

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

RAPPORT DE PROJET PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE À
L'OBTENTION DE LA MAÎTRISE EN GÉNIE,
CONCENTRATION GESTION DE PROJETS D'INGÉNIERIE

PAR
ALEXIS BERTRAND TCHAMANBE

MISE EN PLACE DE LA GESTION DE PROJET AGILE DANS UNE BANQUE
HAÏTIENNE

MONTRÉAL, LE 8 AVRIL 2021



ALEXIS BERTRAND TCHAMANBE, 2021



Cette licence [Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/) signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette œuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'œuvre n'ait pas été modifié.

PRÉSENTATION DU JURY

CE RAPPORT DE PROJET A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

Professeur Alain April, directeur de projet,
Département de Génie logiciel et TI à l'École de technologie supérieure

Professeur François Coallier, jury
Département de Génie logiciel et TI à l'École de technologie supérieure

REMERCIEMENTS

Je remercie tous ceux qui de près ou de loin m'ont aidé à compléter ce projet :

Le professeur Alain April pour son accompagnement et son support.

Ma mère, Delphine Maniassi, pour son soutien inconditionnel.

Ma conjointe et mes deux adorables filles : Kameni Odile, Tchamanbe Queency et Tchamanbe Énora, pour la chaleur quotidienne, qui me donne la force de toujours aller vers l'avant.

MISE EN PLACE DE LA GESTION DE PROJET AGILE DANS UNE BANQUE HAÏTIENNE

ALEXIS BERTRAND TCHAMANBE

RÉSUMÉ

Ce projet d'application de 15 crédits, consiste à étudier les cycles de vie TI actuels d'une banque Haïtienne, et de proposer comment les rendre plus agiles.

Mettre en place l'agilité demande des ajustements à l'ensemble des processus de l'entreprise ainsi qu'aux rôles actuels. Devenir agile demande des changements aussi concernant l'implication future des clients dans les projets TI.

Avant de proposer des solutions, une première étape de cette recherche appliquée vise à identifier, dans la littérature, les enjeux de ce changement et les approches proposées.

Dans une deuxième étape, l'étude des cycles de vie actuels, des rôles, des activités et gabarits actuels sont effectués afin de voir l'impact de l'introduction d'un nouveau rôle d'analyste d'affaires, et en quoi consiste la création d'un nouveau bureau de projet plus agile.

Finalement, ce rapport propose des solutions et avenues pour cette transformation.

Mots-clés : Agile, Scrum, Kanban, Scrumban, Cycles de vie, Processus, développement logiciel

IMPLEMENTATION OF AGILE PROJECT MANAGEMENT IN A HAITIAN BANK

ALEXIS BERTRAND TCHAMANBE

ABSTRACT

This 15-credit application project consists of studying the current IT life cycles of a Haitian bank and propose how to transform them to adopt a more agile approach.

Transforming existing processes so that they become more agile requires adjustments to many business processes as well as current roles. Becoming agile also requires changes in the future involvement of customers in IT projects.

Before proposing solutions, a first step in this applied research aims to identify, in the literature, the challenges associated with such a transformation and the proposed approaches.

In a second step, the study of current IT life cycles, roles, activities and current templates are carried out in order to understand the impact of the introduction of new roles associated with a more agile business analyst, and how to create a more agile IT project office.

Finally, this report proposes solutions and avenues for this transformation.

Keywords: Agile, Scrum, Kanbam, Scrumban, life cycles, Process, software developpement

TABLE DES MATIÈRES

	Page
1.1	Introduction.....3
1.2	Généralité sur les méthodologies agile3
1.2.1	Manifeste agile:..... 3
1.3	Agile et ses méthodologies populaires.....5
1.3.1	Méthodologie SCRUM..... 5
1.3.2	La méthodologie KANBAN 8
1.3.3	La méthodologie Scrumban 9
1.4	Implémentation de l’agilité en industrie10
1.4.1	Création d’un environnement agile..... 10
1.4.2	Déploiement d’un modèle d’affaires opérationnel 10
1.5	Conclusion11
2.1	Introduction.....13
2.2	Structure organisationnelle13
2.2.1	Composition des équipes agiles 13
2.2.2	Organisation des nouveaux rôles 15
2.2.3	Gestion des nouveaux rôles 16
2.2.3.1	Le chef de produit 17
2.2.3.2	Le maître de mêlée..... 18
2.2.3.3	Coach agile..... 19
2.2.3.4	L’équipe de développement..... 20
2.2.4	Impact des nouveaux rôles sur les processus existant de la DIT 20
2.2.5	Stratégie de migration du personnel impacté..... 21
2.2.6	Culture d’entreprise en agilité..... 23
2.3	Analyse des Cycles de vie de la DIT23
2.3.1	Analyse des cycles de vies 23
2.3.2	Processus actuels du CVDL et Agilité..... 24
2.3.2.1	Processus d’Analyse actuel du CDVL..... 25
2.3.2.2	Processus de Conception-Développement actuel du CVDL 26
2.3.2.3	Processus de préparation du test d’acceptation actuel du CVDL 27
2.3.2.4	Processus d’exécution du test d’acceptation actuel du CVDL .. 28
2.3.2.5	Processus d’analyse des résultats du test d’acceptation actuel du CVDL..... 29
2.3.2.6	Processus de mise en production actuel du CVDL..... 30
2.4	Cycle de Vie Scrum pour le CVDL31
2.4.1	Processus de création de la Charte de Projet 32
2.4.2	Processus de documentation des requis 32
2.4.3	Processus de raffinement de l’arriéré du produit 33
2.4.4	Processus de planification du Sprint 35
2.4.5	Processus de développement de produit 36
2.4.6	Processus de revue du Sprint 38
2.4.7	Processus de rétrospective du Sprint..... 38

2.4.8	La réunion quotidienne de Scrum	39
2.5	Scrumban comme Alternative au Scrum pour le CVDL	40
2.5.1	Arriéré du produit	40
2.5.2	Flux de travail Kanban.....	40
2.5.3	Gestion du tableau Kanban	42
2.6	Outils de gestion des configurations	43
2.6.1	Outil de gestion des projets.....	44
2.6.2	Outil de gestion des requis.....	44
2.6.3	Outil de collaboration et de documentation	45
2.6.4	Outil de gestion des versions	45
2.7	Conclusion	45
3.1	Introduction.....	47
3.2	Synthèse du travail complété et revue critique	47
3.2.1	Synthèse du travail complété	47
3.2.2	Revue critique	48
3.3	Améliorations futures.....	49
3.4	Conclusion	49

LISTE DES FIGURES

		Page
Figure 1.1	Méthodologies agiles et Lean	5
Figure 1.2	Cycle de vie Scrum simplifié.....	6
Figure 1.3	Exemple de tableau Kanban.....	9
Figure 2.1	Organigramme des rôles	17
Figure 2.2	Référentiel des processus de la DIT.....	23
Figure 2.3	Processus d'Analyse actuel du CVDL.....	25
Figure 2.4	Processus de Conception Développement actuel du CVDL.....	26
Figure 2.5	Processus préparation du test d'acceptation actuel de la DIT.....	27
Figure 2.6	Processus exécuter test d'Acceptation actuel de la DIT	28
Figure 2.7	Processus d'analyse des résultats du test d'acceptation actuel de la DIT..	29
Figure 2.8	Processus de mise en production actuel de la DIT	30
Figure 2.9	Cycle de vie Scrum de développement logiciel	31
Figure 2.10	Procédure de création de la charte de projet	32
Figure 2.11	Procédure de documentation des requis.....	33
Figure 2.12	Procédure de raffinement de l'arriéré	34
Figure 2.13	Procédure de planification du Sprint.....	36
Figure 2.14	Procédure de développement du produit.....	37
Figure 2.15	Procédure de revue du Sprint.....	39
Figure 2.16	Procédure de rétrospective du Sprint	39
Figure 2.17	Flux de travail Kanban pour développement logiciel en interne	41
Figure 2.18	Exemple de tableau Kanban pour logiciel développé en interne	43

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

CDR:	Centre De Revenu
CM:	Configuration Management
CMMI:	Capability Maturity Model Integration
CVAI:	Cycle de Vie d'Acquisition d'Infrastructure
CVAL:	Cycle de Vie d'Acquisition Logiciel
CVDL:	Cycles de Vie de Développement logiciel
CVDLF:	Cycle de vie de développement logiciel avec fournisseur
CVÉL:	Cycle de Vie Évolution logiciel, demande liée à une interface fournisseur
CVÉL(AI):	Cycles de Vie Évolution Logiciel (Applications internes)
CVMLF (NV):	Cycle de Vie Mise à jour Logiciel Fournisseur (Nouvelle Version)
CVMLF (DI):	Cycle de Vie Mise à jour Logiciel Fournisseur (Demande Interne)
DIT:	Direction Information et Technologie
ISO:	International Organization for Standardization
PMBOK:	Project Management Body of Knowledge
QA:	Quality Assurance
SAFe:	Scaled Agile Framework
XP:	eXtreme Programming

INTRODUCTION

Au cours des récentes années, l'industrie des technologies de l'information a connu beaucoup de bouleversements quant aux méthodologies de développement des applications. La satisfaction des besoins du client étant le but principal recherché dans la plupart des projets, les méthodologies traditionnelles ont beaucoup évolué, et certaines, ont été remplacé par des méthodologies dites modernes, introduits dans l'industrie pour aider les entreprises à mettre l'accent sur les besoins du client, et à demeurer compétitif à travers une production efficace, à des intervalles de temps relativement court, et à des coûts réduits. Les méthodes agiles font partie de ces méthodologies dites modernes, basées sur une approche itérative et collaborative, qui place les besoins du client au centre du cycle de développement du produit logiciel. Le terme « agile » suggère une approche itérative de gestion de projet qui prend le contrepied des approches traditionnelles prédictives et séquentielles de type cycle en V et en cascade (Lothon, 2013). Dans un environnement agile, la notion de gestion de projet est remise en question au profit de la gestion de produit (Lothon, 2013). Une banque haïtienne, consciente de l'apport potentiel des méthodologies modernes, expérimente depuis 2019, la mise en place de l'agilité avec quelques difficultés, notamment en se posant des questions sur l'adaptation des processus actuels des projets vers l'agilité. Il est donc question pour cette banque de prendre du recul et voir comment effectuer cette transformation d'une manière structurée et graduelle, et, entre autres, de revoir les rôles, les processus et gabarits, ainsi que de décider de l'approche d'intervention d'un bureau de projet dans un contexte agile, qui pourra mieux appuyer ces équipes de projets à l'avenir.

Ce rapport de recherche appliquée a pour objectif d'étudier la situation actuelle et de proposer comment rendre les projets technologiques de transformation numérique plus agiles. Un projet de transformation numérique par définition est un projet transversal qui touche plusieurs organisations de la banque. Il sera question, tout au long de cette recherche appliquée, de premièrement faire un survol général de la méthode agile et de sa mise en œuvre, afin que, au regard des processus actuels de la DIT, des améliorations soient proposées pour aider la banque à effectuer le passage efficacement vers l'agilité.

CHAPITRE 1

REVUE DE LA LITTÉRATURE

1.1 Introduction

Cette revue littéraire va présenter un bref résumé de différentes méthodologies agiles actuellement implémentées en industrie. Une analyse de chacune de ces méthodologies permettra de déterminer lesquelles seront plus appropriées pour rendre efficacement les cycles de vie la DIT.

1.2 Généralité sur les méthodologies agile

Une méthodologie agile est une approche de développement s'effectuant par itération successive, qui permet de prendre en compte des changements de requis en cours de développement, de manière à apporter réellement de la valeur au client (Aubry, 2013, p. P.1). Elle se base sur un cycle de développement qui met le client au centre. Le client est impliqué dans la réalisation du début à la fin du projet. Cette implication du client permet à l'équipe de développement d'obtenir une rétroaction régulière afin d'appliquer directement les changements nécessaires. Le principe de base de cette méthode consiste à proposer une version opérationnelle minimale du logiciel puis à intégrer des fonctionnalités supplémentaires à cette version, à l'aide d'un processus itératif (« Définition des méthodes Agiles », 2015).

1.2.1 Manifeste agile:

Une méthodologie agile s'appuie sur quatre valeurs tirées du Manifeste agile (Kent, Beedle, Bennekum, & ..., s.d.-a) :

1. Les individus et leurs interactions plus que les processus et les outils;
2. Des logiciels opérationnels plus qu'une documentation exhaustive;
3. La collaboration avec les clients plus que la négociation contractuelle;
4. L'adaptation au changement plus que le suivi d'un plan.

Une méthodologie agile est basée sur douze principes (Kent, Beedle, Bennekum, & ..., s.d.-b) sous-jacents au manifeste :

1. Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée;
2. Accueillez positivement les changements de besoins, même tard dans le projet. Les processus agiles exploitent le changement pour donner un avantage compétitif au client;
3. Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts;
4. Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet;
5. Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés;
6. La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.
7. Un logiciel opérationnel est la principale mesure d'avancement;
8. Les processus agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant;
9. Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité;
10. La simplicité, c'est-à-dire l'art de minimiser la quantité de travail inutile, est essentielle;
11. Les meilleures architectures, spécifications et conceptions émergent d'équipes autoorganisées;
12. À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

La prochaine section permettra de faire une revue littéraire des différentes méthodologies agiles les plus populaires, et leurs implémentations en industrie.

1.3 Agile et ses méthodologies populaires

L'agilité des processus de projets technologiques est mise en œuvre selon plusieurs méthodologies dont les plus populaires sont : Scrum, Kanban et Scrumban. Ces différentes méthodologies s'inspirent de la philosophie du « Lean » aussi appelée le « Lean thinking » qui consiste à satisfaire le client en lui fournissant des produits et services qui satisfont ses besoins. Il vise à réduire leurs cycles de développement, de livrer aux clients des produits de haute qualité à des coûts réduits, avec le minimum de ressources nécessaires, tout en éliminant toute forme de gaspillage (Sayer & Williams, 2007, p. Chap.1). La figure 1 ci-dessous présente une illustration de l'ensemble des méthodologies agiles, qui émanent toutes de la philosophie du Lean.

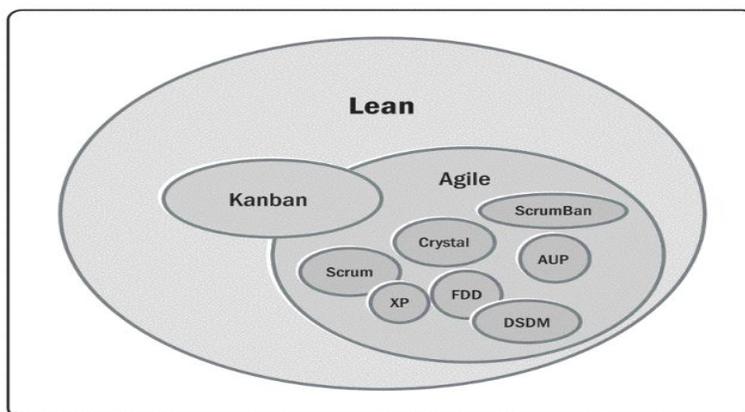


Figure 1.1 : Méthodologies agiles et Lean (Project Management Institute, 2017, p. P.11)

La prochaine section présentera une revue littéraire de la méthodologie Scrum et les moyens utilisés en industrie pour l'implémenter.

1.3.1 Méthodologie SCRUM

Scrum est une méthodologie de développement et de maintenance des logiciels. Elle utilise une approche itérative et incrémentale pour optimiser la prédictibilité et le contrôle des risques. Trois piliers soutiennent l'implémentation du contrôle de processus de cette méthodologie (Project Management Institute, 2017, p. P.4):

1. Transparence : Les aspects importants du processus doivent être visibles à tous ceux qui sont responsables des résultats. La transparence requiert la définition d'un standard commun pour ces aspects afin que les observateurs partagent une compréhension commune de ce qui est observé;
2. Inspection: Les utilisateurs de la méthodologie Scrum doivent fréquemment inspecter les artéfacts et l'état d'avancement par rapport à un Objectif de Sprint (de l'anglais « Sprint Goal ») afin de détecter les écarts indésirables. La fréquence de ces inspections ne devrait pas gêner le travail en cours. Ces inspections sont plus bénéfiques lorsqu'elles sont effectuées avec diligence par des inspecteurs qualifiés sur les lieux de travail;
3. Adaptation: Si un inspecteur détermine qu'un ou plusieurs aspects du processus dérivent hors des limites acceptables, et que le produit qui en résulte ne sera pas acceptable, le processus ou le matériel utilisé par le processus doit être ajusté. Un ajustement doit être fait dès que possible afin de minimiser le risque d'autres dérives.

