

**MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE**

**ÉCOLE NATIONALE SUPÉRIEURE DE MANAGEMENT  
ENSM. Pôle Universitaire de KOLÉA**



**MEMOIRE DE FIN D'ETUDES**

**Master en Entrepreneuriat et Management de Projets EMP**

**L'approche Agile en gestion des projets informatiques :  
évaluation de la prédisposition de la mise en place de l'Agilité au  
sein de la SGA.**

**Elaboré par :**

**MORSLI MOHAMED AMINE**

**Encadré par :**

**Pr. MEZHOUDA ABDELMALIK**

**Année 2018/2019**

## Résumé :

Jusqu'à la fin des années 90, les approches de gestion de projets informatiques dominantes étaient les approches prédictives et séquentiels, ces méthodes dites « classiques » ont été remises en cause par les nouvelles méthodes nommées « Agile ». Les méthodes Agiles sont généralement décrites comme des méthodes itératives séquentielles. L'Agilité a montré ses preuves rapidement et dominait le monde de développement informatique. Malgré le succès phénoménal de ces méthodes la transition des organisations vers l'Agilité nécessite un énorme travail et un changement organisationnel important. Dans ce contexte nous avons tenté, dans ce présent travail, d'évaluer la disposition de Société générale Algérie (SGA) de transiter vers l'adoption de l'Agilité. En utilisant une méthodologie quantitative, l'étude nous a permis de mettre en évidence les pratiques Agiles existantes au sein de la SGA.

**Mots clés :** Méthodes Agiles, Pratiques Agiles, Agilité, SCRUM, XP, Gestion de projets informatiques, Gestion de projets IT.

## Abstract:

Until the end of the 1990s, the predictive and sequential methods were the dominant IT project management methods. Agile methods are usually described as sequential iterative methods. Agility has a phenomenal success quickly and dominated the world of computer development. In spite of the phenomenal success of these methods, the transition of organizations towards agility requires a hard work and an important organizational change. Our research aims at studying the readiness of hosting organization (General Society Algeria) the Agile methods. Using a quantitative method, we checked the ability to use the agility by SGA and we the existent of certain practices of agility in this company.

**Keywords :** Agile Methods, Agile practices, Agilité, SCRUM, XP, IT Project Management.

ملخص:

حتى نهاية التسعينيات، كانت الأساليب التنبؤية المتتابعة هي المهيمنة في تسيير مشروعات تطوير البرمجيات الحاسوبية والتي كانت تعتمد على توقع وسبق كل متطلبات العميل غير أن هذه الأساليب تراجعت بعد ظهور ما يسمى بالأساليب الرشيقية (Agile) والتي عادة ما توصف بالأساليب التكرارية المتتابعة. ومنذ ظهورها، أثبتت الطرق الرشيقية نجاحها بسرعة وسيطرت على عالم تطوير البرمجيات والتكنولوجيا الحديثة. على الرغم من النجاح الهائل لهذه الأساليب، إلا انتقال المنظمات نحوها تطلب الكثير من العمل والتغييرات التنظيمية الهامة. مهمتنا هي جزء من التحضير لهذا الانتقال لاعتماد الأجايل من قبل المنظمة المضيفة، وسوف نتحقق من متطلبات الأجايل في أعمال بنك سوسبيتي جنرال الجزائر وسنحاول تقييم الممارسات الحالية في عملية إدارة المشاريع من طرف الموظفين.

الكلمات المفتاحية: الأجايل – إدارة مشاريع تطوير البرمجيات، سكروم، ممارسات الأجايل.

*A la mémoire de mon père qui nous a quittés tôt le 25 avril 2019. Qu'Allah lui pardonne et l'accueille en son vaste paradis*

## *Remerciement*

En tout premier lieu, je remercie le bon Dieu, tout puissant, de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés.

Je remercie Mon père **Morsli Abdelkader** « que dieu lui pardonne et l'accueille en son vaste paradis » qui était ma source de motivation afin de préparer un excellent travail. Et je remercie la personne qui prie pour moi tous les jours « **Ma mère** » que dieu la protège.

En second lieu Ce travail n'aurait pas vu le jour sans le soutien et l'encadrement de mon encadreur le professeur **Mezhouda Abdelmalik**, que je remercie pour sa présence et son soutien tout au long de la période de préparation de ce travail, et de mon tuteur au sein de la banque **SGA** Mr **Bouktache Karim** qui a travaillé avec moi avec toute transparence sans aucun protocole et qui m'a appris beaucoup de choses sur le management et la gestion des projets.

Je remercie ma chère épouse **Khadidja** de m'avoir supporté dans les moments difficiles tout au long de ma formation à l'ENSM que dieu la protège pour moi, et un grand merci à mes deux enfants la petite **Alaa Fatima Zobra** et mon fils **Mohamed Ibrahim** de m'avoir donné de l'énergie de leur innocence, et que je veux donner un bon l'exemple pour eux.

Je remercie mon partenaire **Mahdjoub Mohamed** pour son support technique, matériel et morale tout au long de ma formation.

Je remercie la banque SGA et les collaborateurs SIOP de m'avoir accepté parmi eux et de n'avoir jamais refusés de répondre à mes requêtes. Et surtout Mme **Madi Nouheila**.

Je remercie notre chère Ecole **l'ENSM** de m'avoir donné la chance de suivre ma formation, en commençant par Madame la directrice Mm **Hasna Mesaid** et le directeur des études Mr **Mustapha Amokrane** de leurs présence et surtout de leur exigence de nous mettre dans les bonnes conditions tout au long de notre formation, ainsi que tout le personnel de l'ENSM. Je remercie tous les professeurs de l'école de leurs disponibilité accessibilité.

Je remercie toute la promotion de master 2017/2019 et spécialement la promotion MPE d'avoir monté le niveau de concurrence aux maximums ainsi de leurs solidarités. *MMA*

# Table des matières

Table des matières .....	III
Liste des tableaux .....	VII
Liste des figures .....	VIII
Introduction .....	1
1. Introduction .....	2
2. Pertinence de la recherche .....	4
2.1. La pertinence théorique .....	4
2.2. Pertinence managériale .....	5
3. Problématique .....	5
4. Hypothèse principale .....	5
4.1. Les hypothèses secondaires .....	5
5. Structure du présent document .....	6
CHAPITRE 1 : Revue de la littérature et cadre conceptuel .....	7
1. Revue de littérature .....	8
1.1. Les méthodes Agiles à l'échelle des grands projets et du portefeuille IT .....	9
1.2. Modèles de transition Agiles .....	10
1.3. Remise en cause de l'Agile .....	11
2. Cadre conceptuel .....	13
2.1. C'est quoi un projet ? .....	13
2.2. Sources des problèmes selon le chaos report .....	16
2.3. Itératif, Incrémentale et Adaptatif .....	16
2.3.1. Le développement itératif VS incrémental .....	16
2.3.2. Développement incrémental (Cascade ou Waterfall) .....	17
2.3.3. Développement itératif .....	18
2.3.4. Développement Itératif Incrémental .....	19
2.3.5. Le développement adaptatif et son cout .....	20
2.4. Approches de gestion de projets .....	20
2.4.1. Les approches prédictives « classiques » .....	20
2.4.2. Limites des approches prédictives .....	24
2.4.3. Une alternative : les méthodes Agiles .....	27
2.4.4. Le manifeste Agiles .....	27
2.4.5. Valeurs et principes Agiles .....	28
2.4.6. Les raisons d'adopter l'agilité .....	32
2.4.7. Les avantages obtenus par l'adoption de l'approche Agile .....	32
2.4.8. Pratiques Agiles réellement utilisés .....	33

2.4.9.	Les difficultés de mise en œuvre de pratiques Agiles .....	34
2.4.10.	Les fondements du paradigme Agile .....	35
2.4.11.	Le consensus des méthodes Agiles .....	36
2.4.12.	Les pratiques communes aux méthodes Agiles.....	37
2.5.	Les méthodes Agiles .....	37
2.5.1.	Les méthodes Agiles Populaires.....	38
2.5.2.	SCRUM .....	38
2.5.3.	KANBAN.....	42
2.5.4.	eXtrem Programing.....	46
3.	Modèle de la recherche .....	51
<b>CHAPITRE 2 : Méthodologie de recherche et contexte organisationnel .....</b>		<b>53</b>
1.	Méthodologie .....	54
1.1.	Approche épistémologique.....	54
1.1.1.	L'épistémologie.....	54
1.1.2.	Le positivisme .....	54
1.1.3.	Le constructivisme .....	55
1.1.4.	Posture épistémologique.....	55
1.2.	Approche méthodologique .....	56
1.2.1.	Méthode de collecte de données .....	57
1.2.1.1.	L'observation participante.....	57
1.2.1.2.	Le questionnaire d'enquête .....	58
1.2.1.3.	Objectifs visés à travers le questionnaire .....	58
1.2.1.4.	Structure du questionnaire et variables .....	58
1.2.1.5.	Le mode d'administration du questionnaire.....	59
1.3.	Population ciblée.....	60
1.3.1.	Taille de l'échantillon .....	61
1.3.2.	La procédure d'analyse des données .....	61
1.4.	Les limites de recherche .....	61
2.	Contexte organisationnel .....	61
2.1.	Présentation de l'entreprise d'accueil SGA .....	62
2.2.	Présentation de la structure d'accueil SIOP.....	62
2.3.	L'organigramme SIOP.....	63
2.4.	La nature des projets SIOP : .....	64
2.5.	La méthode de gestion de projets SGA :.....	65
	Gestion de la demande .....	65
	Pré-cadrage.....	65

Cadrage.....	66
Conception.....	66
Réalisation.....	66
Préparation de l'environnement d'Homologation.....	66
Qualification.....	66
Déploiement.....	66
Clôture.....	66
2.6.    Quelque situation désagréable de projets SGA.....	66
CHAPITRE 3 : Analyse et résultats.....	68
1.    Résultats de l'observation.....	69
2.    Présentation des résultats.....	71
2.1.    Description de l'échantillon :.....	71
2.1.1.    Présentation du profil des répondants :.....	71
2.2.    Traitement et analyse des données :.....	75
2.2.1.    Description des dimensions en fonction des réponses de l'échantillon de l'étude :.....	75
2.2.2.    Analyse de données.....	76
2.2.3.    Test de normalité.....	81
2.2.4.    Test des hypothèses.....	82
2.3.    Fiabilité et validité de l'étude.....	87
2.4.    Discussion des résultats.....	87
2.4.1.    La situation actuelle.....	88
2.4.2.    L'homme est la base.....	88
2.4.3.    Un logiciel opérationnel.....	88
2.4.4.    La collaboration avec le client.....	88
2.4.5.    L'adaptation au changement.....	89
2.4.6.    Les critiques.....	89
2.4.7.    Les quatre valeurs de l'agilité.....	89
2.4.8.    L'utilisation d'une méthode Agile.....	90
2.5.    Validation des hypothèses.....	90
Conclusion.....	92
Bibliographies.....	95
ANNEXE A : Questionnaire.....	97
ANNEXE B : Les méthodes Agiles approfondies.....	102
1.    Les méthodes Agiles.....	103
1.1.    SCRUM.....	103
1.1.1.    Usage de SCRUM.....	104

1.1.2.	<b>Théorie de SCRUM</b> .....	104
1.1.3.	<b>Valeurs SCRUM</b> .....	105
1.1.4.	<b>L'équipe SCRUM</b> .....	105
1.1.5.	<b>Les événements Scrum</b> .....	109
1.1.6.	<b>Les artefacts de Scrum</b> .....	114
1.2.	<b>KANBAN</b> .....	117
1.2.1.	<b>Les Valeurs Kanban</b> .....	118
1.2.2.	<b>Les principes fondamentaux de Kanban</b> .....	119
1.3.	<b>eXtrem Programing</b> .....	121
1.3.1.	<b>Les pratiques de programmation</b> .....	122
1.3.2.	<b>Les pratiques de collaboration</b> .....	123
1.3.3.	<b>Les pratiques de gestion de projet</b> .....	123
1.3.4.	<b>Les quatre valeurs d'XP</b> .....	124

## Liste des tableaux

Tableau 1: Les variables et dimensions du questionnaire .....	59
Tableau 2: Répartition de l'échantillon de l'étude selon le niveau d'études .....	71
Tableau 3: Répartition de l'échantillon de l'étude selon la spécialité/domaine .....	72
Tableau 4: La répartition de l'échantillon de l'étude selon l'expérience professionnelle .....	73
Tableau 5: La répartition de l'échantillon de l'étude selon l'expérience au sein de la SGA ...	74
Tableau 6: la répartition de l'échantillon de l'étude selon l'expérience en gestion de projets.	75
Tableau 7: L'intervalle des moyennes minimales et maximales .....	76
Tableau 8: Description des items de la dimension « La situation actuelle » .....	76
Tableau 9: Description des items de la dimension « L'homme est la base » .....	77
Tableau 10: Description des items de la dimension « Un logiciel opérationnel » .....	78
Tableau 11: Description des items de la dimension « La collaboration avec le client » .....	79
Tableau 12: Description des items de la dimension « L'adaptation au changement » .....	79
Tableau 13: Description des items de la dimension « Les critiques » .....	80
Tableau 14: Description des items de la dimension « L'utilisation d'une méthode Agile » .....	80
Tableau 15: Description des dimensions de l'Agilité « La moyenne globale de l'Agilité » .....	81
Tableau 16: Tests de normalité d'une distribution .....	81
Tableau 17: Test T-Student de l'hypothèse 1 .....	82
Tableau 18: Test T-Student de l'hypothèse 2 .....	83
Tableau 19: Test T-Student de l'hypothèse 5 .....	84
Tableau 20: Test T-Student de l'hypothèse 6 .....	84
Tableau 21: Le test de corrélation de Spearman entre les trois items d'expérience et les différentes dimensions .....	85
Tableau 22: Le niveau d'interdépendance entre l'expérience et les dimensions du questionnaire .....	87

## Liste des figures

Figure 1: Les sources des problèmes des projets Informatiques .....	16
Figure 2: Figure Itératif / incrémental (Jeff Patton) .....	17
Figure 3: Charles Lambdin mars 2016 .....	18
Figure 4: Itératif Incrémentale .....	19
Figure 5: Le développement adaptatif et son cout.....	20
Figure 6 : L'approche en cascade.....	21
Figure 7: Le cycle en V .....	23
Figure 8: Succès et d'échec de projets informatiques .....	25
Figure 9: Les raisons invoquées dans les projets terminés en déplacement .....	25
Figure 10: Les raisons invoquées des projets abandonnés .....	26
Figure 11: Les raisons invoquées des projets réussis .....	26
Figure 12: Les raisons pour adopter l'agilité .....	32
Figure 13: Les résultats d'utilisation des méthode Agiles VS les méthodes classiques .....	32
Figure 14: Les avantages obtenus par l'adoption de l'agilité .....	33
Figure 15:Pratiques Agiles Réellement utilisés.....	34
Figure 16: Les difficultés de mise en œuvre de pratiques Agiles.....	35
Figure 17: Les méthodes Agiles réellement utilisés .....	38
Figure 18: La prédisposition de la banque SGA à l'application de l'agilité .....	52
Figure 19: Historique des visites du questionnaire.....	60
Figure 20: L'organisation SIOP .....	63
Figure 21: Répartition de l'échantillon de l'étude selon le niveau d'études .....	71
Figure 22: Répartition de l'échantillon de l'étude selon la spécialité/domaine .....	72
Figure 23: La répartition de l'échantillon de l'étude selon l'expérience professionnelle .....	73
Figure 24: La répartition de l'échantillon de l'étude selon l'expérience au sein de la SGA....	74
Figure 25: La répartition de l'échantillon de l'étude selon l'expérience en gestion de projets	75
Figure 26: Les trois piliers de l'agilité .....	103
Figure 27: Une présentation de réunion quotidienne SCRUM.....	111
Figure 28: Schéma du processus de la méthode SCRUM .....	117

# **Introduction**

## 1. Introduction

La technologie de l'information se développe d'une manière exponentielle. Dans une année on peut assister à l'apparition de nouveaux produits technologiques (matériel comme gadget électronique, ou logiciel comme les plateformes électroniques) qui changent radicalement le comportement des individus dans cette planète. Depuis leurs apparitions dans les années 1950 les logiciels sont le symbole de l'avancée technologique, qui simplifie la vie des utilisateurs dans tous les domaines. Et qui permettent à la technologie de réaliser ce qu'elle réalise aujourd'hui prenant un simple exemple «un smartphone ne vaut rien sans les applications installées dedans ». Ce développement est dû grâce à l'ambition et la bonne gestion ainsi grâce à la collaboration des contributeurs dans les différents projets de développement de ces produits technologiques (Vance, 2016) (Berry, 2017).

De nombreux modèles et de standards de gestion de projets de développement informatiques ont vu le jour au fil de ces quatre dernières décennies. Ainsi les organisations de développement informatique ont rencontré de sérieux problèmes d'efficacité, de qualité et de productivité, qui ont incités ces organisations à repenser chaque fois leurs modes de réalisation et de gestion des projets informatiques. Jusqu'à la fin des années 90, les méthodes dominantes étaient celle basée sur le développement séquentiel, ce sont des méthodes prédictives dite aussi « classiques » qui anticipent le besoin dans les moindres détails avant de se lancer dans le développement. Le livrable n'est produit qu'à la fin du processus. Ce processus de développement a été symbolisé par la course de relais, ou les étapes se relaient tour à tour. L'approche dite en « cascade » est la plus représentative de ce type de développement. C'est une approche universelle, utilisée dans plusieurs domaines, elle se base sur l'anticipation des demandes de client, la définition complète du cahier de charge et la documentation exhaustive. Toutefois les principes de découpage linéaire de projets portés par les méthodes dites « classiques » ont reçus beaucoup de critiques et ils se sont de plus en plus remis en cause depuis la fin des années 90. (Messenger & Tabaka, 2013), (Vickoff J. P., 2016)

Les organisations de développement informatique et leurs projets se sont retrouvés face à une forte pression et concurrence sur un marché caractérisé par une diversité des offres, une instabilité des demandes et une réduction de délais de mise sur le marché « Time To Market » (Messenger & Tabaka, 2013). L'effet « tunnel » et le changement permanent de fonctionnalités causées par les méthodes classiques ont amenés les praticiens à remettre en question ce modèle de développement et à se focaliser sur les évolutions rapides du marché technologique. Sans oublier les projets échoués ou abandonnés. Dans ce contexte de nouvelles approches plus

flexibles dites « Agiles » ont été développées, elles se sont diffusées progressivement au niveau de l'industrie de logiciels (KHALIL, 2011).

En très peu d'années la notion d'Agilité a connu un succès considérable dans le monde de l'industrie de logiciels après son apparition en 2001 (KHALIL, 2011). Aujourd'hui de nombreuses organisations ont abandonnés ou comptent abandonner les méthodes classiques et opter de devenir « Agile ». Une enquête internationale<sup>1</sup>, auprès de communautés de développement de logiciels à travers le monde, cette enquête a touché près de 1492 répondant et montré que 79% de répondant travaillent dans des équipes Agiles. (12th Annual State of Agile Report, 2018).

L'Agilité est un ensemble de valeurs et principes fondés par les dix-sept signataires de l'Agilemanifesto<sup>2</sup>, la différence entre l'Agilité et les méthodes Agiles est que la première « l'Agilité » est l'adjectif qui représente le manifeste et l'Agilité englobe toutes les méthodes Agile. Par contre les méthodes Agiles sont les cadres de travail dites « Framework » inspirés des valeurs et principes de l'Agilité (Beck & Et, 2001).

Les méthodes Agiles sont généralement décrites comme méthodes itératives et incrémentales qui encouragent l'auto-organisation et qui s'adaptent au changement, se sont généralement des pratiques de bon sens. Contrairement aux approches classiques, se sont des méthodes qui exigent un formalisme léger et des pratiques de développement et de management qui favorise l'ajustement continu au contexte du projet (KHALIL, 2011).

En effet, ces formes innovantes de développement de logiciels reposent sur un ensemble de pratiques d'ingénierie et de gestion de projets bien définies ainsi que sur une philosophie gestionnaire qui visent, d'une part, à améliorer l'adaptabilité et la réactivité des équipes de développement et d'autre part, à réduire, voire à éliminer les divers types de gaspillages (les fonctionnalités inutiles, les anomalies tardivement identifiées, le « retravail », les temps d'attente, dus au chevauchement entre les phases d'un projet, etc.) souvent rencontrés dans les projets informatiques. Cette notion de gaspillage se trouve au cœur des préoccupations des « Agilistes » et en particulier des partisans de l'approche Lean. (KHALIL, 2011)

Les bénéfices de l'Agilité sont reconnus par la majorité des experts du monde de développement informatique. Pourtant on retrouve certains critiques des méthodes Agiles par les praticiens.

---

<sup>1</sup> The 12th State of Agile Report : enquête menée en 2017 et publié en 2018

<sup>2</sup> L'Agilemanifesto : c'est le manifeste Agile créé par les 17 membre en septembre 2001 aux USA, leur site officiel est [www.agilemanifesto.org](http://www.agilemanifesto.org)

Ces praticiens critiquent les méthodes Agiles et dans la majorité des cas continuent à défendre l'agilité, leurs reproches sont en générale sur le fait que la méthode x ou y ne respecte pas les principes de l'Agilité et que certaines méthodes comme SCRUM sont devenues purement commerciales. (Messenger & Tabaka, 2013)

Malgré cela ces méthodes Agiles ont montré leurs efficacités et que l'Agilité est le futur du monde de gestion de projets informatique, mai la transition vers l'Agilité nécessite un effort considérable et l'adoption d'une méthode Agile constitue un réel défi. Avant de chercher à la transition vers l'Agilité il faut chercher les réponses des questions suivantes : quelle méthode utilisée ? Quelle sont les prérequis pour adopter une méthode Agile ? Est-ce que l'organisation peut est prête pour la transition Agile ? Est-ce que l'organisation dispose de ressources humaine et matériel pour pratiquer une méthode Agile ?

Le présent travail consiste vérifier si la SGA possède les prérequis de l'Agilité par l'évaluation des pratiques utilisés au sein de la banque Société Générale Algérie SGA qui cherche à mettre en place une méthode de gestion de projets Agile. Puisque les pratiques Agiles sont des pratiques de bon sens, les organisations peuvent les pratiquer involontairement. Le but du présent travail est de valoriser ces pratiques, estimer le niveau d'Agilité et la prédisposition à l'Agilité de cette organisation estimer l'effort à fournir pour une transition vers l'agilité.

## **2. Pertinence de la recherche**

Notre travail de recherche combine des pertinences, théoriques et managériales.

### **2.1. La pertinence théorique**

La mission de gestion des projets informatiques est généralement confiée à des ingénieurs expérimentés et dans certains cas à des managers qui n'ont pas une formation sur le volet technique IT. Aussi en Algérie les chercheurs dans le domaine de l'informatique ne s'intéressent que sur la recherche technique « développer de nouveau concepts, de nouveau algorithmes, de nouvelles solutions informatiques ... » et ils ne s'intéressent pas aux processus managériaux, et ce n'est pas le cas pour les chercheurs dans la science humaine et de gestion qui s'intéressent sur la recherche sur le volet managérial, mais ces derniers ne maîtrisent pas l'aspect technique pour se lancer dans une recherche qui touche les deux aspects « technique et managériale » qui nécessite une bonne connaissance technique et managériale en même temps. Dans notre revue de littérature nous n'avons pas trouvé des travaux antérieurs sur l'Agilité ni au niveau de notre école ni au niveau des autres universités, de ce fait notre travail va enrichir la bibliographie de la bibliothèque de notre école. Aussi notre formation de base « Master en

informatique » ainsi que notre expérience professionnelle nous a permis de maîtriser l'aspect technique des projets informatique, nous avons acquis des connaissances managériales durant notre formation à l'ENSM qui nous permet de maîtriser les deux aspect « techniques et managérial ». Ce travail va encourager les futurs chercheurs à traiter le sujet de l'agilité.

### **2.2. Pertinence managériale**

Notre étude s'inscrit dans le cadre de préparation d'un changement organisationnel prévu de l'organisme d'accueil la SGA, il s'agit de préparer la transition de la SGA vers l'adoption de l'agilité, considérée comme une décision stratégique. Notre but est de compléter notre formation en management de projet par la maîtrise managérial de gestion de projets informatiques qui différent des projets d'autres domaines.

## **3. Problématique**

Le présent travail cherche à répondre à la question centrale suivante :

Les méthodes de gestion de projets informatiques au sein de la SGA favorisent elles la mise en place de l'approche Agile ?

De cette question centrale découlent les questions secondaires suivantes :

1. En quoi consistent les méthodes Agiles en gestion de projets de développement informatique et quels sont leurs principes directeurs ?
2. Les méthodes Agiles sont- elles la solution idéale pour la gestion des projets de développement informatique ?
3. Quelles sont les spécificités des pratiques de gestion des projets informatiques au sein de la SGA ?
4. Existe-il des pratiques de gestion de projets conformes aux valeurs et aux prérequis de l'Agilité au sein de la SGA ?
5. Quelles sont les valeurs Agiles respectées par les collaborateurs SGA ?

## **4. Hypothèse principale**

La discussion des questionnements précédents sera guidée par l'hypothèse principale suivante :

La méthode de gestion de projet SGA inclue des pratiques de l'agilité.

### **4.1. Les hypothèses secondaires**

De l'hypothèse principale précédente découlent les six hypothèses secondaires suivantes :

1. Le développement de projet au sein de la SGA se base sur l'homme

2. La méthodologie de développement de projets au sein de la SGA vise à produire des logiciels opérationnels de qualité.
3. La collaboration totale avec le client est une condition nécessaire pour la réussite de projets informatiques.
4. Le changement du besoin du client au cours du développement est accepté dans les projets SGA.
5. Les critiques de l'Agilité sont un obstacle de l'adoption de cette dernière au sein de la SGA.
6. Les collaborateurs SGA utilisent certaines pratiques de l'agilité.

## 5. Structure du présent document

Afin de faciliter la lecture du document nous allons présenter sa structure, le document est divisé en trois chapitres :

### Partie 1 : partie théorique

Le chapitre 1 « **revue de littérature et cadre conceptuel** », dans ce chapitre nous allons voir dans sa première partie la revue de la littérature, et dans la deuxième partie nous allons trouver tous les concepts utilisés dans notre recherche.

Le chapitre 2 « **méthodologie de recherche** », dans ce chapitre nous allons voir dans sa première partie la méthodologie adoptée, et dans la seconde partie nous allons voir le contexte organisationnel.

### Partie 2 : partie pratique

Le chapitre 3 « **analyse et résultats** » : ce troisième chapitre est consacré à la présentation des résultats, les analyses et traitements statistiques et la discussion et interprétations de ces derniers.

Et nous allons terminer par une conclusion générale.

# **CHAPITRE 1 : Revue de la littérature et cadre conceptuel**

## 1. Revue de littérature

Vers 1980, l'informatique a pris son envol et nous nous sommes retrouvés face à des opérations de développement de produits intangibles, tels que logiciels ou applications. Les besoins des logiciels sont devenus de plus en plus complexes et leur développement est devenu un casse-tête. Les projets de développement informatique sont devenus plus coûteux et le nombre de contributeur et le périmètre des projets ne cesse d'augmenter. Ces raisons ont rendu les méthodes de gestion de projets classiques « l'approche en cascade et le cycle en V » obsolètes pour les projets IT. Ces méthodes dites classiques enchainent trois grandes phases « phase d'expression du besoin ou ont établies le cahier de charges, la phase réalisation, et la phase test et livraison du produit ». Mais ces méthodes ont conduit plus que la moitié des projets informatiques à l'échec (BERKANI, 2017).

De nombreuses études ont illustrés les échecs des projets IT (Al-Ahmad et al. 2009). D'après plusieurs chercheurs qui ont fait des études approfondies sur les causes principales de ces échecs et les méthodes de gestion traditionnelles sont jugées trop lourdes (Boehm, 2002).

Ces conséquences ont conduit à l'apparition du manifeste Agile en septembre 2001 ou Agile Manifesto, où le monde des projets informatiques a connu les méthodes Agiles, « *ce sont de nouvelles méthodes de gestion et conduite de projets* ». Ce sont des méthodes conçus initialement pour les projets informatiques de grande ampleur, avec un besoin qui évolue rapidement. Les 17 membres fondateurs du manifeste Agile sont devenus des stars dans le monde des projets informatiques. La popularité des méthodes Agiles ne cesse d'augmenter depuis l'apparition du manifeste Agile, et plusieurs travaux de recherche se sont intéressés par ses méthodes. Et peu de chercheurs ont eu le courage de critiquer ces nouvelles méthodes et tester leurs limites. Les méthodes Agiles sont des pratiques basées sur quatre valeurs et douze principes dont le but de gérer des projets ayants un besoin qui change rapidement « on commence avec un besoin initial et on peut terminer le projet avec un autre besoins » ou on commence avec un besoin incomplet. Les utilisateurs de l'Agilité cherchent à optimiser leur gestion de projets et gagner en « effort et en temps ». Cette performance à un coût c'est le changement organisationnel et il faut mettre le paquet.

*« Les méthodes dites Agiles sont un ensemble de propositions de démarches et de pratiques pour la mise en œuvre d'un projet. Elles sont vite adoptées par les entreprises surtout celles spécialisées dans le développement d'applications informatiques »* (Jeffries, 2018)

« Une méthode Agile est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif, avec juste ce qu'il faut de formalisme. Elle génère un produit de haute qualité tout en prenant en compte l'évolution des besoins des clients ». (Messenger & Tabaka, 2013).

Initialement développées et expérimentées de manières isolées et dans une époque connue comme « mutation des comportements » ou la dépendance des logiciels commençait à prendre ampleur, c'est dans ce contexte que les méthodologies Agiles ont pris un fort essor dans le développement de logiciel dans un premier temps (Boehm & Turner, 2005), puis elles se sont logiquement diffusées dans les projets IT qui à l'époque prenaient plus d'importance. Enfin, il est par ailleurs important de souligner les raisons de l'ascension de ces méthodes. De nombreuses recherches se basent sur les avantages des méthodes Agiles, elles permettent à mieux gérer les coûts en réduisant le gaspillage des ressources et permettent d'améliorer la productivité des équipes projets (Laanti et al., 2011).

De multiples travaux de recherches sur les méthodes Agiles sont apparus dans la première décennie des années 2000 (Nerur & Moe, 2012), une revue de publications publiée durant cette première décennie est effectuée par les auteurs en 2012. Une étude empirique sur les travaux de recherches sur les méthodes de développement Agile de logiciel jusqu'à 2005 est effectuée par (Dybå & Dingsøy, 2008), les travaux ont été regroupés en 4 thèmes « introduction et adoption, facteurs humains et sociaux, perception des méthodes Agiles et études comparatifs ».

### **1.1. Les méthodes Agiles à l'échelle des grands projets et du portefeuille IT**

La propagation des méthodes Agiles est le fruit des efforts et l'implication des praticiens et des consultants (Dybå & Dingsøy, 2008), les méthodes Agiles sont mises à l'échelle des programmes de projets de développement de logiciels mobilisant des équipes de plus de 150 personnes qui travaillent sur la même base méthodologique (Laanti, 2008). A partir de 2007 où les chercheurs ont commencés à s'intéresser vraiment à l'Agilité à l'échelle de grands projets qui mobilisent un nombre important de contributeurs (Saeeda et al., 2015). Une étude empirique sur les pratiques Agiles utilisées réellement dans les grands projets « portefeuilles de projets » est effectuée par (Stettina & Hörz, 2015), où les auteurs ont trouvé que certaines méthodes sont retravaillées afin de mettre à l'échelle l'Agilité au niveau du portefeuille de projets d'une organisation. Ces travaux ont donné lieu à la naissance des méthodes Agiles permettant de gérer et réaliser de grands projets, de portefeuille projet et même de programmes (Stettina & Hörz, 2015 ; Dybå & Dingsøy, 2015). Ces nouvelles méthodes Agiles à grand échelle sont encore dans la phase d'adoption au niveau des organisations (Stettina & Hörz, 2015). En effet

certain auteurs considèrent que même les premières méthodes Agiles dites « de première génération » telle que « SCRUM, Kanban et eXtrem Programming » sont en phase d'adaptation par les organisations (Campanelli & Parreiras, 2015). Ces méthodes posent plusieurs challenges aux entreprises, avec une nécessité de changement organisationnelle de plusieurs structures, et changement culturelle afin de réduire la bureaucratie et alléger les processus (Mahanti, 2007). (BERKANI, 2017)

## 1.2. Modèles de transition Agiles

Les méthodes Agiles ne peuvent pas donner les résultats souhaités sans utilisation de modèles de transition Agile qui garantit un passage sûr d'une méthodologie classique à une méthodologie Agile. Des chercheurs se sont intéressés à la transition Agile et principalement aux modèles d'adoption et d'adaptation des méthodes Agiles. Parmi ces travaux on trouve dans la littérature une étude sur la transformation Agile est effectuée par (Javdani Gandomani & Ziaei Nafchi, 2015) qui ont étudiés les facteurs de réussite de la transformation Agile au sein des organisations.

On compte de multiples modèles de transition dans la littérature, ce sont des modèles qui se basent sur la phase d'adoption, les plus populaires sont les suivants :

Nous commençons par le modèle de Boehm & Turner (2003). *« A cette époque ce modèle permettait d'aider les entreprises de développement de logiciel à choisir entre l'Agilité et les approches classiques. Ce modèle permet de mettre en valeur les différents points d'attention sans le choix d'une méthode (la culture, la taille de l'organisation, la taille des équipes, la taille des projets, la criticité). On reproche à ce modèle d'être un outil de diagnostic plutôt qu'un modèle de transition »*. (BERKANI, 2017) ;

Deuxièmement modèle est un modèle plus détaillé élaboré par Sidky et al. (2007) *« découpé en 5 paliers « inspirées des valeurs de l'Agilité » qui permettent d'aligner le choix des pratiques Agiles à l'organisation et aux projets. Un index de 300 indicateurs sont mis en place par les auteurs afin de mesurer le degré d'Agilité de l'organisation de sorte à mieux identifier le passage d'un palier à l'autre. Le modèle et l'index proposés ont fait l'objet de plusieurs critiques, notamment au niveau du manque d'expérimentation et de sa complexité »*. (BERKANI, 2017)

On termine avec le dernier modèle de Javdani Gandomani & Ziaei Nafchi, (2015). *« C'est un modèle plus générique qui cherche à être plus flexible au niveau des organisations qui les*

*mettent en œuvre. C'est un modèle basé sur le modèle « PDCA de Deming (2000) », le modèle met notamment recommande des facilitateurs de transition dans la démarche d'adoption. Or ces facilitateurs en question ne sont pas définis ». (BERKANI, 2017)*

Il est à noter par ailleurs que peu de travaux de recherches se sont intéressés à la mise en place de l'Agilité dans une grande organisation avec un cœur de métier autre que le développement de logiciels et une gestion de projets complexes et la majorité des recherches concentre sur les entreprises de développement de logiciels.

