnes rassemblées pour concevoir une application pendant cinq jours représente une somme considérable en salaires. C'est pour cela qu'il est important que les activités de démarrage se fassent en évitant le gaspillage en temps et en efforts. Voici quelques exemples des types de gaspillage pouvant se produire pendant la période de démarrage :

- une architecture détaillée, qui n'est pas validée par un incrément de logiciel fonctionnel, source de l'effet tunnel ;
- un plan de livraison précis basé sur une capacité d'équipe approximative ;
- la rédaction des spécifications pour des fonctionnalités qui, au final, ne seront jamais produites.

Dans la plupart des organisations où nous intervenons, nous avons l'habitude de tenir les ateliers de démarrage en 5 à 10 jours ouvrables. Quelquefois moins, lorsque tous les participants sont bien préparés ou que le projet n'est pas trop complexe. Mais il ne faut pas prendre cette durée comme étant absolue et limiter les activités à l'extrême en risquant de démarrer le projet dans le chaos. Tous les projets n'ont pas la même

complexité. Il est difficile d'imaginer qu'un projet répondant à un sujet aussi sensible et complexe que l'atterrissage en toute sécurité d'un avion puisse démarrer sans une période d'analyse et de préparation plus longue.

Un indicateur pour s'assurer que l'effort consacré aux ateliers de démarrage est suffisant, est de vérifier qu'au terme de la période, l'équipe de projet est en mesure de s'engager dans une première itération. Une période de démarrage dont la durée est plus grande que celle d'une itération normale est un indice que trop d'efforts y sont consacrés. Selon nous, la production de livrables essentiels est un autre indicateur que les activités de démarrage sont efficaces et concluantes, parce que ce sont des livrables permettant de démarrer un projet avec un haut niveau de confiance :

- une charte de projet ;
- un carnet de produit ordonné et priorisé ;
- un document d'architecture de haut niveau ;
- une définition de terminé ;
- un plan de livraison.

Itération 0 : Plusieurs praticiens de l'agilité nomment la courte période de démarrage l'*Itération 0* (ou *Sprint 0*). Cette itération se distingue des autres parce qu'elle implique souvent plus d'intervenants que la seule équipe de projet et qu'elle est plus courte qu'une itération normale.
Cette appellation est cependant controversée, parce qu'elle n'est pas documentée dans la littérature de Scrum, la méthode agile la plus populaire à ce jour. Pourquoi faire une itération spéciale avant les autres itérations ? Pourquoi ne pas commencer immédiatement les vraies itérations ? Effectivement, il n'y a pas vraiment de logique à nommer cette période « itération », encore moins de lui attribuer un « 0 »[1]. Ne soyons pas puristes, retenons simplement que l'itération 0 est reconnue dans la communauté agile comme une période de préparation au démarrage des projets.

3.4 RÉDIGER UNE CHARTE DE PROJET

La charte de projet est un document décrivant de manière concise la vision, les objectifs, les leviers et les enjeux du projet.

3.4.1 Pourquoi ?

Le document est un guide qui aide l'équipe de projet à prendre les bonnes décisions en cours de projet, en ciblant les objectifs et les conditions de succès.

1. Anecdote : cela dit, nous trouvons encore plus étrange l'appellation « Itération -1 », décrite dans l'un des billets de Scott Ambler (http://www.drdobbs.com/architecture-and-design/209902719, consulté en mars 2011).

3.4.2 Combien ?

La charte de projet doit être concise pour être facile à consulter, voire facile à afficher. Elle peut facilement tenir sur une ou deux pages (voir).

3.4.3 Comment ?

Voici un modèle de structure de charte de projet :