Les valeurs d'engagement, de courage, d'accent, d'ouverture et de respect doivent être incarnées et vécues par une équipe Scrum. La bonne application de la méthodologie repose sur des personnes de plus en plus à même de vivre avec ces cinq valeurs (« 2017-Scrum-Guide-French.pdf », s.d., p. P.5). La figure 3 ci-dessous représente le cycle de vie Scrum simplifié schématisé:

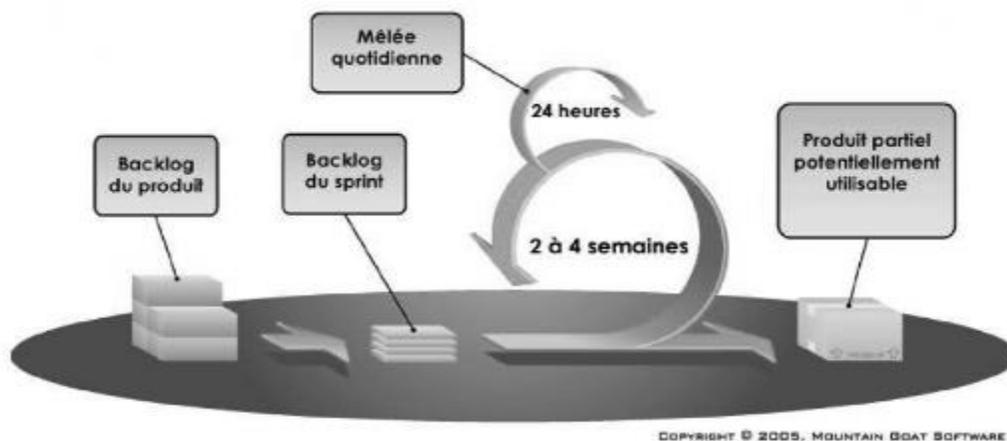


Figure 1.2 : Cycle de vie Scrum simplifié(Aubry, 2013, p. P.19)

Le Sprint est le cœur d'une itération Scrum. Il a une durée typique d'une à quatre semaines au cours de laquelle un incrément produit fonctionnel et potentiellement livrable est créé. Un Sprint peut être terminé avant son échéance, cependant, seul le chef de produit (de l'anglais « Product Owner ») a le pouvoir d'arrêter un Sprint plus tôt, bien qu'il puisse le faire sous l'instruction des parties prenantes ou de l'équipe de développement ou du maître de mêlée (de l'anglais « Scrum Master »). L'objectif du Sprint est typiquement fixe, les changements qui le remettent en cause ne sont que rarement permis. Le périmètre peut toutefois être clarifié et renégocié entre le chef de produit et l'équipe de développement. La planification du Sprint permet de définir le travail à effectuer durant un Sprint (« 2017-Scrum-Guide-French.pdf », s.d., p. P.9).

La mêlée quotidienne (de l'anglais « Daily Scrum ») est une réunion quotidienne, d'environ 15 minutes, destinée à l'équipe de développement. Elle vise à explorer le travail qui a été fait depuis la dernière mêlée quotidienne, et permet de discuter de ce qui sera fait durant les prochaines 24 heures (« 2017-Scrum-Guide-French.pdf », s.d., p. P.12).

La revue de Sprint (de l'anglais « Sprint Review ») est tenue à la fin du Sprint, pour inspecter le travail effectué et adapter la liste de l'arriéré du produit (de l'anglais « Product Backlog ») si nécessaire. Pendant la revue de Sprint, l'équipe Scrum et les parties prenantes échangent sur ce qui a été fait durant le Sprint (« 2017-Scrum-Guide-French.pdf », s.d., p. P.13).

La rétrospective de Sprint (de l'anglais « Sprint Retrospective ») est une opportunité, pour l'équipe Scrum, de s'auto-inspecter et de créer un plan d'améliorations à adopter au cours du prochain Sprint (« 2017-Scrum-Guide-French.pdf », s.d., p. P.14).

L'arriéré du Produit est une liste ordonnée de tous les éléments identifiés comme étant nécessaires à inclure dans le produit. Il constitue l'unique source d'exigences pour tout changement à apporter au produit. Le chef de produit est responsable de l'arriéré du produit, y

compris son contenu, sa disponibilité et son ordonnancement (« 2017-Scrum-Guide-French.pdf », s.d., p. P.15).

L'arrière du Sprint est l'ensemble des éléments sélectionnés pour le Sprint plus un plan pour livrer l'incrément du produit et réaliser l'objectif du Sprint. L'arrière Sprint est une prévision que l'équipe de développement fait de la fonctionnalité qui sera présente dans le prochain incrément et le travail nécessaire pour livrer cette fonctionnalité dans un incrément (« 2017-Scrum-Guide-French.pdf », s.d., p. P.16).

Une équipe Scrum comprend trois rôles/acteurs (« 2017-Scrum-Guide-French.pdf », s.d., p. P.6):

1. **Le chef de produit** est responsable de maximiser la valeur du produit résultant du travail de l'équipe de développement. Il est le seul responsable de la gestion de l'arrière du produit. Ce rôle peut varier grandement selon les organisations, les équipes Scrum et les individus;
2. **L'équipe de développement** se compose de professionnels qui fournissent un incrément potentiellement livrable à la fin de chaque Sprint. Un incrément est requis à la revue de sprint. Seuls les membres de l'équipe de développement créent l'incrément;
3. **Le maître de mêlée** est chargé de promouvoir et supporter le Scrum tel que défini dans le Guide Scrum. Les maîtres de mêlées remplissent leur rôle en aidant tout le monde à comprendre la théorie, les pratiques, les règles et les valeurs de Scrum. C'est un leader-serviteur de l'équipe Scrum.

Maintenant que la méthodologie Scrum a été présentée, la prochaine section présentera la méthodologie Kanban.

1.3.2 La méthodologie KANBAN

Le Kanban est une méthodologie qui n'est pas itérative, mais incrémentale. Le Kanban est aussi une méthode agile, car il respecte le manifeste agile, à travers les douze principes de l'agilité. Dans un processus Kanban, un projet n'a pas de début ni de fin pour l'ensemble des

tâches de développement du logiciel. C'est un travail continu. Chaque tâche/histoire peut commencer et finir indépendamment des autres, sans toutefois avoir de durée précise. Un processus Kanban est constitué de plusieurs histoires placées dans un tableau Kanban et qui se déplacent dans le tableau de gauche à droite du début à la mise en production. Le but de ce processus est de garder un flux constant de travail pouvant être inventorié à la fin (« What is Agile Kanban Methodology? Learn the Methods & Tools », s.d.).

La figure ci-dessous représente un exemple de tableau Kanban. « Un tableau Kanban est un outil de gestion de projet agile conçu pour aider à visualiser le travail, limiter le travail en cours et maximiser l'efficacité (ou le flux) »(Atlassian, s.d.-c).



Figure 1.3 : Exemple de tableau Kanban

Maintenant que la méthodologie Kanban a été présentée, la prochaine section présentera la méthodologie Scrumban.

1.3.3 La méthodologie Scrumban

Le Scrumban est une combinaison du Scrum et Kanban (Paul & Rahman, 2018). Il permet d'utiliser certains éléments du Scrum, inspirés de la méthode Kanban pour ainsi tirer le maximum d'avantage d'un Scrum, qui est lui beaucoup plus structuré. La flexibilité du Kanban lui permet juste de visualiser le flux du travail des Scrums en continu. Cela permet aux équipes Scrum de travailler en toute transparence, et de s'améliorer continuellement. Les changements de requis injectés par le chef de produit (c.-à-d. le client) peuvent être faits à tout moment d'une itération de développement. Cette méthodologie est simple à implémenter et facilement adaptable aux processus existants.

La section suivante permettra de présenter, en général, ce que font les entreprises pour passer à l'agilité.

1.4 Implémentation de l'agilité en industrie

Cette section présentera de manière succincte la création d'un environnement agile et le déploiement de ce modèle d'affaires.

1.4.1 Création d'un environnement agile

La création d'un environnement agile commence par la promotion de l'esprit de l'approche agile au sein de l'entreprise, ainsi que l'acquisition des connaissances, outils et techniques pouvant faciliter l'adoption de cette nouvelle approche au sein des équipes. Les entreprises qui veulent adopter l'agilité essayent d'apporter une réponse claire aux questions suivantes (Project Management Institute, 2017) :

- Comment l'équipe peut-elle fonctionner de manière agile?
- Comment l'équipe peut-elle livrer des produits de manière agile?
- Comment instaurer la transparence au sein des équipes?
- Comment aider l'équipe à mieux prioriser le travail à faire?
- Comment promouvoir des leaders au service des autres, au sein des équipes?

À l'issue de cet exercice, l'entreprise déploie des rôles, processus et outils, basés sur un modèle d'affaires opérationnel, permettant d'implémenter une méthodologie agile (Scrum, Kanban, Scrumban), un standard ou modèle d'assurance qualité et amélioration continue (CMMI, ISO 9001) et un guide de gestion de projet (PMBOK)

1.4.2 Déploiement d'un modèle d'affaires opérationnel

Les méthodologies agiles comme le Scrum et le Kanban s'adaptent très bien à des entreprises ou à des équipes qui font régulièrement le développement et la maintenance des applications.

Cependant, l'un des plus gros enjeux pour les entreprises est de pouvoir intégrer ces méthodologies à d'autres modèles, standard, ou guide de gestion de projet (CMMI, ISO 9000, PMBOK...) pour obtenir un modèle d'affaires opérationnel.

Les entreprises, dans leur démarche pour l'adoption de l'agilité, ne se contentent pas juste de déployer une méthodologie agile, mais plutôt un modèle d'affaires opérationnel. En effet, les méthodologies agiles à elle seule ne permettent pas aux entreprises de développer les produits agiles de meilleure qualité. L'intégration d'un modèle de maturité tel que CMMI, ou d'un standard d'assurance qualité comme ISO 9000, et l'utilisation d'un guide de gestion de projet comme le PMBOK, sont un gage de succès sur le chemin de l'agilité, dans un environnement de qualité et d'amélioration continu, pour une meilleure compétitivité.

En industrie, la qualité est souvent perçue par des acteurs comme un obstacle à l'agilité. Cependant, développer des produits de manière agile sans standard de qualité peut entraîner des pertes énormes en termes de coût et de temps, et voir même l'échec du projet. Le véritable enjeu est donc d'adapter aux processus, des standards de qualité pertinents, sans altérer leur agilité (Amer, Badr, & Hamad, s.d.).

1.5 Conclusion

Les méthodologies agiles Scrum, Kanban, et Scrumban ont été présentés dans cette partie, ainsi que les différents éléments qui rentrent en compte au moment d'adopter l'agilité. La popularité de ces méthodologies en industrie est une preuve de la valeur qu'elles apportent aux entreprises, bien que beaucoup de mesure doit être mise en place pour assurer leur succès. La DIT, ayant compris cet enjeu, pourra adopter une ou plusieurs de ces méthodologies pour rendre effective l'agilité dans ses équipes. La prochaine partie de ce projet recherche appliquée présentera une analyse des différents cycles de vie de la DIT, ainsi qu'une proposition de méthodologie, processus et rôles implémentable pour une adoption efficace de l'agilité.

CHAPITRE 2

PASSAGE À L'AGILITÉ

2.1 Introduction

L'agilité, bien que vendue comme simple et pratique, n'est pas si simple à mettre en place. En plus des nouveaux rôles, processus et potentiellement d'acquérir de nouveaux outils, il faut aussi revoir l'impact de son introduction dans la structure organisationnelle, la culture, et la gestion du changement/formation du personnel impacté. Ce chapitre propose une réorganisation de la structure organisationnelle de la DIT, ainsi que l'ensemble de ses processus actuels et outils qui devront être déployés dans le but d'intégrer progressivement et de manière efficace l'agilité dans les projets hautement technologiques.

2.2 Structure organisationnelle

Cette section aborde la structure organisationnelle que la DIT pourrait mettre en place pour faciliter l'adoption de l'agilité. En effet, seuls les processus et les outils ne suffisent pas à rendre un projet organisationnel agile. La réorganisation et la formation des équipes sont nécessaires, afin de répandre dans toute l'organisation, la culture de l'agilité. Il sera donc question pour la DIT de revoir la structure organisationnelle, la composition et l'organisation des différentes équipes de projets, ainsi que la relation entre le management et les clients.

2.2.1 Composition des équipes agiles

La première valeur du Manifeste agile met l'accent sur les individus et leurs interactions. L'équipe joue donc un rôle primordial et est responsable de la conception et de la livraison du produit. Les décisions liées au travail quotidien du projet sont prises par l'équipe de manière autonome, sans trop d'interférence de la Direction. Les projets agiles sont confiés aux équipes et non à un individu, mais on doit s'assurer qu'il n'y a pas de dérive. C'est l'équipe qui décide de comment travailler, quelle direction prendre pour être plus efficace et s'améliorer

continuellement. La Direction générale confie un projet à l'équipe, qui doit identifier ses membres et travailler de manière autonome. Ces équipes de projets sont constituées des individus qui travaillent ensemble pour atteindre un objectif commun à savoir livrer des produits qui satisfont les besoins du client. Les membres de l'équipe doivent se partager les connaissances, afin d'éviter la problématique du « single point of failure », qui peut être un frein à l'agilité. La constitution d'une équipe de projet doit tenir compte de plusieurs éléments:

- La taille : c'est un facteur très important dans un environnement agile. Plus l'équipe est grande, plus il y a des risques de blocages, notamment dans la prise de décision. Dans un environnement agile, la taille optimale des équipes doit osciller entre quatre et dix personnes. Il faut si possible qu'elle ne soit ni trop petite pour qu'une dynamique de groupe puisse apparaître, ni trop grande pour éviter que des clans n'émergent et que la communication informelle devienne trop complexe (Scalla, 2018, p. P.53);
- Les compétences : L'équipe doit être idéalement pluridisciplinaire (Scalla, 2018, p. P.53). Des compétences variées (c.-à-d. marketing, vente. Opérations, concepteurs, développeurs, testeurs, clients) sont utiles, pour produire de manière itérative des livrables de meilleures qualités et qui répondent bien à un besoin réel;
- Les membres : Les membres de l'équipe doivent intégrer la culture de l'agilité (c.-à-d. le travail collaboratif et l'amélioration continue). La DIT devra soumettre l'ensemble des membres des équipes à une formation concernant la culture de l'agilité. Les membres d'une équipe doivent travailler aussi dans un environnement qui facilite des rencontres en présentiel, et qui favorise les communications informelles. Certains détails de la communication peuvent se perdre, si on se contente de communiquer par courriel ou par téléphone, car la communication face à face rend les échanges entre personnes plus dynamiques et interactives.

La section suivante présente une proposition d'organigramme typique des rôles d'une équipe de projet pour la DIT.

2.2.2 Organisation des nouveaux rôles

L'un des grands défis de la DIT sera de réussir la transition du point de vue du rôle de gestion de projet. En effet dans un environnement agile, un gestionnaire de projet n'a pas le même rôle que dans les méthodologies traditionnelles, car on ne parle plus de gestion de projet, mais plutôt de gestion de produit. Dans la méthodologie Scrum par exemple, le rôle de gestionnaire de projet est partiellement assuré par le chef de produit et le maître de mêlée. Le guide de Scrum ne définit que les fonctions de gestionnaire de projet relatives aux rôles de maître de mêlée et de chef de produit. Toutes les fonctions d'un gestionnaire de projet ne sont donc pas clairement définies dans ce guide. La DIT pourra en plus des différents rôles du Scrum, redéfinir le rôle de gestionnaire de projet, pour des fonctions de gestion de projet qui ne sont pas prises en compte par le Scrum, de manière à éviter tout conflit avec le rôle de chef de produit ou de maître de mêlée (Shastri, Rashina, & Robert, s.d.). En d'autres termes les gestionnaires de projets actuels doivent transférer toutes les responsabilités prévues dans le cadre du Scrum au chef de produit ou au maître de mêlée, et ne conserver que les activités de gestion qui ne sont pas définies dans le cadre du Scrum. La figure ci-dessous est une proposition d'organigramme pour les différents rôles.

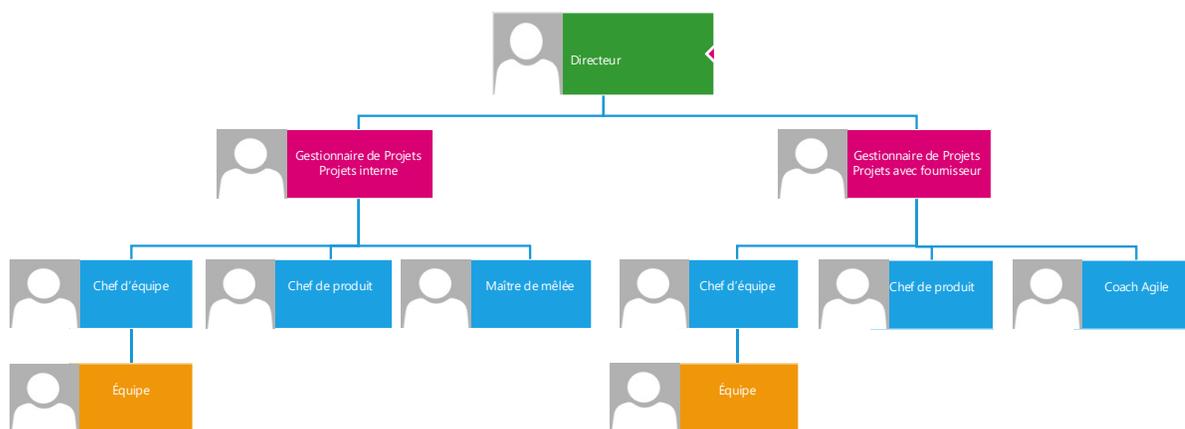


Figure 2.1 : Organigramme des rôles

Le gestionnaire de projet, dans cette nouvelle approche agile, pourra entre autres gérer le contrôle de l'évolution et des statuts du/des projets sous sa responsabilité, effectuer les

planifications budgétaires, gérer du personnel, négocier des ressources à allouer aux projets, et s'occuper de la logistique liée aux projets, qui sont toutes des fonctions omises dans le guide Scrum typiquement (Shastri et al., s.d.).

2.2.3 Gestion des nouveaux rôles

Cette section va aborder les nouveaux rôles que la DIT devrait considérer afin de passer à l'agilité. Une première recommandation pour la DIT est de créer de nouveaux rôles pour chaque produit bancaire. Ceci pose les problématiques suivantes :

- Qu'est-ce qu'un produit bancaire?
- Comment définir ces nouveaux rôles auprès des ressources humaines?
- Quel est l'impact de ces nouveaux rôles sur les processus existants, incluant la DIT?
- Quelle est la stratégie de formation/assignation du personnel impacté et qui devront jouer ces nouveaux rôles?