### **1.3. Remise en cause de l'Agile**

Certes les méthodes Agiles ont révolutionné le développement de projets IT néanmoins elles ont suscité beaucoup de débats et de controverses. Dans son article intitulé : « Agile : Controverses et réflexions », Jean-Pierre Vickoff (2008) a remis en cause les méthodes Agiles dans une rubrique « Questions gênantes pour les méthodes phares » et les questions étaient du genre :

Devant l'aspect actuellement « incomplet » des méthodes Agiles par exemple l'aspect organisationnel de la méthode eXtrem Programming « XP » jugée trop technique, ou l'aspect technique de la méthode SCRUM jugée comme méthode orienté organisation « méthode de gestion de projet »

Aussi dans une autre rubrique « Une remise en cause édifiante », ou l'auteur a parlé de l'apport d'eXtrem Programming proposé en 1999 par Kent Beck dans le livre « XP explained » était la systématisation de certaines pratiques comme la programmation en binôme associé à la rotation des binômes, ces techniques facilitaient le TDD « Test Driven Développement appelé aussi Zéro Défaut » et la montée en compétence de l'équipe.

Ayant compris que le fossé du changement était loin d'être comblé, Kent Beck, très certainement sous la pression commerciale de Scrum, publia, pour tenter de faire adopter plus largement sa méthode, le livre « Planning XP ». Ce document remettait en question l'aspect extrême des pratiques XP.

L'ensemble des pratiques préconisées par XP préexistaient de manière indépendante et de mise en œuvre moins systématique en fonction du contexte ou des ressources. En acceptant une nouvelle voie plus laxiste, c'est une très grande partie de l'apport de XP qui disparaissait.

Un autre article de « Jean-Pierre Vickoff 2013, AGILE, chantiers actuels, gestion des forfaits » qui parle de nombreux articles qui ont traité de la problématique des modifications en cours

de projet appliquées à des contrats particuliers ou à des forfaits. Tous l'ont abordée sous l'angle consensuel de la confiance entre informaticiens et responsable métier. Pour s'assurer de maîtriser les conflits potentiels liés aux demandes d'évolutions, la seule option réaliste consiste à mettre systématiquement en œuvre une mesure immédiate de chaque changement (tracé et éventuellement qualifié par ses causes). En clair faut une métrique formelle de l'impact des modifications.

Dans son livre (Vickoff J. P., 2016) « Méthodes Agiles SCRUM au-delà » il regroupe les méthodes Agiles en versions, en commençant par les méthodes Agiles 1.0 comme SCRUM, XP, puis les Méthodes Agiles 2.0 comme Kanban, Kaisen, Lean et la version 3.0 comme la fameuse méthode SAFe 4.0. En fait les auteurs de ces méthodes s'inspirent des méthodes de génération précédente ou de la même génération.

Il faut souligner que Vickoff n'était pas contre l'Agilité mais plutôt contre les Frameworks et méthodes existantes qui selon lui ont déviés des principes et valeurs de l'agilité. Il parlait aussi dans son livre de penser à créer des méthodes Hybrides selon le besoin comme un self-service des méthodes existantes afin de rendre l'application et l'intégration des pratiques plus facile et plus souple.

Ron Jeffries, un informaticien américain, c'est l'un des 17 signataires du Manifeste Agile de ce fait Son avis est d'autant plus important il dit qu'on devrait abandonner les méthodes Agiles dans les entreprises. « *On voit alors apparaître des alliances pour la promotion des méthodes du Manifeste Agile : la Scrum Alliance, la Kanban Alliance...etc. Des formations et des offres de projets Agiles sont proposées. Toutes ces solutions sont offertes pour assister les entreprises dans leur démarche de développement de logiciel. Même si les méthodes ne sont pas toujours bien respectées, le fait de les essayer est toujours bénéfique pour l'entreprise. Car cela leur aura permis d'avoir une vision globale sur l'évolution des projets et les aider dans leur prise de décision* ». (Jeffries, 2018). Selon (Jeffries, 2018) Lorsque les pratiques Agiles sont mal comprises et appliquées, elles créent souvent une confusion et entraînent une charge de travail à faire supplémentaire en peu de temps et la pression augmente au sein de l'équipe par conséquent. Une situation défavorable ni pour l'entreprise ni pour les développeurs, cette situation rend l'entreprise moins efficace et empêche les équipes d'atteindre leurs objectifs, avant que les méthodes Agiles ne soient adoptées. Selon Ron Jeffries, la cause des problèmes n'est pas les méthodes elle-même, mais plutôt la manière dont elles sont appliquées. Les développeurs devraient adopter les principes de base du Manifeste Agile indépendamment d'un

framework ou d'une méthode dans les projets de développement de logiciels comme cela était pensé initialement par les signataires du Manifeste. Au mieux, (Jeffries, 2018) propose d'utiliser l'eXtreme Programming qui tient compte de la gestion de projet dans la réalisation de l'application. (Jeffries, 2018)

Il faut souligner que la plupart des critiques sont des critiques des méthodes Agiles et non à l'Agilité elle-même. Et les chercheurs qui ont publiés leurs critiques sont en générale des praticiens de l'agilité. Ils reprochent aux méthodes Agiles d'avoir dériver des valeurs et principe du manifeste Agile.

Malgré les critiques cités auparavant personne ne peut nier que les pratiques Agiles sont bénéfiques pour mieux gérer les projets informatiques, les grandes firmes et leaders mondiaux de la technologie de type « Google et Facebook » fonctionnent et gèrent leurs projets en mode Agile, le projet de la voiture autonome de google est menés du début à la fin avec la méthode SCRUM. De ces faits, comme n'importe quel découverte les méthodes Agiles ont pris une part de critique de certains inconvénients que les praticiens doivent les prendre en considération, mais les avantages sont nombreux convaincants à rendre le passage vers l'Agilité obligatoire. Ce passage doit précéder par une étude préalable et l'organisation ne doit pas se limiter sur une seule méthode, même pas deux ou trois, il est recommandé se basé sur les valeurs et principes du manifeste Agile et construire sa propre méthode inspirée des pratiques de méthodes existantes.

## **2. Cadre conceptuel**

Avec l'évolution des nouvelles technologies, l'informatique évolue et les programmes informatiques devient de plus en plus complexes, aussi ces programmes sont devenus indispensables dans notre vie quotidienne que ce soit personnelle ou professionnelle, et ça grâce à la démocratisation de la technologie de l'information et la digitalisation des Systèmes d'information « SI ». Aujourd'hui l'informatique est le support de développement des SI. Les entreprises donnent trop d'importance aux projets informatiques, et l'échec de ces projets devient une hantise justifiée par des statistiques des autres autant que par ses propres expériences fâcheuses. (Vickoff J. P., 2016, p. 17)

### **2.1. C'est quoi un projet ?**

On peut trouver plusieurs définitions du concept projet mai nous avons choisi une définition d'une grande référence.

*« Un projet est une initiative temporaire entreprise dans le but de créer un produit, un service ou un résultat unique. La nature temporaire des projets implique un début et une fin clairement définis. Le fait qu'un projet soit temporaire ne signifie pas nécessairement qu'il est de courte durée. La fin d'un projet est atteinte lorsque les objectifs ont été satisfaits, ou lorsque le projet est arrêté parce que ses objectifs ne seront pas ou ne pourront pas être atteints, ou lorsque le projet n'est plus utile. La décision d'arrêter un projet doit être approuvée et autorisée par une autorité compétente. » (PMBOK, 2017)*

Cela veut dire qu'un projet est une séquence de tâches délimité par le temps, les ressources et les résultats requis (ou espérés), elle a des résultats et des livrables bien définis, elle a un délai et un budget.

Un effort qu'on qualifie de projet diffère d'un effort opérationnel par le fait qu'il est entrepris temporairement jusqu'à atteindre son objectif, après quoi il est terminé, alors qu'un effort opérationnel est un effort permanent ou semi-permanent afin de fournir un service, un produit, ou un résultat.

Les projets informatiques ont une physionomie particulière par rapport aux projets d'autres domaines.

*« Qu'il soit la création d'un nouveau logiciel, la création d'un nouvel espace numérique, ou encore une évolution majeure d'un logiciel ou une infrastructure numérique, l'intégration de logiciels existants. » (SPANNEUT, 2015, p. 19)*

De nombreux facteurs technologiques peuvent caractériser un projet informatique :

- Les systèmes d'exploitation : comme Windows, Android.
- Les technologies de communications : comme la 3G, la 4G.
- Les infrastructures technologiques : comme les data centers et le cloud
- Ou aussi l'utilisation d'outils de développement, l'intégration de logiciels

Une telle diversité qui va apporter des contraintes et des risques. Ces facteurs participent à l'environnement du projet, qui est complexe et changeant, ils contribuent à une incertitude quant au résultat, sans oublier les critères du produit logiciel : « sécurité, fiabilité, temps de réponse, ergonomie, capacité de maintenance, disponibilité, exploitabilité ... » (SPANNEUT, 2015, p. 19).

Un projet informatique « IT » est un projet qui touche ou utilise les technologies d'information et de la communication dans le but de :

- Enabling businesses (permettre aux entreprises de fournir un produit, service ou résultat, améliorer la qualité des livrables, améliorer l'expérience du consommateur, etc.)
- Acquisition ou implémentation de solutions de TIC (développement d'application métier.../ infrastructure IT)
- Développement ou établissement de nouvelles offres de services IT, de nouvelles capacités/moyens.
- Les études de faisabilité sont généralement des Projets, lorsqu'elles concernent des solutions TIC, ça devient un projet IT.

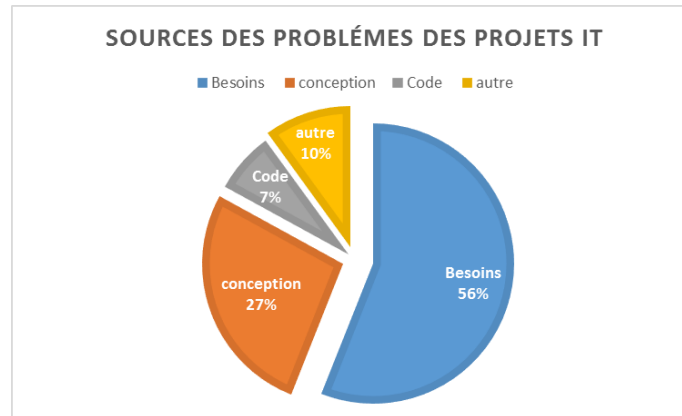
Pourquoi devrions-nous traiter les projets IT séparément des autres projets :

- Un Projet IT est constitué majoritairement d'un effort logique (soft), alors que la plupart des autres projets sont des efforts physiques.
- Le produit d'un projet informatique est généralement intangible donc il est difficile de le mesurer ou valoriser comme les autres produits physiques.
- Variété : souvent, plusieurs équipes techniques d'une variété de spécialités participent au projet IT.
- Complexité : l'identification des besoins pour un projet IT, doit être spécifique et préciser les moindres détails.
- Aspect unique de chaque projet (innovation) : souvent, on regroupe des équipes et des technologies de manière nouvelle, innovante, différente.
- Gestion différente des risques : les défis en complexité et aspect unique de chaque projet introduisent un risque technique différent par rapport aux autres projets, du fait que parfois on ne peut pas identifier ou comprendre la totalité des risques techniques au début du projet
- Défis en communications : les gens spécialisés en technologie généralement ne peuvent pas communiquer avec le gens du business et vice versa.
- Les tests et prototypage : c'est une étape importante dans le cycle de vie d'un projet IT, qui est généralement la première à sacrifier lorsque les délais sont courts ce qui est considéré comme une erreur fatale.

## 2.2.Sources des problèmes selon le chaos report

La **figure 1** est publiée par Standish groupe dans sa première édition de son rapport annuelle publié en 1994 *Chaos Report*<sup>3</sup>. Cette figure nous montre que la source principale des problèmes de projets informatiques est l'expression du besoin.

Figure 1: Les sources des problèmes des projets Informatiques



Source : chaos report 1994

## 2.3.Itératif, Incrémentale et Adaptatif

Dans le développement informatique on distingue trois approches de développement. On va illustrer les trois approches itératif, incrémentale et adaptatif afin de faciliter la compréhension des méthodes de gestion de projet par la suite.

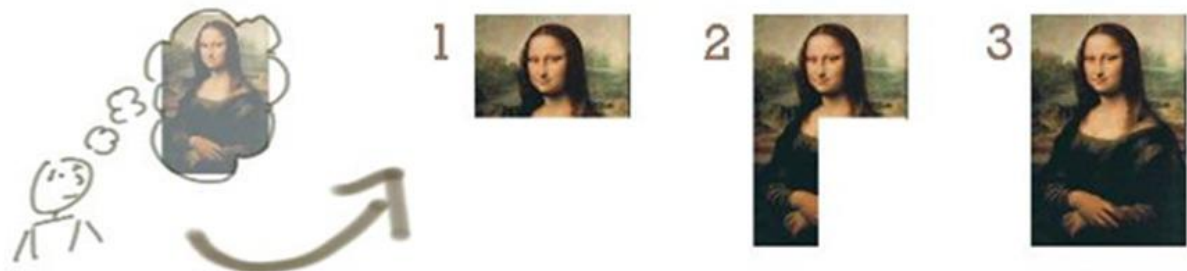
### 2.3.1. Le développement itératif VS incrémental

Les notions sous-jacentes aux principes incrémental et / ou itératif sont fréquemment mal comprises. Pour expliquer la différence entre les deux modes l'Agiliste américain Jeff Patton a utilisé la fameuse peinture de « Mona Lisa » **figure 2** en l'appelant « Mona Lisa Agile », qui a fait le tour du monde. (Vickoff J. P., 2016, p. 44),

<sup>3</sup> Chaos Rapport : est un rapport annuel élaboré par un organisme standish Group, publié par la première fois en 1994. <https://www.standishgroup.com/>

Figure 2: Figure Itératif / incrémental (Jeff Patton)

### Incrémental



### Itératif



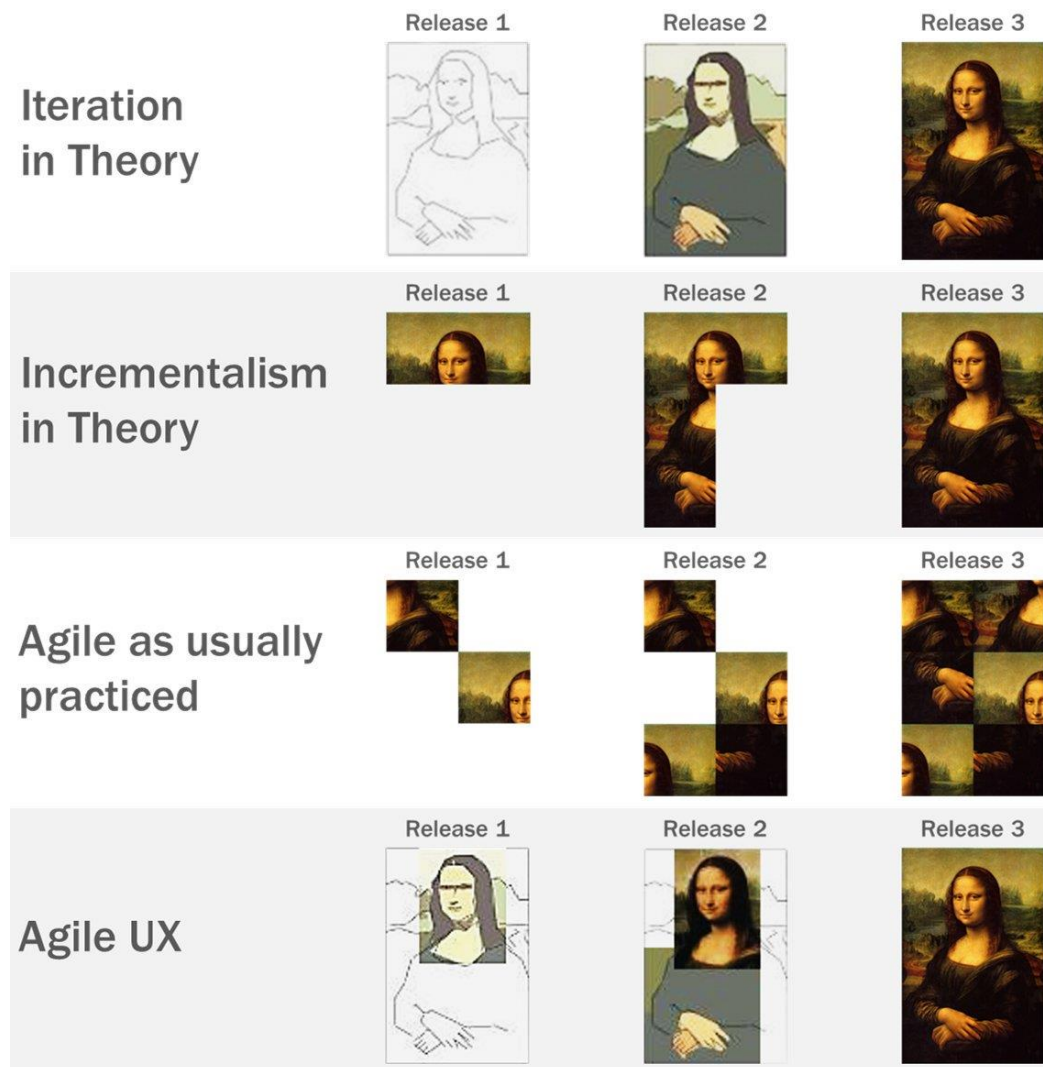
Source : (Vickoff J. P., 2016)

### 2.3.2. Développement incrémental (Cascade ou Waterfall)

Le développement incrémental, typique de l'approche waterfall (cycle en V), s'appuie sur une idée initiale complètement formée, que l'on construit ensuite morceau par morceau, jusqu'à une livraison finale, complète. L'inconvénient est qu'il faut attendre longtemps avant de voir le résultat final.

Le développement incrémental peut, sous prétexte d'agilité, conduire à un résultat dysfonctionnel, comme l'illustre « Charles Lambdin 2016 » : des morceaux sont livrés successivement, mais sans cohérence d'ensemble. C'est ce qui se passe lorsque les méthodes Agiles ne sont pas comprises. La **Figure 3** nous donne plus d'éclaircissement.

Figure 3: Charles Lambdin mars 2016



Source : (Vickoff J. P., 2016)

### 2.3.3. Développement itératif

Le développement itératif, introduit par les méthodes Agiles, part d'une idée grossière, qui est affinée par retouches successives, chacune améliorant la qualité. Dès le début, l'effort porte sur l'ensemble, au risque de ne jamais complètement finir.

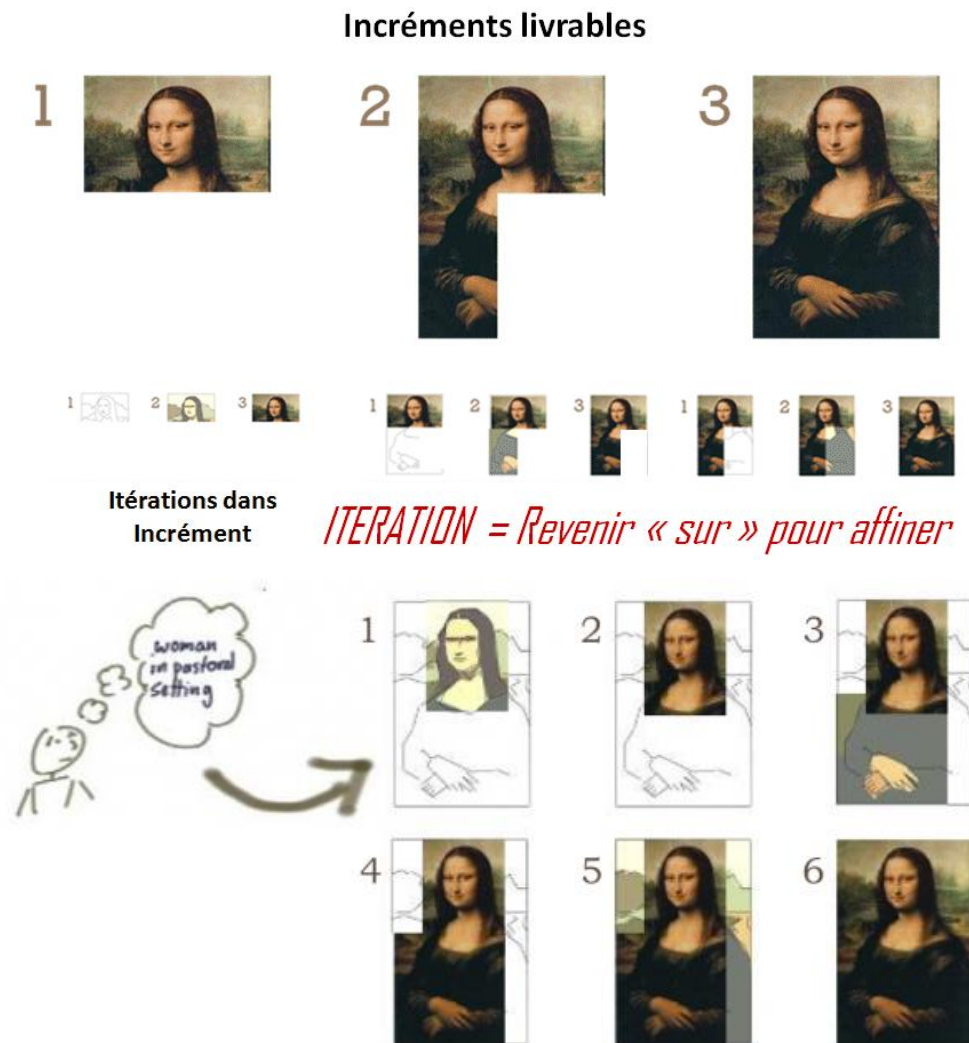
Dans un processus agile, on ne sait pas à l'avance quel sera le résultat final, et c'est ce qui fait sa force : cela permet d'innover constamment. Mais comme l'idée de départ est imprécise, le risque est d'aboutir à tout autre chose, genre Bob l'éponge au lieu d'une Joconde.

Si bien que souvent, seules les premières étapes, de conception, sont itératives, ensuite suivies d'un développement incrémental classique plus sécurisant, mais qui ne laisse plus de place à l'innovation, au risque de ne pas répondre correctement aux besoins qui évoluent entre temps.

### 2.3.4. Développement Itératif Incrémental

En pratique, les équipes Agiles combinent les deux, comme l'explique (Thomas, 2012) par le schéma expliqué dans la **figure 4** :

Figure 4: Itératif Incrémentale



Source : (Vickoff J. P., 2016)

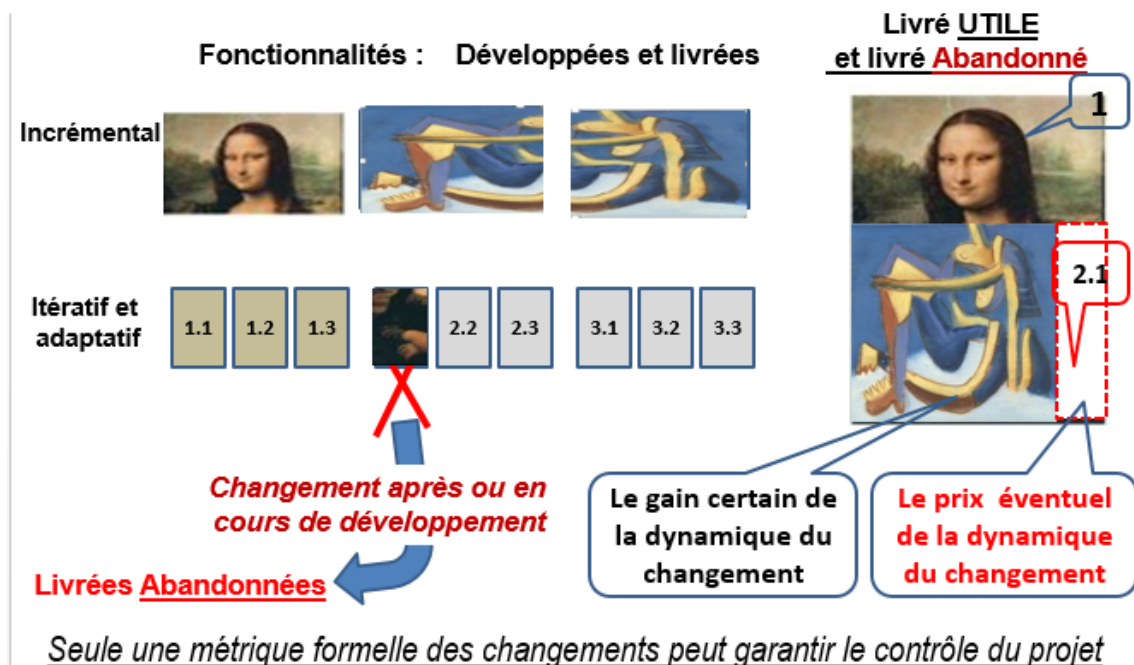
Il s'agit de partir d'une idée, d'abord précisée itérativement, ensuite partiellement développée en un incrément fonctionnel, potentiellement livrable tel quel. Chaque incrément suivant étant susceptible d'affiner les fonctionnalités existantes, est également itératif. Un livrable utilisable est produit à la fin de la première itération. En effet, si la production devait cesser à ce point, le client disposerait d'un portrait au lieu d'un buste, mais au moins il serait achevé. Cette approche combinée est beaucoup plus puissante.

### 2.3.5. Le développement adaptatif et son cout

A la fin de la première itération, un livrable utilisable est produit. En effet, si la production devait cesser à ce point, le client disposerait d'un portrait au lieu d'un buste, mais au moins il serait achevé. La **figure 5** illustre l'addition du concept « adaptatif ». (Vickoff J. P., 2016, p. 43)

L'aboutissement applicatif est parfaitement adapté aux besoins mais pourrait présenter un aspect artistique très avant-gardiste dans le cas de la métaphore de la Joconde.

Figure 5: Le développement adaptatif et son cout



Source : (Vickoff J. P., 2016, p. 47)

## 2.4. Approches de gestion de projets

Nous allons évoquer dans cette rubrique les différentes approches de gestion et de réalisation de projets informatiques.

### 2.4.1. Les approches prédictives « classiques »

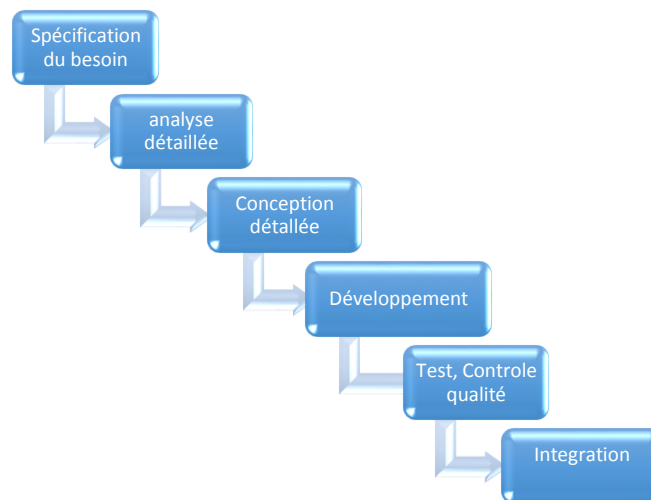
Les approches prédictives de gestion de projet dite « classiques » sont des approches qui ont connus un large succès dans les années 80. Ce sont des méthodes basées sur des activités séquentielles : on commence par le recueil des besoins, on définit le produit, on développe le produit, on effectue les tests et on livre le produit. Les approches les plus utilisées sont l'approche « en cascade » et l'approche « en V »

« *Tout doit être prévisible* » une méthodologie qui prévoit de tout planifier avant le début du projet. C'est pour ça qu'on les qualifie d'approches « prédictives ». Le plan de management du projet doit décrire comment et quand le travail sera réalisé, les modalités de planification, d'exécution, de suivi et de clôture du projet. (Messenger & Tabaka, 2013)

#### 2.4.1.1. L'approche en cascade

Le modèle en cascade fait partie des premiers modèles utilisés lors de la gestion de projet. Il faut représenter un processus sous forme de cascade, qui comporte plusieurs étapes : l'analyse des besoins, l'analyse du système, la conception, l'implémentation, la phase de test, la validation, l'exploitation. Pour pouvoir passer à l'étape suivante, il faut absolument que l'étape courante soit complètement terminée la **figure 6** résume ces étapes. (Messenger & Tabaka, 2013).

Figure 6 : L'approche en cascade



Source : (Messenger & Tabaka, 2013).

Des phases séquentielles caractérisent le cycle en « cascade », se sont des phases qui se succèdent après la validation des livrables produits lors de la phase précédente :

- Tout besoin doit être exprimé et recueilli lors de la première phase, ensuite l'analyse détaillée de ces besoins, puis la conception du système qui répondra à ces besoins.
- Avant le démarrage du développement, la conception du système que ce soit textuelle ou sous forme de diagramme doit être validée.
- L'équipe de testeurs ne pourrait lancer ses campagnes de tests fonctionnels et techniques qu'après l'achèvement du développement.
- On ne peut pas procéder à l'intégration et la mise en production globale et finale avant la correction des anomalies. (Messenger & Tabaka, 2013)

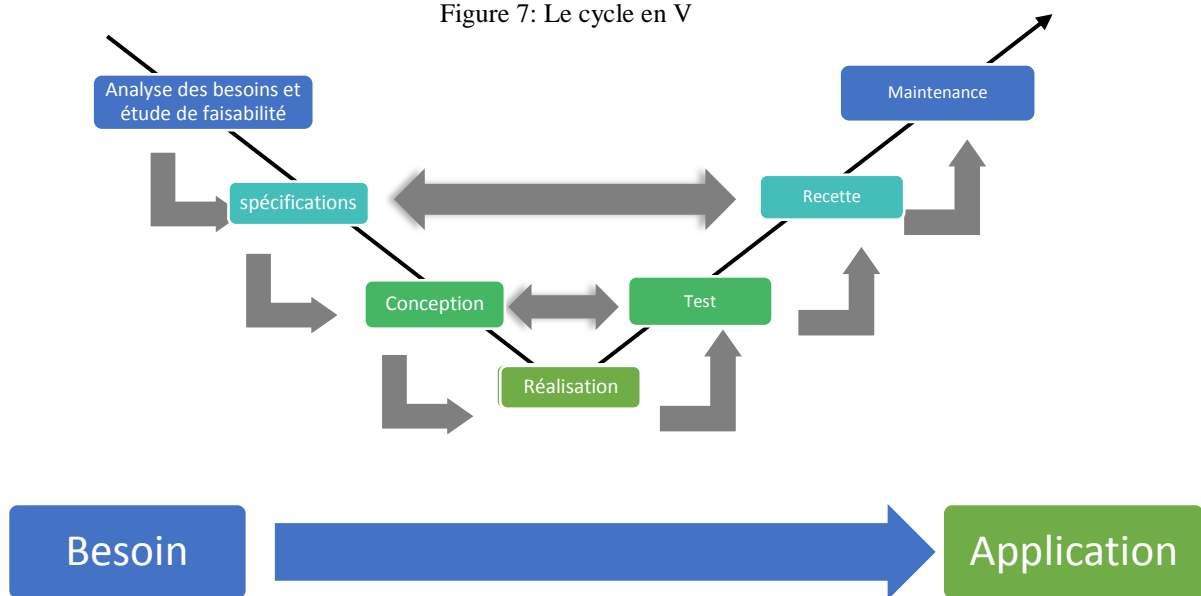
On peut résumer les limites de l'approche en cascade en :

- **Une approche trop rigide.** L'approche « en cascade » est par conséquent trop rigide pour permettre des retours en arrière ; elle suppose que l'on fasse bien du premier coup.
- **L'effet tunnel :** L'effet tunnel est une autre des caractéristiques de l'approche « en cascade » : si un projet dure un an, la phase de recueil des besoins dure deux mois et le client ne voit le résultat que neuf ou dix mois plus tard.
- **Une mauvaise communication :** L'absence de jalons intermédiaires prohibe la validation de ce que sera la version finale du produit. Les mauvaises surprises en fin de cycle de vie et le refus du changement par les équipes de développement pénalisent la qualité des relations avec les utilisateurs. La succession d'intervenants, au travers des différents corps de métier, nuit également à la fluidité de l'information, crée même une déperdition d'information et d'énergie ainsi que de nombreuses ruptures de charge.
- **La levée tardive des facteurs à risques :** Dans un cycle de vie « en cascade », les facteurs à risques sont levés tardivement, puisque les tests de performance ou d'intégration, par exemple, sont reportés après les développements, tout comme l'appréciation des IHM « Interface homme-machines », qui on le sait sont souvent sujettes à d'interminables débats très subjectifs.
- **Une documentation pléthorique :** Afin de se prémunir contre ces risques, l'approche « en cascade » s'attache fortement à la production d'une documentation importante. Aussi on s'interroge sur l'utilité de cette documentation, qui n'est, en outre, pas toujours mise à jour tout au long du projet et devient donc vite inexploitable.

#### 2.4.1.2.Cycle en V

Le modèle dit cycle V en est utilisée au jour d'aujourd'hui, son apparition date des années 80. La phase conception, la phase réalisation suivi par la phase de validation sont les trois phases principales de cette approche la **figure 7** résume ses étapes. (BONACCORSI, 2019)

Figure 7: Le cycle en V



Source : (Messenger & Tabaka, 2013).

- **Etape 1 :** C'est la phase durant laquelle on va collecter les besoins et les transformer en spécifications fonctionnelles. Une fois la première étape complètement réalisée l'on peut passer à la deuxième étape.
- **Etape 2 :** la phase de conception globale, dans cette étape on va réaliser l'architecture du projet. Et à partir de cette dernière qu'on va pouvoir commencer à réaliser les tests d'intégration qui se trouve à la sixième étape.
- **Etape 3 :** la phase de conception détaillée, durant cette étape chaque module dans la deuxième étape sera maintenant sur les différentes équipes travaillent sur le projet. A partir de cette étape on va pouvoir commencer à réaliser les tests unitaires qui se trouvent à la cinquième étape.
- **Etape 4 :** la phase de codage, la phase de développement, c'est l'étape dans laquelle on va réaliser le projet c'est l'étape du codage, le développement du projet.
- **Etape 5 :** la phase des tests unitaires qui permettra de vérifier le bon fonctionnement de chaque fonctionnalité qui aura développé dans l'étape précédente et de vérifier que la dite fonctionnalité répond bien aux spécificités demandées.
- **Etape 6 :** la phase des tests d'intégrations et des tests unitaires ont permis de voir que chaque fonctionnalité fonctionne correctement, dans la phase de test d'intégration l'on va assembler chaque module de l'application et tester si elles ne rentrent pas en conflit les unes avec les autres.