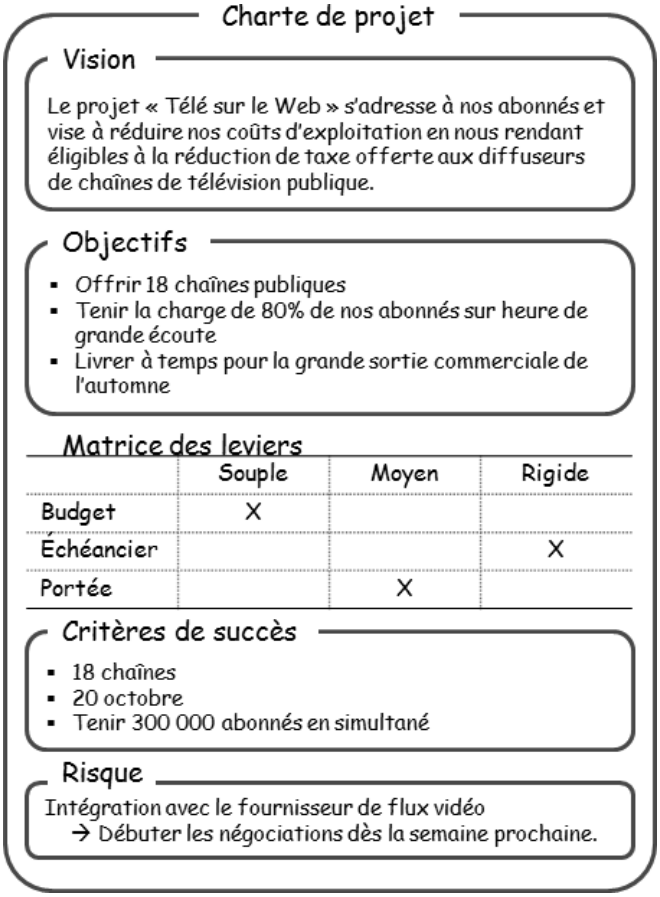
- Une section pour la vision : une phrase résumant la raison d'être du projet. Une bonne formule pour décrire une vision est celle du test de l'ascenseur (voir l'encadré ci-après).
- Une section pour les objectifs : une liste des objectifs à atteindre avec la réalisation du projet.
- Une section pour les mesures de succès : une liste des critères et des indicateurs permettant de vérifier que les objectifs ont été atteints.
- Une section pour de la matrice des compromis : un indicateur de la marge de manœuvre pour chacun des leviers permettant de contrôler la tenue du projet. Certains projets ont une date de livraison inflexible, d'autre ne peuvent pas faire de compromis sur la portée. Un projet où aucun compromis n'est possible sur tous les trois leviers d'ajustement (portée, investissement, date de livraison) est un projet sans marge de manœuvre pour gérer les risques.
- Une section pour les facteurs de risque : les obstacles potentiels ou avérés du projet. L'identification des obstacles permet de les traiter ou de faire une demande d'aide à l'organisation pour ceux dont l'équipe de projet n'a aucun contrôle.

Test de l'ascenseur

Imaginons que vous montiez dans un ascenseur en même temps que le président d'une très grande entreprise. Le président reconnaît votre visage mais ne vous connaît pas. Curieux, il vous demande sur quoi vous travaillez. Vous avez maintenant dix étages pour vous présenter !

Votre présentation pourrait avoir la forme suivante : Suite à l'opportunité <Y>, le projet <X> s'adresse à <public> pour leur offrir un <bénéfice>. Contrairement à <concurrent>, notre projet se distingue par <particularités>.

En pratique, cela pourrait donner ceci : Le projet « Télé sur le web » s'adresse à nos abonnés et vise à réduire nos coûts d'exploitation en nous rendant éligibles à la réduction de taxes offertes aux diffuseurs de chaînes de télévision publique. De plus, cette fonctionnalité nous permettra de vendre un accès aux chaînes privées dont nous sommes les propriétaires exclusifs sur toutes les plates-formes de nos abonnés : téléphone, ordinateur, décodeur numérique.



Charte de projet

Vision

Le projet « Télé sur le Web » s'adresse à nos abonnés et vise à réduire nos coûts d'exploitation en nous rendant éligibles à la réduction de taxe offerte aux diffuseurs de chaînes de télévision publique.

Objectifs

- Offrir 18 chaînes publiques
- Tenir la charge de 80% de nos abonnés sur heure de grande écoute
- Livrer à temps pour la grande sortie commerciale de l'automne

Matrice des leviers

	Souple	Moyen	Rigide
Budget	X		
Échéancier			X
Portée		X	

Critères de succès

- 18 chaînes
- 20 octobre
- Tenir 300 000 abonnés en simultané

Risque

Intégration avec le fournisseur de flux vidéo
→ Débuter les négociations dès la semaine prochaine.