Un produit bancaire peut être considéré comme tout ce qui est livré à un client (c.-à-d. externe) à l'issue d'un processus de développement ou de maintenance logiciel. Il est clair qu'un projet de transformation numérique touche plusieurs organisations internes dans une banque et ainsi nécessite qu'un chef de produit possède une vision transversale des activités bancaires pour un produit donné. Pour chaque produit bancaire, il sera nécessaire de définir comment ce nouveau rôle interagira avec le personnel du bureau de projet de la DIT, et les prérequis nécessaires, avant que ces employés puissent jouer ce rôle pleinement.

L'un des plus gros enjeux pour la DIT sera de mettre sur pied un modèle opérationnel fiable d'un projet agile, qui prend en compte certaines de meilleures pratiques de gestion de projet (c.-à-d. provenant du PMBOK par exemple), une méthodologie agile (c.-à-d. Scrum ou Kanban par exemple), et un modèle ou standard d'assurance qualité et d'amélioration continue (c.-à-d. CMMI ou ISO9001 par exemple). La définition des nouveaux rôles devra tenir compte du modèle opérationnel choisi, car, pour la méthodologie Scrum par exemple, il sera question de savoir clairement, quelle sont les fonctions d'un gestionnaire de projets/analyste

d'affaires/maître de mêlée, et celle d'un chef de produit, car il pourrait avoir conflit entre ces différents rôles. De même, il faudra aussi par exemple définir clairement les fonctions des responsables d'assurance qualité, risques, audit interne et améliorations continues au niveau du processus, par rapport aux livrables obligatoires des projets. Avant de déployer les nouveaux processus et les nouveaux rôles, la DIT doit d'abord clairement définir son modèle d'affaires opérationnel, pour ce qui est de la gestion des projets de développement logiciel à l'avenir.

Les sous-sections suivantes présenteront les différents rôles, ainsi que les stratégies recommandées pour les mettre en place et faciliter leurs implémentations.

2.2.3.1 Le chef de produit

Afin de bien jouer son rôle, un chef de produit doit posséder une très bonne connaissance du produit bancaire dont il est responsable, démontrer une très bonne connaissance des requis de la clientèle, connaître les tendances d'informatisation de ces produits dans son industrie et chez ses concurrents. À l'aide de ces compétences, il/elle sera capable d'aider l'équipe de développement à traduire les exigences/requis des clients en requis techniques implémentables de manière itérative (c.-à-d. progressivement et logiquement). Autrement dit, le chef de produit bancaire assure désormais un rôle d'analyste d'affaires spécialisé dans son produit bancaire, en plus d'effectuer d'autres fonctions dans son unité d'affaires et aussi prévues dans le Scrum d'un projet. Il sera donc plus facile pour la DIT de promouvoir les analystes d'affaires comme chef de produit, en leur procurant une formation adéquate pouvant leur permettre de comprendre toutes les autres fonctions d'un chef de produit, afin d'exercer pleinement ce nouveau rôle, et de pouvoir former plus tard d'autres chefs de produit.

Pour la DIT, un chef de produit pourra être responsable de :

- Collaborer lors de la préparation de la charte du projet ou du cahier des charges;
- Piloter la phase d'analyse du projet, en coordonnant les différents processus de cette étape;
- Être responsable de discuter et prioriser les items de la « mêlée du produit »;

- Aider lors des décisions techniques/fonctionnelles et autres liées au produit, d'un commun accord avec l'équipe de développement;
- Soumettre à l'approbation du comité de direction de projet, les décisions administratives ou financières;
- Être responsable des scénarios de validation pendant le processus de test d'acceptation. Il devrait, en principe, remplacer/complémenter le rôle d'un directeur ou responsable de département opérationnel dans ce processus. Le fait pour la hiérarchie du département d'être impliqué dans ce processus peut constituer un frein à l'agilité, car elle n'est pas proche de l'équipe de développement, et en plus, les décisions ou validations peuvent paraître plus longues, et alourdir ainsi le processus;
- Être responsable de toutes les validations pendant le processus de mise en production/service.

La DIT devrait aider les chefs de produits nouvellement promus à acquérir des compétences pour ce rôle à travers des formations certifiables comme le PMI-ACP (Project Management Institute- Agile Certified Professional) ou le PSPO (Professional Scrum Product Owner).

2.2.3.2 Le maître de mêlée

Le maître de mêlée est un rôle clé pour les projets utilisant une approche Scrum/Kanban. Le succès ou l'échec d'un Scrum repose en grande partie sur les aptitudes du maître de mêlée. Pour la DIT, certains gestionnaires de projets, analystes d'affaires ou responsables des équipes de développement actuel seront de bons candidats pour occuper ce nouveau rôle, car le maître de mêlés est surtout un rôle de gestion. Pour les processus agiles du Scrum, le maître de mêlés pourra être responsable de :

- Appuyer le chef de produit dans ses activités d'analyses de démarrage de projet;
- Promouvoir et supporter le Scrum à la DIT, en aidant tous ceux qui sont impliqués dans les processus Scrum, à comprendre la théorie, les pratiques, les règles et les valeurs de Scrum;

- Planifier et coordonner les différentes réunions de Scrum (Réunion de raffinement, de planification de Sprint, de revue du Sprint, de rétrospective de Sprint, ainsi que la réunion quotidienne de Scrum);
- Aider l'équipe à adopter les bonnes pratiques dans les processus Scrum.

La DIT devrait aider les maîtres de mêlées, nouvellement promus, à acquérir des compétences pour ce nouveau rôle à travers des formations certifiables comme : le PMI-ACP (Project Management Institute- Agile Certified Professional), ou le PSM (Professional Scrum Master).

2.2.3.3 Coach agile

Pour aider les équipes à accepter et adopter l'agilité, il est recommandé que la DIT crée un rôle de coach agile dans son bureau de projet. Certains chefs d'équipes, chargés de projets et/ou analystes d'affaires actuels pourraient être de bon candidat pour assumer ce rôle, à condition de recevoir une formation appropriée. Un coach agile pourra être responsable de:

- Aider les équipes à comprendre, en détail, comment les processus agiles de la DIT fonctionnent, comment être plus productif et adopter une culture d'amélioration continue dans les projets technologiques;
- Aider les membres des équipes à s'impliquer dans les décisions concernant la manière de travailler, et dans les améliorations au niveau des processus, car une équipe agile doit être constituée des personnes qui comprennent les enjeux de l'agilité, et qui se sentent directement concernées par la direction que l'équipe doit prendre;
- Aider l'équipe à adopter les bonnes pratiques.

Il est aussi nécessaire de définir comment ce nouveau rôle sera mis en œuvre au niveau du personnel du bureau de projet de la DIT et les prérequis nécessaires avant que les employés puissent jouer ce rôle pleinement. La DIT devrait aider les coaches agiles nouvellement promus à aller chercher des compétences à l'aide de formations certifiables comme : le PMI-ACP

(Project Management Institute - Agile Certified Professional) ou le PAL (Professional Agile Leadership).

2.2.3.4 L'équipe de développement

Une autre recommandation concerne la transformation du mode de fonctionnement des projets de la DIT vers l'agilité. Quels seront les impacts sur les rôles, les cycles de vie, les activités/livrables? L'équipe de développement sera au cœur de cette transformation et sera responsable de:

- Exécuter les tâches à travers un tableau Scrum/Kanban, ou les tâches planifiées à travers l'arrière du Sprint;
- Prendre toutes les décisions concernant la manière de travailler de l'équipe;
- S'impliquer dans l'amélioration continue des processus de l'équipe;
- Faire l'estimation des tâches et définir les critères de performances.

Cela pose aussi la question, comment seront réparties les rôles entre le bureau de projet et les équipes de développement, quel seront les outils/formations requises?

Une recommandation à la DIT pour ce rôle est d'aider les développeurs à aller chercher des compétences pour ce rôle à l'aide de formations certifiables comme : le PMI-ACP (Project Management Institute - Agile Certified Professional), ou PSD (Professional Scrum Developer).

2.2.4 Impact des nouveaux rôles sur les processus existant de la DIT

Cette section va aborder l'impact des nouveaux rôles sur les processus actuels des projets de la DIT.

Pour les processus qui seront agiles par la méthodologie Scrum, le cycle de vie actuel devra être ajusté afin de tenir compte des rôles et livrables du cycle de vie Scrum. Dans ce dernier, on n'investit pas autant d'effort au début d'un projet pour effectuer une analyse et une conception complète du logiciel avant de commencer le développement. À la Sogebank ceci n'est pas un problème, car il a été observé que les projets passent trop rapidement de l'idée à

la réalisation. Donc une étape d'analyse judicieuse effectuée par le chargé de projet et le chef de produit sera importante pour les grands projets de transformation numérique à l'avenir. Une fois que les requis des clients sont bien documentés et compris par l'équipe de démarrage de projet (c.-à-d. le chargé de projet (maître de mêlée/analyste d'affaires) et le chef de produit), on commence le développement itératif, avec pour objectif de livrer une ou des fonctionnalités à l'issue de chaque itération. Les différents rôles définis dans le processus Scrum sont différents des rôles définis dans le processus actuel. La DIT devra remplacer les rôles des processus actuels par des rôles Scrum, ce qui suppose un ajustement au niveau des équipes, et donc une formation de l'ensemble des employés impactés et qui seront appelé à jouer ces nouveaux rôles.

Concernant les processus d'acquisition de logiciel et ceux qui impliquent des fournisseurs externes, ils pourraient aussi être revus ultérieurement pour y insérer plus d'agilité.

Ainsi, le chef de produit pourra désormais piloter les processus d'analyse, de test d'acceptation et de mise en production en vue de réduire les effets de silos et les engorgements de processus actuels de projets. Il pourra avoir la vue d'ensemble de plus d'un CDR (Centre De Revenu) dans le processus d'analyse, de réalisation, de test d'acceptation, et de préparation à la mise en production (c.-à-d. préoccupations opérationnelles, communications, formation...). La section suivante présentera la stratégie de migration du personnel impacté par l'implémentation des nouveaux rôles.

2.2.5 Stratégie de migration du personnel impacté

Cette section va aborder la gestion des nouveaux rôles au niveau des ressources humaines et la stratégie de migration du personnel impacté.

Pour introduire avec succès le nouveau rôle de chef de produit, il sera nécessaire de réorganiser le rôle d'analyste d'affaires afin de l'intégrer à la fonction de chargé de projets. Le chargé de projet jouera donc le rôle d'analyste d'affaires TI et maître de mêlées et certaines fonctions seront désormais assurées par le chef de produit. Il faudra aussi déterminer le rôle du coach

agile au bureau de projet. Cette modification des fonctions actuelles regroupera des rôles ce qui demandera une formation du personnel.

Pour s'assurer de la motivation réelle des personnes devant occuper les nouveaux rôles, la DIT pourrait publier des offres relatives à ces rôles en interne, et passer les candidats en entrevue pour s'assurer de leurs motivations. Une fois ces personnes sélectionnées, la DIT pourrait organiser plusieurs ateliers, au cours desquels des simulations pourraient être faites pour les préparer par rapport aux nouvelles fonctions et aux nouveaux processus. Pour chaque produit de la banque, la banque devra nommer un chef de produit. Lors du démarrage d'un projet, un chef/chargé de projet (qui joue le rôle d'analyste d'affaires TI + maître de mêlée) sera assigné. Le chargé de projet devra créer une équipe de projet incluant une équipe de développement de la DIT et confirmer le cycle de vie qu'il suivra pour la réalisation de ce produit. Une autre recommandation à la DIT, est de procéder avec une approche de transition progressive vers l'agilité, en effectuant les étapes suggérées suivantes :

- Retravailler les processus/rôles/livrables de cycle de vie actuels (pour les rendre agiles), les documenter et les déployer (c.-à-d. les communiquer aux intervenants) ;
- Continuer le déploiement des outils de gestion de projets agiles nécessaires pour utiliser ces processus dans les équipes de la DIT et les participants tels que les chefs de produits;
- Sélectionner quelques produits dont le développement n'est pas très complexe pour la banque afin de faire des essais graduels;
- Constituer une équipe de personnes qui maîtrise bien le développement de ces produits;
- Nommer un chef de produit et un chargé de projet et un coach agile pour ces projets pilotes.
- Expérimenter l'agilité uniquement sur ces produits dans un premier temps;
- Faire une évaluation des processus/rôles/outils/livrables et apporter au fur et à mesure des correctifs nécessaires pour les améliorer;
- Introduire progressivement d'autres produits et d'autres équipes;
- Évaluer la maturité des processus et appliquer les améliorations nécessaires;
- Répandre l'agilité dans toute l'organisation.

La section suivante abordera la notion de culture d’entreprise dans un environnement agile.

2.2.6 Culture d’entreprise en agilité

Dans un environnement agile, l’esprit d’équipe et la transparence sont de rigueur. La DIT devra définir une politique pour chacun élément suivant, et former les employés concernant: la culture du respect, de la collaboration, de la qualité et de l’apprentissage. La section suivante présentera une analyse des cycles de vie de la DIT.

2.3 Analyse des Cycles de vie de la DIT

Cette section présentera une analyse des différents cycles de vie de développement logiciel de la DIT afin de déterminer la méthodologie agile la plus appropriée pour une transition vers l’agilité.

2.3.1 Analyse des cycles de vies

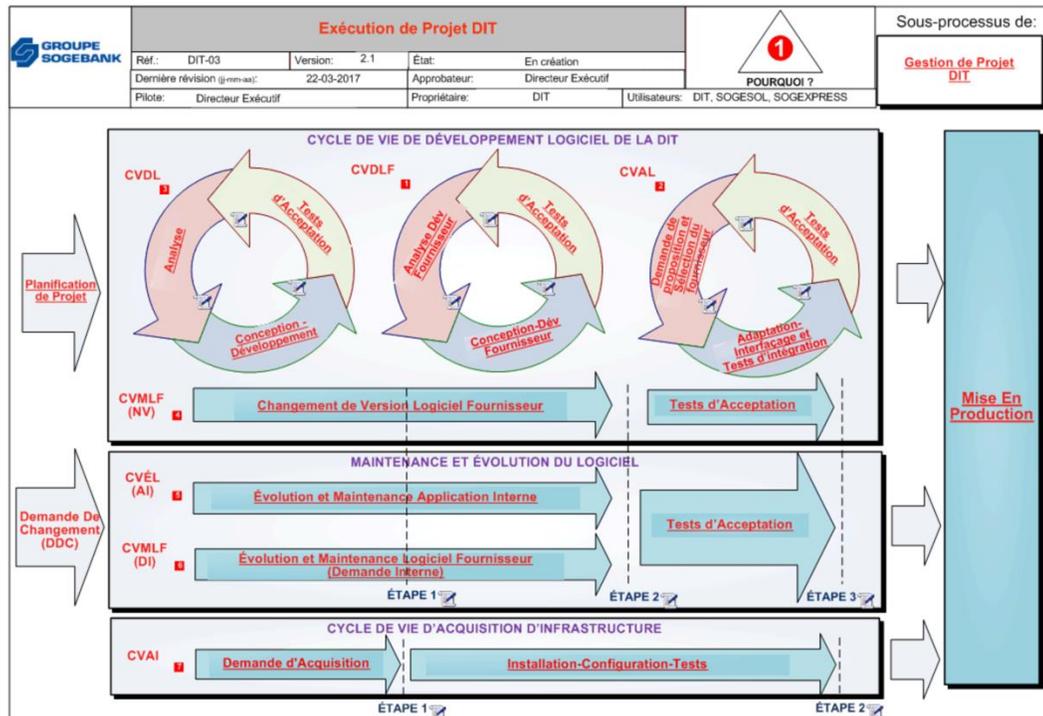


Figure 2.2 : Référentiel des processus de la DIT

La figure 2.2 présente les différents cycles de vie logiciels utilisés par les projets et qui sont cartographiés en détail actuellement à la Sogebank. Pour ces différents cycles de vie, plusieurs processus sont exécutés avec des fournisseurs externes de la DIT. Les processus d'analyse et de développement sont exécutés par les fournisseurs, et les processus de validation du contrat par la DIT, et la direction des affaires juridiques. Il est donc difficile de rendre tous ces processus de cycles de vie agile dans un premier temps.

La DIT a un meilleur contrôle sur la CVDL, car l'ensemble des processus de ce cycle de vie sont exécutés par des acteurs internes à la banque, et n'impliquent pas de fournisseurs externes. Ce cycle de vie peut donc être remplacé par un Cycle de Scrum. En d'autres termes, les logiciels de la Sogebank conçus et maintenus en interne qu'ils soient pour un client interne ou externe seraient de bon candidat pour l'agilité par le Scrum. Cela dit, l'entreprise devra être prête à revoir complètement ses processus et les différents rôles et s'approprier des outils essentiels de gestion des configurations, pour s'accommoder à l'agilité par le Scrum.

La section [2.4](#) permettra de proposer à la Sogebank un nouveau cycle de vie, basée sur le Scrum. Cependant, si la DIT n'est pas prête à revoir ses processus pour l'adapter au Cycle de vie Scrum, elle pourra adopter comme alternative au Scrum, le Scrumban, qui peut être adapté au processus actuel avec l'ajout de quelques éléments Scrum et un tableau Kanban. Un processus Scrumban sera proposé à la DIT dans la section [2.5](#) pour le CVDL.

Il est donc recommandé de démarrer la transition vers l'agilité avec le cycle de vie CVDL qui est utilisé pour les projets internes qui n'impliquent pas de fournisseurs externes. Quand la banque aura plus de maturité, elle pourra penser à faire évoluer les processus des autres cycles de vie vers l'agilité.