- **Etape 7** : la phase des tests de qualifications, aussi appelé tests fonctionnels ou encore tests de validations est l'étape durant laquelle l'on va s'assurer que les demandes du client dans la spécification ont bien été satisfaites. Cette étape est la dernière des phases de tests. (BONACCORSI, 2019)

**Limite :**

- Méthodologie rigide : en cas de mauvaise maîtrise du périmètre, elle sera inadaptée car toute dérive reviendrait à effectuer de nouvelles phases de conception.
- Documentation importante : la formalisation de chaque phase entraîne la production de documents en abondance voire en doublons.
- Organisation lourde : le modèle fait apparaître de nombreuses entités complexifiant l'organisation et ainsi la communication : pour la MOA « maîtrise d'ouvrage », on retrouve la Maitrise d'Ouvrage Déléguée, l'assistance à Maitrise d'Ouvrage, la Maitrise d'Ouvrage Opérationnelle... La partie MOE « Maitrise œuvre » comprend les équipes architectures et techniques. (BONACCORSI, 2019)

**2.4.2. Limites des approches prédictives**

Ces approches qui préconisent de piloter les projets en suivant les plans à la lettre appelés « plan driven développement », ont conduit les acteurs d'un projet à s'opposer à tout changement « en commençant du contenu ou le périmètre du projet jusqu'au processus de développement » tout simplement toute modification des plans initiaux.

Mais en réalité on constate que les plans sont toujours modifiés, et les besoins évoluent en permanence pour répondre aux nouveaux besoins du marché. Alors les approches prédictives sont jugées trop « rigides » et parfois rendent les organisations peu réactives dans le contexte de nouveaux projets statistiques.

**Chaos report**

En 1995 Standish group a publié des chiffres choquants **Figure 8** dans le **chaos report**<sup>4</sup> :

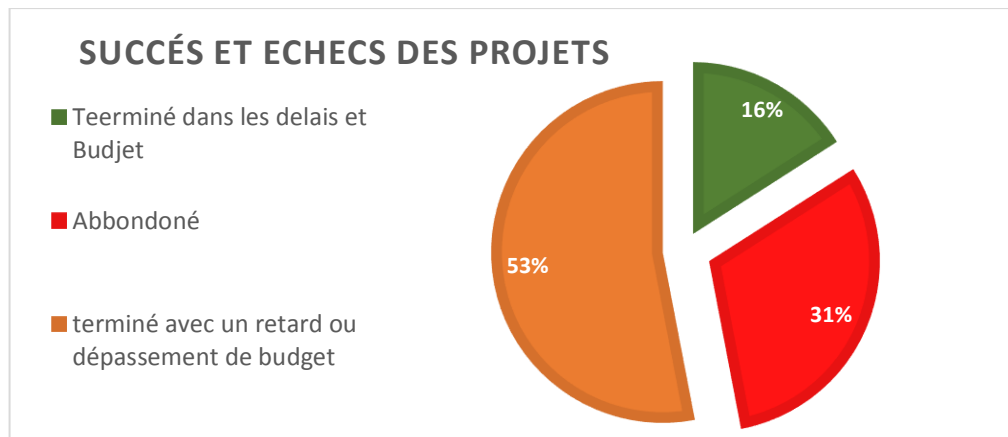
- Seulement 16% des projets IT sont terminés dans le respect des charges et délais.
- 31% de projets n'aboutissent pas.

---

<sup>4</sup> Chaos Report : est un rapport annuel élaboré par un organisme Standish Group, publié par la première fois en 1994. <https://www.standishgroup.com/> l'enquête de 1995 a touché 365 personnes qui ont participé à la création de 8380 applications durant leurs carrières.

- Plus que la moitié des projets sont livrés en retard ou en dépassant le budget.

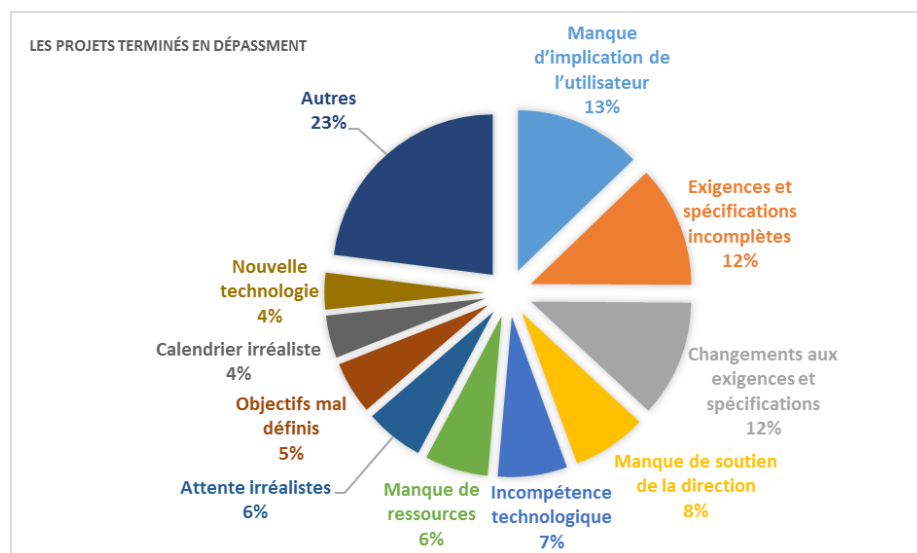
Figure 8: Succès et d'échec de projets informatiques



Source Chaos report 1995

La **figure 9** montre les raisons invoquées des projets terminés en dépassement.

Figure 9: Les raisons invoquées dans les projets terminés en déplacement

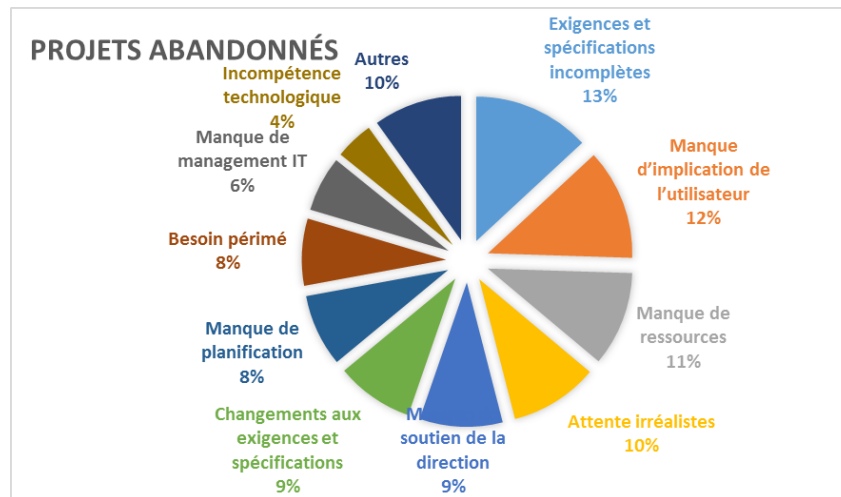


Source : Chaos report 1995

En lisant les statistiques de la **figure 8** et la **figure 9**, nous remarquons que la cause principale qui cause le dépassement de délais est le manque d'implication de l'utilisateur avec 13%, ensuite on trouve en second lieu le problème des exigences et la spécification du besoin qui sont soit incomplètes 12% ou instables 12%. On trouve aussi d'autres causes considérables comme le manque de soutien de la direction, l'incompétence technologique, le manque de ressources, et d'autres sources du problème.

La **figure 10** montre les raisons invoquées des projets abandonnés.

Figure 10: Les raisons invoquées des projets abandonnés



Source : Chaos report 1995

En lisant les statistiques de la **figure 10** nous remarquons que les projets sont abandonnés à cause de plusieurs facteurs. Principalement les exigences et la spécification du besoin incomplètes a 13%, ensuite on trouve le manque d'implication de l'utilisateurs a 12%, aussi le manque de ressources 11% et les attentes irréalistes a 10%. On trouve aussi le manque de soutien de la direction, le changement des exigences et de spécifications, le manque de planification et le manque de management IT.

La **figure 11** montre les raisons invoquées des projets réussit.

Figure 11: Les raisons invoquées des projets réussis



Source : Chaos report 1995

Dans la **figure 11** on remarque que les causes de succès d'un projet informatique est dû à l'implication des utilisateurs, le soutien de la direction, la clarté des exigences et spécifications,

une planification adaptée, des attentes réalistes, et d'autres facteurs comme équipe compétente et clarté de vision et objectifs

### **2.4.3. Une alternative : les méthodes Agiles**

Qu'est-ce qu'une méthode Agile ?

Une méthode Agile est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif, avec juste ce qu'il faut de formalisme. Elle génère un produit de haute qualité tout en prenant en compte l'évolution des besoins des clients (Messenger & Tabaka, 2013).

### **2.4.4. Le manifeste Agiles**

#### **Histoire**

Les méthodes Agiles sont considérées comme une révolution dans le management de projets informatiques. Pour tracer l'évolution de ces méthodes il faut remonter aux années 1950 avec l'introduction de Lean Thinking par la société Toyota. Pui le modèle spiral est apparu avec une première publication d'un article de (Barry W. Boehm, 1987) expliquant le modèle spiral dans le développement logiciel. Le rapport « Standish Group, chaos report 1994 » a changé la vision des chefs de projets informatiques publié en 1994, Standish Group a fait une étude sur la gestion des projets informatiques gérés avec les méthodes de gestions dites traditionnelles, en évaluant ces méthodes, les résultats de chaos report de 1995 figure (8) ont montrés la limite de ces méthodes. Le monde a connu la méthode SCRUM après la publication d'un article de (Ken Schwaber, 1995) qui s'est inspiré de la publication de (Hirotaka Takeuchi et Ikujiro Nonaka, 1986). L'année 1999 a connue l'apparition de la méthode XP ou eXtrem Programing de (Kent Beck, 1999). (Vickoff J. P., 2016, p. 33)

L'année 2001 a connu une réunion de dix-sept figures éminentes du développement de logiciel informatique aux états unis, pour objet « débattre les éléments en communs de leurs méthodes respectives et essayé de remédier aux chiffres publiés dans le chaos report 1994 ». Cette réunion a marqué la naissance du manifeste Agile ou AgileManifesto et de ses principes sous-jacents. Le manifeste Agile est constitué de 4 valeurs et 12 principes.

« *Nous avons trouvé une voie améliorant le développement logiciel en réalisant ce travail et en aidant les autres à le faire. De ce fait nous avons déduit des valeurs communes* » une déclaration de début du Manifeste Agile<sup>5</sup>.

Donc les méthodes sont des pratiques de réalisation et de pilotage de projets. Ces méthodes étaient destinées initialement aux projets de développement de logiciels, mais se sont généralisés à d'autres projets et le champ d'application s'élargit régulièrement « dernièrement on commence à entendre des applications de l'Agilité dans le Marketing ». Les méthodes Agiles se veulent plus pragmatiques que les méthodes dites classiques. Ces méthodes impliquent au maximum le client dans le projet et permettent une grande réactivité à ses demandes. Les méthodes Agiles visent la satisfaction réelle du client en priorité.

Les méthodes Agiles doivent respecter 4 valeurs fondamentales déclinées en 12 principes qui tracent une base de pratiques commune ou complémentaire. Ce sont des méthodes de gestion de projet itérative, incrémentale et récemment adaptative.

### **2.4.5. Valeurs et principes Agiles**

#### **Les 4 valeurs du manifeste agile**

Voici les 4 valeurs proposées par le manifeste Agile :

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

Nous reconnaissons la valeur des seconds éléments, mais privilégions les premiers.

#### **Les 12 principes Agiles sous-jacents du manifeste agile**

Nous allons à présent voir ces 12 principes Agiles très importants dans les projets Agiles. Ses principes suivent les 4 valeurs précédentes tout en précisant certaines choses paraissant essentielles pour leurs créateurs.

---

<sup>5</sup> Une déclaration d'ouverture du manifeste Agile. Une phrase d'ouverture devenue un slogan qui se trouve toujours dans la page d'accueil du site <https://agilemanifesto.org/>

## **1. Satisfaction des clients**

*« Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée. »* (Schwaber & Sutherland, 2017)

Dans les méthodes Agiles, les clients sont au centre des attentions. On fera notre possible pour les faire intervenir au maximum lors des différentes phases de construction du produit (ou offre).

Au pire, on tentera d'avoir de vrais représentants car la présence des clients n'est malheureusement pas toujours possible.

## **2. Accepter le changement du besoin**

*« Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client. »*, (Schwaber & Sutherland, 2017)

Les méthodes Agiles permettent dans l'ensemble d'avoir une véritable capacité de gestion d'un scope variable. Le besoin peut changer et on fait le maximum pour être dans les conditions idéales pour changer une partie de scope en cas de besoin.

Le moindre feedback ou changement de contexte peut imposer des changements au sein du scope du produit (offre).

## **3. Livraison fréquentes**

*« Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts. »* (Schwaber & Sutherland, 2017)

Le fait de livrer régulièrement des fonctionnalités ou morceaux de fonctionnalités permet de pouvoir obtenir des feedbacks plus réguliers ; ceux-ci permettront de mieux réorienter le produit si besoin.

## **4. Travail client-développeur**

*« Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet. »* (Schwaber & Sutherland, 2017)

Pour s'assurer d'avoir un produit (ou offre) plus proche de ce qui est attendu, il est très important que client (ou représentant) et ceux qui développent le produit travaillent de façon collaborative.

« Développeur » signifie ceux qui mettent la solution en place et pas forcément les « programmeurs ».

### **5. Motivation des équipes**

*« 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. »* (Schwaber & Sutherland, 2017)

Les méthodes Agiles privilégient le bien être de chaque membre des équipes. En leur offrant un espace de travail plus convivial et en leur faisant confiance, on a plus de chance d'avoir des équipes productives. Les méthodes Agiles sont très axées sur l'humain parfois trop oublié dans des grands groupes qui font des projets à la chaîne.

### **6. Le dialogue face à face**

*« 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. »* (Schwaber & Sutherland, 2017)

On ne le répètera jamais assez mais l'interaction est essentielle pour une meilleure communication. La communication est la clé de la réussite. La meilleure communication est celle dû face à face bien meilleure et recommandée que le mail, sms ou autres types de communication.

### **7. Opérationnel sinon rien**

*« Un logiciel opérationnel est la principale mesure d'avancement. »* (Schwaber & Sutherland, 2017)

Quand on mesure l'état d'avancement d'un produit ou d'une offre, le manifeste Agile rappelle qu'on ne mesure que ce qui est terminé. Tout ce qui est entamé mais non terminé est considéré comme à 0% d'avancement.

### **8. Rythme soutenable**

*« 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. »* (Schwaber & Sutherland, 2017)

Inutile de travailler jours et nuits pour avancer sur le produit. Les équipes qui font des journées à rallonges ne sont pas plus productives que celles qui travaillent de façon soutenable. En cas de perturbations possibles qui peuvent faire perdre du temps, on fera le nécessaire pour supprimer celles-ci. Travailler plus longtemps ou plus tard ne sera pas une solution viable.

### **9. L'excellence technique**

*« Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité. »* (Schwaber & Sutherland, 2017)

Le manifeste Agile insiste sur la nécessité de mettre de la qualité dans les produits réalisés afin de limiter au maximum les risques de bugs et pour assurer une continuité des futurs développements.

### **10. La simplicité**

*« La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile est essentielle ».* (Schwaber & Sutherland, 2017)

Le manifeste Agile rappelle constamment qu'il ne faut développer que le strict nécessaire.

### **11. Equipes auto-organisées**

*« Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées. »* (Schwaber & Sutherland, 2017)

On laissera l'équipe s'auto-organiser de façon à ce qu'elle trouve elle-même comment être le plus efficace possible. On laissera les choix aux équipes et on ne leur imposera pas des process parfois inutiles. Dans les grandes entreprises, on pourra voir pour que les équipes s'alignent autour des solutions qu'elles choisiront ensemble.

### **12. Amélioration continue**

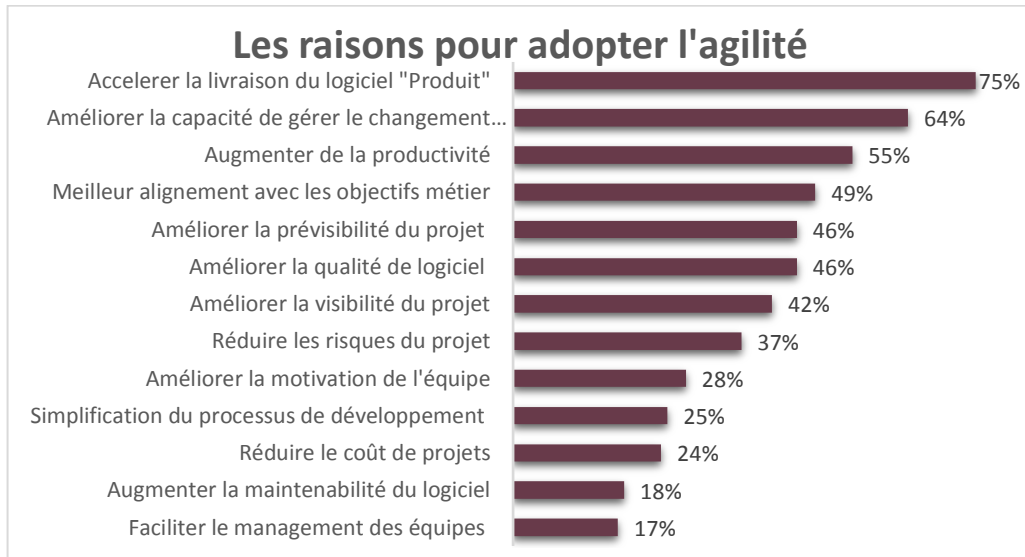
*« À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence. »* (Schwaber & Sutherland, 2017)

L'amélioration continue est essentielle dans le monde de l'agilité. On considère qu'on peut toujours améliorer sa façon de faire et son organisation.

### 2.4.6. Les raisons d'adopter l'agilité

La figure 12 extraite du « 12 th annual state of Agile report<sup>6</sup> » nous montre les raisons de l'adoption de l'Agilité par les entreprises.

Figure 12: Les raisons pour adopter l'agilité



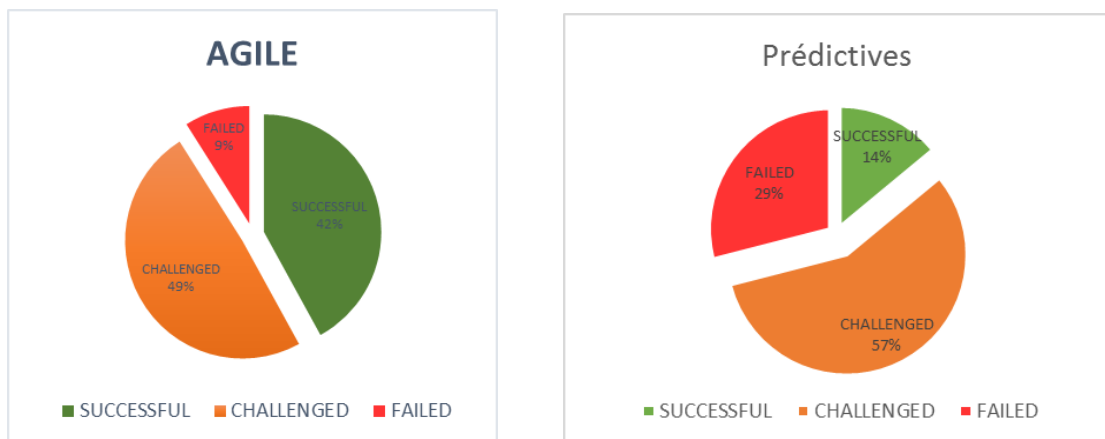
Source: (12th Annual State of Agile Report, 2018)

### 2.4.7. Les avantages obtenus par l'adoption de l'approche Agile

En 2012 Standish groupe publie des chiffres rassurants dans le chaos report.

La figure 13 montre les résultats d'utilisation des méthodes Agiles et les méthodes prédictives

Figure 13: Les résultats d'utilisation des méthode Agiles VS les méthodes classiques



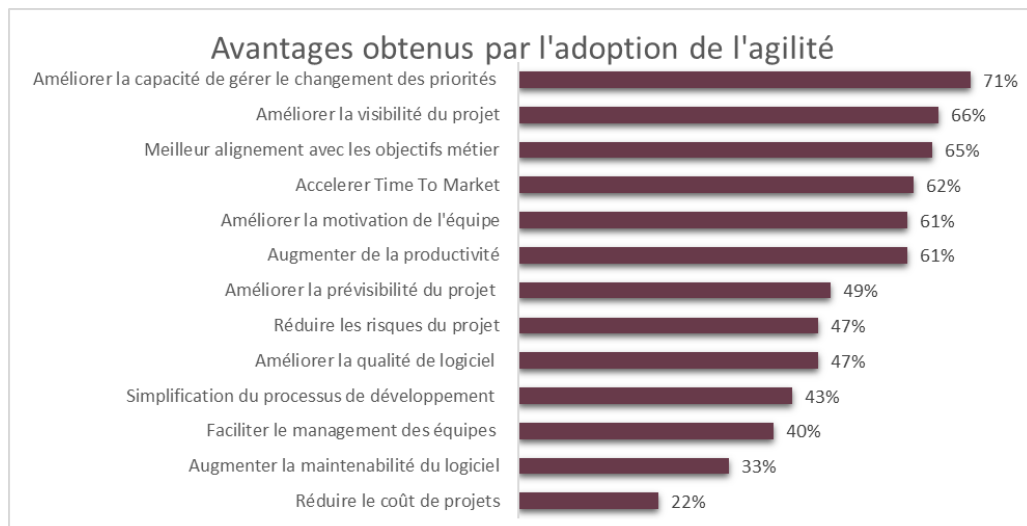
Source : chaos report 2012

<sup>6</sup> The 12th annual state of agile report est un rapport annuel élaboré et publié par [COLLABNET](http://collabnet.com) [VERSIONONE](http://versionone.com)

C'est La mise en œuvre de ces réalités pratiques qui va permettre à l'Agilité de progresser vers les sphères les plus hautes de l'organisation, et non sur la base d'une théorie globale ou structurante.

La figure 14 est extraite du « 12th annual state of Agile report » un rapport annuel sur l'état du développement Agile, qui nous donne une bonne vision sur les avantages obtenus par la mise en œuvre d'une approche Agile.

Figure 14: Les avantages obtenus par l'adoption de l'agilité



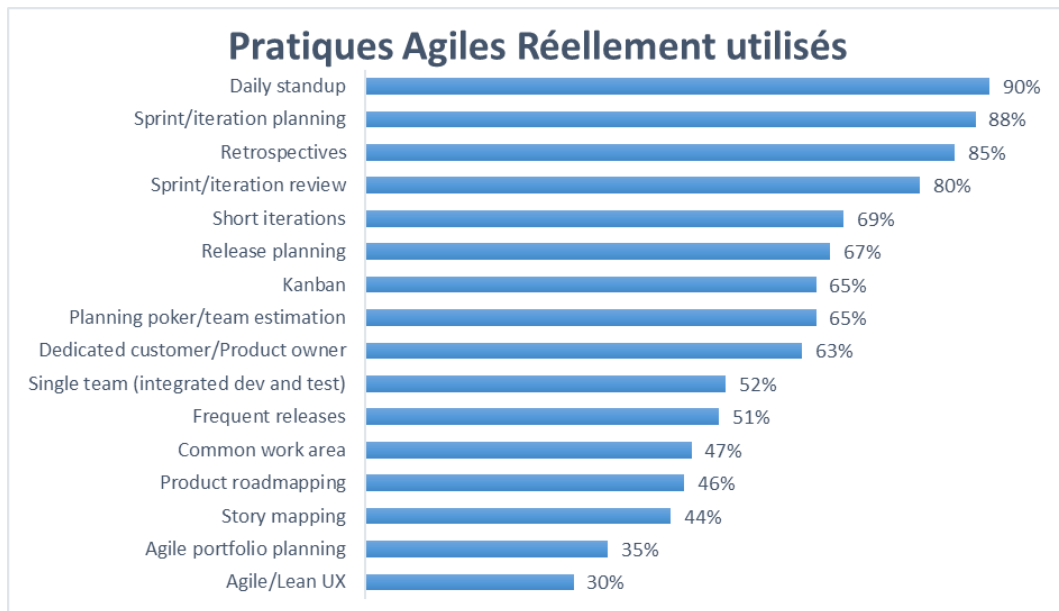
Source: (12th Annual State of Agile Report, 2018)

En analysant les figures précédentes nous constatant qu'année après année les sondages sont de plus en plus populaires et significatifs démontrent que les avantages recherchés et obtenus sont flagrants.

#### 2.4.8. Pratiques Agiles réellement utilisés

Les pratiques Agiles les plus efficaces sont souvent techniquement difficiles à maîtriser, de ce fait les organisations se limitent à organiser de simples réunions le matin et ne se donnent pas la peine à mettre réellement en œuvre des vrais pratiques. La **figure 15** nous montre les pratiques les plus utilisés selon (12th Annual State of Agile Report, 2018).

Figure 15:Pratiques Agiles Réellement utilisés



Source: (12th Annual State of Agile Report, 2018)

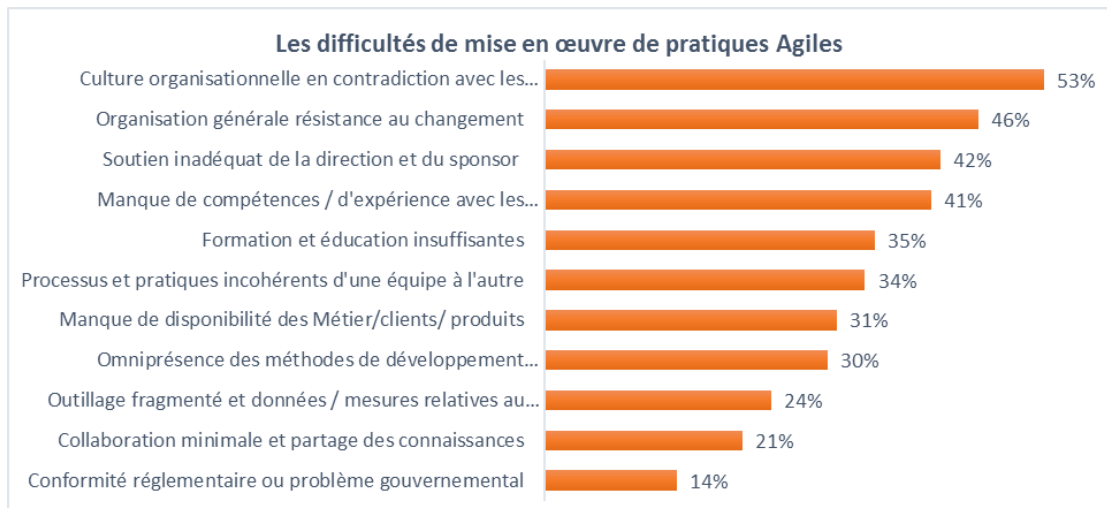
Les méthodes Agiles ne sont qu'un retour d'expérience et formulation de principe issu du bon sens par la pratique de chefs de projets chevronnés et de développeurs expérimentés durant de longues années de travail avant le manifeste agile.

#### 2.4.9. Les difficultés de mise en œuvre de pratiques Agiles

En analysant la **figure 16**, les difficultés rencontrées lors de la mise en œuvre de l'Agilité sont toujours le reflet de l'incapacité de l'organisation à accepter de résoudre ses conflits internes.

Le degré de refus se vari d'une simple réaction a un changement mineur au rejet brutal d'un changement considéré comme culturellement inacceptable. Aussi le temps du changement est une période délicate.

Figure 16: Les difficultés de mise en œuvre de pratiques Agiles



Source: (12th Annual State of Agile Report, 2018)

#### 2.4.10. Les fondements du paradigme Agile

Dans cette rubrique nous allons expliquer les détailler et expliquer les citations du manifeste Agile afin d'extraire des aspects fondamentaux :

- **L'équipe** « *les individus et leurs interactions plus que les processus et les outils* » : l'équipe est plus importante que les outils ou les procédures, il est préférable d'avoir une équipe soudée et qui communique, composé de développeur d'un niveau moyens, plutôt qu'une équipe d'experts et individualiste. Donc la communication est primordiale et considérée comme une notion fondamentale. (Vickoff J. P., 2016, p. 41)
- **L'application** « *Des logiciels opérationnels, plus qu'une documentation exhaustive* » : il est indispensable que l'application fonctionne. C'est bien d'avoir une documentation technique qui pourrais aider mai il ne faut surtout pas qu'elle soit un but ou un objectif. La documentation est considérée comme moyen de communication. Aussi il est important de commenter le code source afin qu'une autre personne pourrait comprendre et le plus important de reprendre le code cela pour transférer les compétences au sein de l'équipe. En plus la documentation représente une charge importante, et si elle n'est pas à jour elle peut être nocive. (Vickoff J. P., 2016, p. 41)
- **La collaboration** « *la collaboration avec le client, plutôt que la négociation contractuelle* » : le client doit être impliqué dans le développement. La collaboration du client avec l'équipe de développement est primordiale, il doit donner son avis et fournir un compte rendu sur sa satisfaction des fonctionnalités et l'adéquation du logiciel avec

ses attentes. Il ne faut pas tomber dans l'erreur de négocier le contrat au début du projet puis négliger les demandes du client. (Vickoff J. P., 2016, p. 41)

- **L'acceptation du changement** « *l'adaptation au changement plus que le suivi d'un plan* » : Selon Peter Drucker « *Il n'y a rien de plus inutile que de faire avec efficacité quelque chose qui ne doit pas du tout être fait* », ce n'est pas parce qu'on a suivi le plan à 100% que finalement ce qu'on faisait eu du sens, le besoin change et s'évolue rapidement et il est primordiale de souligner cet aspect. Afin de permettre l'évolution de la demande du client tout au long du projet, la planification initiale et la structure du projet doivent être flexibles. Aussi les premières livraisons du produit vont souvent provoquer des demandes d'évolution. (Vickoff J. P., 2016, p. 41)

#### 2.4.11. Le consensus des méthodes Agiles

Les modes opératoires en communs des méthodes Agiles sont :

- **Responsable Fonctionnel** : qui définit et ordonne la production des fonctionnalités de l'application.
- **Le rôle du responsable fonctionnel** : il ajuste les besoins entre chaque incrément et valide le travail de l'équipe.
- **La structuration du projet en incrément** : le projet est structuré en incrément d'une à six semaines selon la taille du projet de l'équipe de développement, la réactivité de l'équipe et la visibilité.
- **Réunion initiale** : une réunion initiale s'organise à chaque incrément pour définir les tâches à réaliser.
- **Autonomie de l'équipe** : l'équipe pilote la qualité et la performance du projet de manière consensuelle.
- **Briefing quotidien** : une courte réunion est organisée chaque jour afin de donner à l'équipe une vision globale du projet, qui évoque les éventuels problèmes et permet de trouver des solutions.
- **Reporting Mural** : Un reporting mural comme (tableau, Graphe de progression ...etc.) est mis à jour en temps réel par les membres de l'équipe.
- **Une livraison complète d'incrément** : un incrément achevé doit contenir une livraison complète « développée, approuvée et testée ».
- **La réunion finale** : une rétrospective technique du processus de développement doit être évoquée dans une réunion finale. (Vickoff J. P., 2016, p. 43)

#### 2.4.12. Les pratiques communes aux méthodes Agiles

##### **Pratiques liées au Ressources Humaines « RH » :**

- La participation de l'utilisateur final dans le groupe du travail.
- Groupe de travail disposant du pouvoir de décision.
- Autonomie et organisation centralisée de l'équipe.
- Spécification et validation permanente des exigences

##### **Pratiques liées au pilotage du projet**

- Une adaptation au changement rapide du besoin.
- Niveau méthodologique variable en fonction des enjeux du projet.
- Pilotage par les enjeux et les risques.
- Planification stratégique globale basée sur des itérations rapides.
- Réalisation en jalon par prototypage actif itératif et incrémental.
- Recherche continue d'amélioration des pratiques.

##### **Pratiques liés à la qualité de la production**

- Recherche d'excellence technique de la conception.
- Vision graphique d'une modélisation nécessaire et suffisante.
- Vision de la documentation nécessaire et suffisante.
- Normes et techniques raisonnables de qualité du code.
- Architecture à basée de composants et d'objets
- Gestion des changements automatisée.

#### 2.5. Les méthodes Agiles

Les méthodes Agiles sont dévissées en 3 générations (Vickoff J.-P. , PUMA Essentiel Méthode Agile de 3ème Génération, 2013)

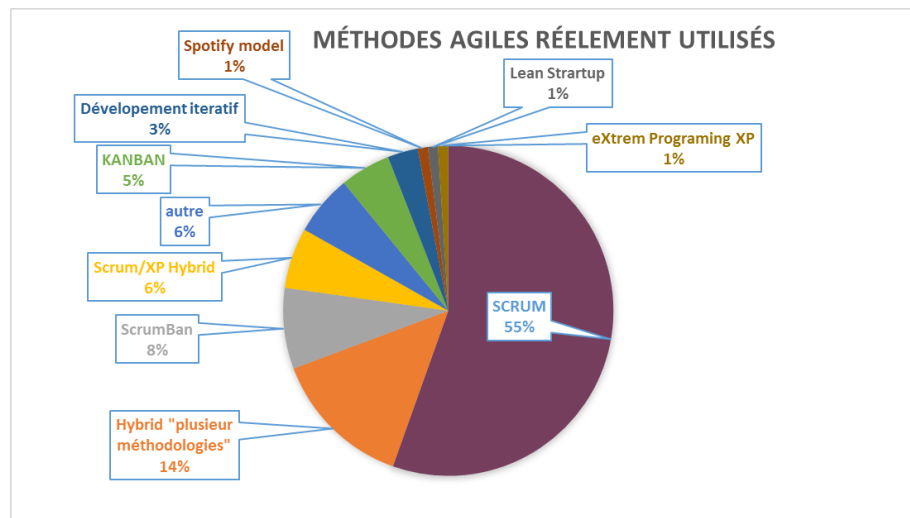
**1ère Génération :** ce sont les méthodes fondatrices conçus dans les années 90 telle que SCRUM, XP, KANBAN...