Figure 3.3 — Exemple d'une charte de projet.

Retour d'expérience

Sur un de nos projets, notre levier le plus flexible était le budget, car le client avait un budget quasi illimité. Aux trois quarts du projet, nous nous sommes aperçus que le projet risquait de ne pas atteindre la portée désirée à la date prévue. Cependant, nous ne savions pas comment utiliser l'argent pour remédier à la situation. Embaucher de nouvelles personnes dans le projet si près de la date finale semblait être contre-productif.

Nous avions appris une leçon. Il ne suffit pas d'identifier les leviers disponibles : certains leviers ont besoin d'une stratégie pour être efficace, et nous aurions dû en prendre conscience pendant nos ateliers de démarrage.

Mathieu

3.5 CONSTRUIRE UN CARNET DE PRODUIT

Un carnet de produit est une liste de besoins d'affaires ordonnés selon leur valeur d'affaires, dont chaque item est décrit sous la forme d'une exigence ou d'une spécification.

- **Pourquoi ?**
  Pendant les ateliers de démarrage, le carnet de produit sert à recueillir toutes les exigences qu'il est prévu de développer pendant le projet. Pendant le développement du projet, le carnet de produit sert d'intrant aux séances de planification. À la fin de chacune des itérations, l'état d'avancement du projet est mesuré en comparant l'état du produit et le contenu restant du carnet de produit.
- **Combien ?**
  Idéalement, il faut suffisamment d'items détaillés pour que le client puisse planifier ses deux premières itérations. L'équivalent de deux itérations permet au client de ne pas être au dépourvu si un item particulier ne peut pas être planifié parce qu'il doit être raffiné ou qu'un préalable manquant empêche son développement.
  Au-delà de deux itérations, les items risquent d'être détaillés prématurément et de devoir être modifiés pour répondre aux besoins changeants du projet. Pire, certains items peuvent être détaillés d'avance et au final ne jamais être développés.
- **Comment ?**
  Le client est responsable de l'entretien du projet. Il utilise le carnet de produit pour gérer, rédiger, prioriser les exigences du produit. Il peut ajouter ou retirer autant d'éléments que nécessaire. Cependant, la stratégie est de concentrer la conception et le raffinement des exigences par ordre de priorité : une stratégie limitant les efforts inutiles.
  Il faut cependant apporter une certaine nuance à cette stratégie, puisque la valeur ajoutée n'est pas le seul élément à surveiller lors de la planification d'un projet, il faut également penser à l'utilisabilité. Prenons l'exemple d'un système de commerce électronique. Même si les méthodes de paiement sont des fonctionnalités à forte valeur ajoutée, tant que le système ne permet pas aux clients de faire des achats, la fonctionnalité de paiement en ligne demeure inutile. Autre exemple, une application de recherche d'emploi qui ne permet pas à un employeur d'afficher des offres d'emploi et de rencontrer des candidats aura une valeur très faible. Une simple annonce dans un journal imprimé aurait déjà plus de valeur pour un employeur. Développer une solution fonctionnelle devrait être le premier objectif de la planification et la priorisation par valeur ajoutée devrait en découler.

Nous vous proposons dans cette section quelques pratiques et conseils pour vous aider à la construction du carnet de produit :

- le *story mapping* pour construire un carnet de produit selon des couloirs fonctionnels ;

- des **pratiques d'ordonnancement** adaptées aux stratégies de planification incrémentale et la gestion du risque ;
- des **pratiques pour calculer la valeur d'affaires** des items du carnet de produit, adaptées aux stratégies de planification de fonctionnalités à grande valeur ajoutée ;
- la rédaction des exigences, adaptée au développement itératif et incrémental d'une solution logicielle.

3.5.1 Story mapping

Le *story mapping*[1] est un exercice proposé par Jeff Patton permettant de construire un carnet de produit en identifiant les couloirs fonctionnels de la solution. Les couloirs fonctionnels sont des processus d'affaires complets, comme la « recherche de candidats pour une offre d'emploi » dans l'exemple du site de recherche d'emploi.

La première étape de l'atelier est de prioriser tous les éléments du carnet de produit selon l'ordre chronologique logique de l'utilisation. Par exemple, dans un système de recherche d'emploi, la publication d'offres d'emploi se produit habituellement avant que les chercheurs d'emploi puissent soumettre leur candidature.