2.3.2 Processus actuels du CVDL et Agilité.

Les différents processus actuels du CVDL seront analysés dans cette section par rapport à une transition vers le Scrum ou le Scrumban.

2.3.2.1 Processus d'Analyse actuel du CDVL

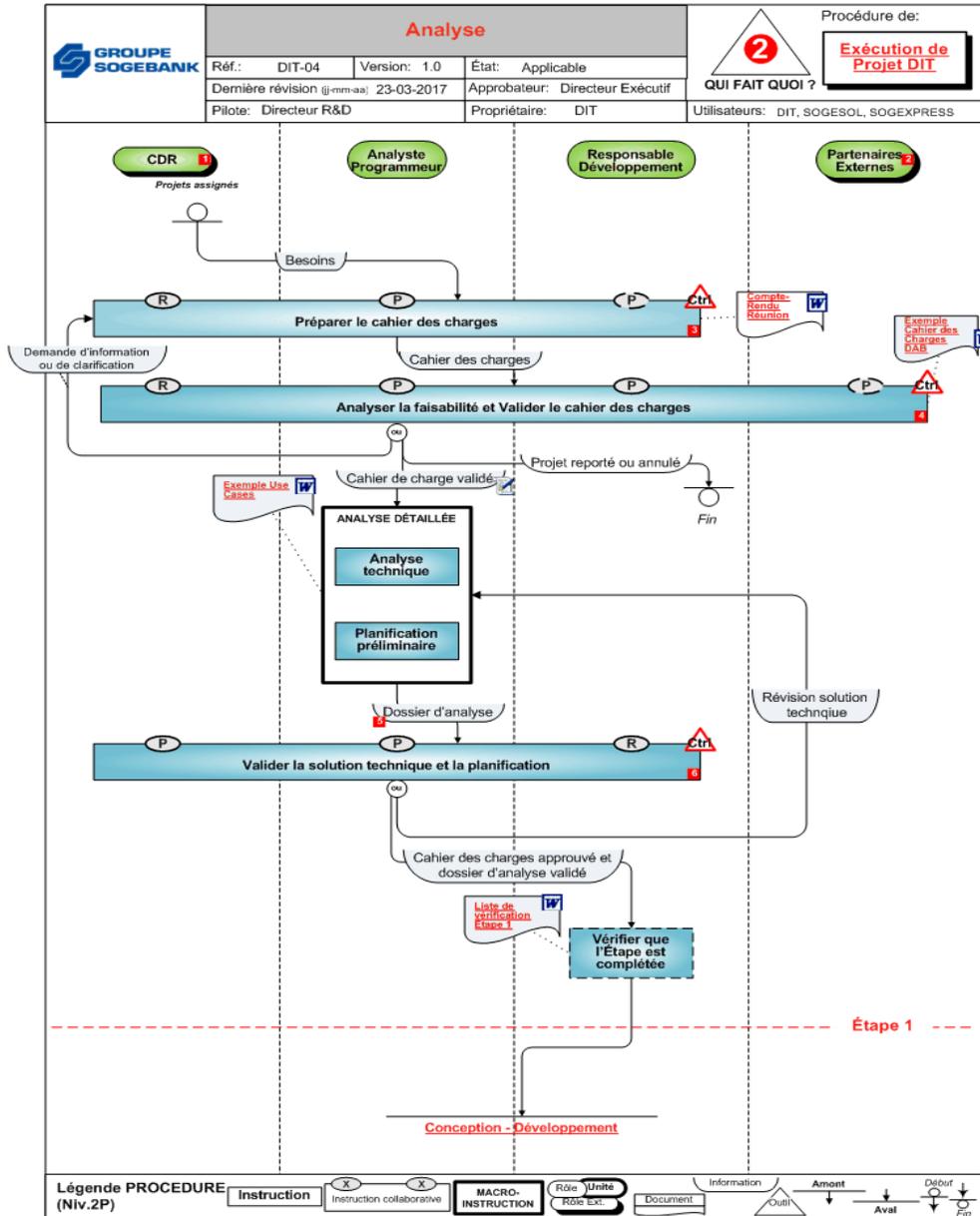


Figure 2.3 : Processus d'Analyse actuel du CVDL

Avec l'adoption du Scrum, ce processus devrait être désormais remplacé par le processus de création de la charte de projet et de documentation des requis. Les nouveaux rôles agiles devraient également être mis en place tels que présentés dans la section 2.4. Le cahier des charges pourra désormais être remplacé par le la charte de projet, qui est document beaucoup plus agile, moins complexe, et efficace pour décrire le projet d'une manière simple. En effet,

la DIT devrait également penser à introduire l'agilité dans ces différents supports de travail, à travers le choix d'une documentation simple, mais efficace. Avec l'adoption du Scrum, ce processus pourra être conservé, avec de légères modifications, notamment l'ajout de quelques éléments Scrum : mise en place des nouveaux rôles (chef de projet, coach agile et l'équipe de développement), et simplification de l'étape d'analyse (documentation des requis, création des « users stories », et mise en place de l'arrière de produit).

2.3.2.2 Processus de Conception-Développement actuel du CVDL

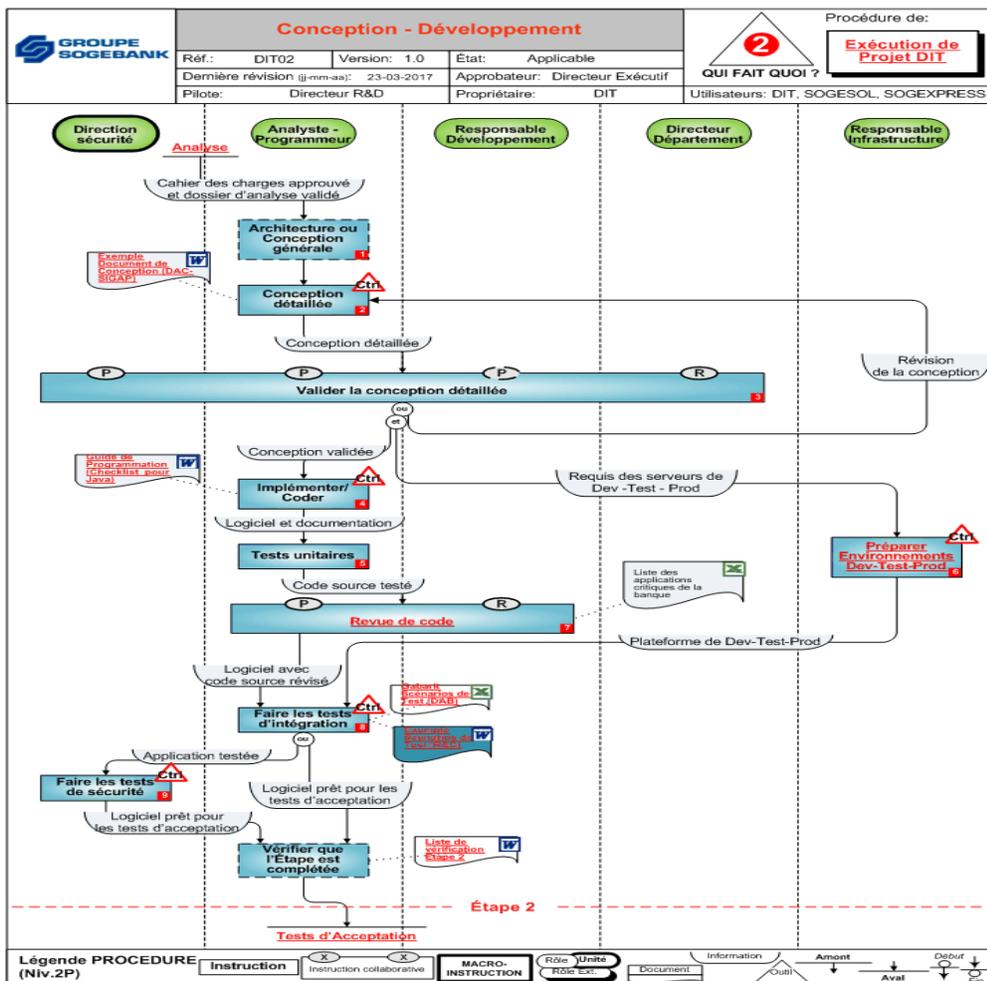


Figure 2.4 : Processus de Conception Développement actuel du CVDL

Avec l'adoption du Scrum, ce processus devrait être désormais remplacé par une partie du processus de développement du produit tel que présenté dans la section 2.4. Le développement se fera désormais dans des Sprints avec comme principaux acteurs les différents rôles Scrum. Avec l'adoption du Scrumban, ce processus pourra être conservé, avec de légères modifications, notamment la mise en place des nouveaux rôles et du tableau Kanban qui permettra à l'équipe de visualiser et de compléter les tâches suivant un ensemble de règles qui seront mises en place, et un flux de travail prédéfini, tel que présenté dans la section 2.5.

2.3.2.3 Processus de préparation du test d'acceptation actuel du CVDL

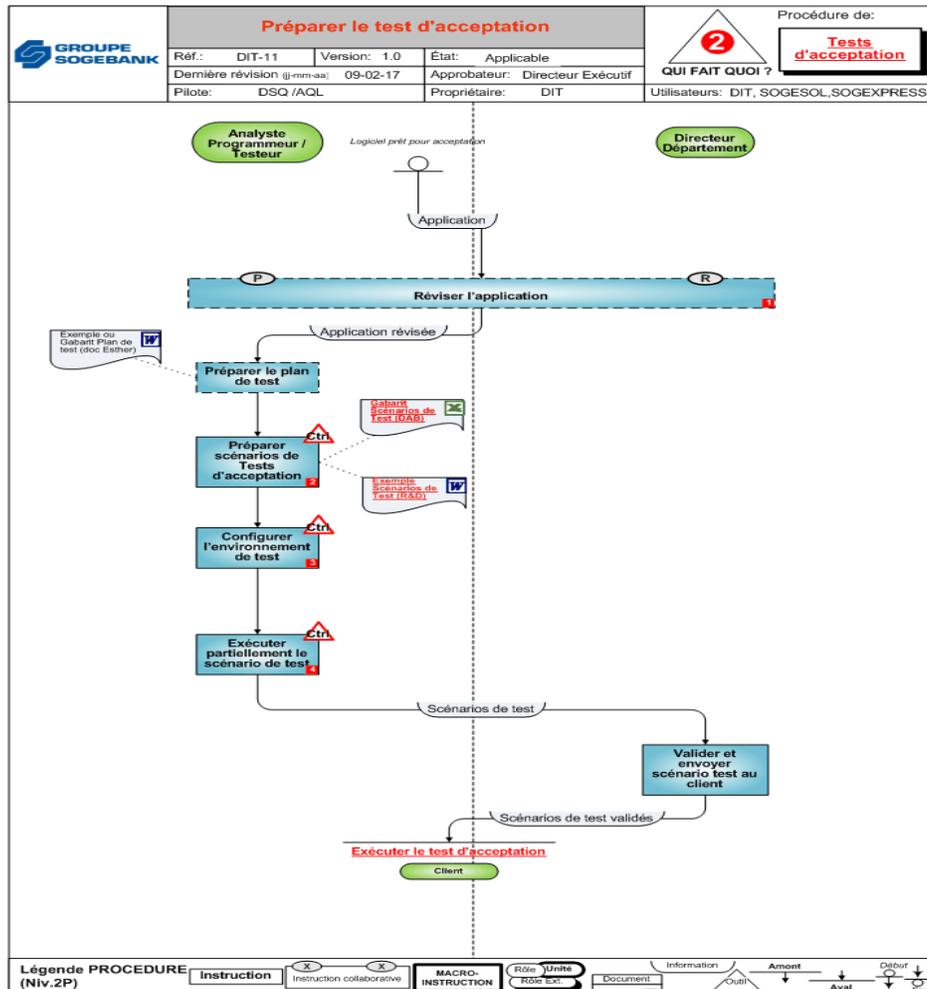


Figure 2.5 : Processus préparation du test d'acceptation actuel de la DIT

Avec l'adoption du Scrum ou du Scrumban, ce processus pourra être conservé, à la seule différence que, pour la rendre agile, le chef de projet devrait désormais remplacer le directeur du département pour toutes les validations, car, le fait pour la hiérarchie du département d'être impliqué dans ce processus peut constituer un frein à l'agilité.

2.3.2.4 Processus d'exécution du test d'acceptation actuel du CVDL

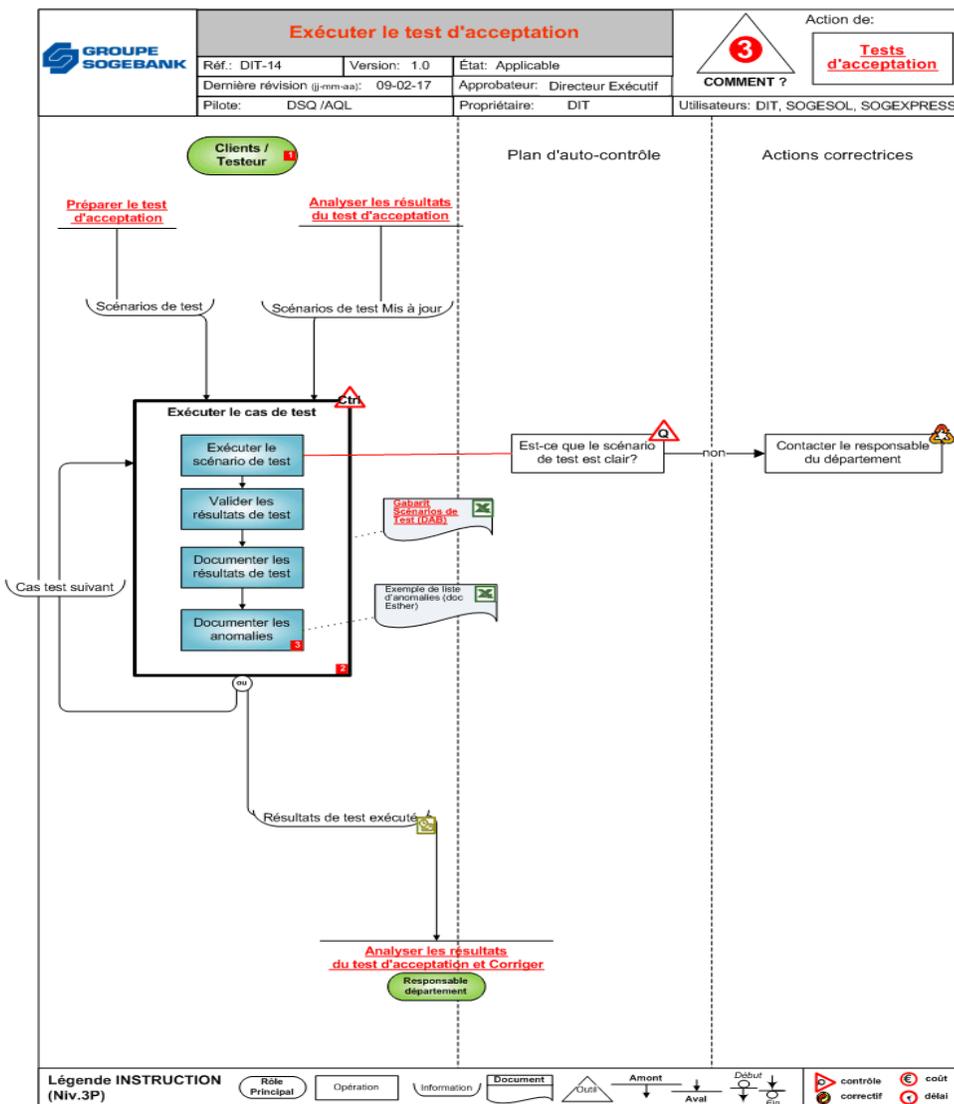


Figure 2.6 : Processus exécuter test d'Acceptation actuel de la DIT

Avec l'adoption du Scrum ou du Scrumban, ce processus pourra être conservé. Aucun changement majeur ne sera alors requis.