**2ème Génération :** ce sont les méthodes sorties après la naissance d'Agile manifesto, telle que DEVOPS, LEAN SOFTWARE

**3ème Génération :** ce sont les méthodes conçus après 2010, telle que SAFe 4.0

Notons que ces méthodes n'ont pas le même degré de maturité de pratiques et le même niveau d'utilisation. La **figure 17** montre les pourcentages l'utilisation des méthodes Agiles (12th Annual State of Agile Report, 2018)

Figure 17: Les méthodes Agiles réellement utilisés



Source: (12th Annual State of Agile Report, 2018)

En analysant cette figure nous constatons que la méthode la plus populaire dans la gestion de projets Informatiques est bien SCRUM. Cette popularité s'étend même sur les méthodes dites hybrides qui combinent plusieurs méthodes Agiles, on trouve la combinaison de SCRUM avec le Kanban nommé **ScrumBan**, ou la combinaison de SCRUM avec eXtrem Programing sous le nom SCRUM/XP. On trouve aussi parmi les méthodes populaires eXtrem Programing XP et KANBAN et le LEAN Software.

### 2.5.1. Les méthodes Agiles Populaires

Nous allons présenter les trois méthodes Agiles les plus populaires « SCRUM, XP et KANBAN » dans le monde de développement informatique.

### 2.5.2. SCRUM

Le mot Scrum provient du terme utilisé en Rugby « La Mêlée », qui représente un mécanisme qui permet de redémarrer le jeu après une erreur commise. L'objectif de l'équipe est d'unir ses forces pour mener le ballon dans l'embut. (Vickoff J. P., 2016, p. 61), (Schwaber & Sutherland, 2017)

Dans le monde du projet, SCRUM est un cadre de travail (Framework) dans lequel les personnes peuvent aborder des problèmes complexes et adaptatifs tout en livrant de manière

efficace et créative des produits de la plus grande valeur possible. (Schwaber & Sutherland, 2017).

L'utilisation de la méthode SCRUM date au début des années 1990 par son créateur **Ken Schwaber**. (Schwaber & Sutherland, 2017). C'est plutôt un cadre de travail qui utilise des processus et techniques afin de mettre en évidence l'efficacité relative à la gestion de projets, et aux techniques du travail. Les gains attendus du SCRUM sont l'amélioration de la gestion de projet, l'amélioration de la qualité du produit, l'amélioration de l'environnement du travail. On peut résumer SCRUM en trois mots « Léger, Simple à comprendre, et difficile à maîtriser ».

### **a. Usage de SCRUM**

Initialement développé pour la gestion et le développement de produits, SCRUM, Scrum a été largement utilisé dans le monde entier depuis sa création en début des années 1990. Scrum a été utilisé pour développer des logiciels, du matériel, des logiciels embarqués, des réseaux de fonctions interactives, des véhicules autonomes et plusieurs d'autres domaines. (Schwaber & Sutherland, 2017).

### **b. Théorie de SCRUM**

Scrum est fondé sur la théorie du contrôle empirique de processus, ou l'empirisme. Scrum utilise une approche itérative et incrémentale pour optimiser la prédictibilité et le contrôle de risque.

Trois piliers soutiennent l'implémentation d'un contrôle empirique de processus : la transparence, l'inspection et l'adaptation. (Schwaber & Sutherland, 2017).

Scrum prescrit quatre événements formels d'inspection et d'adaptation, tels que décrit dans la section événements Scrum de ce document :

- Planification du Sprint (Sprint Planning)
- Mêlée Quotidienne (Daily Scrum)
- Revue de sprint (Sprint Review)
- Rétrospective de Sprint (Sprint Retrospective)

### **c. Valeurs SCRUM**

Les valeurs Scrum sont l'engagement, le courage, la concentration, l'ouverture et le respect. Lorsque ces derniers sont incarnés et vécus par l'équipe, les piliers Scrum « de transparence, d'inspection et d'adaptation » émergent et consolident la confiance entre tout le monde.

**d. L'équipe SCRUM**

Une équipe Scrum comprend un Product Owner, une équipe de développement et un Scrum Master. Les équipes Scrum sont auto-organisées et pluridisciplinaires. Les équipes pluridisciplinaires ont toutes les compétences nécessaires pour effectuer le travail sans dépendre d'autres personnes n'appartenant pas à l'équipe. (Schwaber & Sutherland, 2017).

- **Le Product Owner**

Le propriétaire du produit est le représentant des clients et des utilisateurs. Il est responsable de maximiser la valeur du produit résultant du travail de l'équipe de développement. Il est seul à diriger l'activité de l'équipe de développement. (Schwaber & Sutherland, 2017).

- **L'équipe de Développement**

L'équipe de développement est une équipe hétérogène composée de experts en développement très compétents, et de jeunes développeurs moyennement compétents, leur collaboration consiste à fournir un incrément « Fini<sup>7</sup> » potentiellement publiable à la fin de chaque Sprint.

- **Le Scrum Master**

Il est chargé de promouvoir et supporter Scrum tel que défini dans le Guide Scrum. Les Scrum Masters remplissent leur rôle en aidant tout le monde à comprendre la théorie, les pratiques, les règles et les valeurs de Scrum. Il est le leader-serviteur de l'équipe Scrum

**e. Les événements Scrum**

Les événements prescrits sont utilisés par Scrum pour créer la régularité et minimiser le besoin de réunions non définies par Scrum. Ces événements sont spécifiquement conçus pour permettre la transparence et l'inspection. Ne pas inclure l'un de ces événements entraîne une transparence réduite et constitue une occasion perdue d'inspection et d'adaptation.

- **Le Sprint**

Le cœur de Scrum est le Sprint, qui a une boîte de temps (time-box), une durée fixe « entre 2 et 4 semaines » au cours de laquelle un Incrément Produit « Fini » fonctionnel et potentiellement

---

<sup>7</sup> Lorsqu'un élément du Backlog produit ou un Incrément est décrit comme « Fini », tout le monde doit comprendre ce que « Fini » signifie, Bien que cela puisse varier considérablement d'une équipe Scrum à une autre, les membres doivent avoir une compréhension commune de ce que signifie que le travail soit complet, afin d'assurer la transparence

publiable est créé. Les sprints ont une durée cohérente durant la phase de développement. Un sprint doit être planifié, pour se faire on doit répondre aux questions suivantes :

- Que peut-on livrer comme incrément résultant du Sprint à venir ?
- Comment sera effectué le travail à livrer et nécessaire pour achever l'Incrément ?

L'objectif du Sprint est un but fixé pour le Sprint et peut être réalisé par l'implémentation d'une partie du Backlog Produit. Par ailleurs, l'objectif du Sprint peut être une autre source de cohérence poussant l'équipe de développement à travailler ensemble au lieu d'entreprendre des initiatives distinctes. (Schwaber & Sutherland, 2017).

Un Sprint peut être annulé avant son échéance. Seul le Product Owner a le pouvoir d'annuler le Sprint, bien qu'il ou elle puisse le faire sous l'influence des parties prenantes, de l'équipe de développement ou du Scrum Master (Vickoff J. P., 2016)

- **Daily Scrum**

Ou La mêlée quotidienne, est un événement de 15 minutes destiné à l'équipe de développement. Cette dernière utilise la mêlée quotidienne pour inspecter son avancement vers l'objectif du Sprint et comment cet avancement tend à l'achèvement des travaux prévus dans le Backlog Sprint. (Schwaber & Sutherland, 2017), (Vickoff J. P., 2016)

La structure de la réunion est définie par l'équipe de développement les participants doivent répondre qu'a trois questions :

- Qu'est-ce que j'ai fait hier qui a aidé l'équipe de développement à atteindre l'objectif du Sprint ?
- Que ferai-je aujourd'hui pour aider l'équipe de développement à atteindre l'objectif du Sprint ?
- Est-ce que je vois des obstacles qui m'empêchent ou empêchent l'équipe de développement de respecter l'objectif du Sprint ?
- **Revue de sprint**

Cette réunion se veut informelle, pas une réunion de pilotage, et la présentation de l'incrément est destinée à susciter les réactions et à favoriser la collaboration. Le Scrum Master apprend à tous ceux qui y sont impliqués à le garder dans la boîte de temps. Le résultat de la revue de sprint est un Backlog Produit révisé qui définit les éléments probables pour le prochain Sprint.

Le Backlog Produit peut également être globalement ajusté pour répondre aux nouvelles opportunités d'affaires. (Schwaber & Sutherland, 2017).

- **Rétrospective de Sprint**

C'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. C.-à-d. l'équipe projet doit faire une auto-évaluation sur ce qu'ils ont bien fait et ce qu'ils ont mal fait « ce qu'a bien fonctionné et ce qu'a mal fonctionné ». (Schwaber & Sutherland, 2017). (Vickoff J. P., 2016)

### f. **Les artefacts de Scrum**

Les artefacts de Scrum représentent soit du travail, soit de la valeur fournissant ainsi de la transparence et des opportunités pour l'inspection et d'adaptation. Les artefacts définis par Scrum sont spécialement conçus pour maximiser la transparence d'informations essentielles afin que chacun ait la même compréhension de l'artefact. (Schwaber & Sutherland, 2017).

- **Backlog Produit**

Le Backlog Produit est une liste ordonnée de toutes les éléments ou fonctionnalités identifiées comme nécessaires au produit. Il constitue l'unique source d'exigences pour tout changement à apporter au produit. Le Product Owner est responsable du Backlog produit, y compris son contenu, sa disponibilité et son ordonnancement. Le Backlog Produit évolue au fur et à mesure que le produit et le contexte dans lequel il sera utilisé évoluent. (Schwaber & Sutherland, 2017).

- **Backlog Sprint**

Le Backlog 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. Le Backlog 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 « Fini ».

- **Incrément**

L'incrément est constitué des éléments du Backlog produit « Finis » pendant le sprint ainsi que de la valeur cumulative des incréments livrés dans les sprints précédents. (Vickoff J. P., 2016)

### 2.5.3. KANBAN

Kanban se concentre sur la conception, la gestion et l'amélioration de systèmes en flux de travail de la connaissance tels que du conseil ou de la conception de produits (physiques ou

logiciels). Elle peut être qualifiée comme la méthode du « Commencez par là où vous en êtes ». (Morisseau, 2014), (Anderson & Carmichael, 2016). Kanban peut être qualifié comme « Un catalyseur de changements ciblés et rapides au sein des entreprises de tout type, Qui réduit fortement la résistance au changement tout en respectant les objectifs de l'entreprise. »

La Méthode Kanban s'applique à rendre visible un travail immatériel, et s'assurer que la quantité de travail en cours ne dépasse pas un seuil qui nuit à la productivité « travailler sur ce que le client attend de façon adaptée à la capacité du service ». Pour ce faire nous utilisons un système kanban « un flux d'éléments de travail engendré par une limitation du Travail en cours (WIP pour Work in progress) et des marqueurs visuels ». (Morisseau, 2014), (Anderson & Carmichael, 2016)

Ces marqueurs visuels souvent dénommés KANBANS ou cartes kanban sont affichées sur des tableaux kanban. Ils permettent de voir et alerter lorsque trop ou trop peu de travaux sont dans le système, ce qui améliore le flux de valeur pour les clients. Les règles de limitation du travail en cours créent un système à flux tiré (pull system) : un nouvel élément de travail est « tiré » dans le système uniquement lorsqu'un autre élément vient d'être terminé et libère une capacité de réalisation. Plutôt que de « pousser » les éléments à la demande (et créer une accumulation de travail en cours). (Morisseau, 2014), (Anderson & Carmichael, 2016)

Kanban se focalise sur la livraison de services par une organisation, une ou plusieurs personnes collaborent pour produire des produits d'activité (souvent immatériels). Un service dispose d'un client qui commande le travail dont les besoins sont identifiés et qui accepte.

### a. Les Valeurs Kanban

Kanban est une méthode pilotée par la valeur. Elle est motivée par la conviction que le respect de tous les individus qui contribuent à une entreprise collaborative est nécessaire, non seulement pour le succès de l'entreprise mais pour lui donner du sens. (Morisseau, 2014), (Anderson & Carmichael, 2016)

Les valeurs de Kanban pourraient être résumées par le simple mot « respect ». Cependant il est nécessaire de l'étendre à un ensemble de neuf valeurs encapsulant la raison de l'existence des principes et pratiques de Kanban :

1. **Transparence** : la conviction que le partage public de l'information améliore le flux de valeur. Utiliser un vocabulaire clair et simple fait partie de cette valeur.

2. **Équilibre** : la compréhension que différents aspects, points de vue et capacités doivent être équilibrés entre eux pour accroître son efficacité. Certains aspects (tels que demande et capacité) entraîneront une détérioration s'ils sont déséquilibrés pendant une période prolongée.
3. **Collaboration** : « travailler ensemble » La méthode Kanban a été formulée pour améliorer la façon dont les personnes travaillent ensemble, de ce fait la collaboration est en son cœur.
4. **Centré Client** : comprendre l'objectif du système. Tout « système kanban » coule vers un point de réalisation de la valeur où les clients reçoivent un élément ou un service demandé. C'est le point d'attention naturel dans Kanban.
5. **Flux** : la réalisation de travaux tel qu'un flux de valeur qu'il soit continu ou épisodique. Voir le flux est un point de départ essentiel dans l'utilisation de Kanban.
6. **Direction (leadership)** : habilité à inspirer les autres à agir par des exemples, des mots et une réflexion. La plupart des organisations disposent de certains degrés de structure hiérarchique cependant le leadership est nécessaire dans Kanban à tous les niveaux pour accomplir la livraison de valeur et l'amélioration.
7. **Compréhension** : le savoir de départ (individuel et de l'entreprise) afin d'aller de l'avant. Kanban est une méthode d'amélioration et la connaissance individuelle est un point de départ fondamental.
8. **Accord** : l'engagement de se déplacer ensemble vers des objectifs en respectant et, quand c'est possible, en s'accommodant des différences d'opinions ou d'approches. Ce n'est pas un management par consensus mais un Co-engagement dynamique d'amélioration.
9. **Respect** : valoriser, comprendre et montrer de la considération pour les personnes. Bien à sa place au pied de cette liste, il est le fondement sur lequel les autres valeurs se reposent.

**b. Les principes fondamentaux de Kanban**

Il y a six principes fondamentaux de Kanban pouvant être divisés en deux groupes : les principes de gestion du changement et les principes de livraison de services.

**c. Principes de Gestion du Changement**

Votre organisation est un réseau d'individus, psychologiquement et sociologiquement câblés pour résister au changement. Kanban reconnaît ces aspects humains avec trois principes de conduite du changement :

Les trois principes de conduite du changement de Kanban sont :

1. Commencez par là où vous en êtes :
  - Comprendre les processus existants tels que pratiqués actuellement
  - Respecter les rôles, les responsabilités (fonctions) et les titres existants
2. S'engager à pratiquer l'amélioration par un changement évolutif
3. Encourager les actions de leadership à tous les niveaux : de l'individu à la direction.

**d. Principes de Livraison de Services**

Toute organisation est un écosystème de services interdépendants. Kanban reconnaît cela avec trois principes de livraison de services, applicables non seulement à un service mais à l'ensemble du réseau :

1. Comprendre et se concentrer sur les besoins et les attentes des clients
2. Gérer le travail : laisser les personnes s'auto-organiser
3. Votre organisation est un écosystème de services interdépendants dirigés par ses règles : réfléchissez régulièrement sur leur effectivité et comment les améliorer.

**e. Les pratiques de base de Kanban**

Les pratiques de base de Kanban définissent les activités essentielles pour ceux qui gèrent les systèmes kanban. Il y en a six :

1. Visualiser
2. Limiter le Travail en Cours (WiP : Work in Progress)
3. Gérer le flux
4. Rendre les règles explicites
5. Implémenter des boucles de feedback
6. Améliorer collaborativement, évoluer expérimentalement

Ces pratiques impliquent :

- De voir le travail et les règles qui déterminent quand et comment le traiter.

- D'améliorer le processus de manière évolutionnaire, « que ce soit en conservant et généralisant une améliorant utile, ou en abandonnant et tirant des enseignements de changements inefficaces ».

#### **2.5.4. eXtrem Programing**

L'eXtreme Programing XP est principalement l'œuvre de Kent Beck et Ward Cunningham, deux experts du développement logiciel, et plus particulièrement de la conception objet et des patterns. Avant le développement d'XP, Kent Beck devait surtout sa notoriété dans le monde Smalltalk à son livre, Smalltalk Best Practice Patterns. (Bénard, 2005)

L'eXtreme Programing (XP) est un ensemble de pratiques qui couvre une grande partie des activités de la réalisation d'un logiciel de la programmation proprement dite à la planification du projet, en passant par l'organisation de l'équipe de développement et les échanges avec le client. Ces pratiques n'ont en soi rien de révolutionnaire : il s'agit simplement de pratiques de bon sens mises en œuvre par des développeurs ou des chefs de projet expérimentés, telles que les livraisons fréquentes, la relecture de code, la mise en place de tests automatiques... La nouveauté introduite par XP consiste à pousser ces pratiques à l'extrême d'où le nom de la méthode et à les organiser en un tout cohérent, parfaitement défini et susceptible d'être répété. (Bénard, 2005)

XP se définit comme un processus léger dans lequel l'équipe se focalise sur la réalisation elle-même. Toutes les autres activités sont réduites à leur strict minimum, sachant que la priorité y est donnée à l'Agilité : XP s'adresse principalement à de petites équipes (moins de dix personnes) qui souhaitent réaliser rapidement des logiciels et réagir aisément au changement, sans faire pour autant de concessions sur la qualité du logiciel produit. (Bénard, 2005)

Dans la pratique, ce critère d'Agilité proposé par XP se retrouve à trois niveaux :

- Au niveau du code lui-même, qui est maintenu toujours aussi clair et simple que possible par une activité de remaniement permanente, et qui est soutenu par une batterie de tests automatiques permettant d'y introduire de nouvelles fonctionnalités sans craindre des régressions cachées.
- Au niveau de l'équipe de développement, dont les membres travaillent toujours en commun et sont capables d'intervenir sur toutes les parties de l'application.
- Au niveau de la gestion du projet, à travers une démarche itérative qui permet à tous les intervenants du projet « aussi bien l'équipe de développement que le client d'améliorer continuellement son pilotage en s'appuyant sur l'expérience acquise en cours de route ».

**a. Les pratiques de programmation**

Au cœur des pratiques XP, les pratiques de programmation que nous présentons ci-après permettent d'améliorer continuellement la conception et le code de l'application pour qu'elle reste toujours aussi claire et simple que possible (Bénard, 2005) :

- Conception simple (simple design) : les développeurs implémentent toujours la solution la plus simple qui puisse fonctionner. En particulier, ils n'inventent pas de mécanismes génériques si le besoin immédiat ne l'exige pas.
- Remaniement (refactoring) : les développeurs n'hésitent pas à revenir sur le code écrit pour le rendre plus « propre », le débarrasser d'éventuelles parties inutilisées, et le préparer à l'ajout de la fonctionnalité suivante. D'une manière plus générale, cette pratique propose une démarche de conception continue qui fait émerger la structure de l'application au fur et à mesure du développement.
- Développement piloté par les tests unitaires (test first programming, unit tests, developer tests) : les développeurs écrivent des tests automatiques pour le code qu'ils produisent, et ce au moment même d'écrire le code en question. Cela leur permet d'une part de mieux cerner le problème avant d'écrire le code, et d'autre part de constituer progressivement une batterie de tests qui les autorise ensuite à apporter rapidement des changements dans l'application, tout en conservant une certaine sérénité.
- Tests de recette (acceptance tests, customer tests) : le client précise très explicitement ses besoins et les objectifs des programmeurs « en participant à la rédaction de tests de recette. Comme les tests unitaires, les tests de recette doivent être automatiques afin de pouvoir vérifier tous les jours la non-régression du produit ».

**b. Les pratiques de collaboration**

Dans une équipe XP, tous les développeurs travaillent ensemble et interviennent sur la totalité de l'application. Cela garantit la qualité de cette dernière à travers les relectures croisées que cela engendre, mais cela rend également le travail plus motivant et offre une plus grande souplesse dans l'affectation des tâches. Les pratiques qui régissent cette organisation sont les suivantes (Bénard, 2005) :

- Programmation en binôme (pair programming) : lorsqu'ils écrivent le code de l'application, les développeurs travaillent systématiquement à deux sur la même machine « il s'agit là d'une forme « extrême » de relecture de code, dans laquelle les deux développeurs collaborent activement pour résoudre les problèmes qu'ils rencontrent ».

Les binômes changent fréquemment, ainsi chacun est amené à travailler tôt ou tard avec tous les autres membres de l'équipe.

- Responsabilité collective du code (collective code ownership) : tous les développeurs de l'équipe peuvent être amenés à travailler sur toutes les parties de l'application. De plus, ils ont le devoir d'améliorer le code sur lequel ils interviennent, même s'ils ne sont pas les auteurs initiaux.
- Règles de codage (coding standards) : les développeurs se plient à des règles de codage définies par l'équipe elle-même, de manière à garantir l'homogénéité de leur code avec le reste de l'application, et ainsi à faciliter l'intervention d'autres développeurs.
- Métaphore (metaphor) : les développeurs n'hésitent pas à recourir aux métaphores pour décrire la structure interne du logiciel ou ses enjeux fonctionnels, de façon à faciliter la communication et à assurer une certaine homogénéité de style dans l'ensemble de la conception, l'idéal étant de décrire le système dans son ensemble par une métaphore unique.
- Intégration continue (continuous integration) : les développeurs synchronisent leurs développements aussi souvent que possible « au moins une fois par jour. Cela réduit la fréquence et la gravité des problèmes d'intégration, et permet de disposer à tout moment d'une version du logiciel qui intègre tous les développements en cours ».

### **c. Les pratiques de gestion de projet**

Les pratiques de programmation et de collaboration permettent de créer un contexte dans lequel une démarche de spécifications et de conception purement itérative devient viable. Les pratiques suivantes montrent comment XP exploite cet avantage afin de s'assurer que l'équipe et son client restent en phase tout au long du projet, de façon à converger au plus tôt vers un produit adapté aux besoins du client :

- Livraisons fréquentes (frequent releases) : l'équipe livre des versions du logiciel à un rythme régulier, aussi élevé que possible la fréquence précise étant fixée par le client. Cela permet à l'équipe comme au client de s'assurer que le produit correspond bien aux attentes de ce dernier et que le projet est sur la bonne voie.
- Planification itérative (planning Game) : la planification du projet est réalisée conjointement par le client et l'équipe de développement, au cours de séances dédiées, organisées régulièrement tout au long du projet.
- Client sur site (on-site Customer, whole team) : le client est littéralement intégré à l'équipe de développement pour arbitrer les priorités, et définir précisément ses besoins,

notamment en répondant en direct aux questions des programmeurs et en bénéficiant du feedback immédiat d'une application aux livraisons fréquentes.

- Rythme durable (sustainable pace) : l'équipe adopte des horaires qui lui permettent de conserver tout au long du projet l'énergie nécessaire pour produire un travail de qualité et mettre en œuvre efficacement les autres pratiques.

### **d. Les quatre valeurs d'XP**

Si elles sont indispensables aujourd'hui pour mettre l'eXtreme Programming en œuvre, les pratiques que nous venons de présenter ne suffisent pas pour autant à le définir. Il ne s'agit en définitive que de techniques, destinées à faire émerger un environnement de travail marqué par les quatre qualités érigées en valeurs par XP et qui en font l'essence : la communication, la simplicité, le feedback et le courage.

- **La communication pour une meilleure visibilité**

Du point de vue d'XP, un projet de développement est avant tout un effort collectif de création, dont le succès dépend d'une part de la capacité de ses différents intervenants à s'accorder sur une vision commune de ce qui doit être produit, et d'autre part de leur capacité à synchroniser leurs actions individuelles pour atteindre l'objectif commun. Or, ces deux conditions dépendent en majeure partie de la qualité de la communication qui lie ces intervenants entre eux. Mais sur le fond, il n'y a finalement pas grand-chose de spécifique à XP sur ce point.

Ce qui démarque l'approche XP dans ce domaine, c'est l'accent mis sur la communication directe, sur le contact humain. Certes, la communication orale présente des faiblesses en termes de structuration de l'information et de traçabilité, mais elle tire sa force de sa simplicité et des interactions rapides entre les interlocuteurs qui permettent de converger rapidement sur les informations essentielles. Au quotidien, la communication directe permet donc d'obtenir une « bande passante » nettement supérieure à l'écrit, et XP exploite largement cet avantage dans la plupart de ses pratiques. Mais elle apparaît sous une forme différente : toutes les informations ayant trait à l'implémentation et la conception se retrouvent dans le code lui-même dont la clarté fait l'objet de nombreux efforts et dans la batterie de tests qui l'accompagne, et celles relatives aux besoins du client sont consignées d'une manière on ne peut plus formelle, sous la forme de tests de recette automatiques.

- **La simplicité comme garantie de productivité**

Une personne qui arrive sur un projet XP ne doit pas s'étonner de s'entendre répéter les quelques « mantras » suivants : « La chose la plus simple qui puisse marcher » (The simplest thing that could possibly work), « Tu n'en auras pas besoin » (You ain't gonna need it et son acronyme YAGNI), ou encore « Une fois et une seule » (Once and only once). Tous trois sont l'expression d'une même valeur : la simplicité.

Cette simplicité touche d'abord à la réalisation même du logiciel, dans laquelle un soin particulier est apporté au code pour le débarrasser de toute complexité superflue. Les développeurs sont donc invités à implémenter la chose la plus simple qui puisse marcher et, en ce qui concerne ces fameux mécanismes génériques, ou encore ces outils « magiques » qui en font plus que ce qu'on leur demande, ils n'auront pas besoin ! En revanche, simple ne veut pas dire simpliste : les solutions adoptées doivent être aussi simples que possible, mais toutes les duplications doivent être éliminées de sorte que chaque information, chaque mécanisme ne soit exprimé qu'une fois et une seule ce principe est garant de la facilité de modification du code sur le long terme.

La simplicité est également recherchée dans le choix des outils et dans la méthode de travail elle-même.

En définitive, XP est fondé sur un pari : « faire simple un jour, avec la possibilité de revenir en arrière le lendemain si un besoin différent apparaît. L'éventuel coût supplémentaire induit sera, d'une part réduite parce que l'application est restée assez simple pour évoluer facilement, et d'autre part sera bien peu de choses au regard des économies réalisées à ne pas faire ce qui n'aurait servi à rien. » C'est au succès de ce pari qu'une équipe XP doit sa vitesse de développement et son ouverture au changement.

- **Le feedback comme outil de réduction du risque**

Malgré ce que peuvent laisser supposer son nom et ses apparences immédiates, l'eXtreme Programing est avant tout un processus de réduction du risque dans le projet. Le risque est en effet soigneusement contrôlé à tous les niveaux, par la mise en place de boucles de feedback qui permettent à l'équipe de développement, comme à son client, de savoir à tout moment dans quel état se trouve réellement le projet, et de pouvoir rectifier le tir au fur et à mesure pour mener le projet à son terme avec succès.

Mais le feedback ne se borne pas à l'observation : la pratique de la planification itérative permet de tirer parti des informations recueillies pour, à la fois, améliorer la planification elle-même et faire converger le produit vers une solution mieux adaptée aux besoins réels du client.

L'activité de programmation fait également l'objet de divers mécanismes de feedback, tout d'abord à travers les tests unitaires mis en place, qui donnent aux développeurs des indications immédiates sur le fonctionnement du code qu'ils écrivent. Enfin, les développeurs s'appuient en permanence sur le feedback de leur binôme pour s'assurer de la validité et de la qualité du code qu'ils produisent.

Ce feedback permanent est un facteur de qualité, puisque les intervenants du projet améliorent sans cesse leur travail à partir de l'expérience qu'ils accumulent.

- **Le courage de prendre les bonnes décisions**

L'expérience fait apparaître que la mise en œuvre des pratiques XP requiert une certaine dose de cran. En effet, il faut du courage pour se lancer dans un projet sans avoir au préalable tout spécifié et conçu dans le détail, même si l'on sait que le processus suivi comporte de nombreux mécanismes de feedback.

Il faut également du courage pour se borner à réaliser des choses simples, se focaliser uniquement sur les besoins du moment en se disant qu'on pourra adapter l'application à de nouveaux besoins, le moment venu. Enfin, il faut du courage pour appliquer les principes de communication et de feedback, en particulier lorsqu'il s'agit de maintenir une transparence complète, même lorsque les nouvelles ne sont pas bonnes, ou encore lorsqu'il s'agit de travailler ouvertement avec son binôme en acceptant de lui montrer nos propres limites ou lacunes.

Ces trois méthodes sont bien détaillées dans l'annexe B

### **3. Modèle de la recherche**

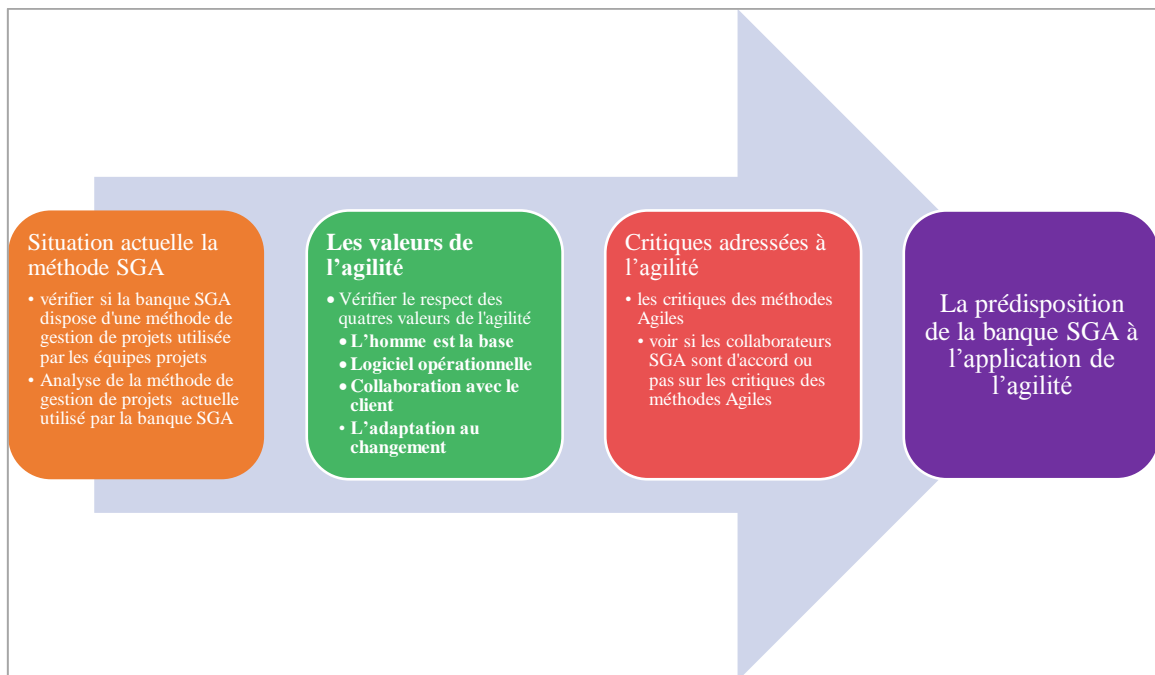
Adopter l'Agilité dans une organisation est une décision stratégique, et la performance et l'efficacité de l'utilisation d'une méthode Agile n'est perceptible qu'avec la maturité et la maîtrise de cette méthode. Pour cela et avant de choisir une méthode Agile il faut faire un diagnostic au sein de l'organisation avant de valoriser les prérequis et pratiques Agiles déjà mis en place dans cette organisation.

Dans ce contexte, notre étude est de faire une enquête sur les pratiques Agiles utilisée au sein de l'organisme d'accueil « la banque Société Générale Algérie SGA », et de valoriser les

pratiques Agiles utilisés par les acteurs projet de la banque, vérifier la prédisposition à l'application de l'Agilité et mesurer l'effort à fournir pour mieux appliquer cette méthode.

Aussi dans notre travail allons essayer de sortir avec des recommandations sur la façon que nous jugeons optimale pour adopter l'agilité. Aussi une feuille de route sera tracée afin d'assurer une transition optimale. La **figure 18** est un schéma qui résume notre modèle de la recherche, l'enchaînement de notre enquête et la finalité de notre travail.

Figure 18: La prédisposition de la banque SGA à l'application de l'agilité



Source : par nos soins

# **CHAPITRE 2 : Méthodologie de recherche et contexte organisationnel**

Dans la première partie de ce chapitre nous allons détailler la posture épistémologique qui encadre notre recherche ainsi que la démarche méthodologique adoptée. Ensuite nous allons aborder les outils de collecte et de traitement de données, nous terminons par l'échèle de mesure. Dans la deuxième partie nous allons expliquer le contexte organisationnel

### **1. Méthodologie**

Dans cette partie nous allons voir l'aspect méthodologique en commençons par l'épistémologie, en passant par l'approche méthodologique et la population ciblée, et nous terminons par évoquer les limites de la recherche.

#### **1.1. Approche épistémologique**

##### **1.1.1. L'épistémologie**

Le terme épistémologie (ou plus précisément le terme anglais epistemology) a été employé par le métaphysicien James Frederick Ferrier (1808-1864) pour désigner une théorie de la connaissance. L'épistémologie est construite à partir des termes grecs épistémé (connaissance théorique, savoir) et logos (discours rationnel, langage, jugement). Il apparaît pour la première fois dans un ouvrage de langue française en 1901, dans la traduction d'un ouvrage de Bertrand Russel. L'épistémologie vise fondamentalement à caractériser les sciences existantes, en vue de juger de leur valeur et notamment de décider si elles peuvent prétendre se rapprocher de l'idéal d'une connaissance certaine et authentiquement justifiée. (MOUKALA , 2011)

Tout travail de recherche repose sur une certaine vision du monde, utilise une méthodologie, propose des résultats visant à prédire, prescrire, comprendre ou expliquer. Une explication de ces présupposés épistémologiques permet de contrôler la démarche de recherche, et d'accroître la validité de la connaissance, De là deux grands paradigmes se sont constitués :

##### **1.1.2. Le positivisme**

Le positivisme est le paradigme philosophique fondé par Auguste Comte (1798-1857) qui considère que l'homme ne peut atteindre les choses en elle-même (leur être, leur essence) et que seuls les faits expérimentés ont une valeur universelle. Il a pour but de codifier les connaissances dites "positives", celles qui découlent directement de l'observation et de l'expérience et d'éliminer tout ce qui subit l'influence de la métaphysique. Le positivisme établit une hiérarchie entre les sciences qui part de l'étude des corps bruts et s'élève jusqu'aux corps organisés, (MOUKALA , 2011), (Esfeld , 2009)

La vision positiviste par rapport à « Pourquoi », « Quoi » et « Comment » de la recherche d'une connaissance peut être résumée de façon suivante :

Le raisonnement du positivisme est un raisonnement déductif qui part du général au particulier. La déduction correspond au processus qui permet de conclure (déduire) une affirmation à partir d'hypothèses, de prémisses ou d'un cadre théorique : les conclusions résultent formellement de ces prémisses ou de cette théorie.