La seconde étape est de classer les items selon leur indice de criticité. La criticité se définit comme étant une combinaison de la fréquence d'utilisation de la fonctionnalité et de son importance. Toujours avec l'exemple d'un site de recherche d'emploi, joindre un *curriculum vitae* (CV) au moment de la candidature est probablement plus critique que d'être capable d'éditer son CV en ligne.

La cartographie représentant la combinaison de la priorité et de la criticité permet d'identifier les couloirs fonctionnels du carnet de produit. La première tranche de la cartographie représente probablement l'ensemble minimal de fonctionnalités permettant à tous les types d'utilisateurs du système d'utiliser le produit dans sa forme la plus simple. La construction du carnet de produit autour des couloirs fonctionnels facilite la construction d'une solution utile (voir la Figure 3.4).

3.5.2 Ordonner le carnet de produit

Dans son livre *Agile Estimating and Planning*[2], Mike Cohn, propose un modèle de priorisation prenant en compte la valeur d'affaires et la gestion du risque. Dans la Figure 3.5, ce modèle propose de commencer par les éléments les plus risqués ayant également la plus forte valeur ajoutée (1er quadrant, en haut à droite). C'est la stratégie qui est adoptée lorsque le premier objectif est la réalisation d'une solution fonctionnelle. Le développement d'un premier couloir fonctionnel, comme proposé dans l'atelier *Story Mapping* est un bon moyen d'atteindre cet objectif.

1. http://www.agileproductdesign.com/blog/the_new_backlog.html, consulté en mars 2011.
2. Cohn, Mike, *Scrum : Agile estimating and planning*, Prentice Hall, 2005.