2.3.2.5 Processus d'analyse des résultats du test d'acceptation actuel du CVDL

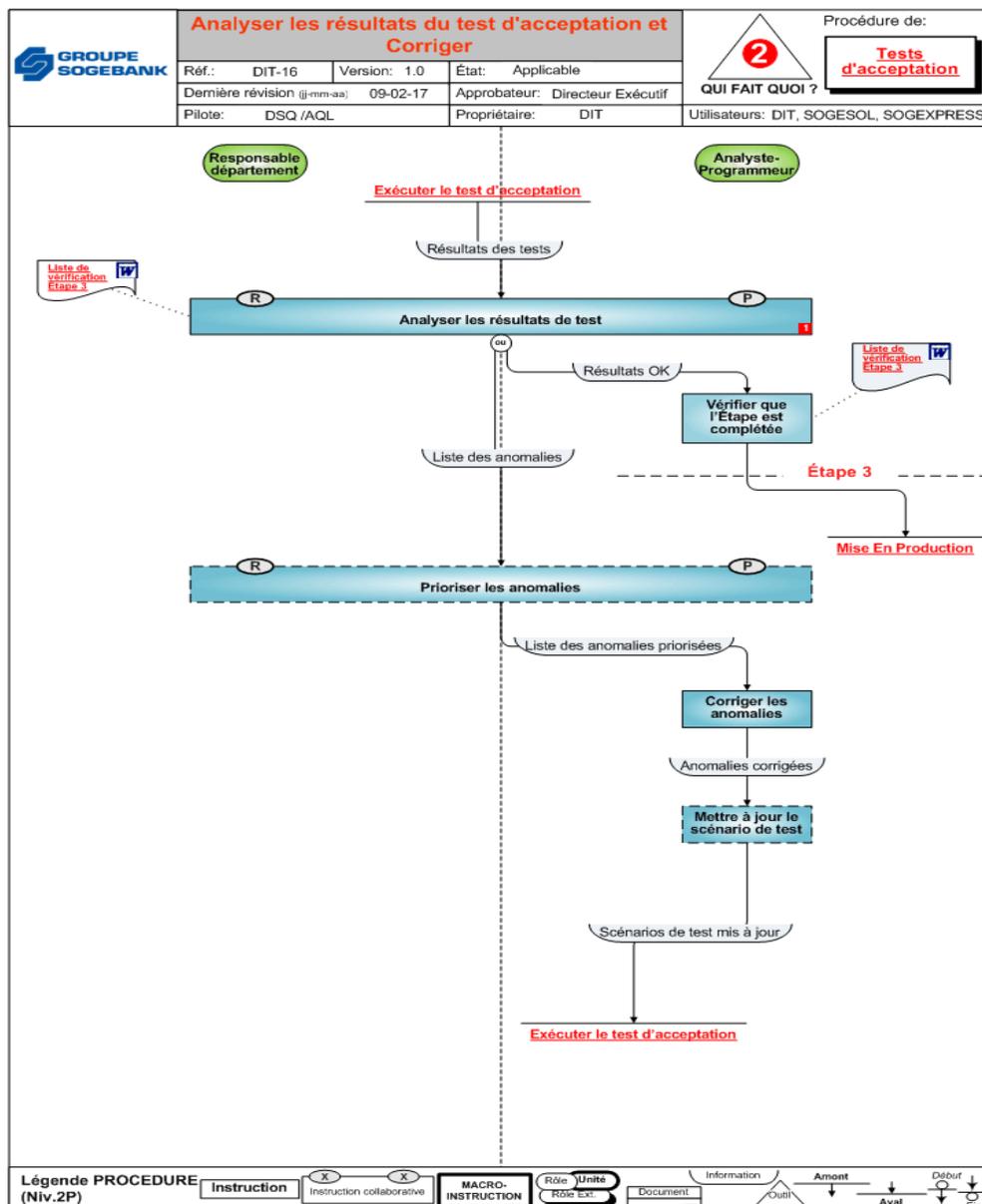


Figure 2.7 : Processus d'analyse des résultats du test d'acceptation actuel de la DIT

Avec l'adoption du Scrum ou du Scrumban, ce processus pourra être conservé, à la seule différence que, pour la rendre agile, le chef de projet devrait désormais remplacer le responsable du département dans toutes les étapes, car, le fait pour la hiérarchie du département d'être impliqué dans ce processus peut constituer un frein à l'agilité.

2.3.2.6 Processus de mise en production actuel du CVDL

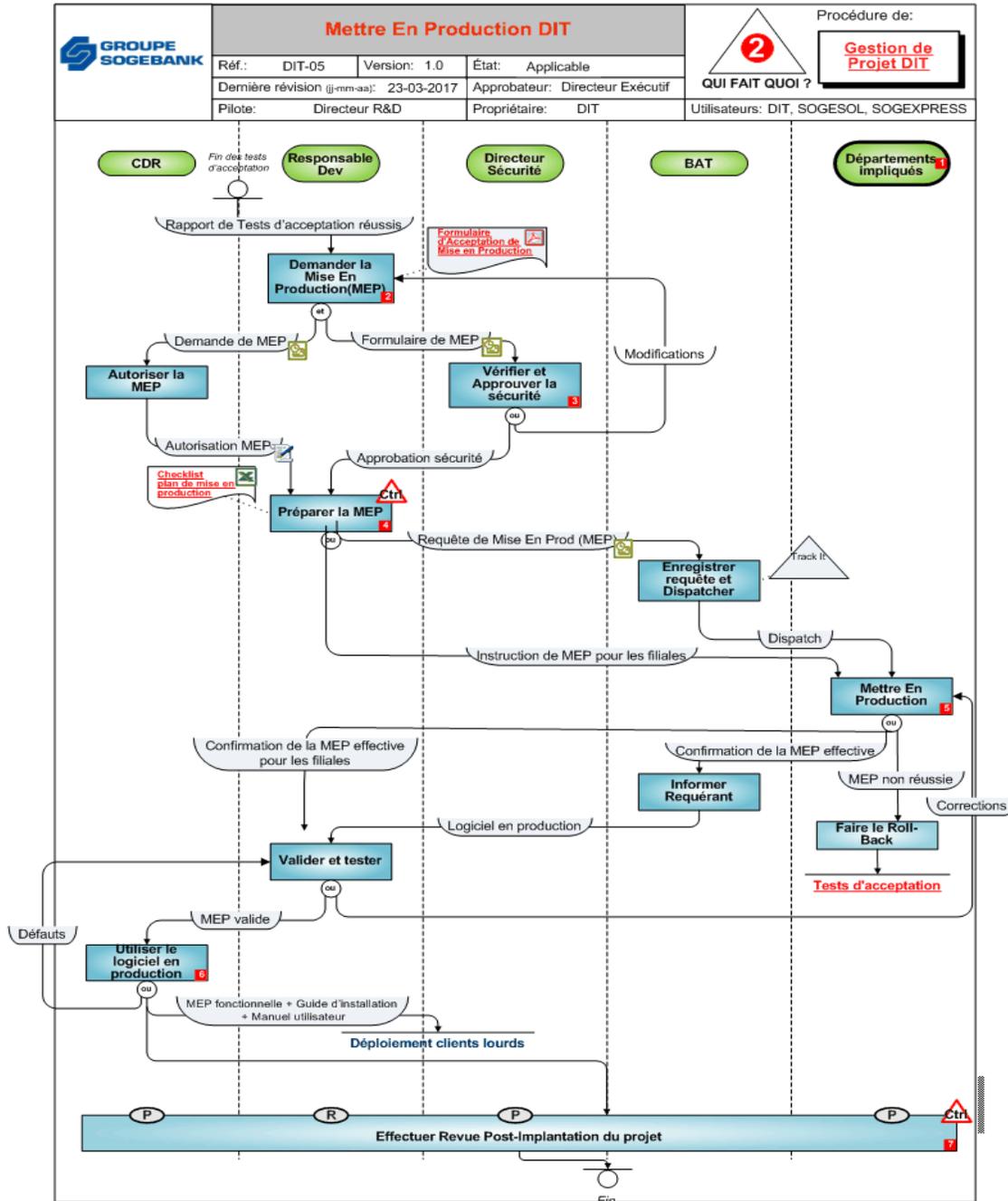


Figure 2.8 : Processus de mise en production actuel de la DIT

Avec l'adoption du Scrum ou du Scrumban, ce processus pourra être conservé, aucun changement majeur ne sera alors requis. Dans la section suivante, les différents processus du cycle de vie Scrum seront présentés.

2.4 Cycle de Vie Scrum pour le CVDL

Toutes les applications développées et supportées en interne pourront suivre le cycle vie Scrum présenté à la figure 7 ci-dessous. Pour ce cycle de vie, tout nouveau projet de développement logiciel, commence par l'établissement de la charte des projets, puis de la documentation des requis. Par la suite, les procédures du Scrum sont les mêmes, qu'on soit dans nouveau logiciel, ou une simple maintenance applicative.

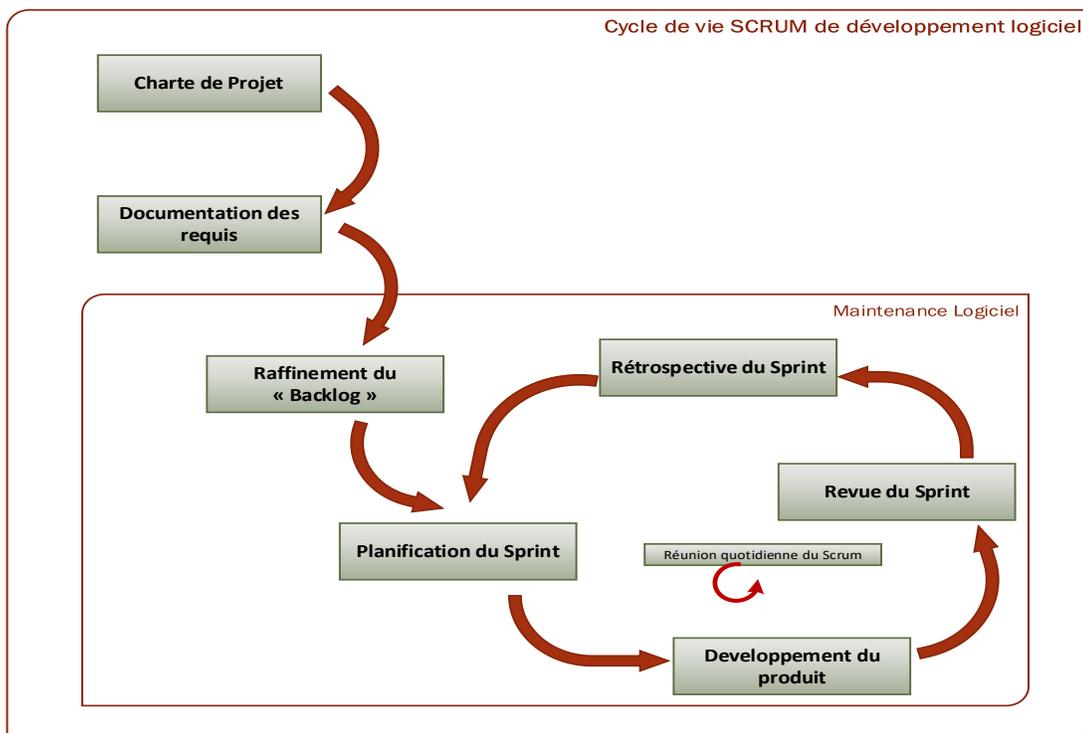


Figure 2.9 : Cycle de vie Scrum de développement logiciel

L'ensemble des procédures présentées dans les étapes de ce cycle de vie seront détaillées dans les sous-sections suivantes.

2.4.1 Processus de création de la Charte de Projet

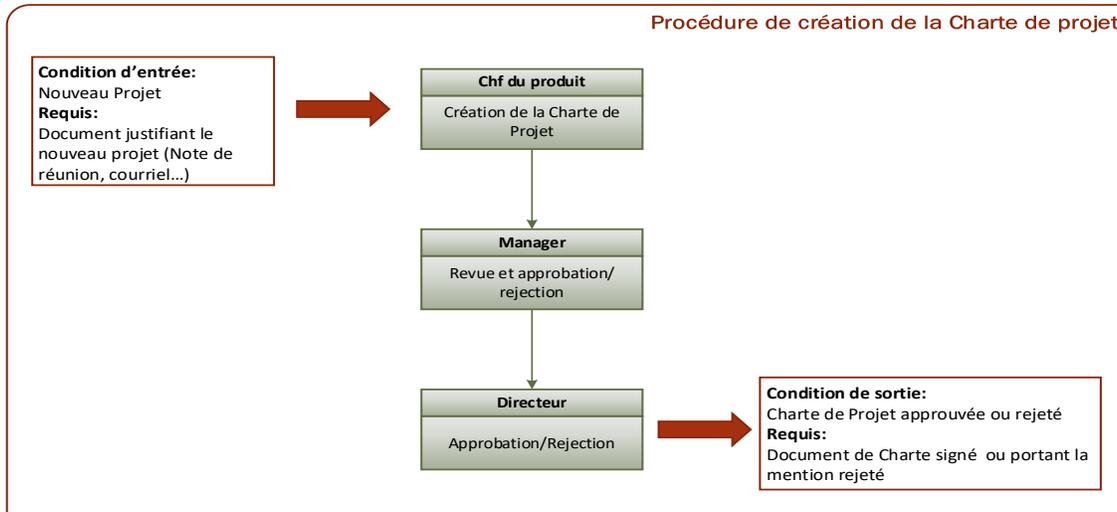


Figure 2.10 : Procédure de création de la charte de projet.

Le processus de création de la Charte de Projet est le seul qui requiert l'intervention de la haute hiérarchie. Dès que le projet est planifié et approuvé par l'équipe de gestion des projets, la direction de la DIT nomme le chef de produit qui sera le principal responsable du projet. Il documente la Charte de Projet, et le soumet pour revue et approbation aux parties prenantes, et ensuite au directeur du département des TI. À l'issue de ce processus, la charte du projet est soit approuvée ou rejetée. Dans un environnement agile, ce document devrait être le plus simplifié possible. Un exemple de Gabarit de Charte de Projet qui pourrait inspirer la DIT se trouve en [Annexe V](#).

2.4.2 Processus de documentation des requis

Une fois la charte de projet approuvée, le chef de produit procède à la documentation des requis. Il a la responsabilité de documenter les requis du client dans un l'outil de gestion des configurations. Ensuite, il fait une réunion de démarrage du projet pour présenter le projet aux parties prenantes (requis, jalons, responsabilités ...). Une fois la réunion de démarrage terminée et le projet créé dans l'outil de gestion des configurations, il crée l'arrière du produit, regroupant l'ensemble des fonctionnalités de l'application, sous forme d'histoires ou de cas

d'utilisation (de l'anglais « Users Stories ») identifiables et compréhensibles, en définissant la priorité de chacune (exemple en [Annexe I.](#)).

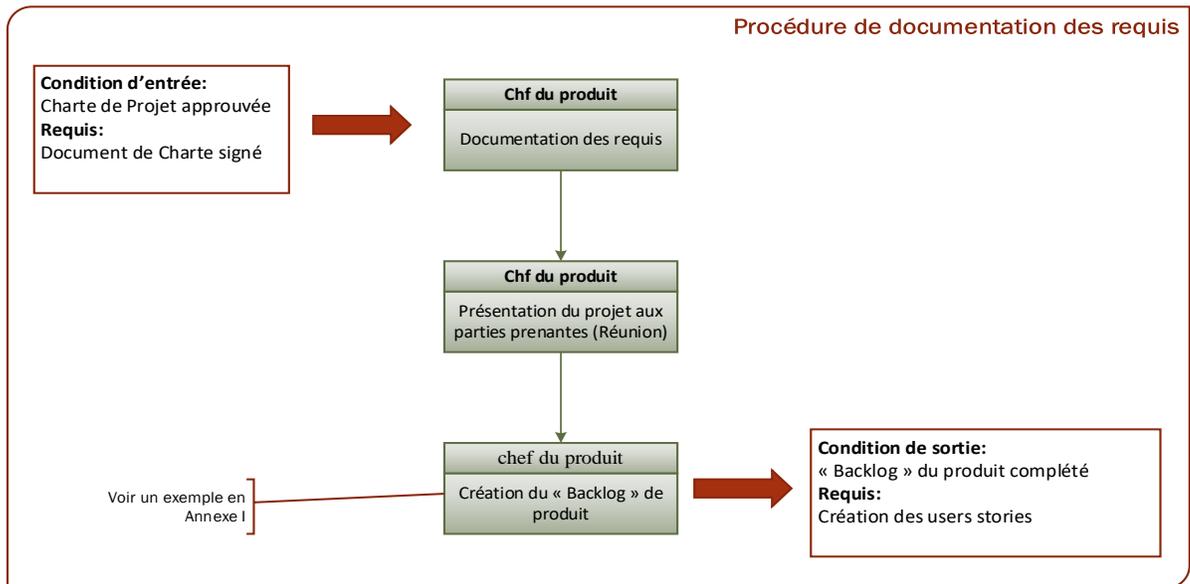


Figure 2.11 : Procédure de documentation des requis.

La gestion de ces histoires et leur planification à travers les Sprints peuvent s'avérer essentielles pour la réussite ou l'échec du projet. Le chef de produit devra consulter régulièrement l'équipe de développement, le maître de mêlée et si nécessaire le client, pour mieux réussir cet exercice. Le véritable enjeu est d'écrire les histoires pouvant être complètement implémentées à travers un Sprint. Il peut être démotivant pour les développeurs de devoir passer plusieurs Sprints sur le même User Story, d'où la nécessité de devoir souvent subdiviser un User Story complexe en plusieurs histoires pouvant chacun être implémenté à travers un Sprint.

2.4.3 Processus de raffinement de l'arriéré du produit

Le chef de produit identifie toutes les histoires de l'arriéré du produit à raffiner et les communique à l'équipe Scrum, puis demande une réunion de raffinement au maître de mêlée.

Certains outils de Gestion des configurations permettent au chef du produit de juste placer un flag sur les tâches à raffiner pour qu'elles soient visibles par toute l'équipe.

Le maître de mêlée organise la réunion de raffinement, et invite toute l'équipe Scrum. Pendant la réunion de raffinement, l'équipe évalue chaque User Stories en fonction de sa complexité et de l'effort global nécessaire pour implémenter la tâche, et attribut un score appelé story points.