### **1.1.3. Le constructivisme**

Pour la philosophie de la science et l'épistémologie, le constructivisme ou le constructivisme épistémologique est une approche de la connaissance qui est apparue vers la moitié du XXe siècle. D'après ce courant, la réalité est une construction créée par celui ou celle qui l'observe. Le constructivisme est issu, entre autres, des travaux de Jean Piaget (1964). (MOUKALA , 2011)

Le constructivisme est une posture épistémologique qui prétend qu'une personne développe son intelligence et construit sa connaissance en action et en situation et par la réflexion sur l'action et ses résultats. La personne appréhende et comprend les situations nouvelles à travers ce qu'elle sait déjà et modifie ses connaissances antérieures afin de s'y adapter. Chaque adaptation à une situation permet d'élargir et d'enrichir le réseau de connaissances antérieures dont dispose une personne et cette progression continue du réseau lui permet de traiter des situations de plus en plus complexe. (MOUKALA , 2011)

Le raisonnement du constructivisme est un raisonnement inductif : il part d'observations particulières pour aboutir à une conclusion de portée générale. L'induction correspond à un processus qui permet de passer du particulier (faits observés, cas singuliers, données expérimentales, situations) au général (une loi, une théorie, une connaissance générale), la posture inductive accorde la primauté à l'enquête, à l'observation, voire à l'expérience et essaie d'en tirer des leçons plus générales, des constats universaux : le chercheur cherche à établir quelques énoncés dont la validité dépasse le cadre de ses seules observations. (MOUKALA , 2011)

### **1.1.4. Posture épistémologique**

La posture épistémologique c'est prendre de la distance par rapport à son objet de recherche, afin de l'observer et de l'analyser avec le plus d'objectivité possible. Nos orientations de recherche dépendent en effet de notre posture épistémologique et influencent en retour le

positionnement méthodologique. Mettre à jour sa posture permet de justifier ses choix, de fournir un cadre à sa recherche. (Esfeld , 2009)

Pour réaliser ce mémoire nous avons adopté une posture épistémologique positiviste déductive, qui part d'observations Générale pour obtenir une conclusion de portée particulière. Un choix justifié par la nature de l'objet de recherche qui consiste à mesurer le niveau de l'Agilité et valoriser les pratiques Agiles au sein de l'organisation.

Ainsi, la nature de notre problématique qui s'intéresse à l'existence des valeurs et des pratiques Agiles dans les processus de gestion de projets informatiques au sein de la SGA nous impose de faire appel aux méthodes quantitatives plus appropriées à ce type de problématiques.

### **1.2. Approche méthodologique**

Notre travail s'inscrit dans un paradigme positiviste déductif, qui va déduire une affirmation à partir d'hypothèses et apporter les éléments de réponses à notre problématique. Du cadre théorique, nous avons choisi une méthode de collecte d'informations quantitative dans le but de collecter des données sur les pratiques et les orientations Agiles des équipes projet de la SGA. Ainsi que la prédisposition de la SGA a adopter une méthode Agile.

Partant du principe que les méthodes augmentent le taux de réussite des projets de développement informatiques comme il a été prouvé par les travaux de (12th Annual State of Agile Report, 2018) et chaos Report 2012 et chaos report 2015.

Le choix d'une méthode quantitative est justifié par la nature de la recherche ou on veut analyser les pratiques utilisées ainsi que les orientations de tous les collaborateurs de la direction SIOP<sup>8</sup>, de ce fait nous avons jugés qu'un appel à une méthode quantitative basée sur les statistiques est la meilleure solution pour déduire les réponses nécessaires à notre problématique.

Nous tentons dans notre présente recherche de s'assurer de la mise en pratique des valeurs Agiles au sein de la SGA en se basant sur la généralisation de certaines pratiques Agiles de base chez les acteurs de projets de développement informatiques au sein de la SGA.

Nous avons jugé que la démarche déductive va nous permettre d'analyser les pratiques des collaborateurs d'une manière générales puis sortir avec des résultats en mesurant le taux d'utilisation de l'Agilité en essayant de mesurer l'effort à fournir pour s'aligner aux valeurs et principes de l'agilité.

---

<sup>8</sup> Direction SIOP : Direction Système d'information, Organisation et projets

### 1.2.1. Méthode de collecte de données

La collecte de données repose toujours sur trois assises. Premièrement, des sources multiples doivent être utilisées afin de permettre l'analyse d'une variété de données, de tracer des lignes de convergence et de renforcer la validité de construit. Deuxièmement, il faut créer une base formelle de données afin de les rendre accessibles à d'autres chercheurs désirant vérifier les analyses et les résultats de l'étude. Troisièmement, il est nécessaire de maintenir une chaîne d'évidences pour assurer la cohérence et démontrer la fiabilité des données. On doit aussi y retrouver les circonstances de la collecte de ces données. Une personne externe peut ainsi suivre le cheminement des évidences depuis la question de recherche jusqu'aux conclusions. (Gagnon, 2012, p. 57)

Une collecte efficace d'informations est nécessaire afin d'obtenir les informations appropriées, bien cerner l'objet de recherche et arriver à des résultats crédibles. Pour cela nous avons choisis de collecter les informations sur le sujet à travers une diversité de sources en relation avec notre sujet : livres, articles, rapports, thèses, mémoires, publications, conférences, manuelles de références des méthodes etc. En plus notre connaissance technique du sujet et notre formation de base « Master en Informatique » nous a facilité l'accès à l'information.

Et afin de valoriser les pratiques Agiles déjà utilisées par les collaborateurs nous avons choisis deux modes de collecte de données.

#### 1.2.1.1. L'observation participante

En tant que stagiaire, nous avons commencé notre démarche de recherche par l'observation participante ou nous étions impliqués avec l'équipe de pilotage de projets responsable de validation, affectation et suivie de tous les projets de la direction SIOP. Nous avons observé le mode de fonctionnement de la direction SIOP de la SGA, et plus particulièrement le département de Pilotage, Conformité et communication. L'observation a duré près de deux mois entre le 25 décembre 2018 et le 20 février 2019 et durant cette période nous avons assistés à des ateliers de négociation, de lancement et de clôtures de projet en plus nous avons suivie quotidiennement les échanges entre le PMO<sup>9</sup> et les chefs de projet. Cette observation nous a permis de construire une idée globale sur le fonctionnement de cette direction, et en particulier nous avons observés les différents processus de gestion de projet. Et nous avons assisté à

---

<sup>9</sup> PMO Project Management Officer.

l'implémentation de la nouvelle solution de planification et de gestion de projets collaboratif nommée « SCIFORMA » qui préconise d'utiliser le mode Agile.

#### **1.2.1.2. Le questionnaire d'enquête**

L'utilisation du questionnaire correspond toujours à une volonté de mesurer quelque chose. Son usage s'inscrit dans une logique d'études à vocation descriptive ou explicative et de nature quantitative :

- Pour décrire une population ou un groupe ciblé sur un certain nombre de critères « dans notre cas le profil du répondant ».
- Pour estimer une valeur absolue ou relative.
- Pour tester des relations entre variables afin de vérifier et de valider des hypothèses.

Notre questionnaire a été conçu afin de valoriser les pratiques de l'Agilité utilisées au sein des équipes projet, et voir les pratiques que les collaborateurs préfèrent utiliser par la suite. Et tester la prédisposition de l'Agilité au sein de la SGA.

#### **1.2.1.3. Objectifs visés à travers le questionnaire**

- Connaître les différents profils des collaborateurs.
- Connaître si les équipes projet utilisent la méthode de gestion de projet SGA ou chaque chef de projet utilise sa propre méthode « c'est la demande de l'entreprise via notre tuteur le PMO qui voulais connaître la situation actuelle et le processus de gestion de projet utilisé au sein de la direction et nous avons inclus cette partie dans notre questionnaire ».
- Mesurer l'adhésion et le respect des 4 valeurs et les 12 principes de l'agilité.
- Mesurer l'appréciation et la position des répondants sur les critiques de l'agilité.

#### **1.2.1.4. Structure du questionnaire et variables**

Le questionnaire est inspiré des valeurs et des principes de l'Agilité que nous avons extrait 4 variables, l'administration était impliquée dans la conception du questionnaire, on était obligé de modifier et valider le questionnaire à plusieurs reprises avec la direction.

Le questionnaire est devisé en 2 parties avec 43 items (expression).

La première partie contient six (06) questions d'ordre général à choix multiple afin d'identifier le profil du répondant.

La deuxième partie représente le cœur de notre questionnaire composé de 37 questions.

Toutes les variables de la deuxième partie seront mesurées avec une échelle de Likert en cinq points. Les répondants doivent donner leurs opinions en cochant de « pas du tout d'accord » à « tout à fait d'accord » codifié de comme suit :

Tableau 1: Les variables et dimensions du questionnaire

Variables	Questions
La situation actuelle	La section 1 : (Q7 - Q14) : regroupe des questions pour la situation actuelle de l'organisation « la méthode de gestion de projet utilisé réellement au sein de la SGA »
L'homme est la base	
Les individus et leurs interactions plus que les processus et les outils	La section 2 : (Q15 – Q22) : regroupe des questions sur la préférence de baser sur l'homme plutôt que sur les outils.
Logiciel opérationnelle	
Des logiciels opérationnels plus qu'une documentation exhaustive	La section 3 : (Q23 – Q29) : regroupe des questions sur la préférence de baser sur la livraison de logiciels opérationnels plutôt qu'une bonne documentation.
Collaboration avec le client	
Collaboration avec les clients plus que la négociation contractuelle	La section 4 : (Q30 – Q33) : regroupe de questions sur la préférence de baser sur la collaboration avec le client plutôt de négocier un contrat ou demander une expression détaillée du besoin.
L'adaptation au changement	
L'adaptation au changement plus que le suivi d'un plan	La section 5 : (Q34 – Q38) : regroupe de questions sur la préférence de baser sur la tolérance du changement plutôt de suivre un plan.
Les critiques	
Les critiques des méthodes Agiles	La section 6 : (Q40 – Q44) : regroupe de questions sur les critiques de méthodes Agiles.

Pour la **question 39** : nous l'avons mis pour tester la volonté des collaborateurs pour faire une transition Agile.

#### 1.2.1.5. Le mode d'administration du questionnaire

Le fait que l'outil Google Forms est inaccessible via le réseau de la banque SGA à cause de restrictions recommandés par le règlement du groupe Société Générale, nous nous sommes retrouvés obliger d'acheter un hébergement de site web où nous avons développé et hébergé notre questionnaire. Après validation technique le lien questionnaire en ligne a été envoyé par mail à tous les responsables des équipes impliquées dans la gestion de projet de l'entreprise SGA et ces responsables ont envoyés le lien à leurs collaborateurs le 16 mai 2019.

Après 4 jours nous avons constatés que seulement 7 collaborateurs ont répondu à notre questionnaire. De ce fait nous avons installé un mouchard dans notre site pour calculer le nombre de visite, afin de cerner la cause de taux réduit de répondant à notre questionnaire.

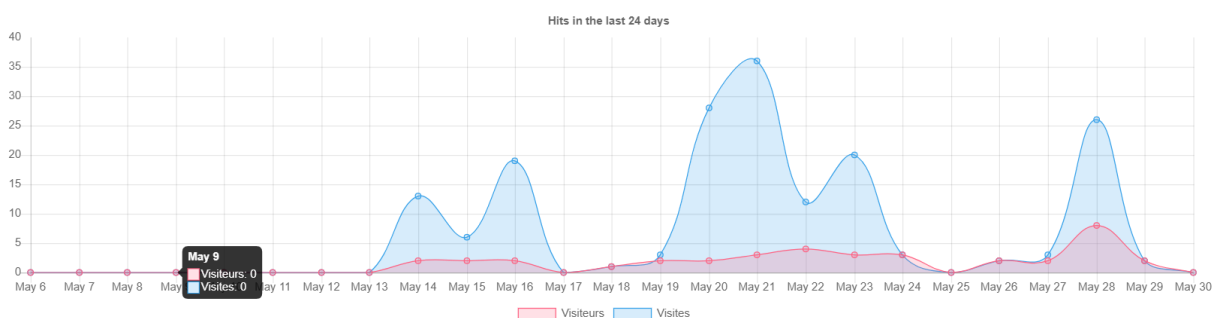
Il fallait voir si les collaborateurs ont accédé à notre questionnaire et ils n'ont pas validés à cause de mal compréhension ou le questionnaire était un peu long, ou bien ils n'ont pas accédé au questionnaire.

Par la suite le 20 mai 2019 nous avons utilisé nos relations afin de se rapprocher aux équipes et les demander un par un de remplir le questionnaire. Le nombre est de visite est sauté à 28 visites le jour même et on a eu 36 visites le lendemain, nous avons cumulés 90 visites dans deux jours. Mai le nombre de répondant était 23.

Donc on a constaté que les collaborateurs ont accédés mai ils n'ont pas rempli le formulaire. Nous avons cherché à comprendre la cause du problème et nous avons appris qu'ils ont pensé que le questionnaire est effectué par la direction dans le but de les contrôler et il ne s'agit pas d'une enquête scientifique. Nous avons envoyé un email de rappel en expliquant le but de notre enquête le 28 mai 2019 et on les a appelés un par un par téléphone afin qu'ils répondent à notre questionnaire. On a eu au totale 35 réponses à notre questionnaire sont récoltées.

La **figure 19** est extraite du mouchard qui montre le nombre de visites en bleu et le nombre de validations en rouge.

Figure 19: Historique des visites du questionnaire



Source : image extraite de notre site « mouchard »

### 1.3. Population ciblée

Le questionnaire est adressé à l'ensemble de l'équipe SIOP « Direction Système d'information, Organisation et Projet », formée des collaborateurs impliqués directement ou indirectement dans la gestion de tous les projets informatiques de la banque 82 collaborateurs.

### 1.3.1. Taille de l'échantillon

Nous avons ciblé les 69 collaborateurs de la structure SIOP en éliminant les collaborateurs de support utilisateur car ils ne sont jamais impliqués dans les projets plus 5 nouveaux collaborateurs qui n'ont même pas fini leur premier mois dans l'entreprise.

### 1.3.2. La procédure d'analyse des données

Les données collectées ont été analysées en utilisant le calcul de la moyenne et de l'écart type pour l'analyse des réponses des collaborateurs SIOP pour chaque item et dimension du questionnaire.

Aussi puisqu'il s'agit d'une approche hypothético-déductive, nous avons effectué le test T-moyenne nommé aussi « T-student » afin de valider les hypothèses.

Nous avons opté pour Le logiciel SPSS (Statistical Package for the Social Sciences) version 25.0 comme outil de traitement des données collectées pour son efficacité à faire les traitements statistiques. Ainsi que l'Excel pour convertir les données avant de les importer a SPSS.

## 1.4. Les limites de recherche

Concernant les limites, premièrement le temps n'était pas à notre faveur car l'adoption de l'Agilité se réalise à moyen terme et les résultats se mesurent au minimum après deux années de pratique, aussi l'échantillon reste minime et nécessite encore plus de répondant. Aussi le mono cas « une seule organisation » est une autre limite.

## 2. Contexte organisationnel

La nouvelle organisation SIOP a mis en place une solution de planification et de gestion de projets collaboratifs nommée « Sciforma ». Cette dernière est achetée chez l'entreprise française Sciforma. Et cette solution recommande d'utiliser le mode Agile dans la gestion de projets. De ce fait la direction SIOP a décidé de mettre en place une méthode de gestion de projets Agiles, inspirée des méthodes Agiles, mai avant de se lancer, il fallait valoriser les prérequis organisationnelles de l'Agilité et pratiques Agiles déjà utilisées au sein de l'entreprise. Et tester la prédisposition d'adopter l'agilité.

Nous étions attachés à l'équipe **Pilotage, qualité et communication**, responsable de validation, suivi, affectation et pilotage de tous les projets SIOP. Encadré par le PMO « Project management Officer ».

Dans le début du stage nous étions impliqués avec l'équipe pilotage afin de comprendre le fonctionnement de SIOP. Ensuite nous avons assisté le PMO sur la consolidation des retours de chefs de projets sur les tâches effectués durant la semaine pour présenter l'état d'avancement de tous les projets aux managers. Ensuite nous avons analysé le processus de gestion de projets de la banque SGA.

### **2.1. Présentation de l'entreprise d'accueil SGA**

La Société Générale Algérie SGA est une banque multinationale française de droit Algérien créée en 1999 et a commencé à travailler officiellement en 2001, c'est la première banque privée en Algérie. La SGA compte 1450 employés et 100 Agence Bancaire dans tout le territoire Algérien, avec un chiffre d'affaire de 19.5 milliards de dinars en 2017 et 5.4 Milliard de dinars de bénéfice.

### **2.2. Présentation de la structure d'accueil SIOP**

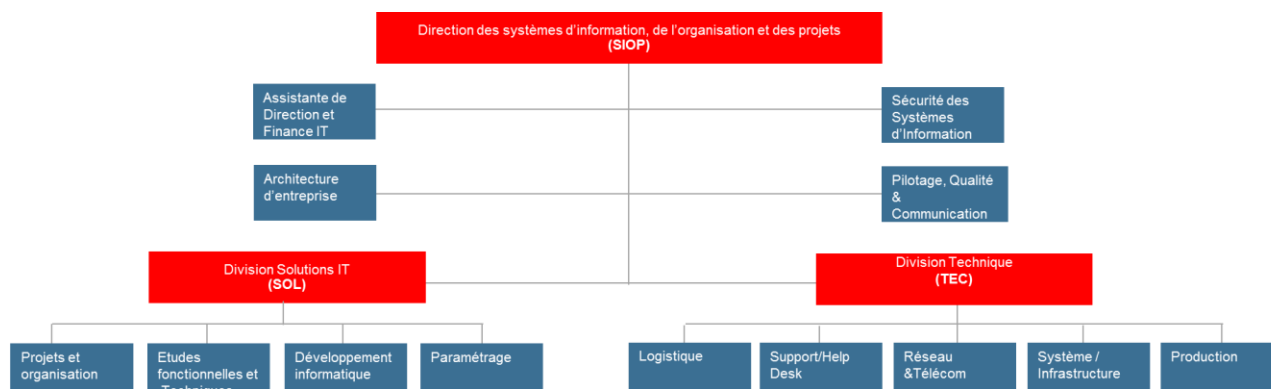
La Direction SIOP est Créée en septembre 2018 en regroupant deux directions :

- La Direction de Système d'information DSI : chargée de l'infrastructure technologique de la banque SGA. La mission de la DSI est d'administrer le Data center de la banque et maintenir toutes les applications informatiques de la banque.
- La Direction Projet et Organisation DPO : chargé de la gestion de tous les projets de la banque SGA. La mission de la DPO est de valider et assurer la gestion et le pilotage des projets de la banque. La DPO adhère les chefs de projets de la banque.

Cette nouvelle direction la SIOP est divisée en deux sous directions Sol et Tech plus de trois services « Pilotage, Architecture d'entreprise et Sécurité ».

## 2.3. L'organigramme SIOP

Figure 20: L'organisation SIOP



Source : document interne

La **figure 20** montre l'organigramme de SIOP, est divisée en deux sous directions :

- Sous-direction Technique « TECH » : regroupe les services :
  - **Logistique** : responsable de l'achat et fourniture des équipement IT « Ordinateurs, écrans, imprimantes, Serveurs ... ». 04 collaborateurs
  - **Support Help-Desk** : responsable de support et assistance des utilisateurs « tous les collaborateurs de la banque les sollicitent en cas de problème dans un équipement « ordinateurs, imprimante, scanner ... » 08 collaborateurs
  - **Réseau et télécom** : c'est le service responsable d'assurer la liaison de communication entre différents sites « Agences » de la banque « c'est le service responsable de lignes Téléphoniques, liaisons spécialisées, et accès à internet » 07 collaborateurs.
  - **Système et Infrastructure** : responsable de maintenir toute la plateforme informatique « Data center » et de fournir l'accès à tous les serveurs de la banque. C'est les administrateurs du Data center de la banque. 09 collaborateurs
  - **Production** : responsable de support et maintenance de toutes les applications de la banque. C'est les administrateurs des solutions mises en production. 08 collaborateurs.
- Sous-direction Solution IT « SOL » :
  - **Projet et organisation** : Responsable de la gestion de projets de la banque. C'est les chefs de projets. 11 collaborateurs.

- **Etude fonctionnelle et technique** : ce service est responsable des études fonctionnelles et techniques des projets. Ils sont impliqués dans tous les projets pour faire leurs diagnostics sur les impacts du nouveau projet sur les différentes applications mise en production. 16 collaborateurs.
- **Développement informatique** : lorsqu'il s'agit du développement interne d'une solution c'est le service développement qui s'en occupe de la conception et le développement de cette solution. 04 collaborateurs.
- **Paramétrage** : lorsqu'il y a une nécessité de changement de paramètres c'est le service qui s'en occupe du paramétrage de toutes les solutions informatiques de la banque. 04 collaborateurs.
- Il y a trois services attachés directement au directeur SIOP :
  - **Pilotage, Qualité et communication** : c'est le service responsable de validation, suivi, affectation et pilotage de tous les projets SIOP. Le PMO « Project Management Officer » fait partie de ce service. 03 collaborateurs
  - **Sécurité des systèmes d'information** : Responsable de la Sécurité des données de la banque contre les cyberattaques et les vulnérabilités informatiques. 05 collaborateurs.
  - **Architecture d'entreprise** : Responsable des processus et les interactions entre les différentes solutions de la banque. 03 collaborateurs.

La mission de SIOP est divisée en deux grandes missions :

- **Opérationnelle** : Assurer le bon fonctionnement de l'infrastructure Technologique et Digitale de la banque, « Division TECH »
- **Projets** : Valider, Piloter et Gérer tous les projets de la banque SGA « Division SOL »

### 2.4. La nature des projets SIOP :

La direction SIOP gère plus de 140 projets par an, et tous les collaborateurs de tous les services peuvent être impliqués dans les projets. On distingue trois types de projets :

- Projets de développement de nouveaux logiciels : il s'agit des projets que le PMO a opté pour un développement interne au sein de l'entreprise. Ce sont soit de nouveaux projets ou bien des projets d'extension « ajout de nouvelles fonctionnalités ». On compte plus de 5 projets par mois.
- Projets de paramétrage : ce sont des projets de paramétrage des applications mises en production. On compte entre 3 et 4 projets par mois.

- Projets d'acquisition et paramétrage de solution informatiques : il s'agit d'achat de solutions informatiques développées en externe. On compte entre 5 à 7 projets par mois.
- Projets d'acquisition de matérielle informatique : ce sont des projets d'achat et configuration de matériels informatiques. On compte une dizaine de projets par ans.

## 2.5. La méthode de gestion de projets SGA :

### Gestion de la demande

Une demande de projet est exprimée par le correspondant **métier**<sup>10</sup> en remplissant une fiche d'opportunité le PMO reçoit cette note d'opportunité, après analyse de l'ensemble de fiches d'opportunités reçus une réunion de « **préparation du comité d'arbitrage des projets** » est organisée chaque 15 jours par le PMO avec tous les managers SIOP « responsables de services, Chefs de départements, et directeurs » pour présenter tous les nouveaux projets. Certains projets sont refusés ou jumelés avec d'autres, et les autres sont validés.

### Pré-cadrage

Affectation et pilotage projet : les membres du Comité de pilotage sont désignés et compris un sponsor est pour valider toutes les étapes du projet.

Désigner le chef de projet : Un chef de projet est désigné par le PMO dans cette réunion selon sa charge de projets en prenant en considération sa connaissance du sujet. Le chef de projet commence son travail par le rapprochement du correspondant métier et commence à constituer son équipe projet en se rapprochant à toutes les structures impliquées dans le projet pour demander une ressource humaine.

Un développeur est désigné pour faire partie de l'équipe projet. « Peu de projets ont eu plus qu'un développeur ». On peut faire appel à l'outsourcing si tous les développeurs sont chargés de travail.

Le chef de projet commence à collecter les informations sur le projet et les processus impactés, décrire le contexte du projet, apprécier les gains attendus, définir les enjeux identifier les objectifs et la finalité du projet, délimiter le périmètre et planifier et animer le Quikoff<sup>11</sup>.

---

<sup>10</sup> Chaque direction à un correspondant métier : ou les collaborateurs de cette direction lui remonte le besoin du projet et c'est à lui de faire la demande de lancement de projet au niveau du PMO

<sup>11</sup> Quikoff : c'est le copil de lancement de projet

### **Cadrage**

**Lancement des travaux d'analyse :** le chef de projet commence à identifier les processus modélisés, les processus en cours, les processus non modélisés, et les processus inexistant à la SGA, ensuite rédiger le processus, et définir les solutions possibles, chiffrer les solutions, réaliser une étude comparative entre les solutions, effectuer le planning et animer le copil pour avoir le Go or not Go

### **Conception**

Le chef de projet rédige les spécifications fonctionnelles détaillées du produit, et revoir les spécifications fonctionnelles avec les métiers et la DSIOP.

### **Réalisation**

S'il s'agit d'un projet de développement on commence le développement séquentiel avec la méthode en cascade ou le produit n'est livré qu'à la fin du projet.

### **Préparation de l'environnement d'Homologation**

Préparer les serveurs nécessaires à accueillir la solution développée. Héberger la solution afin de commencer les tests.

### **Qualification**

Passer une batterie de tests par les utilisateurs afin de détecter les erreurs.

### **Déploiement**

Mettre la solution en production après correction des erreurs.

### **Clôture**

Faire un bilan du projet, compléter la documentation, et clôturer le projet par un copil de clôture.

## **2.6. Quelques situations désagréables de projets SGA**

### **Projets développés mais ils ne sont jamais utilisés**

Une application de gestion de temps « pointage » a été développée en interne à SGA suite à une demande DRH « direction de Ressources Humaines ». Le projet a consommé plus de 100 Jour homme de développement mais il n'est jamais utilisé par DRH. Document interne

Une autre application de gestion de projets a été développée par un prestataire externe pour la SGA suite à une demande de la DPO « direction Projet et organisation l'ancienne structure projets », le projet a duré plus de 2 ans mais la solution n'est jamais utilisée. Document interne

### **Projets qui ont connus des retards considérables :**

Un projet qui permet aux clients de consulter leurs comptes en ligne : sa livraison initiale prévue était 2014 mais le produit est livré en 2017. Document interne

Un autre projet nommé Genesys : sa livraison initiale prévue était en fin de 2014 mais il n'est livré qu'à la fin de 2016. Source Document interne

Un projet d'une application mobile : est commencé en juin 2017 prévue pour se terminer dans trois mois, l'application est livrée en début de 2019. Document interne

### **Projets de maintenance :**

On compte plus de 124 projets de maintenance dans deux ans au sein de la banque SGA. Source Document interne

## **CHAPITRE 3 : Analyse et résultats**

Dans ce chapitre nous allons commencer par la présentation de résultats de notre enquête et notre observation avec la description de l'échantillon recueilli selon les variables du questionnaire. En utilisant le logiciel SPSS version 25.0 et le logiciel Excel pour analyser ces résultats. On commence par une analyse par tris à plat puis une analyse des résultats par des tri croisés pour confirmer ou infirmer nos hypothèses.

Ensuite nous allons présenter nos décisions et nous allons essayer de faire des recommandations et tracer une feuille de route.

## **1. Résultats de l'observation**

Dès que nous avons obtenu l'accord de passer notre stage au sein de la banque SGA nous avons commencé notre travail, nous avons intégré le département Pilotage, communication et conformité nous avons commencé notre enquête par la collecte de données, de documents, et tout autre source qui pourrait nous aider dans notre travail. Nous avons commencé par l'analyse du processus de gestion de projet de la banque SGA à partir du manuel de la méthode gestion de projets. Après notre tuteur nous a impliqué sur un projet de mise en place d'une solution de gestion de projets, et nous étions impliqués dans le pilotage de tous les projets par la consolidation de toutes les tâches effectuées durant la semaine par tous les collaborateurs de la direction SIOP. Nous avons participé à quelques tâches du PMO de la banque. Nous avons assisté aux réunions de lancement, de validation et de réception de projets.

Après d'avoir passé plus de trois mois au sein de la SIOP nous avons constatés certaines pratiques de l'Agilité et certaines pratiques que nous jugeons obstacles de la mise en place de l'agilité, on va citer les points essentiels et nous détaillons par la suite :

Les pratiques de l'Agilité utilisées intentionnellement/involontairement :

1. L'implication du client : le métier fait partie de l'équipe projet et valide les étapes de projet.
2. Les réunions quotidiennes : dans certain projet et en face des situations bloquantes les équipes projets font des réunions tous les jours.
3. Les chefs de projets et les développeurs prennent en considération les retours du client.
4. La division des projets en lots est une pratique utilisée par les développeurs de SGA.

5. L'aménagement : un aménagement qui encourage la communication et la collaboration recommandé par l'Agilité est mis en place par la banque dans le cadre de la nouvelle réorganisation.

#### Les obstacles de l'Agilité

1. Le manque d'implication : ou tout le monde repousse la balle loin de lui et le soucie chaque collaborateur qu'il exécute sa tâche et dès qu'il y a un incident la chose la plus importante c'est que tout le monde cherche le coupable et soulage lorsque le problème ne vient pas de lui ou de son équipe. On retrouve même une sorte de concurrence et conflits entre les services.
2. Le manque de motivation : ou tous les collaborateurs ne font pas d'effort pour intervenir sur des problèmes en utilisant leurs compétences hors de leurs tâches et refusent de partager leurs savoir-faire avec leurs collègues. Ces comportements sont le résultat des pratiques des managers qui sanctionnent leurs éléments lors de dépassement mais pas d'encouragement s'ils se débrouillent bien.
3. La SGA a mis en place une base de connaissance pour saisir les incidents et le processus suivis afin de le résoudre puis la solution trouvée. Mais les ingénieurs refusent de remplir cette base par peur de devenir inutile dans l'équipe et sa structure reste dépendante de lui.
4. Un problème de communication, ou l'entreprise a mis en place un système d'information composé des outils et des processus de collaboration afin de faciliter la circulation de l'information mais les collaborateurs le considèrent comme une charge supplémentaire puisqu'ils sont obligés de remonter leurs tâches quotidiennes via l'outil. Ces comportements ont rendu ces outils inutiles.
5. La résistance aux changements : tout le monde veut rester dans sa zone de confort et considère chaque nouveauté ou amélioration comme une charge supplémentaire.
6. L'exigence du besoin détaillé avant de lancer un projet.

Nous avons présenté les observations jugées bloquantes ou obstacles de l'adoption de l'Agilité qu'il faut impérativement les corriger avant de penser à l'agilité.

Nous allons analyser les résultats de notre enquête et essayer de faire la liaison entre notre observation et notre enquête quantitative.

## 2. Présentation des résultats

Dans cette partie nous allons présenter les résultats de notre recherche, en commençons par la description de l'échantillon, en passant par le traitement et analyse des données et en terminant par une discussion des résultats et validation des hypothèses.

### 2.1. Description de l'échantillon :

#### 2.1.1. Présentation du profil des répondants :

##### 2.1.1.1. Répartition des enquêtés par Niveau d'étude :

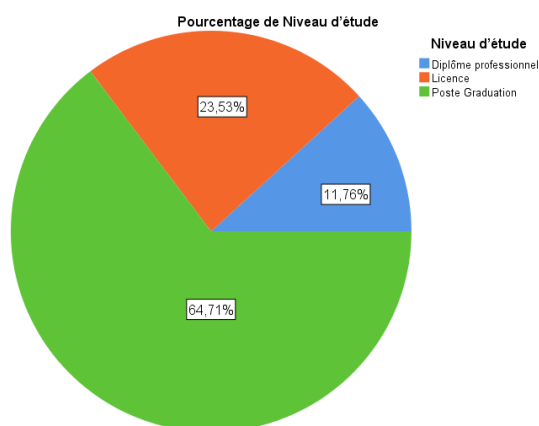
Les caractéristiques personnelles de l'échantillon de l'étude selon la variable Niveau d'études se présente dans le **tableau 2** comme suit :

Tableau 2: Répartition de l'échantillon de l'étude selon le niveau d'études

		Niveau d'étude			
		Fréquence	Pourcentage	Pourcentage valide	Pourcentage cumulé
Valide	Diplôme professionnel	4	11,4	11,8	11,8
	Licence	8	22,9	23,5	35,3
	Poste Graduation	22	62,9	64,7	100,0
	Total	34	97,1	100,0	
Manquant	9,00	1	2,9		
Total		35	100,0		

Source : Tiré des résultats de la recherche

Figure 21: Répartition de l'échantillon de l'étude selon le niveau d'études



On constate que la grande majorité des sondés ont un niveau de poste graduation soit 64.71%, 23.53% de répondant ont un niveau de licence. On constate que 88.2% de répondant ont un niveau de licence et plus ce qui place l'organisation dans un niveau d'étude considérable.

11.76% de répondants ont un diplôme professionnel et 3% n'ont pas répondu à cette question. Ces résultats sont justifiés par le besoin de la banque SGA qui veut être compétitive et gagner une longueur d'avance sur ses concurrents dans le volet investissement technologique « infrastructure numérique » cette dernière nécessite certaines compétences, trouvée chez les diplômés d'une licence et plus.

### 2.1.1.2. Répartition des enquêtés par la spécialité/domaine :

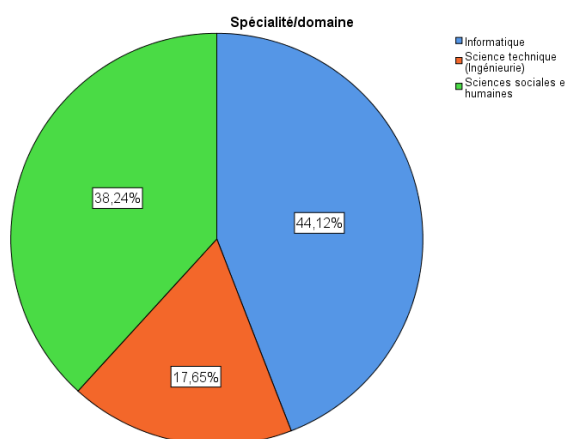
Le **tableau 3** et la **figure 22** ci-dessous décrivent la répartition de l'échantillon de l'étude selon la spécialité/domaine :

Tableau 3: Répartition de l'échantillon de l'étude selon la spécialité/domaine

		Spécialité/domaine			
		Fréquence	Pourcentage	Pourcentage valide	Pourcentage cumulé
Valide	Informatique	15	42,9	44,1	44,1
	Science technique (Ingénierie)	6	17,1	17,6	61,8
	Sciences sociales et humaines	13	37,1	38,2	100,0
	Total	34	97,1	100,0	
Manquant	9,00	1	2,9		
Total		35	100,0		

Source : Tiré des résultats de la recherche

Figure 22: Répartition de l'échantillon de l'étude selon la spécialité/domaine



On constate que la majorité des répondants ont un diplôme informatique soit 44.12% et 17.65% ont un diplôme d'ingénierie science technique « mathématique, statistiques, électronique, électrotechnique ... » Et 38.24% ont un diplôme issu de science social et humaine. On a 61.8% des répondants ont un diplôme technique, un résultat justifié par la nature des activités de

l'organisation SIOP qui nécessite l'administration de la maintenance de matériels technologiques. Les 38.24% sont en général certains managers chefs de projets.