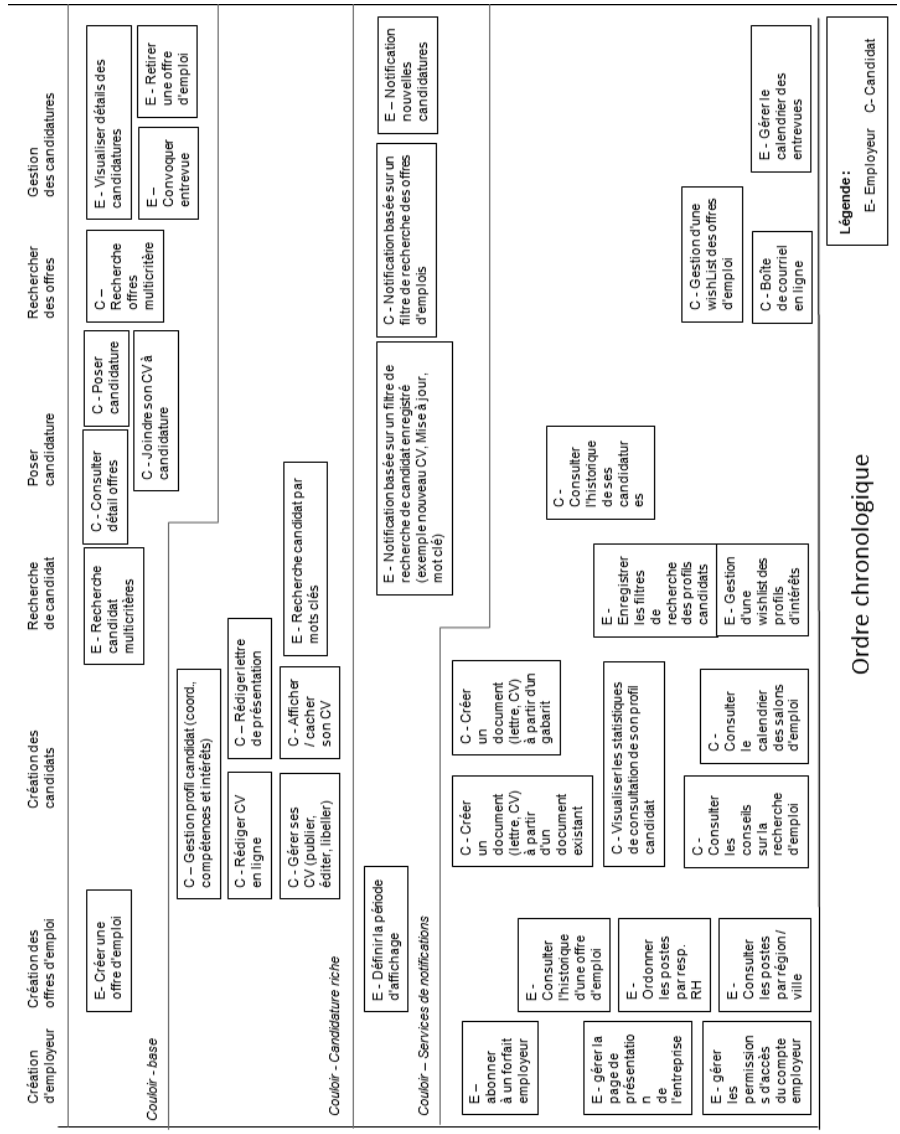


Figure 3.4 — Exemple du résultat final de l'exercice du *Story Mapping*.

Le 2e quadrant (coin inférieur droit) est celui des éléments à forte valeur ajoutée mais dont les facteurs de risque sont faibles. Ces éléments sont intéressants puisqu'ils ont un fort retour sur investissement : beaucoup de valeur pour peu de risque.

Le 3e quadrant (coin inférieur gauche) représente les éléments ayant moins de valeur d'affaire et étant moins risqués. Ces éléments seront probablement les derniers à être développés.

Ce modèle ne propose pas de planifier les éléments risqués ayant peu de valeur d'affaires. En effet, si on veut éviter de développer des fonctionnalités qui seront peu ou pas utilisées, il faudrait éviter de planifier ces éléments même s'ils faisaient partie de la portée initiale du projet[1].

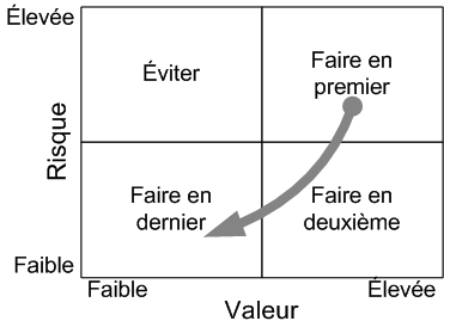


Figure 3.5 — Stratégie de priorisation des exigences.

3.5.3 Calculer la valeur d'affaires des items du carnet

Toutes les organisations TI visent à livrer de la valeur pour leurs clients, sans qu'elle soit systématiquement quantifiée ou rendue visible à tous. Les approches agiles incitent explicitement les équipes à planifier le développement logiciel par

ordre d'importance et insistent pour que chaque livraison ajoute une valeur d'affaires au produit développé de manière incrémentale. Toutefois, les approches agiles ne préconisent pas de technique particulière pour mesurer la valeur d'affaires des fonctionnalités d'un projet.

Il est possible, au début d'une transition agile, que les clients ne quantifient pas la valeur d'affaires des fonctionnalités qu'ils priorisent. Typiquement, ils ne l'ont jamais fait, soit parce qu'ils ne savent pas comment faire, soit que ce n'est tout simplement pas dans leurs habitudes de travail. Les méthodes agiles prônant la transparence, il est normal que cette transparence s'applique aussi à la valeur d'affaires attribuée à chacun des items d'un carnet de produit. C'est pourquoi il est fortement encouragé

1. Rappelons la troublante statistique du rapport du Standish Group qui mentionne que 45 % des fonctionnalités développées ne sont jamais utilisées. La planification par priorité permet de diminuer ce pourcentage, en éliminant les éléments à faible valeur ajoutée.

d'accompagner les responsables de produit à quantifier cette valeur et de l'utiliser dans les activités de gouvernance. Ainsi, les gestionnaires, dirigeants et responsables de produits seront motivés à fournir et maintenir cette valeur.

Dans certaines grandes entreprises où la valeur d'affaires de chaque fonctionnalité et caractéristique est quantifiée depuis de nombreuses années, on va jusqu'à en quantifier le rendement. Cette analyse permet de vérifier l'exactitude de la quantification de la valeur, servant ainsi à alimenter et améliorer le processus de quantification. Nous ne disons pas qu'il faille se rendre là dès le début d'une transition agile, mais seulement qu'avec le temps, les organisations désirant augmenter la maturité à leur processus en ce sens vont jusqu'à mesurer le rendement réel.