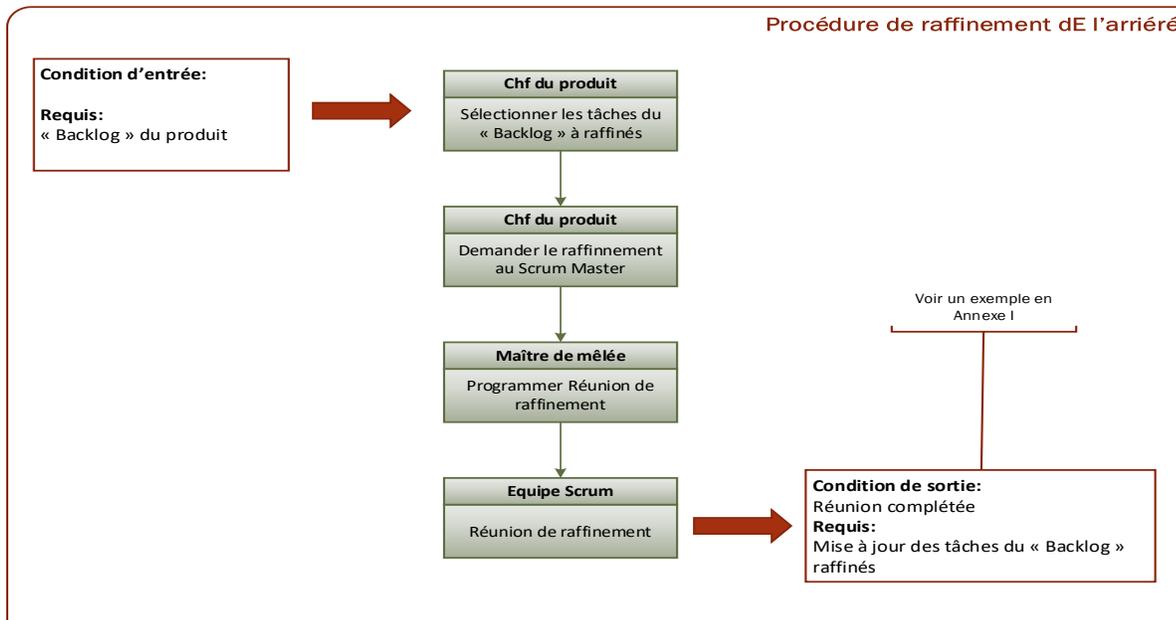


Figure 2.12 : Procédure de raffinement de l'arrière.

Les story points sont des unités de mesure qui permettent de faire un estimé l'effort nécessaire pour implémenter une tâche. Dans les projets agiles, on évite généralement de les exprimer en heures ou en jour, car il est plus optimal d'évaluer les tâches en fonction de leur complexité et non en fonction de leur durée. L'une des techniques communément utilisées pour évaluer les story points est le « planning poker » qui consiste à prendre une tâche/histoire de l'arrière, de discuter brièvement de celle-ci, de permettre à chaque membre de l'équipe de faire mentalement une estimation et de proposer un story point (chiffre sur une carte), puis de le montrer à l'équipe. Si tout le monde n'a pas la même estimation, on essaye de discuter rapidement pour trouver l'estimation la plus raisonnable. Un prototype ou une clarification de requis ou analyse de risques peut être nécessaire si l'équipe rencontre certaines difficultés dans

cet exercice. Une fois le raffinement de la tâche complété, le maître de mêlée met à jour la tâche raffinée avec les informations sur l'évaluation de la complexité et l'effort.

L'évaluation des stories points permet au chef de produit et au maître de mêlée d'évaluer les ressources nécessaires dans la planification des futurs Sprints. Cet exercice peut sembler complexe au début, mais avec le temps et de l'expérience, l'équipe finit par bien comprendre les enjeux (Atlassian, s.d.-b).

2.4.4 Processus de planification du Sprint

Le chef du produit définit l'objectif du Sprint et sélectionne par ordre de priorité les tâches qu'il aimerait placer dans le Sprint, puis fait une demande de planification de Sprint au maître de mêlée. Le maître de mêlée planifie la réunion de planification de Sprint, et invite toute l'équipe Scrum. Pendant la réunion, le maître de mêlée consulte le dernier Sprint pour savoir si toutes les tâches ont été accomplies et s'il n'y a rien à transférer dans le nouveau Sprint. Le maître de mêlée vérifie la disponibilité des membres de l'équipe de développement, et planifie en conséquence les tâches à mettre dans le Sprint. Il est important de confronter la disponibilité des ressources et l'effort à mettre dans chaque tâche pour s'assurer qu'il y aura suffisamment de ressource pour effectuer le travail tout au long du Sprint, ou que les ressources ne sont pas sous-utilisées. À l'issue de cet exercice, les tâches priorisées par le chef du produit peuvent être réduites ou augmentées dépendamment de la disponibilité des ressources, entraînant ainsi une modification de l'objectif du Sprint. Après la réunion de planification, le maître de mêlée démarre le Sprint dans l'outil de gestion des configurations avec une date de début et une date de fin. Dépendamment de l'organisation, la durée d'un Sprint peut être d'une à quatre semaines. Il est cependant important que cette durée soit constante pour tous les Sprints. Trois semaines semblent une durée raisonnable.

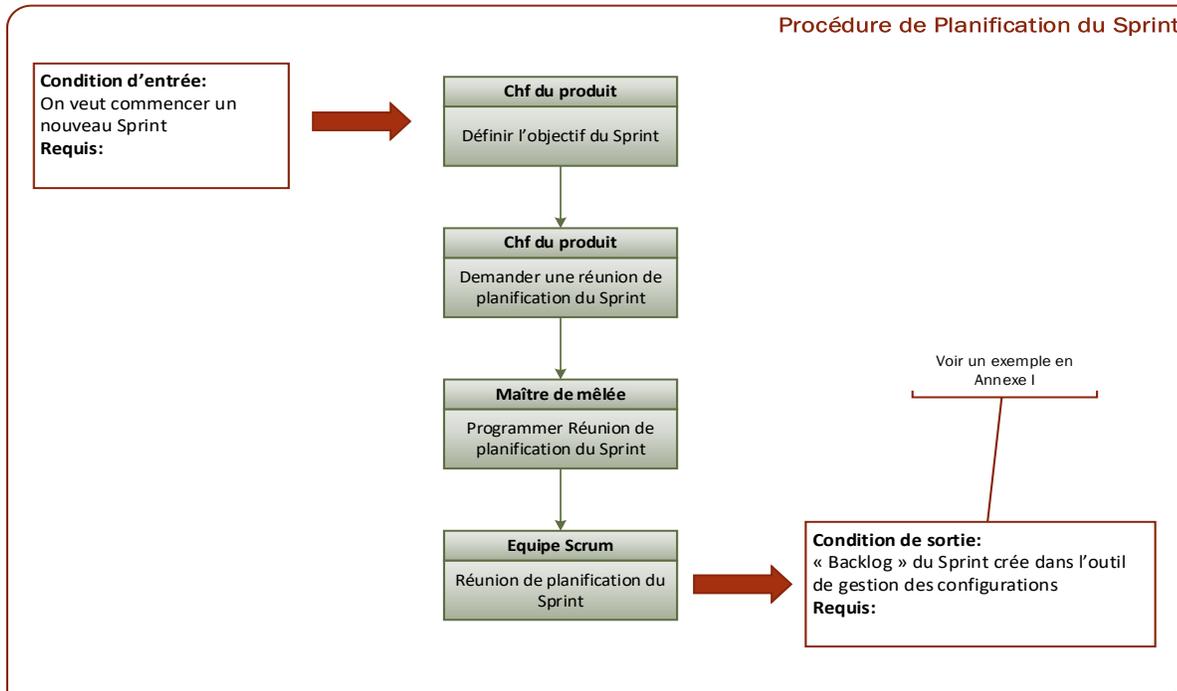


Figure 2.13 : Procédure de planification du Sprint.

À la fin de ce processus, toutes les tâches et sous-tâches à exécuter pendant le Sprint sont créées dans l'outil de gestion de configuration. La figure de l'[Annexe I](#) montre les tâches créées dans l'arrière du Sprint, indiquant l'ensemble des actions à compléter pour implémenter le Sprint A. La figure de l'[Annexe II](#) montre les sous-tâches créées dans chaque story point, indiquant l'ensemble des actions à compléter pour l'implémentation d'un story point.

2.4.5 Processus de développement de produit

L'équipe de développement, documente les requis techniques, l'architecture de l'application et les cas tests dans l'outil de gestion de la configuration, puis établit une traçabilité entre ces différents documents et les User Stories. Elle implémente ensuite le code ainsi que les tests unitaires. Une revue du code par un membre de l'équipe autre que celui qui fait le codage est nécessaire pour s'assurer que les bonnes pratiques de codage établies par l'équipe sont respectées. Une fois le code complété, l'équipe déploie l'application en environnement de test, pour les tests (vérification, validation, régression et sécurité), puis en environnement de postproduction

pour les tests d'acceptation. Ces derniers doivent désormais être coordonnés et validés par le chef de produit, ainsi que le processus de mise en production. Pour éviter des obstacles à l'agilité, les responsables de la DIT ne devraient plus être impliqués dans les processus de développement des produits, le chef de produit assume toutes les responsabilités jusqu'à la mise en production. Pour rappel, dans une méthodologie Scrum, les seuls acteurs sont le chef de produit, le maître de mêlée et l'équipe.

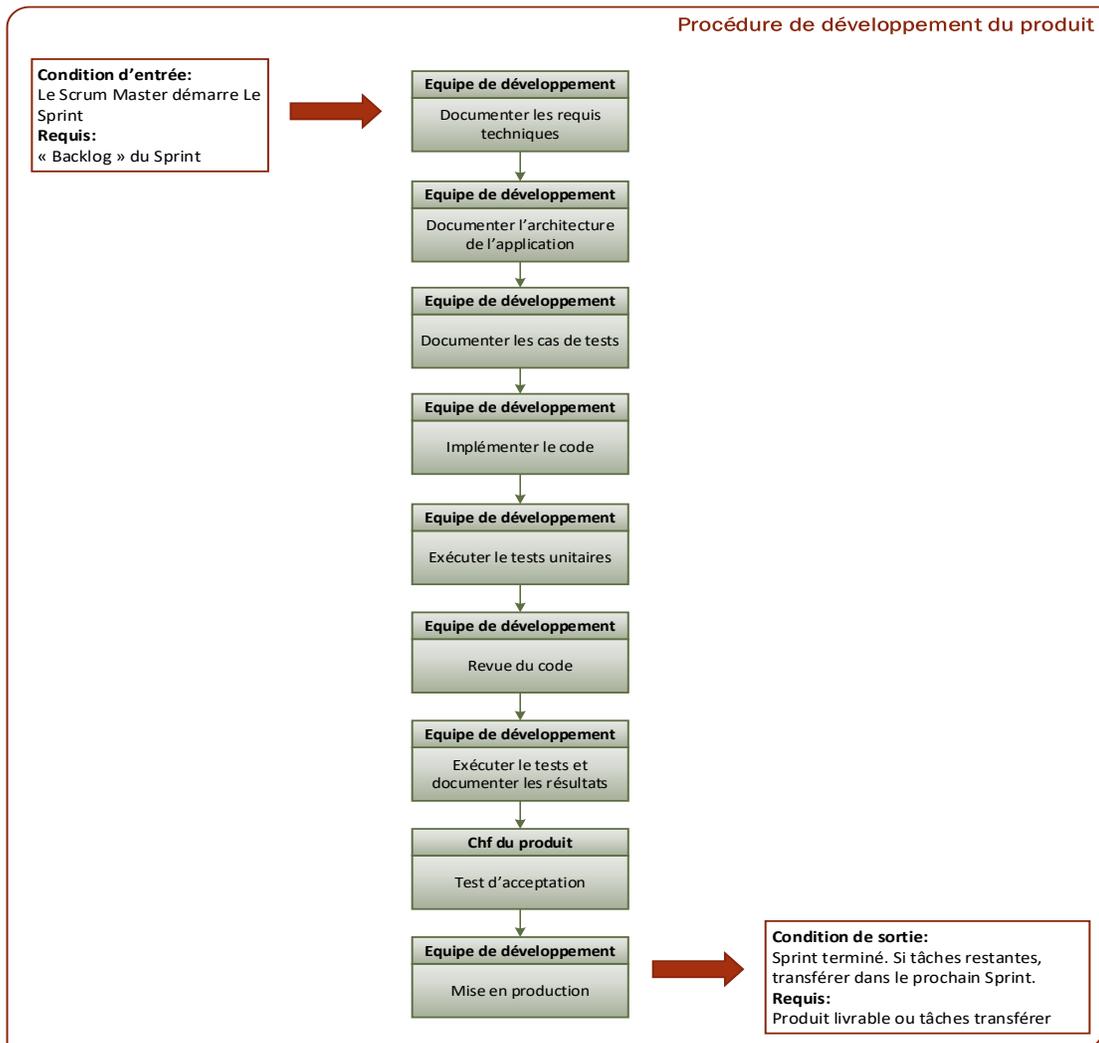


Figure 2.14 : Procédure de développement du produit

Une fois les tests complétés, le produit est livré au client. Si toutes les tâches n'ont pas pu être accomplies, elles seront analysées pendant la réunion de revue de Sprint et éventuellement transférées au prochain.

2.4.6 Processus de revue du Sprint

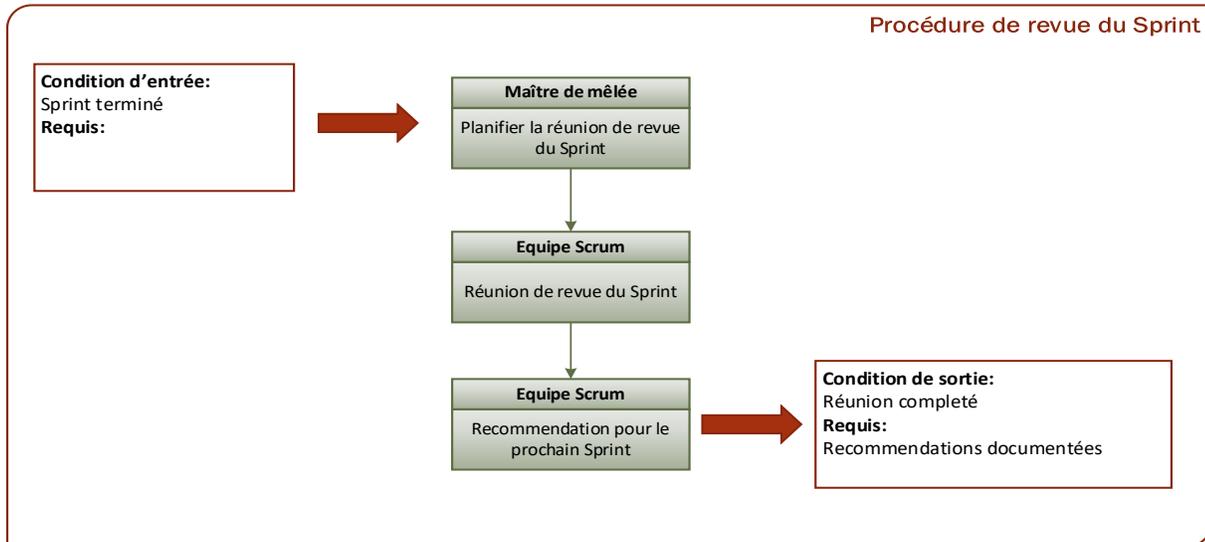


Figure 2.15: Procédure de revue du Sprint

À la fin de chaque Sprint, le maître de mêlée planifie la réunion de revue du Sprint. Toute l'équipe Scrum et éventuellement le client sont présents à cette réunion. Pendant la réunion, l'équipe demande aux clients si les fonctionnalités complétées sont satisfaisantes, passe en revue les tâches qui ont été accomplies ou pas. Il est question ici de savoir si l'objectif du Sprint a été atteint, et sinon on discute sur ce qu'il y a lieu de faire pour améliorer le prochain Sprint. Le chef de produit présente ensuite l'arrière du produit aux parties prenantes pour discuter des priorités et des tâches à placer dans le futur Sprint. Cette réunion doit se faire en toute transparence, car toutes les parties prenantes doivent savoir ce qui a bien fonctionné, ce qui n'a pas bien fonctionné et ce qu'il faut faire pour améliorer le prochain Sprint (Atlassian, s.d.-a).

2.4.7 Processus de rétrospective du Sprint

Après la réunion de revue du Sprint, le maître de mêlée planifie la réunion de rétrospective du Sprint. Toute l'équipe Scrum est invitée à cette réunion. Pendant la réunion, l'équipe passe en revue le processus d'exécution du Sprint et même une réflexion sur des améliorations à faire pour être plus efficace dans les futurs Sprints. Une réflexion est faite autour des questions suivantes (s.d.) :

- Qu'est-ce qui a bien marché pendant le Sprint?
- Qu'est-ce qui n'a pas bien marché pendant le Sprint?
- Quelles sont les leçons apprises pendant le Sprint?
- Quelles sont les améliorations à faire au prochain Sprint?

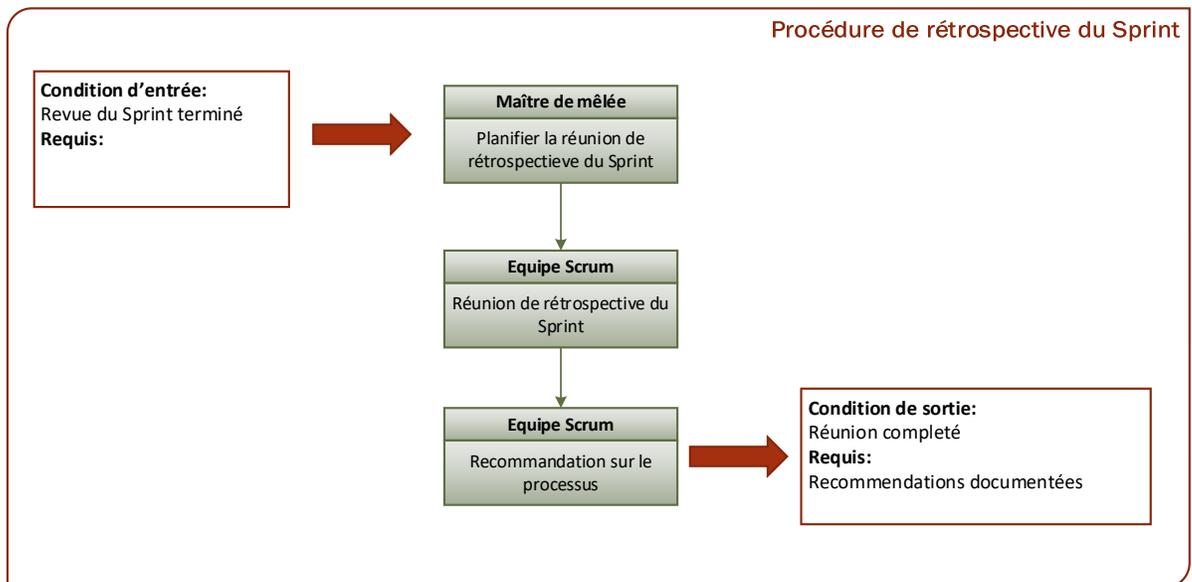


Figure 2.16: Procédure de rétrospective du Sprint

Cette étape permet de mesurer la motivation, la compétence des membres de l'équipe, les bonnes pratiques, la transparence à travers le processus, les besoins de l'équipe en formation et en communication, et bien d'autres. À l'issue de l'étape, le maître de mêlée doit documenter et assigner aux personnes concernées toutes les actions qui doivent être complétées pour mieux réussir le prochain Sprint.