### 2.1.1.3. Répartition de l'échantillon par expérience professionnelle :

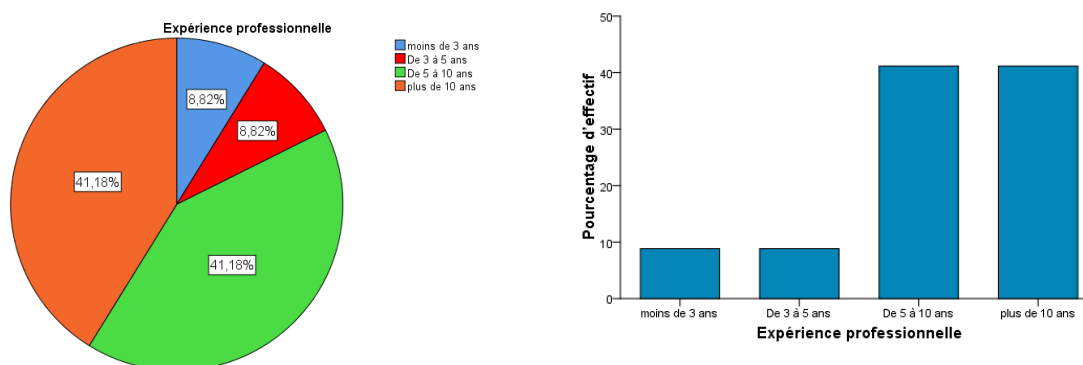
Le **tableau 4** et la **Figure 23** ci-dessous décrivent la répartition de l'échantillon de l'étude selon l'expérience professionnelle :

Tableau 4: La répartition de l'échantillon de l'étude selon l'expérience professionnelle

		Expérience professionnelle			
		Fréquence	Pourcentage	Pourcentage valide	Pourcentage cumulé
Valide	Moins de 3 ans	3	8,6	8,8	8,8
	De 3 à 5 ans	3	8,6	8,8	17,6
	De 5 à 10 ans	14	40,0	41,2	58,8
	Plus de 10 ans	14	40,0	41,2	100,0
	Total	34	97,1	100,0	
Manquant	9,00	1	2,9		
Total		35	100,0		

Source : Tiré des résultats de la recherche

Figure 23: La répartition de l'échantillon de l'étude selon l'expérience professionnelle



Nous constatons que 41% des répondants ont plus de 10 ans et 41% ont une expérience entre 5 et 10 ans soit 82% des répondants ont une expérience professionnelle de plus de 5 ans. D'après ces résultats on considère que l'organisation a une expérience de plus de 5 ans.

### 2.1.1.4. Répartition de l'échantillon par expérience au sein de la SGA :

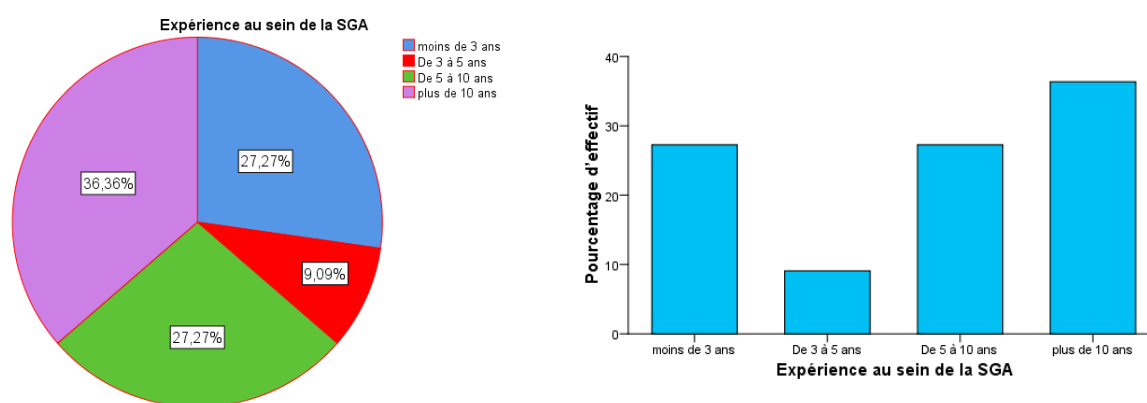
Le **tableau 5** et la **figure 24** ci-dessous décrivent la répartition de l'échantillon de l'étude sur l'expérience au sein de la SGA :

Tableau 5: La répartition de l'échantillon de l'étude selon l'expérience au sein de la SGA

Expérience au sein de la SGA					
		Fréquence	Pourcentage	Pourcentage valide	Pourcentage cumulé
Valide	Moins de 3 ans	9	25,7	27,3	27,3
	De 3 à 5 ans	3	8,6	9,1	36,4
	De 5 à 10 ans	9	25,7	27,3	63,6
	Plus de 10 ans	12	34,3	36,4	100,0
	Total	33	94,3	100,0	
Manquant	9,00	2	5,7		
Total		35	100,0		

Source : Tiré des résultats de la recherche

Figure 24: La répartition de l'échantillon de l'étude selon l'expérience au sein de la SGA



La représentation graphique de l'échantillon selon l'expérience au sein de la SGA nous montre que la moyenne la plus significative est celle de plus de 10 ans avec 34% en plus deux grande moyennes significatives de 26% tous les deux se sont celle d'une expérience entre 5 et 10 ans et moins de 3 ans. On peut considérer que 60% des répondant ayant une expérience de 5 ans et plus. Ce qui implique qu'il y a une stabilité au sein de l'organisation et la majorité de collaborateurs se connaissent depuis 5 ans et plus et ils ont l'habitude de travailler ensemble.

#### 2.1.1.5. Répartition de l'échantillon par expérience en gestion de projets :

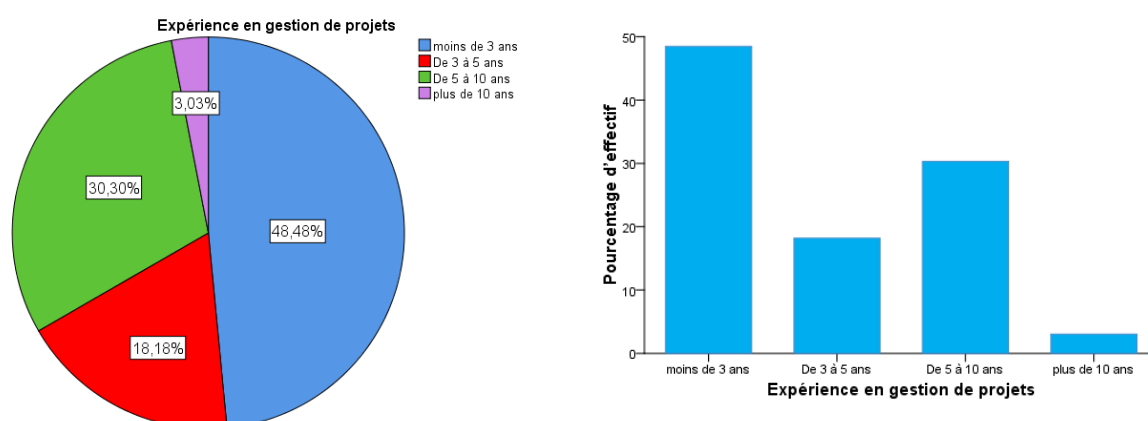
Le **tableau 6** et la **figure 25** ci-dessous décrivent la répartition de l'échantillon de l'étude selon l'expérience professionnelle :

Tableau 6: la répartition de l'échantillon de l'étude selon l'expérience en gestion de projets

Expérience en gestion de projets					
		Fréquence	Pourcentage	Pourcentage valide	Pourcentage cumulé
Valide	Moins de 3 ans	16	45,7	48,5	48,5
	De 3 à 5 ans	6	17,1	18,2	66,7
	De 5 à 10 ans	10	28,6	30,3	97,0
	Plus de 10 ans	1	2,9	3,0	100,0
	Total	33	94,3	100,0	
Manquant	9,00	2	5,7		
Total		35	100,0		

Source : Tiré des résultats de la recherche

Figure 25: La répartition de l'échantillon de l'étude selon l'expérience en gestion de projets



D'après la lecture des résultats obtenu nous constatons que la majorité des répondants ont moins de 3 ans en gestion de projets soit 48.48%, 30.3% ont une expérience entre 5 et 10 ans en gestion de projets et 18.18% ont une expérience entre 3 et 5 ans ce qui fait un équilibre d'expérience. En additionnant les deux sections nous constatons que 48.5% des répondant ont une expérience entre 3 et 10 ans. Ce qui signifie que l'échantillon a une expérience moyenne en gestion de projets.

## 2.2. Traitement et analyse des données :

### 2.2.1. Description des dimensions en fonction des réponses de l'échantillon de l'étude :

La valeur minimale que peut obtenir un item est 1 et la valeur maximale est 5, donc on a 4 degrés de variations sur 5 ce qui donne un pas de 0.8 en commençons par la valeur minimale le « 1 ». Le tableau suivant détermine les moyennes minimales et maximales des intervalles et le poids relatif qui leur correspond :

Tableau 7: L'intervalle des moyennes minimales et maximales

L'intervalle	[1 – 1.8[	[1.8 – 2.6[	[2.6 – 3.4[	[3.4 – 4.2[	[4.2 – 5]
Le poids relatifs	Pas du tout d'accord	Pas d'accord	Neutre	D'accord	Tout à fait d'accord

Source : Tiré des résultats de la recherche

### 2.2.2. Analyse de données

Les tableaux suivants démontrent la description statistique des réponses données par les collaborateurs SGA pour chaque item et dimension du questionnaire et le poids relatifs leurs correspondant :

#### 2.2.2.1. Situation actuelle :

Tableau 8: Description des items de la dimension « La situation actuelle »

	Moyenne	Ecart type	Poids relatif
<b>La situation actuelle</b>	3,1856	0,51284	Neutre
La méthodologie de gestion de projet au sein de la SGA est clairement définie.	3,5294	0,92884	D'accord
La documentation de référence de gestion de projets SGA prend en charge toutes les questions soulevées durant les différentes phases de projet	3,3939	0,99810	Neutre
Les collaborateurs SIOP sont bien formés sur la méthodologie SGA de gestion de projet	3,2424	1,34699	Neutre
Les chefs de projets de SIOP maîtrisent en général les outils de planification de projet	2,8235	1,11384	Neutre
Les équipes projets SIOP manquent de collaboration entre eux et avec les parties prenantes	2,7812	1,06965	Neutre
Les solutions développées et mises en production font recours a plusieurs actions de maintenance afin de corriger les bugs	3,1765	0,99911	Neutre
Beaucoup d'étapes peuvent être éliminées dans le processus de gestion projets SGA	3,3235	0,84282	Neutre
Dans chaque projet de développement, la vision est suffisamment et clairement définie et mise en évidence	3,2059	0,97792	Neutre

Source : Tiré des résultats de la recherche

Le **tableau 8** de données ci-dessus nous montre que le niveau de réponse donnée par les collaborateurs SIOP sur la situation actuelle « la méthode de gestion de projets utilisée par la direction SIOP » était partagée. Autant que la moyenne arithmétique de la dimension situation actuelle est **3.18** et l'écart type **0.51** indiquant que les collaborateurs n'ont pas une opinion claire

sur la méthode de gestion de projets actuelle, relatif entre ces derniers. D'autre part la moyenne arithmétique des items varie entre **2.78** et **3.529** et l'écart type varie entre **0.842** et **1.346** ce qui affirme ce qui précède. Ça nous amène à déduire qu'il existe une méthode de gestion de projets actuellement utilisée par la direction SIOP mais les collaborateurs n'ont pas une opinion claire sur le sujet. Ces résultats signifient qu'ils n'ont pas d'opinion sur la situation actuelle. Cette situation mène à conclure que les collaborateurs sont conscients qu'il existe une méthode mais ils n'arrivent pas à la maîtriser.

### 2.2.2.2. L'homme est la base

Tableau 9: Description des items de la dimension « L'homme est la base »

	Moyenne	Ecart type	Poids relatif
<b>L'homme est la base</b>	3,6864	0,43860	D'accord
Les espaces de travail mis à la disposition équipes de projets sont bien aménagés	4,0882	0,83003	D'accord
Une équipe auto-organisée peut gérer mieux un projet sans besoins d'une structure hiérarchique	3,0882	1,26414	Neutre
Le recours à la validation hiérarchique « Manager direct ou indirect » est indispensable dans la gestion d'un projet	3,8529	1,10460	D'accord
La clé de réussite d'un projet est l'amélioration continue des processus de gestion de projet	4,1471	0,82139	D'accord
Il est préférable de laisser le choix d'outils de développement aux développeurs	3,7647	1,07475	D'accord
L'accent est mis sur les interactions entre membres de l'équipe plus que sur les processus	3,2647	1,02422	Neutre
La meilleur Equipe est une équipe avec des compétences moyenne engagée ensemble sur thématique à la fois	3,1765	0,93649	Neutre
La meilleur Equipe est une équipe avec des compétences variables et hétérogènes	4,1250	0,87067	D'accord

Source : Tiré des résultats de la recherche

Le **tableau 9** de données ci-dessus nous montre que le niveau de réponse donnée par les collaborateurs SIOP sur l'homme est la base « la première valeur d'Agilité » était partagé. La moyenne arithmétique de la dimension « l'homme est la base » est **3.68** et l'écart type de **0.43** indiquent qu'il y a un accord relatif entre les sondés. Aussi la moyenne arithmétique de tous les items varie entre **3,08** et **4,14** avec une variation de **1.125** montre que l'appréciation des items était partagé par l'ensemble des collaborateurs. Cependant si on regarde l'écart type qui varie

entre **0,82** et **1,26**, on comprend qu'il y a une dispersion des appréciations de l'échantillon d'étude. Le résultat signifie qu'il y a un accord que **l'homme est la base** dans un projet.

### 2.2.2.3. Un logiciel opérationnel

Tableau 10: Description des items de la dimension « Un logiciel opérationnel »

	Moyenne	Ecart type	Poids relatif
<b>Un logiciel Opérationnel</b>	3,2889	0,46248	Neutre
Le plus important est de livrer un logiciel (un produit) qui fonctionne peu importe la méthode et l'organisation	2,1176	1,06642	Plutôt pas d'accord
Il est préférable de diviser et livrer un produit (logiciel) en lots.	3,6471	0,88360	D'accord
Un développement incrémental peut améliorer la qualité de livrable et réussir le projet	3,8485	0,61853	D'accord
Les livraisons fréquentes à intervalle temporelle bien défini assurent une meilleure gestion du projet et une meilleure qualité du livrable	4,0588	0,54723	D'accord
Il est préférable de produire la documentation au fur et à mesure tout au long du projet.	4,3030	0,63663	Tout à fait d'accord
Un projet peut être lancé même sans qu'il y ait une vision claire	2,3438	1,26004	Plutôt pas d'accord
Un projet peut être lancé même avec le minimum d'information possible	2,7273	1,25680	Neutre

Source : Tiré des résultats de la recherche

Le **tableau 10** de données ci-dessus nous montre que le niveau de réponse donnée par les collaborateurs SIOP sur la dimension « **Un logiciel Opérationnel** », était varié entre Tout à fait d'accord, d'accord, Neutre, plutôt pas d'accord mais la réponse finale est penché vers Neutre avec une moyenne arithmétique de **3.328** et un l'écart type de **0.462**. A propos des items la moyenne arithmétique varie entre **2.11** et **4.34**, et l'écart type varie entre **0.54** et **1.26** ce qui nous mène à conclure qu'il y a une dispersion des opinions de l'échantillon de l'étude. Le résultat signifie qu'il n'y a pas une opinion claire à propos de la dimension « logiciel Opérationnel ».

#### 2.2.2.4. La collaboration avec le client

Tableau 11: Description des items de la dimension « La collaboration avec le client »

	Moyenne	Ecart type	Poids relatif
<b>La collaboration avec le client</b>	4,0833	0,46631	D'accord
Le développement dans un projet est un travail collaboratif avec le client plus qu'un contrat négocié	3,9091	0,91391	D'accord
Le feedback de l'utilisateur est très utile au cours de développement	4,1818	0,95048	D'accord
La réalisation d'un projet nécessite un cadrage en temps réel avec les « stakeholders » les parties prenantes	3,8182	0,72692	D'accord
L'implication du client/métier est nécessaire tout au long de projet	4,4242	0,79177	Tout à fait d'accord

Source : Tiré des résultats de la recherche

Le **tableau 11** de données ci-dessus nous montre que le niveau de réponse donnée par les collaborateurs SIOP sur la dimension « **La collaboration avec le client** », était un accord avec une moyenne arithmétique de **4.083** et un l'écart type de **0.466**. A propos des items la moyenne arithmétique varie entre **3.818** et **4.42**, et l'écart type varie entre **0.726** et **0.95** ce qui signifie que les répondants étaient d'accord sur tous les items. Le résultat signifie que les répondants sont d'accord sur la dimension « **La collaboration avec le client** ».

#### 2.2.2.5. L'adaptation au changement

Tableau 12: Description des items de la dimension « L'adaptation au changement »

	Moyenne	Ecart type	Poids relatif
<b>L'adaptation au changement</b>	3,8788	0,59359	D'accord
Accepter le changement du besoin au cours de développement dans un projet pourrait améliorer le résultat final	3,9697	1,18545	D'accord
Les processus de gestion de projet doivent faire l'objet d'une révision périodique et adaptation avec le contexte	4,2121	0,85723	Tout à fait d'accord
Un rythme soutenable est nécessaire tout au long du projet avec adaptation aux nouveaux changements	4,0303	0,84723	D'accord
Si on est pressé par les délais on doit doubler nos efforts afin de terminer à temps	3,5455	1,12057	D'accord
Le délai ne doit pas être fixé au début du projet « on doit planifier chaque lot »	3,6364	1,22010	D'accord

Source : Tiré des résultats de la recherche

Le **tableau 12** de données ci-dessus nous montre que le niveau de réponse donnée par les collaborateurs SIOP sur la dimension « **L'adaptation au changement** », était un accord avec

une moyenne arithmétique de **3.878** et un l'écart type de **0.593**. A propos des items la moyenne arithmétique varie entre **3.545** et **4.212**, et l'écart type varie entre **0.847** et **1.220** ce qui signifie que les répondants étaient d'accord sur tous les items. Le résultat signifie que les répondants sont d'accord sur la dimension « **L'adaptation au changement** ».

### 2.2.2.6. Les critiques

Tableau 13: Description des items de la dimension « Les critiques »

	Moyenne	Ecart type	Poids relatif
Les critiques	3,7955	0,58113	D'accord
Ce n'est pas évident de pouvoir toujours diviser un projet en lots	3,6667	1,05079	D'accord
L'incertitude du besoin exacte du client peut mener à un conflit	4,1563	0,91966	D'accord
Le changement fréquent du besoin « des fonctionnalités » par le client peut provoquer des retards	4,5758	0,61392	Tout à fait d'accord
Les réunions quotidiennes sont une perte du temps	2,7879	1,16613	Neutre

Source : Tiré des résultats de la recherche

Le **tableau 13** de données ci-dessus nous montre que le niveau de réponse donnée par les collaborateurs SIOP sur la dimension « **les critiques** », était un accord avec une moyenne arithmétique de **3.795** et un l'écart type de **0.581**. A propos des items la moyenne arithmétique varie entre **2.787** et **4.575**, et l'écart type varie entre **0.613** et **1.166** ce qui signifie que les répondants étaient d'accord la majorité des items. Le résultat signifie que les répondants sont d'accord sur la dimension « **les critiques** » ce qui cela veut dire qu'il faut prendre en considération ces critiques si non au démarrage de l'application de l'Agilité il y aurait une forte résistance au changement.

### 2.2.2.7. L'utilisation d'une méthode Agile

Tableau 14: Description des items de la dimension « L'utilisation d'une méthode Agile »

	Moyenne	Ecart type	Poids relatif
L'utilisation d'une méthode Agile est une solution pour gérer mieux les projets de développement SIOP	3,9394	0,86384	D'accord

Source : Tiré des résultats de la recherche

Le **tableau 14** de données ci-dessus montre que le niveau de réponse donnée par les répondants sur la citation « L'utilisation d'une méthode Agile est une solution pour gérer mieux les projets de développement SIOP » est un accord avec une moyenne arithmétique de **3.939** et un l'écart

type de **0.863**. Le résultat signifie que les répondants sont d'accord sur le fait qu'utiliser une méthode Agile peut améliorer la gestion de projets au niveau de la direction SIOP.

### 2.2.2.8. Les 4 valeurs de l'agilité

Tableau 15: Description des dimensions de l'Agilité « La moyenne globale de l'Agilité »

	Moyenne	Ecart type	Poids relatif
Moyenne globale Agilité	3,7236	0,34926	D'accord
L'homme est la base	3,6864	0,43860	D'accord
Un logiciel Opérationnel	3,2889	0,46248	Neutre
La collaboration avec le client	4,0833	0,46631	D'accord
L'adaptation au changement	3,8788	0,59359	D'accord

Source : Tiré des résultats de la recherche

Le **tableau 15** de données ci-dessus nous montre que le niveau de réponse donnée par les collaborateurs SIOP sur les quatre dimensions relatives aux variables de l'Agilité était un accord avec une moyenne arithmétique de **3.723** et un l'écart type de **0.349**. A propos des dimensions la moyenne arithmétique varie entre **3.288** et **4.083**, et l'écart type varie entre **0.438** et **0.593** ce qui signifie que les répondants étaient d'accord trois des quatre valeurs de l'Agilité sauf celui de « **Logiciel Opérationnel** ». Le résultat signifie que les répondants sont d'accord sur l'ensemble des valeurs et principes Agiles.

### 2.2.3. Test de normalité

L'étude comprend six hypothèses avant de vérifier celles-ci, un test des variables a été mené afin de voir si les résultats répondent à la distribution normale ou pas, pour pouvoir choisir le test approprié paramétrique ou non paramétrique.

Tableau 16: Tests de normalité d'une distribution

	Tests de normalité					
	Kolmogorov-Smirnov			Shapiro-Wilk		
	Z	DDL	Sig.	W	DDL	Sig.
La situation actuelle	0,146	33	0,070	0,971	33	0,510
L'homme est la base	0,122	33	0,200	0,968	33	0,416
Un logiciel Opérationnel	0,121	33	0,200	0,963	33	0,310
La collaboration avec le client	0,207	33	0,001	0,934	33	0,045
L'adaptation au changement	0,187	33	0,005	0,934	33	0,046
Les critiques	0,122	33	0,200	0,958	33	0,222
Moyenne globale des quatre valeurs de l'agilité	0,113	33	0,200	0,985	33	0,919

Source : Tiré des résultats de la recherche

Le **tableau 16** ci-dessus montre le test de normalité de « Kolmogorov-Smirnov et Shapiro-Wilk » une distribution n'est pas normale pour les dimensions « la collaboration avec le client, l'adaptation aux changements » car leurs valeurs significatives asymptotiques (bilatérales) correspondante atteignent dans chaque dimension une valeur inférieur au niveau de signification approuvée de 0, 05, pour le reste la distribution est normale car leurs valeurs significatives asymptotiques (bilatérales) correspondante atteignent dans chaque dimension une valeur supérieur au niveau de signification approuvée de 0, 05 ce qui permet d'utiliser le test de T Student pour la vérification des hypothèses pour les distributions normale et traiter les deux dimensions qui ont une distribution anormale.

Cela signifie qu'on peut choisir le test paramétrique pour les dimensions qui suivent la loi normale et le non paramétrique pour les autres.

## 2.2.4. Test des hypothèses

### 2.2.4.1. Test de l'hypothèse 1 de l'étude

H1 : Le développement de projet au sein de la SGA se base sur l'homme.

Les résultats du test de Student de cette hypothèse se présentent ainsi :

Tableau 17: Test T-Student de l'hypothèse 1

L'homme est la base	Valeur de test = 3			
	T	DDL	Sig. (Bilatéral)	Différence moyenne
	9,126	33	0,000	0,68645

Source : Tiré des résultats de la recherche

Le **tableau 17** ci-dessus montre que la valeur de T Student est 9.126 et que le niveau de signification est de 0.000 soit inférieur au seuil de signification de 5% donc on accepte l'hypothèse alternative, qui consiste que la moyenne arithmétique des réponses soit différente de 3 avec une différence moyenne de 0.686. Le résultat vient pour confirmer les résultats exprimés précédemment dans le tableau relatif à la description des items de la dimension « l'homme est la base » qui est égale à 3,6758 et le poids relatif et « d'accord ». Ce qui confirme que la méthode de gestion de projets de SGA se base sur l'homme. Donc nous considérons que l'hypothèse H1 est confirmée.

### 2.2.4.2. Test de l'hypothèse 2 de l'étude

H2 : La méthodologie de développement de projets au sein de la SGA vise à produire des logiciels opérationnels de qualité.

Les résultats du test de Student de cette hypothèse se présentent ainsi :

Tableau 18: Test T-Student de l'hypothèse 2

Un logiciel Opérationnel	Valeur de test = 3			
	T	DDL	Sig. (Bilatéral)	Différence moyenne
	3,642	33	0,001	0,28887

Source : Tiré des résultats de la recherche

Le **tableau 18** ci-dessus montre que la valeur de T Student est 3.642 et que le niveau de signification est de 0.001 soit inférieur au seuil de signification de 5% donc on accepte l'hypothèse alternatif, qui consiste que la moyenne arithmétique des réponses soit différente de 3 avec une différence moyenne de 0.2888. Le résultat vient pour confirmer les résultats exprimés précédemment dans le tableau relatif à la description des items de la dimension « **Un logiciel Opérationnel** » qui est égale à 3,288 et le poids relatif et « Neutre ». Ce qui confirme que la méthode de gestion de projets vise à produire un logiciel opérationnel de qualité. Donc nous considérons que l'hypothèse H2 est partiellement confirmée.

### 2.2.4.3. Test l'hypothèse 3 de l'étude

H3 : La collaboration totale avec le client est une condition nécessaire pour la réussite de projets informatiques.

La dimension collaboration avec le client ne suis pas une loi normale ce qui nous mène à appuyer sur les résultats de la moyenne arithmétique des items de cette dimension égale à 4.083 et l'écart type est de 0.4663, de ces résultats nous jugeons que l'hypothèse H3 est confirmée.

### 2.2.4.4. Test l'hypothèse 4 de l'étude

H4 : Le changement du besoin du client au cours du développement est accepté dans les projets SGA.

L'adaptats au changement ne suis pas une loi normale ce qui nous mène à appuyer sur les résultats de la moyenne arithmétique des items de cette dimension égale à 3.8788 et l'écart type est de 0.59359, de ces résultats nous jugeons que l'hypothèse H4 est confirmée.

### 2.2.4.5. Test l'hypothèse 5 de l'étude

H5 : Les critiques de l'Agilité sont un obstacle de l'adoption de cette dernière au sein de la SGA.

Les résultats du test de Student de cette hypothèse se présentent ainsi :

Tableau 19: Test T-Student de l'hypothèse 5

Les critiques	Valeur de test = 3			
	T	DDL	Sig. (Bilatéral)	Différence moyenne
	7,863	32	0,000	0,79545

Source : Tiré des résultats de la recherche

Le **tableau 19** ci-dessus montre que la valeur de T Student est 7.863 et que le niveau de signification est de 0.000 soit inférieur au seuil de signification de 5% donc on accepte l'hypothèse alternatif, qui consiste que la moyenne arithmétique des réponses soit différente de 3 avec une différence moyenne de 0.795. Le résultat vient pour confirmer les résultats exprimés précédemment dans le tableau relatif à la description des items de la dimension « les critiques » qui est égale à 3,79 et le poids relatif et un « accord ». Ce qui confirme que les collaborateurs SGA sont d'accord sur les critiques de l'agilité. Donc nous considérons que l'hypothèse H5 est valide.

### 2.2.4.6. Test l'hypothèse 6 de l'étude

H6 : Les collaborateurs SGA utilisent certains pratiques de l'agilité.

Les résultats du test de Student de cette hypothèse se présentent ainsi :

Tableau 20: Test T-Student de l'hypothèse 6

Moyenne globale des quatre dimensions de l'agilité	Valeur de test = 3			
	T	DDL	Sig. (Bilatéral)	Différence moyenne
	12,081	33	0,000	0,72361

Source : Tiré des résultats de la recherche

Le **tableau 20** ci-dessus montre que la valeur de T Student est 12.081 et que le niveau de signification est de 0.000 soit inférieur au seuil de signification de 5% donc on accepte l'hypothèse alternatif, qui consiste que la moyenne arithmétique des réponses soit différente de

3 avec une différence moyenne de 0.7236. Le résultat vient pour confirmer les résultats exprimés précédemment dans le tableau relatif à la description des items de la dimension « la moyenne globale de l'Agilité » qui est égale à 3,723 et le poids relatif et un « accord ». Ce qui confirme que les collaborateurs SGA utilisent certaines pratiques de l'agilité. Donc nous considérons que l'hypothèse H6 est confirmée.

#### 2.2.4.7. Vérification de l'impact de l'expérience sur les différentes dimensions

Tableau : le **tableau 21** nous montre le teste de corrélation de Spearman entre les trois variables de l'expérience « expérience professionnelle, expérience au sein de la SGA, expérience en gestion de projets ».

Tableau 21: Le test de corrélation de Spearman entre les trois items d'expérience et les différentes dimensions

Corrélation Spearman								
		La situation actuelle	L'homme est la base	Un logiciel Opérationnel	Collaboration avec le client	L'adaptation au changement	Critiques	Moyenne globale
Expérience en gestion de projets	Coef de corrélation	0,088	0,093	0,317	0,610*	0,388*	0,117	0,452*
	Sig. (bilat)	0,626	0,606	0,072	0,000*	0,028*	0,522	0,008*
	N	33	33	33	32	32	32	33
Expérience professionnelle	Coef de corrélation	-0,091	0,078	0,150	0,195	0,308	0,243	0,286
	Sig. (bilat)	0,610	0,660	0,399	0,276	0,082	0,174	0,101
	N	34	34	34	33	33	33	34
Expérience au sein de la SGA	Coef de corrélation	-0,073	0,148	0,167	0,104	0,151	0,180	0,195
	Sig. (bilat)	0,692	0,417	0,360	0,576	0,416	0,333	0,285
	N	32	32	32	31	31	31	32

Source : Tiré des résultats de la recherche

Le tableau ci-dessus montre les observations suivantes :

1. Il y a une forte corrélation entre l'expérience en gestion de projets et la dimension la collaboration avec le client avec un coefficient de 0.61, et le niveau de signification (bilatérales) est de 0.00 implique qu'il existe une relation entre la variable expérience en gestion de projets et la dimension « la collaboration avec le client ». Ce qui implique que la variable « expérience en gestion de projets » à un impact sur la dimension « la collaboration avec le client »

2. Il y a une forte corrélation entre l'expérience en gestion de projets et la dimension la « L'adaptation au changement » avec un coefficient de 0.388, et le niveau de signification (bilatérales) est de 0.028 implique qu'il existe une relation entre la variable expérience en gestion de projets et la dimension « L'adaptation au changement ». Ce qui implique que la variable « expérience en gestion de projets » à un impact sur la dimension « L'adaptation au changement »
3. Il y a une forte corrélation entre l'expérience en gestion de projets et la dimension la « Moyenne globale de l'Agilité » avec un coefficient de 0.452, et le niveau de signification (bilatérales) est de 0.008 implique qu'il existe une relation entre la variable expérience en gestion de projets et la dimension « Moyenne globale de l'Agilité ». Ce qui implique que la variable « Moyenne globale de l'Agilité » à un impact sur la dimension « la collaboration avec le client ».
4. Il y a une corrélation moyenne entre l'expérience en gestion de projets et la dimension la « Un logiciel Opérationnel » avec un coefficient de 0.317, et le niveau de signification (bilatérales) est de 0.072 supérieur à 0.05 implique qu'il n'existe pas une relation entre la variable expérience en gestion de projets et la dimension « Un logiciel Opérationnel ».
5. La corrélation est faible entre l'expérience en gestion de projets et les autres dimensions. Et il n'existe pas une liaison entre eux l'expérience en gestion de projets et ces dimensions.
6. La corrélation est faible entre l'expérience professionnelle et les dimensions « La situation actuelle, L'homme est la base, Un logiciel Opérationnel et Collaboration avec le client ». Et il n'existe pas une relation entre l'expérience professionnelle et ces dimensions.
7. Il y a une corrélation moyenne entre l'expérience professionnelle et les dimensions « L'adaptation au changement, Critiques et la Moyenne globale de l'Agilité ». Et il n'existe pas une relation entre la variable expérience professionnelle ces dimensions.
8. La corrélation est faible entre la variable expérience au sein de la SGA et toutes les dimensions. Et il n'existe pas une relation entre la variable te ces dimensions.

D'après ces observations nous pouvons dire que l'expérience en gestion de projets à un impact sur l'utilisation des pratiques de l'Agilité pour les deux expériences « expérience au sein de la SGA et l'expérience professionnelle » il n'y a pas un impact.

### 2.3. Fiabilité et validité de l'étude

Pour s'assurer de la validité de construit du questionnaire les coefficients de corrélation des rangs de Spearman ont été calculé entre le score moyen de chaque dimension et le score moyen total du questionnaire, Comme le montre le tableau 22 :

Tableau 22: Le niveau d'interdépendance entre l'expérience et les dimensions du questionnaire

La dimension	Coefficient de corrélation de sperman	Le score moyen de tout le questionnaire
La situation actuelle	Coefficient de corrélation	0,626
	Sig. (Bilatéral)	0,000
	N	34
L'homme est la base	Coefficient de corrélation	0,595
	Sig. (Bilatéral)	0,000
	N	34
Un logiciel Opérationnel	Coefficient de corrélation	0,541
	Sig. (Bilatéral)	0,001
	N	34
La collaboration avec le client	Coefficient de corrélation	0,571
	Sig. (Bilatéral)	0,001
	N	33
L'adaptation au changement	Coefficient de corrélation	0,767
	Sig. (Bilatéral)	0,000
	N	33
Les critiques	Coefficient de corrélation	0,222
	Sig. (Bilatéral)	0,213
	N	33
Moyenne globale de l'Agilité	Coefficient de corrélation	0,894
	Sig. (Bilatéral)	0,000
	N	34

Source : Tiré des résultats de la recherche

Le **tableau 22** ci-dessus montre que tous les niveaux significatifs sont inférieurs à 5%, ce qui indiquerait une corrélation positive entre eux, le calcul des Coefficient de corrélation de Spearman a permis de vérifier l'interdépendance entre les dimensions au niveau de corrélation interne, leurs valeurs se situant entre (0.222 et 0.894) ce qui traduit une forte corrélation inter dimensions positives. Pour la validité structurale du questionnaire, nous avons trouvé un bon niveau d'interdépendance entre les dimensions du questionnaire. Ce qui témoigne d'une bonne validité structurale de l'outil.