Même convaincue de la pertinence de cette mesure, une organisation n'ayant jamais mesuré la valeur d'affaires des fonctionnalités d'un projet peut se sentir démunie face à cette pratique de gestion. Bien qu'il soit possible de déterminer la valeur d'affaires de manière intuitive, nous présentons ici un résumé de quelques pratiques permettant de mesurer la valeur d'affaires des fonctionnalités et caractéristiques.

La valeur financière

Nous avons travaillé avec plusieurs compagnies d'assurance où des actuaires étaient en mesure de calculer financièrement le rendement de chacune des fonctionnalités prévues d'un système. Sachant qu'il existe peu de domaines d'affaires bénéficiant des services d'un actuaire, nous vous présentons un résumé des étapes à suivre pour calculer cette valeur :

1. **Identifier les revenus potentiels** : il existe plusieurs types de catégorisation possibles pour mesurer les sources de revenus que peut engendrer le développement d'une fonctionnalité. Elle peut être une source de nouveau revenu en attirant de nouveaux clients. Elle peut être une source de revenu supplémentaire si elle permet d'augmenter le prix qu'un client existant est prêt à investir dans les services. Elle peut être un manque à gagner si l'absence de la fonctionnalité ne retient pas les clients existants. Et finalement elle peut être une source d'économie si elle permet d'améliorer l'efficacité des opérations.
2. **Comparer les revenus aux coûts** : une fois estimés, les revenus peuvent être comparés avec les coûts requis à la livraison de la fonctionnalité. Le ratio entre les revenus et les coûts est déjà un indicateur permettant de prioriser les fonctionnalités à développer.
3. **La valeur nette actualisée (VAN)**[1] : les revenus estimés pour une fonctionnalité ne sont pas les mêmes lorsqu'un projet s'étend sur une longue période. En effet, les revenus qu'il n'est pas possible de générer aujourd'hui représentent un manque à gagner sur des investissements. Le calcul de la VAN permet de tenir compte de l'amortissement des revenus à travers le temps.

1. VAN, traduction de l'anglais *Net Present Value, NPV.*

4. **Le retour sur investissement (RSI)**[1] : Tandis que la VAN permet de connaître le retour net d'un investissement dans le futur, le RSI permet de savoir le taux de retour sur investissement à travers le temps. Autrement dit, plus le RSI est grand, plus vite le capital sera remboursé et plus rapidement l'investissement rapportera des revenus.

L'évaluation du potentiel monétaire est un élément de planification très efficace et sans ambiguïté. Cependant, il n'est pas simple car ses calculs demandent des compétences en mathématiques financières.

La valeur de la satisfaction

Le modèle de Kano[2] permet de mesurer la valeur d'affaires des fonctionnalités selon la satisfaction qu'elles apportent aux utilisateurs. Voici un résumé des étapes à suivre pour déterminer la satisfaction potentielle des fonctionnalités :

1. À partir d'un questionnaire à l'intention des utilisateurs, évaluer le degré de satisfaction que l'utilisateur retirerait si la fonctionnalité était disponible et le degré d'insatisfaction si la fonctionnalité était absente.
2. Le résultat des deux questions pour chaque fonctionnalité permet de catégoriser le degré de satisfaction selon trois groupes :

   a) les **fonctionnalités essentielles** sont considérées comme implicites par les utilisateurs, qui retirent peu de satisfaction lorsqu'elles sont présentes et beaucoup de frustration lorsqu'elles sont absentes ;

   b) les **fonctionnalités exprimées** sont proportionnelles à leur présence. Plus un produit contient de fonctionnalités exprimées, plus le client est prêt à payer cher pour l'utiliser ;

   c) les **fonctionnalités attrayantes** procurent beaucoup de satisfaction aux utilisateurs qui n'avaient aucune attente particulière à leur sujet.