2.4.8 La réunion quotidienne de Scrum

Le maître de mêlée, le chef de produit et les membres de l'équipe de développement se rencontrent quotidiennement pendant tout le Sprint pour une réunion qui dure maximum 15 minutes. Pendant cette réunion, on identifie les points de blocage, trouve rapidement des solutions, et planifie le travail à faire avant la prochaine réunion. Le but est de rendre transparent le travail entre les différents membres de l'équipe, de s'assurer que tout le monde travaille pour l'atteinte des objectifs du Sprint et de permettre à l'équipe de décider elle-même

comment faire le travail pour atteindre ces objectifs. En principe, chaque membre de l'équipe doit pouvoir répondre aux questions suivantes : qu'ai-je complété comme tâche depuis la dernière réunion? Sur quelle tâche vais-je travailler jusqu'à la prochaine réunion? Qu'est-ce qui me bloque dans mon travail? L'équipe doit trouver la meilleure formule pour rendre cette réunion utile et efficace(Atlassian, s.d.-a).

2.5 Scrumban comme Alternative au Scrum pour le CVDL

Même si le Scrum semble plus structuré, avec des rôles précis, et des livraisons à chaque Sprint, il requiert un certain temps d'adaptation et une certaine routine aux équipes, avant d'atteindre la vitesse de croisière. De plus pour la DIT, il sera nécessaire de faire des changements importants par rapport aux processus actuels, pour adopter le Cycle de vie Scrum. Si la DIT n'est pas prête à répondre aux exigences du Scrum et à faire de grands changements dans ses processus pour ce qui est de la CVDL, le Scrumban, une méthodologie hybride du Scrum et du Kanban, pourrait être une alternative intéressante. Il pourra donc s'adapter au processus existant de la DIT avec l'ajout de quelques éléments Scrum et un tableau Kanban.

2.5.1 Arriéré du produit

À l'issue du processus d'analyse (voir section 2.3.2.1), le chef de projet documente l'ensemble des requis du projet, et crée tous les « users stories », et les priorise dans l'arriéré du produit. Une fois les tâches priorisées, il les déplace dans le l'arriéré du Sprint, qui correspond à la colonne « À faire » du tableau kanban dans Jira.

2.5.2 Flux de travail Kanban

La figure ci-dessous représente une proposition du flux de travail Kanban, pour les projets de développement et maintenance logicielle en interne :

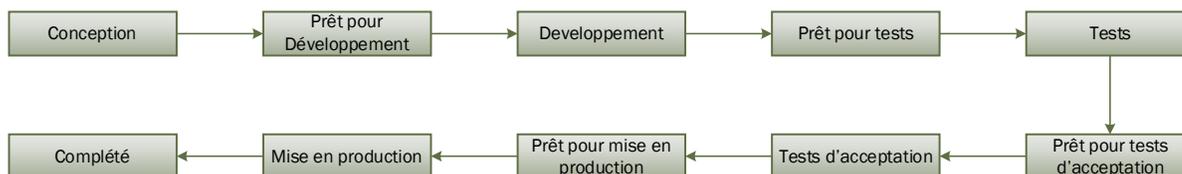


Figure 2.17 : Flux de travail Kanban pour développement logiciel en interne

- La phase de conception permet de compléter et de valider la conception détaillée d'une tâche, et de le pousser ensuite dans la colonne « prêt pour développement ». Les tâches prêtes pour l'étape de conception sont dans la colonne « À faire » du tableau kanban;
- La phase de développement permet d'implémenter le code, et de pousser la tâche ensuite dans la colonne « prêt pour tests »;
- Pour l'étape « prêt pour tests » : Les testeurs de l'équipe s'assignent des tâches à partir colonne « Prêt pour tests », puis change le statut à « Tests » et la tâche se déplace désormais dans la colonne « Tests »;
- Pour l'étape « Tests » : Les testeurs de l'équipe exécutent les tests, puis change le statut à « Prêt pour tests d'acceptation » et la tâche se déplace désormais dans la colonne « Prêt pour tests d'acceptation »;
- Pour L'étape « Prêt pour tests d'acceptation » : Une fois les tests complétés, les testeurs mettent le statut de la tâche à « Prêt pour tests d'acceptation » pour que le responsable des tests d'acceptation voie que la tâche est prête pour les tests d'acceptation;
- Pour L'étape « Tests d'acceptation » : Le responsable du test d'acceptation, s'intéresse plus à la colonne « Prêt pour tests d'acceptation » et s'assignent des tâches à partir de cette colonne, puis change le statut à « Tests d'acceptation » et la tâche se déplace désormais dans la colonne « Tests d'acceptation »;
- Pour L'étape « Prêt pour mise en production » : Une fois les tests d'acceptation complétés, les testeurs mettent le statut de la tâche à « Prêt pour mise en production » pour que le responsable de la mise en production voie que la tâche est prête pour la mise en production;
- Pour L'étape « Mise en production » : Le responsable de la mise en production, s'intéresse plus à la colonne « Prêt pour mise en production » et s'assignent des tâches

à partir de cette colonne, puis change le statut à « Mise en production » et la tâche se déplace désormais dans la colonne « Mise en production »;

- Pour L'étape « Complété » : Le responsable de la mise en production change le statut de la tâche à « complété », après la mise en production.

La figure ci-dessous représente un exemple du tableau Kanban pour le CVDL, dans l'outil Jira.

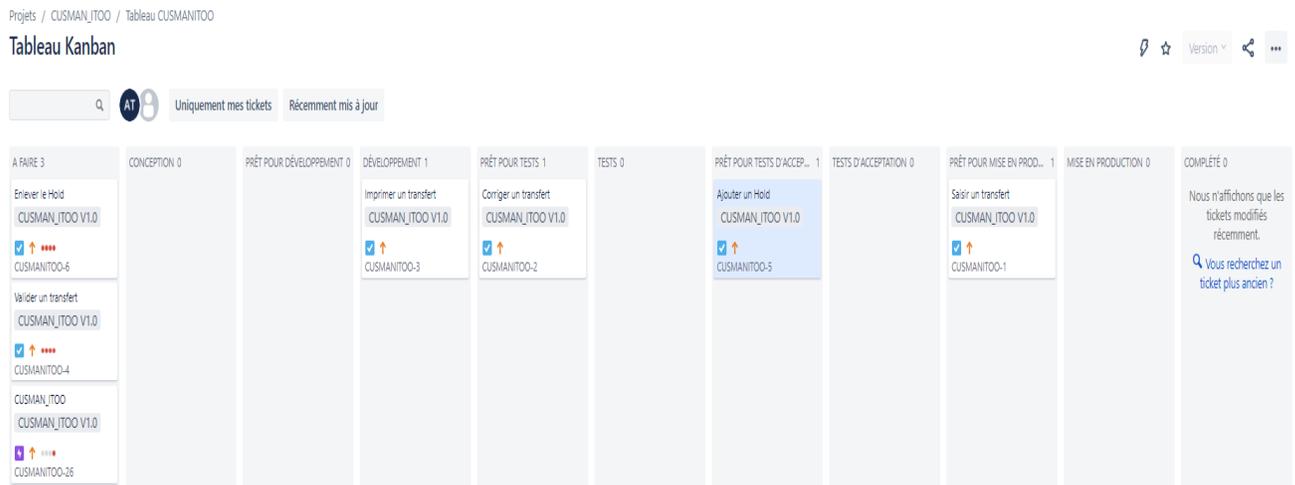


Figure 2.18 : Exemple de tableau Kanban pour logiciel développé en interne

2.5.3 Gestion du tableau Kanban

- Le tableau Kanban est la propriété de l'équipe de développement.
- Les membres de l'équipe doivent se rencontrer au moins trois fois par semaine dans des réunions courtes (30 minutes max), pour faire le point sur la situation des tâches dans le tableau Kanban. Au cours de ces réunions, chaque membre de l'équipe doit présenter les tâches sur lesquelles il travaille, ainsi que les difficultés dont il fait face. L'équipe réfléchit ensemble sur les solutions à apporter pour faire avancer les tâches bloquées, et les améliorations possibles sur la manière de travailler, et les performances de l'équipe. Chaque membre de l'équipe doit participer activement aux échanges d'idées concernant l'amélioration continue. Le Coach agile doit aider l'équipe à comprendre cet enjeu. Par exemple, le développeur, en plus de faire du code, doit

participer à toutes les discussions et décisions concernant l'équipe, les améliorations à faire, et les orientations à prendre :

- Pour éviter des engorgements dans le tableau Kanban, la définition des contraintes par colonne du tableau Kanban est importante, car elle permet d'éviter que plusieurs tâches ne traînent pendant longtemps dans une colonne, et affecter ainsi la performance de l'équipe. Une des contraintes est par exemple la limitation du nombre de tâches par colonne, qui permet de s'assurer que les membres de l'équipe accomplissent une tâche avant de prendre la suivante, et éviter que certaines tâches ne soient négligées ou abandonnées pendant trop longtemps dans une même colonne;
- Pour des besoins de planification, et pour aider à mesurer la performance de l'équipe, une estimation de chaque tâche doit être faite. Le chef de produit doit s'assurer que toutes les tâches placées dans la colonne « À faire » du tableau Kanban, sont estimées;
- Le chef de produit organise des réunions de raffinement du produit avec l'équipe de développement, pour évaluer l'effort de chaque tâche, et en faire une estimation, qui permettra de mesurer régulièrement les performances de l'équipe;
- À chaque étape du Kanban, des sous-tâches pourront être créées dans la tâche, pour matérialiser et documenter les différentes tâches complétées dans l'étape ([voir Annex IV](#)).

La section suivante présentera les outils indispensables pour une bonne exécution des processus agiles.

2.6 Outils de gestion des configurations

Dans cette section, les différents outils de gestion des configurations seront abordés. Ces outils sont indispensables pour une bonne exécution des processus agiles.

De nos jours, plusieurs outils sont disponibles sur le marché, et s'adaptent très bien aux processus agiles dans des entreprises. Très souvent, il faut combiner plusieurs de ces outils pour une bonne maîtrise de l'agilité dans son cycle de vie entier. Ainsi dans le processus d'adoption de l'agilité, il faut un outil de gestion de projets agiles, un outil de gestion des requis, un outil de collaboration et de documentation, et un outil de gestion des versions.

2.6.1 Outil de gestion des projets

Les outils de gestion de projets agiles permettent aux équipes agiles d'être plus efficaces dans l'exécution de leurs tâches quotidiennes, et une livraison régulière et efficace. Le logiciel Jira, l'un des plus populaires en industrie de nos jours, permet d'automatiser les tâches de planification, de suivi du travail, et de livraison. La DIT, ayant déjà déployé cette application, doit essayer de tirer le maximum de son potentiel. Elle pourra l'utiliser pour :

- Créer un flux de travail pour chaque projet;
- Gérer les versions du produit;
- Gérer et prioriser les différentes tâches relatives à un processus;
- Gérer les différentes étapes du Sprint;
- Gérer l'arriéré du produit;
- Créer et gérer le tableau Kanban;
- Gérer les tâches à compléter pour une version du produit;
- Assigner les tâches aux équipes;
- L'intégrer à d'autres outils de développement.

2.6.2 Outil de gestion des requis

La documentation des requis est une étape très importante de la gestion des configurations. Afin de s'assurer que les requis du client ont été bien implémenté, il est nécessaire de faire une traçabilité entre les requis du client, les requis techniques, les cas de tests et les résultats de tests, avant la livraison du produit au client. Pour la DIT, si ce n'est pas encore fait, il est recommandé d'acquérir un outil de gestion des requis. Il existe plusieurs outils sur le marché, IBM DOORS en est un exemple.

2.6.3 Outil de collaboration et de documentation

La collaboration est un élément très indispensable dans un environnement agile, et la possibilité de documenter les processus dans un environnement de collaboration peut aider les différentes équipes à être plus proches des processus qu'elles suivent, et à pouvoir facilement les améliorer. Plusieurs outils de collaboration existent de nos jours sur le marché. Confluence, qui se connecte très bien à Jira, est l'un des outils les plus populaires en industrie. La DIT peut acquérir un outil pareil si ce n'est pas fait, pour des besoins de collaboration et de documentation au sein des équipes.

2.6.4 Outil de gestion des versions

Automatiser la gestion des versions dans un environnement agile à l'aide d'un outil approprié est très important. En effet, en agilité, les produits sont livrés en plusieurs itérations, et donc en plusieurs versions. L'utilisation d'un outil permettant aux équipes de travailler de manière collaborative sur les livrables, et de faire un Baseline de chaque version livrée au client est essentielle pour une bonne gestion des configurations. Des outils comme GitHub ou Subversion sont parmi les plus populaires en industrie. Si la DIT utilise déjà l'un de ces outils, il faudra l'adapter au rythme des livrables dans un environnement agile.

2.7 Conclusion

Pour le passage à l'agilité, une réorganisation de la structure organisationnelle et un cycle de vie Scrum et scrumban ont été proposés à la DIT. Pour ce qui est de la structure organisationnelle, il a été suggéré à la DIT de mettre sur pied des équipes agiles en valorisant le travail collaboratif, et de créer de nouveaux rôles, tout en déployant une stratégie de migration du personnel impacté. Un Cycle de vie Scrum a été proposé pour rendre agile le CVDL, avec comme alternative le Scrumban, si la DIT n'est pas prête à faire des changements majeurs dans ses processus. Il a également été recommandé à la DIT de démarrer la transition vers l'agilité uniquement avec le cycle de vie CVDL car elle a un meilleur contrôle dessus. Les autres cycles de vie étant complexes à rendre agile, la banque pourra attendre d'atteindre une

certaine maturité, avant de penser à faire évoluer les processus de ces cycles de vie vers l'agilité. En plus de revoir sa structure organisationnelle et ses processus, la DIT devrait également acquérir des outils pour faciliter le travail en équipe et la gestion des configurations.

CHAPITRE 3

REVUE CRITIQUE ET AMÉLIORATIONS

3.1 Introduction

L'implémentation de l'agilité dans une structure est le résultat de plusieurs démarches. En effet, il est important de connaître les processus et le contexte dans lequel ils sont exécutés, ainsi que la structure organisationnelle, et la culture d'entreprise. La compréhension de l'environnement de production est donc importante pour une bonne compréhension des enjeux de l'agilité. Ce rapport de recherche appliquée s'est basé essentiellement sur les processus actuels de la DIT pour proposer des méthodologies pouvant aider à les rendre agiles. Dans ce chapitre, une synthèse du travail qui a été fait sera proposée, ainsi qu'une revue critique, et des améliorations futures nécessaires pour améliorer les solutions proposées.

3.2 Synthèse du travail complété et revue critique

Cette section présentera une synthèse du travail qui a été effectué dans ce projet de recherche, ainsi qu'une revue critique des différents résultats obtenus.

3.2.1 Synthèse du travail complété

La démarche adoptée dans ce projet consistait dans un premier temps, à faire une revue littéraire des méthodologies agiles les populaires en industrie, ensuite, explorer les différents processus de la DIT, et proposer comment les rendre agiles.

Dans la revue littéraire, après avoir présenté le manifeste agile, les méthodologies agiles populaires telles que le Scrum, le Kanban et le Scrumban, ont été sommairement décrites, ainsi que les éléments et les étapes qu'une entreprise doit considérer au moment de passer à l'agilité. Il a été question par la suite de proposer comment rendre agiles les processus de la DIT. Une réorganisation de la structure organisationnelle de la DIT a été proposée, avec une nouvelle philosophie de constitution des équipes, la création de nouveaux rôles et la gestion de leur

impact sur les processus actuels, ainsi qu'une stratégie de migration du personnel impacté par ces nouveaux rôles. Ensuite, les différents cycles de vie de la DIT ont été explorés, afin de déterminer quelle méthodologie agile sera plus appropriée pour rendre les processus agiles. Après analyse de l'ensemble des processus du CVDL, la méthodologie Scrum a été recommandée, avec comme méthodologie alternative le Scrumban, car avec le Scrum, la DIT devra faire des changements importants dans son processus actuel, alors qu'avec le Scrumban la DIT peut adapter son processus actuel, en opérant quelques changements importants (création des nouveaux rôles, et gestion du tableau Kanban). Pour chaque méthodologie agile proposée, un nouveau processus agile a également été proposé à la DIT. Un point a également été fait sur l'ensemble des outils de gestion des configurations à déployer pour mieux réussir le passage à l'agilité.