### 2.4. Discussion des résultats

Interprétation des résultats :

Il faut mentionner que l'objectif de notre étude est l'évaluation et valoriser les pratiques Agiles utilisés par les équipes projet de la banque SGA.

#### **2.4.1. La situation actuelle**

Suite à la demande de la direction nous avons intégré cette partie afin de voir s'il existe une méthode de gestion de projet utilisée par les équipes projet. Les résultats ont montré qu'il n'y a pas une vision claire sur la méthode de gestion de projet SGA. L'ensemble des répondants n'ont pas d'opinion sur l'utilisation de cette méthode.

D'après notre observation et les documents fournis par notre tuteur, la SGA a mis en place une méthode de gestion de projets avec un processus détaillé avec une documentation accessible, et les équipes projets ne la consultent pas.

Notre observation sur le sujet nous confirme qu'il y a une méthode de gestion de projets mis en place par SGA.

#### **2.4.2. L'homme est la base**

L'homme est la base est la première valeur de l'Agilité qui stipule que « Les individus et leurs interactions plus que les processus et les outils ». D'après les analyses descriptives de cette dimension et après confirmation de l'hypothèse H2, on peut déduire que les équipes projets ont une vision positive sur cette dimension ce qui implique que les collaborateurs pratiquent certains principes de la première valeur de l'agilité. Ce résultat nous permet de dire qu'il ne faut pas un effort important pour maîtriser correctement la première valeur de l'agilité.

#### **2.4.3. Un logiciel opérationnel**

Cette dimension est la deuxième valeur Agile qui stipule « Des logiciels opérationnels plus qu'une documentation exhaustive ». Le résultat de cette dimension nous montre que l'ensemble des répondants n'ont pas une opinion claire sur le sujet. Ce qui signifie qu'il n'y a pas une réponse claire sur l'objectif de lancer un projet, et le plus important est de suivre les règles plutôt que livrer un bon produit et l'hypothèse H3 viens pour appuyer le résultat. Cette valeur d'Agilité nécessite un effort important afin que l'organisation maîtrise cette dimension.

#### **2.4.4. La collaboration avec le client**

La collaboration avec le client stipule « Collaboration avec les clients plus que la négociation contractuelle ». Les résultats de cette dimension nous montrent que les équipes projet maîtrise cette valeur de l'agilité, il faut juste une valorisation de leurs pratiques. En plus nous

considérons que cette dimension est acquise au sein de l'organisation. Plus de maîtrise va être au fur et à mesure après l'adoption d'une méthode Agile.

Notre observation sur le sujet nous confirme qu'il y a une implication du client au cours du projet.

#### **2.4.5. L'adaptation au changement**

L'adaptation au changement stipule « L'adaptation au changement plus que le suivi d'un plan ». Les résultats de cette dimension nous montrent que les équipes projet maîtrisent cette valeur de l'agilité, il faut juste une valorisation de leurs pratiques. En plus nous considérons que cette dimension est acquise au sein de l'organisation. Plus de maîtrise va être au fur et à mesure après l'adoption d'une méthode Agile.

Notre observation sur le sujet nous confirme qu'il y a une adaptation au changement au cours du projet.

#### **2.4.6. Les critiques**

Les résultats de cette dimension nous montrent que les collaborateurs sont d'accord sur les critiques fréquentes de l'agilité. Ces critiques sont des observations et des constats des praticiens de certaines méthodes Agiles. Ce qui signifie qu'il faut prendre en considération ces critiques avant de se lancer dans l'adoption de l'agilité, si non le projet de l'adoption va échouer. Et surtout qu'il faut du temps avant que l'organisation maîtrise une méthode agile.

#### **2.4.7. Les quatre valeurs de l'agilité**

C'est une dimension créée par le calcul de la moyenne des quatre dimensions des valeurs de l'agilité. Le résultat de cette dimension nous montre que l'organisation utilise bel et bien les pratiques Agiles. Cela est confirmé par l'hypothèse H6. Et comme nous avons vu dans le chapitre 2, que les principes de l'Agilité sont inspirés des pratiques du bon sens, alors nous concluons que l'organisation a beaucoup de points positifs et elle est à mi-chemin de l'agilité. La suite est facile à implémenter.

Nos observations initiales nous confirment qu'il y a une implication du client au cours du projet.

#### **2.4.8. L'utilisation d'une méthode Agile**

C'est une variable que nous avons mis dans le questionnaire afin de tester la volonté des collaborateurs d'utiliser une méthode Agile pur et dur, le résultat nous montre qu'il y a un accord sur le sujet, ce qui signifie que les équipes projet veulent pratiquer un des méthodes Agiles.

#### **2.5. Validation des hypothèses**

D'après les résultats obtenus, l'hypothèse générale de notre recherche qui suppose que « La méthode de gestion de projet SGA inclue des pratiques de l'Agilité » est validé. En vue que la moyenne globale du questionnaire obtenue est de 3.58 qui a un poids relatif a « d'accord ».

L'hypothèse H1 : Le développement de projet au sein de la SGA se base sur l'homme est confirmée.

L'hypothèse H2 : La méthodologie de développement de projets au sein de la SGA vise à produire des logiciels opérationnels de qualité est confirmée.

L'hypothèse H3 : La collaboration totale avec le client est une condition nécessaire pour a réussite de projets informatiques est confirmée.

L'hypothèse H4 : Le changement du besoin du client au cours du développement est accepté dans les projets SGA est confirmée.

L'hypothèse H5 : Les critiques de l'agilités sont un obstacle de l'adoption de cette dernière au sein de la SGA est confirmée.

L'hypothèse H6 : Les collaborateurs SGA utilisent certains pratiques de l'Agilité est confirmée.

# **Conclusion**

### Conclusion

Les méthodes Agiles sont des méthodes révolutionnaires dans la gestion de projets informatique, les résultats de « chaos report 2015 et (12th Annual State of Agile Report, 2018) » ont montrés que le taux de réussite des projets conduits avec l'Agilité est très élevé par rapport aux méthodes classiques. En plus de très grands projets informatiques ont connus un succès en utilisant une méthode agile, à titre d'exemple le projet de la voiture autonome de google conduit par la méthode SCRUM, en plus toutes les grandes firmes de la technologie « Google, Facebook, Microsoft » utilisent l'Agilité dans la gestion de leurs projets.

D'autre part certaines organisations, et généralement dans les multinationales ou les processus sont bien conçus, il est qu'on trouve certains pratiques d'Agilité utilisés généralement par les personnes les plus expérimentés. De cette raison dans notre travail nous avons supposés que c'est le cas dans la banque Société Générale Algérie SGA et nous voulons évaluer les pratiques des équipes projet.

Pour y parvenir nous avons commencé par une partie théorique qui est débuté par une revue de littérature qui permet de cadrer le travail à travers les études antérieurs et relatives au sujet de l'agilité, ensuite nous avons présenté tous les concepts utilisés dans notre travail, puis nous avons présentés la méthodologie de la recherche utilisée et le contexte de l'organisme d'accueil. Dans la seconde partie dédiée à la pratique nous avons présenté les résultats de notre recherche, l'analyse et le traitement statistique de ces résultats et nous avons effectué une discussion de ces résultats.

L'objectif principale du présent travail était l'évaluation de la prédisposition de la mise en place l'agilité au sein de la SGA.

Les données ont été recueillies à l'aide d'un questionnaire en ligne ou le lien est communiqué par mail, après l'analyse des résultats par le traitement des données, les statistiques descriptives, et la vérification des hypothèses par T-moyen, nous avons obtenus les résultats suivants :

La méthode de gestion de projets de la banque SGA inclue des pratiques Agiles, et les collaborateurs utilisent les pratiques de l'Agilité involontairement. Aussi les quatre valeurs de l'Agilité sont respectées au sein de la banque, et le taux de l'Agilité au sein de la banque SGA est moyennement accepté. Et la SGA est prédisposé à adopter l'agilité.

Pour les critiques des méthodes Agiles les collaborateurs sont d'accord que ces critiques sont des points négatifs qu'il faut prendre en considération.

## Conclusion Générale

Pour adopter l'Agilité, la banque SGA n'a pas besoin de dépenser un grand effort, et elle est à mi-chemin, il faut valoriser les pratiques acquises, mettre en place d'un système de motivation efficace et faire des formations sur la méthode Agile adoptée, bien communiquer cette méthode.

En d'autres termes, la transition vers l'agilité au sein de la SGA nécessite :

- Plus de communication sur la méthode de gestion de projets SGA va permettre aux collaborateurs mieux comprendre cette méthode.
- Lancer des sessions de formation sur le processus de gestion de projets est recommandé afin de maîtriser la méthode gestion de projets SGA.
- Il est préférable de prendre en considération les recommandations des collaborateurs sur la modification et l'adaptation du processus de gestion de projets.
- Encourager l'esprit d'équipe et renforcer la collaboration entre les équipes projet et les parties prenantes pourrait donner plus d'implication des équipes et les parties prenantes.
- Donner plus d'autonomie aux équipes projet est recommandé.
- Il est recommandé de se baser sur la communication entre les équipes projet en premier lieu, et après on se base sur le suivi du processus.
- Il est préférable que l'équipe projet se concentre sur une seule thématique à la fois afin de produire un résultat de qualité et dans les délais.
- Il est recommandé de communiquer l'objectif et le but de lancer un projet.
- Nous recommandons Plus de souplesse sur les exigences de détails excessives d'avant-projet, expression trop détaillée, et cahier de charges avec le moindre détail.
- L'engagement d'un expert en Agilité afin de gagner du temps et éviter les problèmes fréquemment rencontrés par d'autres organisations est souhaitable.
- Il faut du temps pour maîtriser l'agilité, c'est considéré comme un investissement à moyen terme. Pour cette raison il est nécessaire de bien choisir la méthode adéquate.
- Bien sélectionner des équipes ayant une certaine complicité et compétence, va améliorer la prédisposition de l'agilité.
- Surtout l'implication du métier/client dans les formations afin qu'il comprenne les enjeux et éviter d'utiliser la valeur « accepter le changement » pour mener le projet à l'échec.

### Les perspectives de recherche :

Dans ce présent travail nous avons réalisé une évaluation des pratiques et les prédispositions d'Agilité au sein de l'organisme d'accueil qui rentre dans le cadre de préparation de l'adoption

## Conclusion Générale

de l'agilité, on voulait faire une étude sur la meilleur méthode Agile que la banque SGA pourrait adopter, aussi il y a une tendance vers les méthodes Agiles Hybrides au lieu de se baser sur une seule méthode « des travaux très intéressants de (Vickoff J. P., 2016) ».

D'autre part la généralisation de l'Agilité pour d'autre domaine tel que le marketing, on parle aussi du Management Agile sont des sujets très intéressants dans le management du futur.

## Bibliographies

- 12th Annual State of Agile Report. (2018, avril 9). Récupéré sur CollabNet VersionOne: <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report>
- AFITEP. (2010). *Dictionnaire de management de projet* (éd. 4). Paris: AFNOR.
- Anderson, D., & Carmichael, A. (2016). *Essential Kanban Condensed*. LeanKanban.
- Beck, K., & Et, A. (2001, septembre). *agilemanifesto*. Récupéré sur [agilemanifesto.org](http://agilemanifesto.org/): <http://agilemanifesto.org/>
- Bénard, J.-L. (2005). *Gestion de projet : EXtreme Programming*. Paris: Eyrolles.
- BERKANI, A. (2017). Rôle des centres dédiés à la diffusion des méthodes Agiles : le cas d'une organisation complexe. *Paris Dauphine University*.
- Berry, G. (2017). *L'Hyperpuissance de l'informatique Algorithmes, données, machines, réseaux*. Paris: Odile Jacob.
- BONACCORSI, A. (2019, 03 20). *4356--gestion-projet-methodes-predictives*. Récupéré sur [www.supinfo.com](http://www.supinfo.com): <https://www.supinfo.com/articles/single/4356--gestion-projet-methodes-predictives>
- Esfeld, M. (2009). *Philosophie des sciences : Une introduction*. Lausanne: PPUR Presses polytechniques et universitaires romandes.
- Gagnon, Y.-C. (2012). *L'étude de cas comme méthode de recherche*. Québec: Presses de l'Université du Québec.
- Hunt., A. (2015, Mai 6). *The Failure of Agile*. Récupéré sur [toolshed](https://toolshed.com/2015/05/the-failure-of-agile.html): <https://toolshed.com/2015/05/the-failure-of-agile.html>
- Jeffries, R. (2018, Mai 10). *Developers Should Abandon Agile*. Récupéré sur [ronjeffries.com](https://ronjeffries.com/articles/018-01ff/abandon-1/): <https://ronjeffries.com/articles/018-01ff/abandon-1/>
- KHALIL, C. (2011). *Les méthodes "Agiles" de management de projets informatiques : une analyse "par la pratique"*. paris: Télécom ParisTech.
- Meijer, E. (2014). One Hacker Way Rational alternative of Agile Dev Day. Récupéré sur <https://vimeo.com/110554082>
- Messenger, V., & Tabaka, J. (2013). *Gestion de projet Agile : avec Scrum, Lean, eXtreme Programming*. Paris: Eyrolles.
- Morisseau, L. (2014). *Kanban pour l'IT Une nouvelle méthode pour améliorer les processus de développement* (éd. 2). Paris: Dunod.
- MOUKALA, N. (2011). *Introduction à l'épistémologie des Sciences Humaines et Sociales*. Saint-Denis: EDILIVRE APARIS.
- PMBOK. (2017). *Guide Du Corpus Des Connaissances En Management De Projet: Guide Pmbok* (éd. 6). Project Management Institute PMI.
- Schwaber, K., & Sutherland, J. (2017). *Le Guide de Référence de Scrum: Les Règles de Jeu*.

## Bibliographies

- SPANNEUT, H. (2015). *Projets informatiques, S'approprier le Guide PMBOC pour réussir sa gestion de projet*. Saint-Herblain: Editions ENI.
- Thomas, S. (2012, 12 03). *Revisiting the Iterative Incremental Mona Lisa*. Récupéré sur itsadeliverything.com: <http://itsadeliverything.com/revisiting-the-iterative-incremental-mona-lisa>
- Vance, A. (2016). *Elon Musk: Tesla, Paypal, SpaceX : l'entrepreneur qui va changer le monde*. Paris: Eyrolles.
- Vickoff, J. P. (2016). *Agile Scrum et au-delà - Organisation - Management, Pilotage des Projets*. Paris: Bookelis.
- Vickoff, J.-P. (2008). *PUMA Essentiel Agilité globale et Système d'information*. Paris: Entreprise-Agile.
- Vickoff, J.-P. (2008). *Puma et les méthodes Agiles*. Paris: Agile-Entreprise.
- Vickoff, J.-P. (2013). *PUMA Essentiel Méthode Agile de 3ème Génération*. Paris: Agile-Entreprise.
- Zapata, A. (2004). *L'épistémologie des pratiques : pour l'unité des savoirs*. Paris: L'Harmattan .

## **ANNEXE A : Questionnaire**

Madame/Monsieur

Cette enquête rentre dans le cadre de la préparation d'un mémoire de fin de cycle de Master en Management à l'Ecole Nationale supérieure de Management (Kolea) [ensm.dz](http://ensm.dz), intitulé :

**L'approche Agile en gestion des projets informatiques : évaluation de la prédisposition de la mise en place de l'Agilité.**

Votre contribution constitue un apport important dans l'amélioration de la qualité de ce dernier.

Merci de bien vouloir prendre quelques minutes de votre précieux temps pour renseigner soigneusement le questionnaire ci-dessus qui prendra entre 10 et 15 minutes. En donnant votre appréciation sur les différentes expressions.

**Remarque :**

*Le sondage est anonyme, et les résultats seront exploités dans un cadre purement académique*

**Etudiant:**

Morsli Mohamed Amine

ENSM - Kolea – Tipaza

**Encadreur**

Pr. Abdelmalik Mezhouda

ENSM - Kolea – Tipaza

**Partie 1 : Informations générales** (merci de cocher la case convenable)

**1. Niveau d'étude**

Diplôme professionnel	Licence	Poste Graduation
-----------------------	---------	------------------

**2. Spécialité/domaine**

Science Technique	Science humaine et sociale
-------------------	----------------------------

**3. Expérience professionnelle**

<3 ans	De 3 à 5 ans	De 5 à 10 ans	> 10 ans
--------	--------------	---------------	----------

**4. Expérience en gestion de projets**

<3 ans	De 3 à 5 ans	De 5 à 10 ans	> 10 ans
--------	--------------	---------------	----------

**5. Expérience au sein de la SGA**

<3 ans	De 3 à 5 ans	De 5 à 10 ans	> 10 ans
--------	--------------	---------------	----------

**6. Poste occupé**

Développeur	Ingénieur	Administrateur	Chef de projet	Manager	Autre
-------------	-----------	----------------	----------------	---------	-------

**Partie 2 : expressions du questionnaire**

	<b>Expressions</b>	<b>Pas du tout d'accord</b>	<b>Pas d'accord</b>	<b>Neutre</b>	<b>D'accord</b>	<b>Tout à fait D'accord</b>
<b>La situation actuelle</b>						
7.	La méthodologie de gestion de projet au sein de la SGA est clairement définie.					
8.	La documentation de référence de gestion de projets SGA prend en charge toutes les questions soulevées durant les différentes phases de projet					
9.	Les collaborateurs SIOP sont bien formés sur la méthodologie SGA de gestion de projet					
10.	Les chefs de projets de SIOP maîtrisent en général les outils de planification de projet					
11.	Les équipes projets SIOP manquent de collaboration entre eux et avec les parties prenantes					
12.	Les solutions développées et mises en production font recours à plusieurs actions de maintenance afin de corriger les bugs					
13.	Beaucoup d'étapes peuvent être éliminées dans le processus de gestion projets SGA					
14.	Dans chaque projet de développement, la vision est suffisamment et clairement définie et mise en évidence					
<b>Pratiques Agiles</b>						
<b>L'homme est la base</b>						
15.	Les espaces de travail mis à la disposition équipes de projets sont bien aménagés					
16.	Une équipe auto-organisée peut gérer mieux un projet sans besoins d'une structure hiérarchique					
17.	Le recours à la validation hiérarchique « Manager direct ou indirect » est indispensable dans la gestion d'un projet					
18.	La clé de réussite d'un projet est l'amélioration continue des processus de gestion de projet					
19.	Il est préférable de laisser le choix d'outils de développement aux développeurs					
20.	L'accent est mis sur les interactions entre membres de l'équipe plus que sur les processus					
21.	La meilleur Equipe est une équipe avec des compétences moyenne engagée ensemble sur thématique à la fois					
22.	La meilleur Equipe est une équipe avec des compétences variables et hétérogènes					
<b>Un logiciel opérationnel</b>						
23.	Le plus important est de livrer un logiciel (un produit) qui fonctionne peu importe la méthode et l'organisation					

	<b>Expressions</b>	<b>Pas du tout d'accord</b>	<b>Pas D'accord</b>	<b>Neutre</b>	<b>D'accord</b>	<b>Tout à fait D'accord</b>
24.	Il est préférable de diviser et livrer un produit (logiciel) en lots.					
25.	Un développement incrémental peut améliorer la qualité de livrable et réussir le projet					
26.	Les livraisons fréquentes à intervalle temporelle bien défini assurent une meilleure gestion du projet et une meilleure qualité du livrable					
27.	Il est préférable de produire la documentation au fur et à mesure tout au long du projet.					
28.	Un projet peut être lancé même sans qu'il y ait une vision claire					
29.	Un projet peut être lancé même avec le minimum d'information possible					
<b>La collaboration avec le client</b>						
30.	Le développement dans un projet est un travail collaboratif avec le client plus qu'un contrat négocié					
31.	Le feedback de l'utilisateur est très utile au cours de développement					
32.	La réalisation d'un projet nécessite un cadrage en temps réel avec les « stakeholders » les parties prenantes					
33.	L'implication du client/métier est nécessaire tout au long de projet					
<b>L'adaptation au changement</b>						
34.	Accepter le changement du besoin au cours de développement dans un projet pourrait améliorer le résultat final					
35.	Les processus de gestion de projet doivent faire l'objet d'une révision périodique et adaptation avec le contexte					
36.	Un rythme soutenable est nécessaire tout au long du projet avec adaptation aux nouveaux changements					
37.	Si on est pressé par les délais on doit doubler nos efforts afin de terminer à temps					
38.	Le délai ne doit pas être fixé au début du projet « on doit planifier chaque lot					
<b>Les critiques</b>						
39.	L'utilisation d'une méthode Agile est une solution pour gérer mieux les projets de développement SIOP					
40.	Ce n'est pas évident de pouvoir toujours diviser un projet en lots					
41.	L'incertitude du besoin exacte du client peut mener à un conflit					
42.	Le changement fréquent du besoin « des fonctionnalités » par le client peut provoquer des retards					
43.	Les réunions quotidiennes sont une perte de temps					

## **ANNEXE B : Les méthodes Agiles approfondies**

Dans cette partie nous allons décrire en détail les trois méthodes Agiles les plus utilisés dans le monde.

## 1. Les méthodes Agiles

La **figure 17** présentée dans le chapitre 1 nous montre l'utilisation des méthodes Agiles (12th Annual State of Agile Report, 2018)

Les méthodes Agiles sont dévissées en 3 générations

**1ère Génération** : ce sont les méthodes fondatrices conçus dans les années 90 telle que SCRUM, XP, KANBAN...

**2ème Génération** : ce sont les méthodes sorties après la naissance d'Agile manifesto, telle que DEVOPS, LEAN SOFTWARE

**3ème Génération** : ce sont les méthodes conçus après 2010, telle que SAFe 4.0

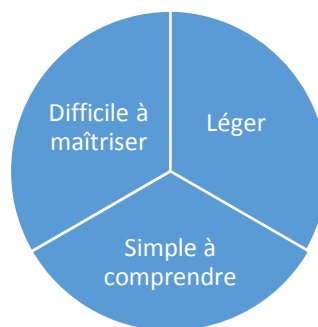
### 1.1. SCRUM

Scrum provient du terme rugbystique « Scrum » (La mêlée) qui est un mécanisme qui permet de redémarrer le jeu après une erreur commise. L'objectif de l'équipe est d'unir ses forces pour mener le ballon dans l'embut. (Schwaber & Sutherland, 2017) **Et** (Vickoff J. P., 2016, p. 61)

C'est un cadre de travail (framework) dans lequel les personnes peuvent aborder des problèmes complexes et adaptatifs tout en livrant de manière efficace et créative des produits de la plus grande valeur possible. (Schwaber & Sutherland, 2017) **Et** (Vickoff J. P., 2016, p. 61)

Sacrum résumé en trois mots dans la **figure 26** :

Figure 26: Les trois piliers de l'agilité



Source : par nos soins

Scrum est un cadre de processus qui a été utilisé pour gérer le travail sur des produits complexes depuis le début des années 1990. C'est plutôt un cadre de travail dans lequel vous pouvez utiliser différents processus et techniques. Scrum met en évidence l'efficacité relative à la gestion de votre produit et aux techniques de travail afin que vous puissiez continuellement améliorer le produit, l'équipe et l'environnement de travail. (Schwaber & Sutherland, 2017)

### **1.1.1. Usage de SCRUM**

Scrum a été initialement développé pour la gestion et le développement de produits. Depuis le début des années 1990, Scrum a été largement utilisé dans le monde entier pour :

- Rechercher et identifier des marchés, des technologies et des caractéristiques produit viables.
- Développer des produits et des améliorations.
- Publier des produits et des améliorations, jusqu'à plusieurs fois par jour.
- Développer et maintenir des environnements Cloud (en ligne, sécurisé, à la demande) et d'autres environnements d'exploitation de produits.
- Maintenir et renouveler des produits.

Scrum a été utilisé pour développer des logiciels, du matériel, des logiciels embarqués, des réseaux de fonctions interactives, des véhicules autonomes, des écoles, des gouvernements, du marketing, de la gestion opérationnelle des organisations et presque tout ce que nous utilisons dans notre vie quotidienne. Plus il y a une augmentation des complexités sur le plan technologique, marché et environnemental ainsi leurs interactions, plus l'utilité de Scrum, pour faire face à cette complexité, est quotidiennement confirmée. Maintenant, Scrum est largement utilisé pour la gestion de l'organisation, de ses produits et ses services. Lorsque les mots « développer » et « développement » sont utilisés dans le Guide Scrum, ils font référence à des travaux complexes, tels que les types identifiés ci-dessus. (Schwaber & Sutherland, 2017)

### **1.1.2. Théorie de SCRUM**

Scrum est fondé sur la théorie du contrôle empirique de processus, ou l'empirisme. L'empirisme affirme que la connaissance provient de l'expérience et la prise de décisions est basée sur des faits connus. Scrum utilise une approche itérative et incrémentale pour optimiser la prédictibilité et le contrôle de risque. (Schwaber & Sutherland, 2017)

Trois piliers soutiennent l'implémentation d'un contrôle empirique de processus : la transparence, l'inspection et l'adaptation.

- **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.
- **Inspection** : Les utilisateurs de Scrum doivent fréquemment inspecter les artefacts Scrum et l'état d'avancement par rapport à un Objectif de Sprint afin de détecter les écarts indésirables. La fréquence de ces inspections ne devrait pas gêner le travail en cours.
- **Adaptation** : Si un problème ou une faiblesse est constaté pendant l'inspection, le processus doit alors être adapté.

Scrum prescrit quatre événements formels d'inspection et d'adaptation, tels que décrit dans la section événements Scrum de ce document :

- Planification du Sprint (Sprint Planning)
- Mêlée Quotidienne (Daily Scrum)
- Revue de sprint (Sprint Review)
- Rétrospective de Sprint (Sprint Retrospective)

### 1.1.3. Valeurs SCRUM

La bonne application de Scrum repose sur des personnes de plus en plus à même de vivre avec ces cinq valeurs. Ces personnes s'engagent, personnellement, à atteindre les objectifs de l'équipe Scrum. Les membres de l'équipe Scrum ont le courage de faire les bonnes choses et de résoudre les problèmes difficiles. Tout le monde se concentre et se focalise sur le travail à faire durant le Sprint et les objectifs de l'équipe Scrum. (Schwaber & Sutherland, 2017)

L'équipe Scrum et ses parties prenantes acceptent d'être ouvertes sur tout le travail et les défis impliqués par l'accomplissement de ce travail.

### 1.1.4. L'équipe SCRUM

Une équipe Scrum comprend un Product Owner, une équipe de développement et un Scrum Master. Les équipes Scrum sont auto-organisées et pluridisciplinaires. Les équipes pluridisciplinaires ont toutes les compétences nécessaires pour effectuer le travail sans dépendre

d'autres personnes n'appartenant pas à l'équipe. L'équipe de Scrum s'est révélée être de plus en plus efficace dans toutes les précédentes applications, ainsi que dans tout autre travail complexe. (Schwaber & Sutherland, 2017)

Les livraisons incrémentales d'un produit « Fini » assurent la disponibilité d'une version potentiellement utile du produit fonctionnel.

- **Le Product Owner**

Ou le propriétaire du produit est le représentant des clients et des utilisateurs. Il est responsable de maximiser la valeur du produit résultant du travail de l'équipe de développement. La façon de jouer ce rôle peut varier grandement selon les organisations, les équipes Scrum et les individus. Il s'agit « d'une personne et non d'un comité ». Il est seul à diriger l'activité de l'équipe de développement à qui il n'est « pas permis de suivre les instructions d'une autre personne ».

Le Product Owner est le seul responsable de la gestion du Backlog Produit (Product Backlog). La gestion du Backlog Produit comprend :

- L'expression claire des éléments du Backlog produit.
- L'ordonnement des éléments dans le Backlog produit pour mieux réaliser les objectifs et les missions.
- L'optimisation de la valeur du travail effectué par l'équipe de développement.
- L'assurance que le Backlog produit est visible, transparent et clair pour tous, et montre sur quoi l'équipe de développement travaillera prochainement.
- L'assurance que l'équipe de développement comprend adéquatement les éléments du Backlog produit.

Le Product Owner peut lui-même accomplir les tâches susmentionnées ou les déléguer à l'équipe de Développement. Afin que le Product Owner réussisse dans sa démarche, toute l'organisation doit respecter ses décisions. Les décisions du Product Owner sont visibles dans le contenu et l'ordonnement du Backlog produit. Nul ne peut forcer l'équipe de développement à travailler à partir d'un autre ensemble d'exigences. (Schwaber & Sutherland, 2017)

- **L'équipe de Développement**

L'équipe de développement se compose de professionnels qui fournissent un incrément « Fini » potentiellement publiable à la fin de chaque Sprint. Un incrément « Fini » est requis à la revue

de sprint. Les équipes de développement sont structurées et habilitées par l'organisation à s'organiser et gérer leur propre travail. La synergie résultante optimise l'efficacité et l'efficacité globale de l'équipe de développement. (Schwaber & Sutherland, 2017)

Les équipes de développement ont les caractéristiques suivantes :

- Elles sont auto-organisées. Nul (pas même le Scrum Master) n'indique à l'équipe de développement comment transformer les éléments du Backlog Produit en incréments de fonctionnalités potentiellement publiables.
- Elles sont pluridisciplinaires, avec toutes les compétences nécessaires, en tant qu'équipe, pour créer un incrément produit.
- Scrum ne reconnaît aucun titre aux membres de l'équipe de développement, indépendamment du travail effectué par une personne.
- Scrum ne reconnaît pas d'équipes au sein de l'équipe de développement indépendamment des domaines qui doivent être couverts tels que l'exécution de tests, l'architecture, la gestion opérationnelle ou l'analyse fonctionnelle.
- Les membres de l'équipe de développement peuvent détenir individuellement des compétences et des centres d'intérêt spécifiques, mais c'est l'équipe de développement dans son ensemble qui est tenue responsable.
- **Taille de l'équipe de développement**

La taille optimale de l'équipe de développement doit être suffisamment petite pour rester réactive et assez grande pour accomplir un travail significatif durant le Sprint. Le fait d'avoir une équipe de développement de moins de trois membres, réduit l'interaction et produit de faibles gains de productivité. Durant un Sprint, les petites équipes de développement peuvent rencontrer des contraintes liées aux compétences ce qui les empêchent à livrer un incrément potentiellement publiable. Les rôles de Product Owner et Scrum Master ne sont pas inclus dans l'équipe de développement, à moins qu'ils aient également des tâches dans le Backlog Sprint. (Schwaber & Sutherland, 2017)

- **Le Scrum Master**

Ou le Maître de mêlée, il est chargé de promouvoir et supporter Scrum tel que défini dans le Guide Scrum. Les Scrum Masters remplissent leur rôle en aidant tout le monde à comprendre la théorie, les pratiques, les règles et les valeurs de Scrum. (Schwaber & Sutherland, 2017)

Le Scrum Master est un leader-serviteur de l'équipe Scrum. Le Scrum Master assiste les personnes externes à l'équipe Scrum pour identifier quelles sont les interactions bénéfiques avec elle. Le Scrum Master aide tout le monde à adapter leurs interactions avec l'équipe Scrum pour maximiser la valeur créée par cette équipe. (Schwaber & Sutherland, 2017)

- **Missions du SCRUM Master :**
  - **Le Scrum Master au service du Product Owner**

Le Scrum Master sert le Product Owner de plusieurs façons, y compris :

- S'assurer que les objectifs, le périmètre et le domaine du produit sont compris par tous les membres de l'équipe Scrum de la meilleure façon possible.
- Trouver des techniques pour une gestion efficace du Backlog produit.
- Aider l'équipe Scrum à comprendre le besoin de clarté et concision des éléments du Backlog produit.
- Comprendre la planification de produits dans un contexte empirique.
- S'assurer que le Product Owner sait comment organiser le Backlog produit pour maximiser la valeur.
- Comprendre et mettre en œuvre l'Agilité.
- Faciliter les événements Scrum, en cas de demande ou nécessité.
  - **Le Scrum Master au service de l'équipe de Développement**

Le Scrum Master sert l'équipe de Développement de plusieurs façons, y compris :

- Coacher l'équipe de développement en matière d'auto-organisation et de pluridisciplinarité.
- Aider l'équipe de développement à créer des produits de grande valeur.
- Supprimer les obstacles à la progression de l'équipe de développement.
- Faciliter les événements Scrum, en cas de demande ou nécessité.
- Coacher l'équipe de développement dans des environnements organisationnels où Scrum n'est pas encore complètement adopté et compris.
  - **Le Scrum Master au service de l'Organisation**

Le Scrum Master sert l'organisation de plusieurs façons, y compris :

- Accompagner l'organisation dans son adoption de Scrum.
- Planifier les implémentations de Scrum au sein de l'organisation.

- Aider les employés et les parties prenantes à comprendre et adopter Scrum ainsi que le développement empirique de produits.
- Provoquer les changements qui augmentent la productivité de l'équipe Scrum.
- Collaborer avec d'autres Scrum Masters pour accroître l'efficacité de l'application de Scrum au sein de l'organisation.

### 1.1.5. Les événements Scrum

Les événements prescrits sont utilisés par Scrum pour créer la régularité et minimiser le besoin de réunions non définies par Scrum. Une fois qu'un Sprint commence, sa durée est fixe et ne peut être écourtée ou prolongée. Ces événements sont spécifiquement conçus pour permettre la transparence et l'inspection. Ne pas inclure l'un de ces événements entraîne une transparence réduite et constitue une occasion perdue d'inspection et d'adaptation. (Schwaber & Sutherland, 2017)

- **Le Sprint**

Le cœur de Scrum est le Sprint, qui a une boîte de temps (time-box), une durée, d'un mois ou moins au cours de laquelle un Incrément Produit « Fini » fonctionnel et potentiellement publiable est créé. Les sprints ont une durée cohérente durant la phase de développement. Un nouveau Sprint commence immédiatement après la conclusion du Sprint précédent.