L'évaluation de la satisfaction est à la portée de tous et permet de faire une évaluation fiable de la valeur d'affaires des fonctionnalités d'un carnet de produit. Elle permet en outre de connaître le minimum de fonctionnalités attendues par les clients, de savoir pour quelles fonctionnalités les utilisateurs sont prêts à payer plus et comment attirer des clients enthousiastes à l'aide de fonctions attrayantes.

Pour plus de détails et de référence sur le calcul de la valeur financière et du degré de satisfaction, nous vous invitons à consulter l'ouvrage *Agile Estimating and planing*[3]. Évidemment ce ne sont pas les seuls moyens d'évaluer la valeur d'affaires des fonctionnalités d'un produit, mais cet ouvrage de Mike Cohn a popularisé ces techniques auprès d'une portion significative de praticiens agiles.

1. RSI, traduction de l'anglais *Return on Investment, ROI*.
2. Source : http://fr.wikipedia.org/wiki/Mod%C3%A8le_de_Kano, consulté en mars 2011.
3. *Scrum : Agile estimating and planning* de Mike Cohn, Prentice Hall, 2005.

3.5.4 Détailler les exigences

Une fois que le carnet de produit décrit grossièrement l'ensemble de toutes les exigences du projet, les items sont détaillés dans l'ordre des priorités. La description des items du carnet de produit peut prendre plusieurs formes : scénario utilisateur) (*user story*), cas d'utilisation (*use case*) ou autres. Il faut s'assurer que la forme choisie se prête bien à la planification incrémentale. L'acronyme INVEST (Indépendant, Négociable, Valeur d'affaires, Estimable, *Small* [petit], Test fonctionnel), que Mike Cohn propose dans son ouvrage *User Stories Applied*[1] rappelle les qualités des items se prêtant bien au développement agile. Pour plus de détails sur la rédaction du carnet de produit en mode agile, vous pouvez consulter les livres de Mike Cohn et Alistair Cockburn, *User Stories Applied for Agile Software Development*[2] et *Rédiger des cas d'utilisation efficaces* [3].

3.6 FIXER LA DÉFINITION DE TERMINÉ

La définition de « terminé » est une description sans équivoque décrivant tout ce qui doit être fait pour considérer un élément du carnet de produit comme terminé. Elle sert de référence pour les membres de l'équipe de projet pour assurer que tout ce qui est terminé répond aux mêmes critères de qualité, peu importe qui a contribué à sa réalisation. La définition de terminé représente le contrat de qualité entre le client et l'équipe de réalisation.

3.6.1 Pourquoi ?

> **Principe agile** : Un logiciel fonctionnel est la meilleure unité de mesure de la progression du projet.

Les approches agiles mesurent l'avancement des projets par la proportion de fonctionnalités d'un produit opérationnelles à un moment donné. Selon ce principe, une fonctionnalité qui serait seulement documentée dans un dossier d'analyse ne pourrait être compilée dans l'état d'avancement d'un projet, puisqu'elle n'est toujours pas opérationnelle au sein du produit. Pour devenir fonctionnelle, les autres disciplines de développement comme le code, l'intégration et les tests doivent avoir été complétées.

Avec une définition de terminé, les membres de l'équipe s'entendent sur toutes les activités requises pour compléter les items du carnet de produit. Lorsqu'un membre annonce qu'un item est terminé, tous ont la confiance qu'il ne reste pas de travail à accomplir pour être en mesure de la livrer.

---

1. *User Stories Applied : For Agile Software Development*, de Mike Cohn, Addison-Wesley Professional, 2004.
2. *Scrum: Agile estimating and planning* de Mike Cohn, Prentice Hall, 2005.
3. *Rédiger des cas d'utilisation efficaces* de Alistair Cockburn, Eyrolles, 2001.

Durant les ateliers de démarrage, l'équipe devra estimer les éléments du carnet de produit pour être en mesure de rédiger un plan de livraison. Au moment d'estimer, la définition de terminé ne devrait pas être négociable. Alors que la portée d'une exigence peut varier, les activités de développement devraient être fixes. Cette définition de la qualité peut faire varier la capacité d'une équipe à livrer des incréments de logiciel fonctionnel, mais elle n'affecte pas la valeur des estimations.