3.2.2 Revue critique

Les travaux effectués dans ce rapport de recherche appliquée sont un peu limités, car, les recommandations faites sont basées uniquement sur les processus de la DIT, sans prendre en compte les conditions réelles d'exécution de ces processus, l'environnement de travail, la culture d'entreprise, et la gestion des ressources humaines, qui sont tous des facteurs importants à analyser et à prendre en compte, pour augmenter les chances de succès dans la transition vers l'agilité. La migration vers l'agilité étant quelque de complexe à mettre en place, seule l'exploitation des processus n'est pas suffisante pour faire des recommandations implémentable avec succès dans l'environnement de production de l'entreprise, car, chaque entreprise a ses réalités et ses particularités, qui ne sont pas perceptibles et analysables juste à travers les différents processus. Idéalement, avant de faire un travail pareil, il serait plus fructueux d'effectuer un mandat en tant que consultant au sein de la DIT, pour comprendre l'environnement de travail, les processus et les difficultés liées à leurs exécutions, comprendre la culture d'entreprise et interviewer les différents acteurs des processus, afin de faire des recommandations beaucoup plus proches des réalités de l'entreprise.

3.3 Améliorations futures

Cette section abordera les différentes améliorations futures de notre travail.

Plusieurs améliorations pourront être apportées à ce projet de recherche appliquée :

- Développer un plan d'assurance qualité et d'amélioration continue, incluant les audits QA (Quality Assurance) et CM (Configuration Management);
- Proposer un Framework permettant l'évaluation de la maturité des processus, deux à trois années après le passage à l'agilité, permettant d'identifier les pratiques liées aux nouveaux Rôles, et aux différentes étapes du cycle de vie Scrum ou Scrumban;
- Développer la possibilité pour la banque Haïtienne de répandre l'agilité dans tous les autres équipes ou départements de l'entreprise, car le Scrum est limité aux équipes, et n'est pas adapté aux projets inter équipes. Une méthodologie permettant de déployer l'agilité à l'échelle de l'entreprise telle que le SAFe (Scaled Agile Framework) sera à envisager dans le futur;
- En raison de la pandémie récente du Covid-19, il serait important de proposer une pratique permettant d'adapter les équipes agiles au travail à distance, de manière à ce qu'elles demeurent productives dans ce contexte, car, idéalement, une équipe agile devrait être logée dans un même espace de travail de manière à faciliter les communications entre personnes physiques.

3.4 Conclusion

Une revue critique du travail complété dans ce rapport de recherche appliquée montre qu'il faudrait davantage d'information sur la banque pour comprendre toutes les difficultés qu'a la DIT à mettre sur pied l'agilité, et faire des recommandations en fonction de ces difficultés. De même, plusieurs améliorations pourront être faites dans le futur pour aborder d'autres aspects de l'agilité tels que la gestion de l'assurance qualité, l'évaluation de la maturité des processus, et la promotion de l'agilité à l'échelle de la banque.

CONCLUSION

Ce projet de recherche appliquée avait pour objectif principal d'analyser les différents cycles de vie de développement logiciel de la DIT d'une banque Haïtienne, et de proposer comment les rendre agiles. Dans un premier temps, une revue littéraire des méthodologies agiles les plus populaire de l'industrie a été faite, pour comprendre comment les entreprises procèdent pour adopter efficacement l'agilité. Par la suite, une analyse des cycles de vie de la DIT a été faite, suivie de plusieurs recommandations sur sa structure organisationnelle, et les outils de gestion des configurations. Enfin, pour passer des processus actuels de la DIT au processus agile, un cycle de vie Scrum a été proposé pour le CVDL, avec comme alternative le Scrumban. Les autres cycles de vie étant difficile à rendre agile, en raison d'une importante intervention des partenaires externes, il est recommandé à la DIT d'atteindre une certaine maturité en agilité avec le CVDL, avant d'entreprendre de rendre les autres cycles de vie (CVDLF et CVAL) agiles.

Plusieurs étapes sont donc à considérer par la DIT pour réussir son passage à l'agilité : déploiement des processus agile, réorganisation des équipes, création des nouveaux rôles, et acquisition des outils de collaboration et de gestion des configurations.

Une revue critique de ce projet montre plusieurs limites dans le travail effectué, car les recommandations faites sont basées uniquement sur les processus de la DIT, sans prendre en compte les conditions réelles de leurs exécutions. De plus plusieurs améliorations pourront être apportées à ce travail dans le futur, pour aborder la gestion de l'assurance qualité, l'agilité pour les projets inter équipes à travers une méthodologie permettant de déployer l'agilité à l'échelle de l'entreprise, et l'évaluation de la maturité des processus agiles, deux ou trois années après leurs déploiements.

ANNEXE I : Exemple d'Arrière de Produit et de Sprint

The screenshot displays the Jira Software interface for a project named 'Projet A'. The main view is the 'Backlog' for 'Tableau PA'. The interface is divided into two main sections: 'Sprint A' and 'Backlog'.

Sprint A (7 tickets): This section is titled 'Sprint A' and includes the description 'Implémenter les fonctionnalités des stories points B, C et G' and the dates '18/janv./21 10:31 AM - 06/févr./21 10:31 AM'. It contains the following items:

Item	Priority	Points
User story C	PA-7	5
User story D	PA-3	8
User story E	PA-2	5
Tests verification et validation	PA-17	
Tests de sécurité	PA-19	
Tests d'acceptation	PA-16	
Mise en production	PA-18	

Backlog (12 tickets): This section is titled 'Backlog' and contains the following items:

Item	Priority	Points
User story A	PA-1	-
User story B	PA-4	-
User story F	PA-5	-
User story H	PA-6	-
User story G	PA-8	-
User story I	PA-9	5
User story J	PA-10	-
User story K	PA-11	-
User story L	PA-12	-
User story M	PA-13	-

Annotations on the left side of the image point to the 'Backlog du Sprint' (highlighting the Sprint A section) and the 'Backlog du produit' (highlighting the main Backlog section).

ANNEXE II : Exemple sous-tâches d'un story point dans le Sprint A

Projets /  Projet A /  PA-7

Story Point G

 Joindre  Créer une sous-tâche  Associer un ticket  

Description

Comme utilisateur, j'aimerais pouvoir ...

Sous-tâches

Progression : 0%

 PA-20	Documentation des requis techniques	 À FAIRE
 PA-21	Documentation de l'architecture de l'application	 À FAIRE
 PA-22	Documentation des cas de tests	 À FAIRE
 PA-23	Implémentation du code	 À FAIRE
 PA-25	Exécution des tests unitaire	 À FAIRE
 PA-24	Revue du code	 À FAIRE
 PA-26	Tests informel	 À FAIRE

À faire 

Responsable  Non assigné

Rapporteur  Alexis TCHAMANBE

Étiquettes Aucun

Story Points  5

Sprint Sprint A

Priorité  High

 **Afficher 5 champs supplémentaires**

Estimation originale, Suivi temporel, Epic Link, Composants

Date de création avant-hier

Date de mise à jour hier

ANNEXE III : Exemple d'arriéré de produit pour le projet CUSMAN_ITOO (Projet décrit dans le Gabarit du Cahier de Charge attachée aux processus de la DIT)

Projets / CUSMAN_ITOO / Tableau CUSMANITOO

Backlog Partager ...

Rechercher AT Uniquement mes tickets Récemment mis à jour

A faire 7 tickets

<input checked="" type="checkbox"/>	Saisir un transfert	CUSMAN_ITOO V1.0	CUSMANITOO-1	↑
<input checked="" type="checkbox"/>	Corriger un transfert	CUSMAN_ITOO V1.0	CUSMANITOO-2	↑
<input checked="" type="checkbox"/>	Imprimer un transfert	CUSMAN_ITOO V1.0	CUSMANITOO-3	↑
<input checked="" type="checkbox"/>	Ajouter un Hold	CUSMAN_ITOO V1.0	CUSMANITOO-5	↑
<input checked="" type="checkbox"/>	Enlever le Hold	CUSMAN_ITOO V1.0	CUSMANITOO-6	↑
<input checked="" type="checkbox"/>	Valider un transfert	CUSMAN_ITOO V1.0	CUSMANITOO-4	↑
<input checked="" type="checkbox"/>	CUSMAN_ITOO	CUSMAN_ITOO V1.0	CUSMANITOO-26	↑

Backlog 12 tickets

<input checked="" type="checkbox"/>	Débiter un compte client	CUSMANITOO-7	↑
<input checked="" type="checkbox"/>	Approuver un transfert	CUSMANITOO-8	↑
<input checked="" type="checkbox"/>	Annuler un transfert	CUSMANITOO-9	↑
<input checked="" type="checkbox"/>	Vérifier le status d'un transfert	CUSMANITOO-10	↑
<input checked="" type="checkbox"/>	Envoyer un transfert en correction	CUSMANITOO-11	↑
<input checked="" type="checkbox"/>	Annuler un transfert	CUSMANITOO-12	↑
<input checked="" type="checkbox"/>	Changer les données SWIFTRef Service	CUSMANITOO-13	↑
<input checked="" type="checkbox"/>	Poster un transfert sur SWIFT Alliance Access	CUSMANITOO-14	↑
<input checked="" type="checkbox"/>	Vérifier le numéro de confirmation	CUSMANITOO-15	↑
<input checked="" type="checkbox"/>	Télécharger le numéro de confirmation	CUSMANITOO-16	↑
<input checked="" type="checkbox"/>	Autoriser les transferts dépassant la limite	CUSMANITOO-17	↑
<input checked="" type="checkbox"/>	Générer les rapports de transferts expédiés	CUSMANITOO-18	↑

ANNEXE IV : La figure ci-dessous montre un exemple (dans l'outil Jira) de la tâche « Saisir un transfert »

Elle contient l'ensemble des sous-tâches à exécuter tout au long des différentes étapes du Kanban pour compléter et livrer la fonctionnalité. Cette tâche est à l'étape « Prêt pour tests d'acceptation », c'est pour cela que toutes les sous-tâches concernant les étapes précédentes sont à l'état « Terminé(e) ».

Saisir un transfert

- Joindre
- Créer une sous-tâche
- Associer un ticket
- ...

Description

Ajouter une description...

Sous-tâches

Progression : 42%

CUSMANITOO-19	Développement des interfaces	↑	TERMINÉ(E)
CUSMANITOO-20	Tests d'intégration	↑	TERMINÉ(E)
CUSMANITOO-21	Tests de regression	↑	TERMINÉ(E)
CUSMANITOO-22	Préparation du test d'acceptation	↑	BACKLOG
CUSMANITOO-23	Exécution du test d'acceptation	↑	BACKLOG
CUSMANITOO-24	Analyser les resultats du test d'acceptation	↑	BACKLOG
CUSMANITOO-25	Compléter la mise en production	↑	BACKLOG

Prêt pour tests d'acceptation

- Responsable: Non assigné
- Rapporteur: Alexis TCHAMANBE
- Étiquettes: Aucun
- Priorité: Medium

Afficher 5 champs supplémentaires
Estimation originale, Suivi temporel, Epic Link, Composants e

Date de création il y a 8 heures
Date de mise à jour il y a 1 heure

ANNEXE V : Pposition de Gabarit pour le project Charter

Charte de Projet		Version 0.1
Nom du projet		
Date de début du projet		
Date de fin du projet		
Directeur		
Manager		
Project Owner		
Les parties prenantes		
Détails du projet		
Objectif du projet		
Context & Problème		
Opportunités		
Portée du projet		
Les livrables du projet		
Indices de performance		
ROI		
Contraintes		
Risques		
Ressource et investissements		
Les grands jalons du projet		
Jalons	Date	
Equipe du projet		
Cycle de vie		
Approbation		
	Manager	Directeur
Signature		
Date		

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- 2017-Scrum-Guide-French.pdf. (s.d.). Repéré à <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-French.pdf>
- Amer, S. K., Badr, N., & Hamad, A. (s.d.). *Combining CMMI Specific Practices with Scrum Model to Address Shortcomings in Process Maturity*. (S.l.) : The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019).
- Atlassian. (s.d.-a). Quatre réunions Agile démystifiées. *Atlassian*. Repéré à <https://www.atlassian.com/fr/agile/scrum/ceremonies>
- Atlassian. (s.d.-b). Que sont les story points ? Comment les estimer ? *Atlassian*. Repéré à <https://www.atlassian.com/fr/agile/project-management/estimation>
- Atlassian. (s.d.-c). Qu'est-ce qu'un tableau Kanban ? *Atlassian*. Repéré à <https://www.atlassian.com/fr/agile/kanban/boards>
- Aubry, C. (2013). *Scrum - 3e éd.: Le guide pratique de la méthode agile la plus populaire*. (S.l.) : Dunod.
- Définition des méthodes Agiles. (2015, 21 janvier). *Idéematic*. Repéré à <https://www.ideematic.com/actualites/2015/01/methodes-agiles-definition/>
- Kent, B., Beedle, Bennekum, & ... (s.d.-a). Manifeste pour le développement Agile de logiciels. Repéré à <https://agilemanifesto.org/iso/fr/manifesto.html>
- Kent, B., Beedle, Bennekum, & ... (s.d.-b). Principes sous-jacents au manifeste. Repéré à <https://agilemanifesto.org/iso/fr/principles.html>
- Lothon, F. (2013, 3 juin). Introduction aux méthodes agiles et Scrum. *L'Agiliste*. Repéré à <https://agiliste.fr/introduction-methodes-agiles/>
- Paul, A. J., & Rahman, S. K. (2018). STUDY ON AGILE MANAGEMENT IN CONSTRUCTION PROJECT USING SCRUMBAN METHODOLOGY, 05(11), 4.
- Project Management Institute (Éd.). (2017). *The Agile practice guide*. Newtown Square, Pennsylvania : The Project Management Institute.
- Sayer, N. J., & Williams, B. (2007). *Lean for dummies*. Hoboken, N.J : Wiley.
- Scalla, A. (2018). Les méthodes agiles en bibliothèque, 104.
- Shastri, Y., Rashina, H., & Robert, A. (s.d.). The role of the project manager in agile software development projects. *The Journal of Systems & Software*. Repéré à <https://pdf.sciencedirectassets.com/271629/1-s2.0-S0164121220X00124/1-s2.0-S0164121220302612/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEA0aCXVzLWVhc3QtMSJHMEUCIFePgbRsVero3GMpSUc4OCH34sxWRLogVlbUIx4I8EoOAIeA3LF05vjtKaNs7T%2FikUmYMfPDKHg69htJClhHCO44tIqtAMINhADGgwwNTkwMDM1NDY4NjUiDJHizq3Peq2IqRQpryqRAwVkdP8I71w5OAwSxhTUNUdhgRLzNvu%2FoeJiUyexkSs0Pi0EUQEpPxQ7JoeU1aZF1W9Qb7DXS%2Fkz6gjik6MC3ovcys32UJGcK6iC%2BR2qqovwAt96Aqy5JFMSFiGNO%2BhxzZYNhT%2F555umdkh690%2FPLFTDO%2Bar9uIzyf2wLyyTbqrNI68v3GsWWlZq6yidZsBoc%2BHu4SK0AaKEiAzAuiYpOxC69kPjwNYBvEjnyFvWcFxnAUAl3%2B5r983EIUIVBSruprZqGJuddOS4uLzDlFvuzEwxRzRDbU7bNAqSmb0YCFuqezQbynr3Ivj6Rsvh8F%2FNra0cOxlkIVAp0%2Bv0nrobaaZ8gZZaTqveSMymIoWykGkH%2FYb4VQvXS7DMjNHHK77z43iYTGvMAxykd5F>

HdkHVuKN2sHpsakqGE43KyhLAqnuni1MJ2tuAbxKkT9A7xstKujnnScMbCcssdO
 E93QBCceUHB9eyK2%2BFmrPwgOqI3D98aDJ7gR1aSB86k9KC65YaFry3HgtW
 mYQ47Z2H6nlFaMnr%2BIIIGousB2NdOvkMLFZPvxt7tF6lFK4sdDNSytYO%2B
 SKpTvbSNKPy%2B9YdHsveHI9UjygidauwddCF4hetEbGw8omaxGpTyi9H4ftcPJV
 0ggMIS6lRQPDugndU83tBQ9C90KutdToqm5FS8n8JATq1J7i4qzeqkGVte41Y4Uw
 ChHsZjtirPSvOeCWdPrjdIGOSxAkhlUwJqfEtsyOchyonB14X5WO5bmvaATGk0x0
 LSc02%2FC21QB240zYGWKQ2IXSkKhyR6NjzjPsoQ45e0POwUVUQT6pvIYZN
 qTnlw5sm5b1F4VJJhhYvhGAG7WFnsiDG3JA%3D%3D&X-Amz-
 Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20210307T215531Z&X-Amz-
 SignedHeaders=host&X-Amz-Expires=300&X-Amz-
 Credential=ASIAQ3PHCVTY6TQ2WZ6C%2F20210307%2Fus-east-
 1%2Fs3%2Faws4_request&X-Amz-
 Signature=4d83a90a03d8f2b3fa8f10f33b38747a284387ce59b1c69d4e6c7bcb8386e5
 de&hash=71e21de89c5dadf2837548a91d74cfef8973fbf43df1f3ccb8e44899a43d952e
 &host=68042c943591013ac2b2430a89b270f6af2c76d8dfd086a07176afe7c76c2c61&
 pii=S0164121220302612&tid=spdf-dca4cbc8-8704-496d-a5c3-
 5826b4c39522&sid=eb7bca681bcf024cc10b09e9c9cfb9c13d07gxrqa&type=client

Visual Paradigm. (s.d.). What is Sprint Retrospective Meeting in Scrum? Repéré à
<https://www.visual-paradigm.com/scrum/what-is-sprint-retrospective-meeting/>

What is Agile Kanban Methodology? Learn the Methods & Tools. (s.d.). Repéré à
<https://www.inflextra.com/methodologies/kanban.aspx>