3.6.2 Combien ?

Plus la définition de terminé couvre tout le travail à faire pour livrer un item du carnet de produit, plus les chances de découvrir de mauvaises surprises seront réduites lorsque le produit sera prêt à être déployé en production. Il faut donc répertorier toutes les activités de développement de l'organisation et les inscrire à la définition de terminé du projet. Cela couvre les activités de rédaction jusqu'à celles de stabilisation. D'ailleurs, une définition complète et respectée réduira le temps nécessaire pour stabiliser les produits avant leur mise en production.

3.6.3 Comment ?

Les éléments de la définition ne devraient pas laisser place à interprétation. Par exemple, un élément qui s'intitulerait « Les tests unitaires sont écrits », ne permet pas de vérifier si le niveau de qualité défini a été rencontré ou pas. Chacun des membres peut avoir une conception différente de ce que tester le code représente.

Pour prévenir cette mauvaise interprétation, Amr Elssamadisy, dans son ouvrage *Agile Adoption Patterns*[1], propose d'adopter des objectifs dont les éléments qui sont « SMART », c'est-à-dire **S**pécifique, **M**esurable, **A**tteignable, pe**R**tinent (***r**elevant*) et limité dans le **T**emps. En ajoutant un seuil de couverture de tests par module de l'application, l'élément « Les tests unitaires sont écrits » pourrait devenir « Les tests unitaires couvrent à 100 % le code du module de service et le code des modules de la couche du domaine ». Ce genre d'élément ne laisse plus place à interprétation, et la vérification d'une fonctionnalité se basera sur des critères mesurables.

Pour poursuivre votre réflexion à propos de la définition de terminé, vous pouvez vous référer aux ouvrages *Scrum, le guide pratique de la méthode agile la plus populaire*[2] de Claude Aubry et *Agile Estimating and Planning*[3] de Mike Cohn.

3.7 BÂTIR UN PLAN DE LIVRAISON

Rédiger un plan de livraison suite à l'adoption d'un processus de planification itératif est un véritable défi. Comment estimer la taille d'un projet alors que les items du

1. *Agile Adoption Patterns* de Amr Elssamadisy, Addison-Wesley Professional, 2008.
2. *Scrum : Le guide pratique de la méthode agile la plus populaire* de Claude Aubry, Dunod, 2010.
3. *Scrum : Agile estimating and planning* de Mike Cohn, Prentice Hall, 2005.

INFOPRO

TYPE D'OUVRAGE

L'ESSENTIEL | **SE FORMER** | RETOURS D'EXPÉRIENCE

Mathieu Boisvert
Sylvie Trudel

MANAGEMENT DES SYSTÈMES D'INFORMATION

APPLICATIONS MÉTIERS

ÉTUDES, DÉVELOPPEMENT, INTÉGRATION

EXPLOITATION ET ADMINISTRATION

RÉSEAUX & TÉLÉCOMS

CHOISIR L'AGILITÉ

Du développement logiciel à la gouvernance

Cet ouvrage s'adresse à tous ceux qui participent à la conception de logiciels, qu'ils soient gestionnaires, développeurs, chefs de projet, DSI, ou qu'ils soient « côté métier », initiateurs et futurs utilisateurs de ces logiciels.

Quand une organisation choisit d'adopter des **méthodes agiles** pour développer de nouvelles applications, plusieurs de ses **processus métiers** sont concernés et remis en cause.

Cet ouvrage analyse et décrit l'impact de ces changements en suivant quatre niveaux concentriques :

- Les projets
- L'encadrement des projets
- Les processus et les individus
- La gouvernance.

Son atout principal est qu'il s'appuie sur de nombreux **retours d'expériences** vécues par les auteurs au cours de leur vie professionnelle.

Son objectif est de fournir aux lecteurs une **méthode de réflexion** aussi complète et efficace que possible pour aider une équipe à trouver la meilleure façon de réussir sa transition vers les méthodes agiles.

MATHIEU BOISVERT

Est conseiller en adoption des méthodes agiles à Pyxis Technologies (Québec) depuis 2004. ScrumMaster et membre actif de la communauté agile.

SYLVIE TRUDEL

Est spécialiste en développement logiciel et amélioration des processus avec plus de vingt-six années d'expérience. Consultante à Pyxis Technologies. ScrumMaster depuis 2006.

9 782100 558506

6917033

ISBN 978-2-10-055850-6

DUNOD

www.dunod.com