Les Sprints contiennent et consistent en une planification du Sprint (Sprint Planning), des mêlées quotidiennes (Daily Scrums), des activités de développement, une revue de sprint (Sprint Review) et une rétrospective de Sprint (Sprint Retrospective). (Schwaber & Sutherland, 2017)

Pendant le Sprint :

- L'objectif du sprint est fixe ; les changements qui le remettent en cause ne sont donc pas permis.
- Les objectifs de qualité sont maintenus ; ils ne sont jamais revus à la baisse.
- Le périmètre peut être clarifié et renégocié entre le Product Owner et l'équipe de développement selon ce que l'équipe Scrum apprend.

Chaque sprint peut être considéré comme un projet n'ayant qu'un horizon d'un mois. À l'instar d'un projet, un Sprint est utilisé pour accomplir quelque chose. Lorsque l'échéance d'un Sprint

est trop longue, la définition de ce qui est en cours de construction peut changer, la complexité peut augmenter et le risque peut s'accroître. (Schwaber & Sutherland, 2017)

- **Planification d'un Sprint**

La Planification du Sprint doit répondre aux questions suivantes :

- Que peut-on livrer comme incrément résultant du Sprint à venir ?
- Comment sera effectué le travail à livrer et nécessaire pour achever l'Incrément ?

Le travail à effectuer durant un Sprint est défini durant la réunion de Planification du Sprint (Sprint Planning). Ce plan est créé de manière collaborative par tous les membres de l'équipe Scrum. La réunion de planification du Sprint dure au maximum huit heures pour un Sprint d'un mois. Pour les sprints plus courts, l'événement est généralement plus court. Le Scrum Master veille à ce que l'événement ait lieu et que les participants en comprennent le but. Le Scrum Master apprend à l'équipe Scrum à le garder dans la boîte de temps (time-box). (Schwaber & Sutherland, 2017)

- **Annulation d'un sprint**

Un Sprint peut être annulé avant son échéance. Seul le Product Owner a le pouvoir d'annuler le Sprint, bien qu'il ou elle puisse le faire sous l'influence des parties prenantes, de l'équipe de développement ou du Scrum Master. Un Sprint serait annulé si l'objectif du Sprint devient obsolète. Cela pourrait se produire si l'organisation change de direction ou si les conditions de marché ou celles technologiques changent. (Schwaber & Sutherland, 2017)

En général, un Sprint devrait être annulé s'il n'a plus de sens compte tenu des circonstances. Lorsqu'un Sprint est annulé, tous les éléments achevés et « Finis » du Backlog produit sont examinés. Tous les éléments du Backlog produit incomplets sont estimés à nouveau et remis au Backlog produit. Les annulations Sprint consomment des ressources, car tout le monde se regroupe dans une autre réunion de planification du Sprint pour commencer un autre Sprint.

- **Objectif du Sprint**

L'objectif du Sprint est un but fixé pour le Sprint et peut être réalisé par l'implémentation d'une partie du Backlog Produit. Par ailleurs, l'objectif du Sprint peut être une autre source de cohérence poussant l'équipe de développement à travailler ensemble au lieu d'entreprendre des initiatives distinctes. (Schwaber & Sutherland, 2017)

- **Daily Scrum**

Ou La mêlée quotidienne, est un événement de 15 minutes destiné à l'équipe de développement. L'équipe de développement utilise la mêlée quotidienne pour inspecter son avancement vers l'objectif du Sprint et comment cet avancement tend à l'achèvement des travaux prévus dans le Backlog Sprint.

La structure de la réunion est définie par l'équipe de développement et peut être conduite de différentes manières si elle se concentre sur la progression vers l'objectif du Sprint. Certaines équipes de développement utiliseront les questions, d'autres seront plus basées sur la discussion. Voici un exemple de ce qui pourrait être utilisé :

- Qu'est-ce que j'ai fait hier qui a aidé l'équipe de développement à atteindre l'objectif du Sprint ?
- Que ferai-je aujourd'hui pour aider l'équipe de développement à atteindre l'objectif du Sprint ?
- Est-ce que je vois des obstacles qui m'empêchent ou empêchent l'équipe de développement de respecter l'objectif du Sprint ?

Figure 27: Une présentation de réunion quotidienne SCRUM



Source (Vickoff J. P., 2016)

L'équipe de développement ou les membres de l'équipe se rencontrent souvent immédiatement après la mêlée quotidienne pour des discussions détaillées ou pour adapter ou planifier à nouveau le reste du travail du Sprint. Le Scrum Master apprend à l'équipe de développement à

limiter la mêlée quotidienne à 15 minutes. Les mêlées quotidiennes améliorent la communication, éliminent les autres réunions, identifient les obstacles à éliminer qui perturbent le développement, mettent en avant et encouragent la prise de décision rapide tout en améliorant le niveau de savoir au sein l'équipe de développement. Il s'agit d'un point clé d'inspection et d'adaptation.

- **Revue de sprint**

Cette réunion se veut informelle, pas une réunion de pilotage, et la présentation de l'incrément est destinée à susciter les réactions et à favoriser la collaboration. Le Scrum Master apprend à tous ceux qui y sont impliqués à le garder dans la boîte de temps. (Schwaber & Sutherland, 2017)

À la fin du sprint, l'équipe et les parties prenantes invitées se réunissent pour effectuer la revue de sprint, qui dure au maximum quatre heures. L'objectif de la revue de sprint est de valider l'incrément de produit qui a été réalisé pendant le sprint. L'équipe énonce les éléments du carnet de produit sélectionnés en début de sprint. (Schwaber & Sutherland, 2017)

La Revue de sprint comprend les éléments suivants :

- Les participants incluent l'équipe Scrum et les principales parties prenantes invitées par le Product Owner.
- Le Product Owner indique quels éléments du Backlog Produit ont été « Finis » et ceux qui n'ont pas été « Finis ».
- L'équipe de développement discute de ce qui s'est bien passé pendant le Sprint, quels problèmes ont été rencontrés, et comment ces problèmes ont été résolus.
- L'équipe de développement démontre le travail « Fini » et répond aux questions sur l'incrément.
- Le Product Owner discute de l'état actuel du Backlog Produit tel qu'il est. Il ou elle projette les dates prévisionnelles et celles de livraison en fonction des progressions réalisées à ce jour (si nécessaire).
- L'ensemble du groupe convient de ce qu'il faut faire pour la suite, de sorte que la revue de sprint fournisse une contribution précieuse à la prochaine réunion de Planification du Sprint.
- La revue de la façon dont les conditions de marché ou un usage potentiel du produit pourrait avoir dicté ce qu'il conviendrait mieux de faire dorénavant.

- La revue des délais, budget, fonctionnalités potentielles et conditions de marché pour les prochaines versions prévues de la fonctionnalité du produit.

Le résultat de la revue de sprint est un Backlog Produit révisé qui définit les éléments probables pour le prochain Sprint. Le Backlog Produit peut également être globalement ajusté pour répondre aux nouvelles opportunités d'affaires.

- **Rétrospective de Sprint**

C'est 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. C a d L'équipe projet doit faire une auto-évaluation sur ce qu'ils ont bien fait et ce qu'ils ont mal fait « ce qu'a bien fonctionné et ce qu'a mal fonctionné ». (Schwaber & Sutherland, 2017)

La rétrospective de Sprint se produit après la revue de sprint et avant la prochaine réunion de planification du Sprint. Il s'agit au maximum d'une réunion de trois heures pour les Sprints d'un mois. Pour les sprints plus courts, l'événement est généralement plus court. Le Scrum Master s'assure que l'événement ait lieu et que les participants en comprennent son but.

Le Scrum Master s'assure que la réunion est positive et productive. Le Scrum Master apprend à tous ceux qui y sont impliqués à la garder dans la boîte de temps (time-box). Le Scrum Master participe en tant que membre de l'équipe Scrum et y amène le point de vue du responsable du processus Scrum. (Schwaber & Sutherland, 2017)

Le but de la rétrospective de Sprint est de :

- Inspecter la manière dont le dernier Sprint s'est déroulé en ce qui concerne les personnes, les relations, les processus et les outils.
- Identifier et ordonner les principaux éléments qui ont bien fonctionné et des améliorations potentielles.
- Créer un plan pour mettre en œuvre des améliorations sur la façon dont l'équipe Scrum fait son travail.

Le Scrum Master encourage l'équipe Scrum à améliorer, dans le cadre de travail Scrum, son processus de développement et ses pratiques afin de les rendre plus efficaces et agréables pour le prochain Sprint. À la fin de la rétrospective de Sprint, l'équipe Scrum devrait avoir identifié les améliorations qu'elle mettra en œuvre dans le prochain Sprint. Bien que des améliorations

puissent être mises en œuvre à tout moment, la rétrospective de Sprint offre une opportunité formelle d'inspection et d'adaptation. (Schwaber & Sutherland, 2017)

### **1.1.6. Les artefacts de Scrum**

Les artefacts de Scrum représentent soit du travail, soit de la valeur fournissant ainsi de la transparence et des opportunités pour l'inspection et d'adaptation. Les artefacts définis par Scrum sont spécialement conçus pour maximiser la transparence d'informations essentielles afin que chacun ait la même compréhension de l'artefact. (Schwaber & Sutherland, 2017)

- **Backlog Produit**

Le Backlog Produit est une liste ordonnée de tous les éléments identifiés comme nécessaires au produit. Il constitue l'unique source d'exigences pour tout changement à apporter au produit. Le Product Owner est responsable du Backlog produit, y compris son contenu, sa disponibilité et son ordonnancement. Le Backlog Produit évolue au fur et à mesure que le produit et le contexte dans lequel il sera utilisé évoluent. (Schwaber & Sutherland, 2017)

Si un produit existe, son Backlog Produit existe. Le Backlog Produit liste toutes les fonctionnalités, les fonctions, les exigences, les améliorations et les corrections qui constituent des modifications à apporter au produit dans les versions futures. (Schwaber & Sutherland, 2017)

À mesure qu'un produit est utilisé, que sa valeur augmente et que l'on commence à recevoir des retours sur son utilisation, le Backlog produit devient une liste plus vaste et plus exhaustive. Les exigences ne cessent jamais de changer, de sorte qu'un Backlog produit est un artefact vivant. Les changements de besoins de l'organisation, des conditions de marché ou de la technologie peuvent entraîner des changements dans le Backlog produit. On peut alors ajouter un attribut aux éléments du Backlog Produit pour les regrouper. (Schwaber & Sutherland, 2017)

Le raffinement du back log de produit consiste en l'ajout de détails, d'estimations et de l'ordonnancement des éléments du Backlog Produit. Il s'agit d'une activité régulière dans laquelle le Product Owner et l'équipe de développement collaborent pour détailler les éléments du Backlog Produit. (Schwaber & Sutherland, 2017)

Les éléments qui occuperont l'équipe de développement durant le prochain Sprint sont affinés au point que n'importe lequel peut être raisonnablement « Fini » dans un Sprint. Le Product Owner peut influencer l'équipe de développement en l'aidant dans sa compréhension et le choix

des compromis, mais les personnes qui effectueront le travail ont le mot final sur les estimations. (Schwaber & Sutherland, 2017)

- **Suivi de la progression vers les objectifs**

À tout moment, la somme de travail restant pour atteindre un objectif de développement peut être calculée. Diverses pratiques de projections des tendances telles que les « burn downs », les « burn- ups » ou les « cumulative flow » ont été utilisées afin de prévoir la progression. Ces pratiques ont prouvé leur utilité. Seul ce qui s'est déjà passé peut-être utiliser pour la prise de décision prévisionnelle. (Schwaber & Sutherland, 2017)

- **Backlog Sprint**

Le Backlog 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. Le Backlog 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 « Fini ». Pour assurer une amélioration continue, il comprend au moins une amélioration de processus, hautement prioritaire, identifiée lors de la précédente réunion de rétrospective de Sprint. Le Backlog Sprint est un plan suffisamment détaillé pour que la progression soit compréhensible lors de la mêlée quotidienne. (Schwaber & Sutherland, 2017)

- **Suivi de la progression du Sprint**

À n'importe quel moment d'un sprint, la somme totale du travail restant dans le Backlog Sprint peut être calculée. L'équipe de développement fait le suivi de cette somme de travail restant au moins à chaque mêlée quotidienne pour évaluer la probabilité d'atteindre l'objectif du Sprint. En effectuant le suivi du travail restant tout au long du sprint, l'équipe de développement peut gérer son avancement. (Schwaber & Sutherland, 2017)

- **Incrément**

L'incrément est constitué des éléments du Backlog produit « Finis » pendant le sprint ainsi que de la valeur cumulative des incréments livrés dans les sprints précédents. À la fin d'un Sprint, le nouvel incrément doit être « Fini », ce qui implique qu'il doit être dans un état publiable et qu'il correspond à la définition de « Fini » (Definition of Done) de l'équipe de développement. Un incrément est la partie d'un travail fait, qui prend en charge l'empirisme, et vérifiable à la fin du Sprint. L'incrément est un pas vers une vision ou un but. L'incrément doit être dans un

état publiable, sans égard à la décision du Product Owner de le publier ou non. (Schwaber & Sutherland, 2017)

- **Artefact de Transparence**

Les décisions d'optimiser la valeur et contrôler le risque sont prises en se basant sur l'état perçu des artefacts. Dans la mesure où les artefacts ne sont pas totalement transparents, ces décisions peuvent être faussées, la valeur peut diminuer et le risque peut augmenter. Le Scrum Master doit travailler avec le Product Owner, l'équipe de développement et les autres parties impliquées pour comprendre si les artefacts sont complètement transparents. Un Scrum Master peut détecter une transparence incomplète en inspectant les artefacts, en identifiant certains usages récurrents, en écoutant attentivement ce qui est dit et en décelant les différences entre les résultats escomptés et réels. (Schwaber & Sutherland, 2017)

La responsabilité du Scrum Master consiste à travailler avec l'équipe Scrum et l'organisation afin d'accroître la transparence des artefacts.

- **Définition de « Fini »**

Lorsqu'un élément du Backlog produit ou un Incrément est décrit comme « Fini », tout le monde doit comprendre ce que « Fini » signifie, Bien que cela puisse varier considérablement d'une équipe Scrum à une autre, les membres doivent avoir une compréhension commune de ce que signifie que le travail soit complet, afin d'assurer la transparence. Il s'agit de la définition de « Fini » pour l'équipe Scrum. Celle-ci est utilisée pour évaluer si le travail est terminé dans un incrément produit. (Schwaber & Sutherland, 2017)

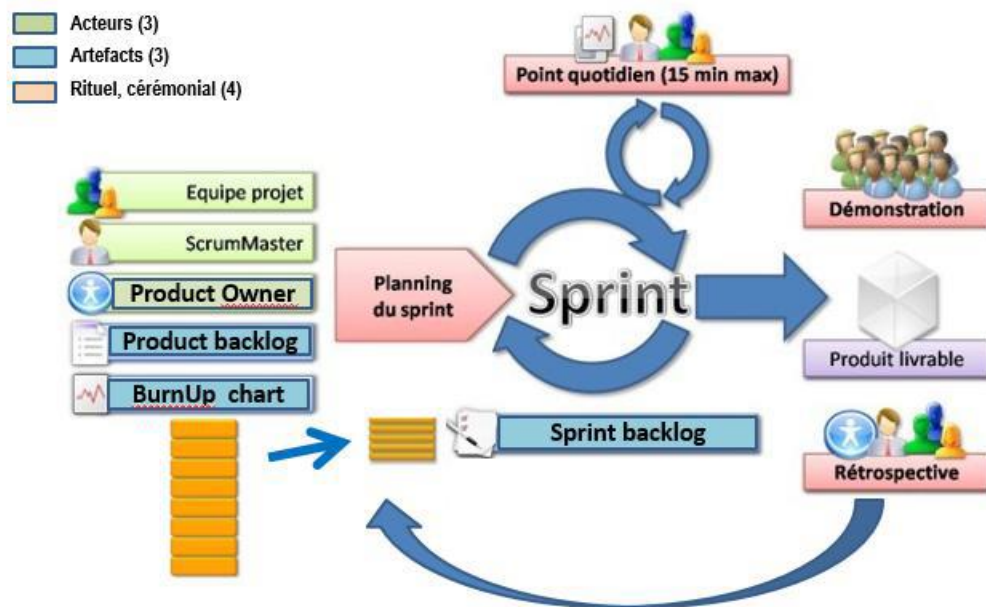
La même définition guide l'équipe de développement en sachant combien d'éléments de Backlog produit elle peut choisir au cours de la Planification du Sprint. L'objectif de chaque Sprint est de fournir des Incréments de fonctionnalités potentiellement publiables qui adhèrent à la définition de « Fini » actuelle de l'équipe Scrum. (Schwaber & Sutherland, 2017)

Les équipes de développement fournissent un incrément de la fonctionnalité du produit à chaque Sprint. Cet incrément est publiable, de telle sorte qu'un Product Owner peut choisir de le publier immédiatement. Si la définition de « Fini » d'un incrément fait partie des conventions, standards ou lignes directrices de l'organisation de développement, toutes les équipes doivent au minimum la respecter. (Schwaber & Sutherland, 2017)

Chaque Incrément est ajouté à tous les Incréments précédents et testé de manière approfondie, tout en veillant à ce que tous les Incréments fonctionnent ensemble.

Au fur et à mesure que les équipes de Scrum mûrissent, on s'attend à ce que leurs définitions de « Fini » se développent pour inclure des critères plus stricts pour une meilleure qualité. L'utilisation d'une nouvelle définition de « Fini » peut révéler le travail à faire dans les incréments précédemment « Finis ». Tout produit ou système devrait avoir une définition de « Fini » qui est une norme pour tout travail effectué. (Schwaber & Sutherland, 2017)

Figure 28: Schéma du processus de la méthode SCRUM



Source : (Vickoff J. P., 2016)

## 1.2. KANBAN

Kanban se concentre sur la conception, la gestion et l'amélioration de systèmes en flux de travail de la connaissance tels que du conseil ou de la conception de produits (physiques ou logiciels). Elle peut être qualifiée comme la méthode du "Commencez par là où vous en êtes". (Morisseau, 2014), (Anderson & Carmichael, 2016). Kanban peut être qualifié comme « Un catalyseur de changements ciblés et rapides au sein des entreprises de tout type, Qui réduit fortement la résistance au changement tout en respectant les objectifs de l'entreprise. »

La Méthode Kanban s'applique à rendre visible un travail immatériel, et s'assure que la quantité de travail en cours ne dépasse pas un seuil qui nuit à la productivité « travailler sur ce que le

client attend de façon adaptée à la capacité du service. Pour ce faire nous utilisons un système kanban » « un flux d'éléments de travail engendré par une limitation du Travail en cours (WIP pour Work in progress) et des marqueurs visuels ». (Morisseau, 2014), (Anderson & Carmichael, 2016)

Ces marqueurs visuels souvent dénommés kanbans ou cartes kanban sont affichées sur des tableaux kanban. Ils permettent de voir et alerter lorsque trop ou trop peu de travaux sont dans le système, ce qui améliore le flux de valeur pour les clients. Les règles de limitation du travail en cours créent un système à flux tiré (pull system) : un nouvel élément de travail est "tiré" dans le système uniquement lorsqu'un autre élément vient d'être terminé et libère une capacité de réalisation. Plutôt que de « pousser » les éléments à la demande (et créer une accumulation de travail en cours). (Morisseau, 2014), (Anderson & Carmichael, 2016)

Kanban se focalise sur la livraison de services par une organisation « une ou plusieurs personnes collaborent pour produire des produits d'activité (souvent immatériels). Un service dispose d'un client qui commande le travail dont les besoins sont identifiés et qui accepte ».

### 1.2.1. Les Valeurs Kanban

Kanban est une méthode pilotée par la valeur. Elle est motivée par la conviction que le respect de tous les individus qui contribuent à une entreprise collaborative est nécessaire, non seulement pour le succès de l'entreprise mais pour lui donner du sens. (Morisseau, 2014), (Anderson & Carmichael, 2016)

Les valeurs de Kanban pourraient être résumées par le simple mot « respect ». Cependant il est nécessaire de l'étendre à un ensemble de neuf valeurs<sup>1</sup> (incluant respect) encapsulant la raison de l'existence des principes et pratiques de Kanban :

1. **Transparence** : la conviction que le partage public de l'information améliore le flux de valeur. Utiliser un vocabulaire clair et simple fait partie de cette valeur.
2. **Équilibre** : la compréhension que différents aspects, points de vue et capacités doivent être équilibrés entre eux pour accroître son efficacité. Certains aspects (tels que demande et capacité) entraîneront une détérioration s'ils sont déséquilibrés pendant une période prolongée.
3. **Collaboration** : travailler ensemble. La méthode Kanban a été formulée pour améliorer la façon dont les personnes travaillent ensemble, de ce fait la collaboration est en son cœur.

4. **Centré Client** : comprendre l'objectif du système. Tout « système kanban » coule vers un point de réalisation de la valeur où les clients reçoivent un élément ou un service demandé. C'est le point d'attention naturel dans Kanban.
5. **Flux** : la réalisation de travaux tel un flux de valeur qu'il soit continu ou épisodique. Voir le flux est un point de départ essentiel dans l'utilisation de Kanban.
6. **Direction (leadership)** : habilité à inspirer les autres à agir par des exemples, des mots et une réflexion. La plupart des organisations disposent de certains degrés de structure hiérarchique cependant le leadership est nécessaire dans Kanban à tous les niveaux pour accomplir la livraison de valeur et l'amélioration.
7. **Compréhension** : le savoir de départ (individuel et de l'entreprise) afin d'aller de l'avant. Kanban est une méthode d'amélioration et la connaissance individuelle est un point de départ fondamental.
8. **Accord** : l'engagement de se déplacer ensemble vers des objectifs en respectant et, quand c'est possible, en s'accommodant des différences d'opinions ou d'approches. Ce n'est pas un management par consensus mais un Co-engagement dynamique d'amélioration.
9. **Respect** : valoriser, comprendre et montrer de la considération pour les personnes. Bien à sa place au pied de cette liste, il est le fondement sur lequel les autres valeurs se reposent.

### 1.2.2. Les principes fondamentaux de Kanban

Il y a six principes fondamentaux de Kanban pouvant être divisés en deux groupes : les principes de gestion du changement et les principes de livraison de services.

#### 1.2.2.1. Principes de Gestion du Changement

Votre organisation est un réseau d'individus, psychologiquement et sociologiquement câblés pour résister au changement. Kanban reconnaît ces aspects humains avec trois principes de conduite du changement :

Les trois principes de conduite du changement de Kanban sont :

1. Commencez par là où vous en êtes :
  - Comprendre les processus existants tels que pratiqués actuellement
  - Respecter les rôles, les responsabilités (fonctions) et les titres existants
2. S'engager à pratiquer l'amélioration par un changement évolutif
3. Encourager les actions de leadership à tous les niveaux : de l'individu à la direction.

Il y a deux raisons principales qui font que « commencer par là où vous en êtes » est une bonne idée. La première est que minimiser la résistance au changement en respectant les pratiques actuelles et ceux qui les pratiquent est cruciale pour engager chacun à relever les défis de l'avenir. La seconde est que les processus actuels, avec leurs lacunes évidentes, contiennent la sagesse et la résilience que même ceux qui les utilisent peuvent ne pas apprécier pleinement. Puisque le changement est essentiel, nous ne devons pas imposer des solutions à partir de contextes différents, mais plutôt accepter de poursuivre une amélioration évolutionniste à tous les niveaux de l'organisation. En démarrant de la pratique actuelle, on établit le niveau de référence de la performance et de l'efficacité qui permettra d'évaluer les changements futurs. (Anderson & Carmichael, 2016)

### **1.2.2.2. Principes de Livraison de Services**

Toute organisation est un écosystème de services interdépendants. Kanban reconnaît cela avec trois principes de livraison de services, applicables non seulement à un service mais à l'ensemble du réseau :

1. Comprendre et se concentrer sur les besoins et les attentes des clients
2. Gérer le travail : laisser les personnes s'auto-organiser
3. Votre organisation est un écosystème de services interdépendants dirigés par ses règles : réfléchissez régulièrement sur leur effectivité et comment les améliorer.

Ces principes s'alignent étroitement avec le programme d'orientation services et la valeur de l'orientation client. Lorsque les éléments de travail et le flux de valeur client qu'ils représentent ne sont pas clairement visibles, les entreprises se concentrent alors sur ce qui reste visible : les personnes qui travaillent. Sont-elles toujours occupées ? Sont-elles suffisamment compétentes ? Pourraient-elles travailler plus dur ? Le client et les éléments de travail apportant de la valeur pour le client reçoivent alors moins d'attention. C'est ce que ces principes mettent en lumière : la priorité doit revenir aux clients du service et la valeur qu'ils reçoivent de ce service. (Anderson & Carmichael, 2016)

### **1.2.2.3. Les pratiques de base de Kanban**

Les pratiques de base de Kanban définissent les activités essentielles pour ceux qui gèrent les systèmes kanban. Il y en a six :

1. Visualiser.
2. Limiter le Travail en Cours (WiP : Work in Progress).

3. Gérer le flux.
4. Rendre les règles explicites.
5. Implémenter des boucles de feedback.
6. Améliorer collaborativement, évoluer expérimentalement.

Ces pratiques impliquent :

- De voir le travail et les règles qui déterminent quand et comment le traiter ; puis
- D'améliorer le processus de manière évolutionnaire « que ce soit en conservant et généralisant une améliorant utile, ou en abandonnant et tirant des enseignements de changements inefficaces ».

### **1.3. eXtrem Programing**

XP est principalement l'œuvre de Kent Beck et Ward Cunningham, deux experts du développement logiciel, et plus particulièrement de la conception objet et des patterns. Avant le développement d'XP, Kent Beck devait surtout sa notoriété dans le monde Smalltalk à son livre, Smalltalk Best Practice Patterns. (Bénard, 2005)

L'eXtreme Programming (XP) est un ensemble de pratiques qui couvre une grande partie des activités de la réalisation d'un logiciel de la programmation proprement dite à la planification du projet, en passant par l'organisation de l'équipe de développement et les échanges avec le client. Ces pratiques n'ont en soi rien de révolutionnaire : il s'agit simplement de pratiques de bon sens mises en œuvre par des développeurs ou des chefs de projet expérimentés, telles que les livraisons fréquentes, la relecture de code, la mise en place de tests automatiques... La nouveauté introduite par XP consiste à pousser ces pratiques à l'extrême d'où le nom de la méthode et à les organiser en un tout cohérent, parfaitement défini et susceptible d'être répété. (Bénard, 2005)

Par opposition aux méthodes de développement dites « lourdes », qui imposent des activités et des produits indirects consommateurs de temps et de ressources, XP se définit comme un processus léger dans lequel l'équipe se focalise sur la réalisation elle-même. Toutes les autres activités sont réduites à leur strict minimum, sachant que la priorité y est donnée à l'Agilité : XP s'adresse principalement à de petites équipes (moins de dix personnes) qui souhaitent réaliser rapidement des logiciels et réagir aisément au changement, sans faire pour autant de concessions sur la qualité du logiciel produit. (Bénard, 2005)

Dans la pratique, ce critère d'Agilité proposé par XP se retrouve à trois niveaux :

- Au niveau du code lui-même, qui est maintenu toujours aussi clair et simple que possible par une activité de remaniement permanente, et qui est soutenu par une batterie de tests automatiques permettant d'y introduire de nouvelles fonctionnalités sans craindre des régressions cachées.
- Au niveau de l'équipe de développement, dont les membres travaillent toujours en commun et sont capables d'intervenir sur toutes les parties de l'application.
- Au niveau de la gestion du projet, à travers une démarche itérative qui permet à tous les intervenants du projet « aussi bien l'équipe de développement que le client d'améliorer continuellement son pilotage en s'appuyant sur l'expérience acquise en cours de route ».

### 1.3.1. Les pratiques de programmation

Au cœur des pratiques XP, les pratiques de programmation que nous présentons ci-après permettent d'améliorer continuellement la conception et le code de l'application pour qu'elle reste toujours aussi claire et simple que possible :

- Conception simple (simple design) : les développeurs implémentent toujours la solution la plus simple qui puisse fonctionner. En particulier, ils n'inventent pas de mécanismes génériques si le besoin immédiat ne l'exige pas.
- Remaniement (refactoring) : les développeurs n'hésitent pas à revenir sur le code écrit pour le rendre plus « propre », le débarrasser d'éventuelles parties inutilisées, et le préparer à l'ajout de la fonctionnalité suivante. D'une manière plus générale, cette pratique propose une démarche de conception continue qui fait émerger la structure de l'application au fur et à mesure du développement.
- Développement piloté par les tests unitaires (test-first programming, unit tests, developer tests) : les développeurs écrivent des tests automatiques pour le code qu'ils produisent, et ce au moment même d'écrire le code en question. Cela leur permet d'une part de mieux cerner le problème avant d'écrire le code, et d'autre part de constituer progressivement une batterie de tests qui les autorise ensuite à apporter rapidement des changements dans l'application, tout en conservant une certaine sérénité.
- Tests de recette (acceptance tests, customer tests) : le client précise très explicitement ses besoins et les objectifs des programmeurs « en participant à la rédaction de tests de recette. Comme les tests unitaires, les tests de recette doivent être automatiques afin de pouvoir vérifier tous les jours la non-régression du produit ».

### 1.3.2. Les pratiques de collaboration

Dans une équipe XP, tous les développeurs travaillent ensemble et interviennent sur la totalité de l'application. Cela garantit la qualité de cette dernière à travers les relectures croisées que cela engendre, mais cela rend également le travail plus motivant et offre une plus grande souplesse dans l'affectation des tâches. Les pratiques qui régissent cette organisation sont les suivantes : (Bénard, 2005)

- Programmation en binôme (pair programming) : lorsqu'ils écrivent le code de l'application, les développeurs travaillent systématiquement à deux sur la même machine – il s'agit là d'une forme « extrême » de relecture de code, dans laquelle les deux développeurs collaborent activement pour résoudre les problèmes qu'ils rencontrent. Les binômes changent fréquemment, ainsi chacun est amené à travailler tôt ou tard avec tous les autres membres de l'équipe.
- Responsabilité collective du code (collective code ownership) : tous les développeurs de l'équipe peuvent être amenés à travailler sur toutes les parties de l'application. De plus, ils ont le devoir d'améliorer le code sur lequel ils interviennent, même s'ils n'en sont pas les auteurs initiaux.
- Règles de codage (coding standards) : les développeurs se plient à des règles de codage définies par l'équipe elle-même, de manière à garantir l'homogénéité de leur code avec le reste de l'application, et ainsi à faciliter l'intervention d'autres développeurs.
- Métaphore (metaphor) : les développeurs n'hésitent pas à recourir aux métaphores pour décrire la structure interne du logiciel ou ses enjeux fonctionnels, de façon à faciliter la communication et à assurer une certaine homogénéité de style dans l'ensemble de la conception, l'idéal étant de décrire le système dans son ensemble par une métaphore unique.
- Intégration continue (continuous integration) : les développeurs synchronisent leurs développements aussi souvent que possible « au moins une fois par jour. Cela réduit la fréquence et la gravité des problèmes d'intégration, et permet de disposer à tout moment d'une version du logiciel qui intègre tous les développements en cours ».

### 1.3.3. Les pratiques de gestion de projet

Les pratiques de programmation et de collaboration permettent de créer un contexte dans lequel une démarche de spécifications et de conception purement itérative devient viable. Les pratiques suivantes montrent comment XP exploite cet avantage afin de s'assurer que l'équipe

et son client restent en phase tout au long du projet, de façon à converger au plus tôt vers un produit adapté aux besoins du client : (Bénard, 2005)

- Livraisons fréquentes (frequent releases) : l'équipe livre des versions du logiciel à un rythme régulier, aussi élevé que possible la fréquence précise étant fixée par le client. Cela permet à l'équipe comme au client de s'assurer que le produit correspond bien aux attentes de ce dernier et que le projet est sur la bonne voie.
- Planification itérative (planning Game) : la planification du projet est réalisée conjointement par le client et l'équipe de développement, au cours de séances dédiées, organisées régulièrement tout au long du projet.
- Client sur site (on-site Customer, whole team) : le client est littéralement intégré à l'équipe de développement pour arbitrer les priorités, et définir précisément ses besoins, notamment en répondant en direct aux questions des programmeurs et en bénéficiant du feedback immédiat d'une application aux livraisons fréquentes.
- Rythme durable (sustainable pace) : l'équipe adopte des horaires qui lui permettent de conserver tout au long du projet l'énergie nécessaire pour produire un travail de qualité et mettre en œuvre efficacement les autres pratiques.

#### 1.3.4. Les quatre valeurs d'XP

Si elles sont indispensables aujourd'hui pour mettre l'eXtreme Programming en œuvre, les pratiques que nous venons de présenter ne suffisent pas pour autant à le définir. Il ne s'agit en définitive que de techniques, destinées à faire émerger un environnement de travail marqué par les quatre qualités érigées en valeurs par XP et qui en font l'essence : la communication, la simplicité, le feedback et le courage. (Bénard, 2005)

- **La communication pour une meilleure visibilité**

Du point de vue d'XP, un projet de développement est avant tout un effort collectif de création, dont le succès dépend d'une part de la capacité de ses différents intervenants à s'accorder sur une vision commune de ce qui doit être produit, et d'autre part de leur capacité à synchroniser leurs actions individuelles pour atteindre l'objectif commun. Or, ces deux conditions dépendent en majeure partie de la qualité de la communication qui lie ces intervenants entre eux. Mais sur le fond, il n'y a finalement pas grand-chose de spécifique à XP sur ce point. (Bénard, 2005)

Ce qui démarque l'approche XP dans ce domaine, c'est l'accent mis sur la communication directe, sur le contact humain. Certes, la communication orale présente des faiblesses en termes

de structuration de l'information et de traçabilité, mais elle tire sa force de sa simplicité et des interactions rapides entre les interlocuteurs qui permettent de converger rapidement sur les informations essentielles. Au quotidien, la communication directe permet donc d'obtenir une « bande passante » nettement supérieure à l'écrit, et XP exploite largement cet avantage dans la plupart de ses pratiques. Mais elle apparaît sous une forme différente : toutes les informations ayant trait à l'implémentation et la conception se retrouvent dans le code lui-même dont la clarté fait l'objet de nombreux efforts et dans la batterie de tests qui l'accompagne, et celles relatives aux besoins du client sont consignées d'une manière on ne peut plus formelle, sous la forme de tests de recette automatiques. (Bénard, 2005)

- **La simplicité comme garantie de productivité**

Une personne qui arrive sur un projet XP ne doit pas s'étonner de s'entendre répéter les quelques « mantras » suivants : « La chose la plus simple qui puisse marcher » (The simplest thing that could possibly work), « Tu n'en auras pas besoin » (You ain't gonna need it et son acronyme YAGNI), ou encore « Une fois et une seule » (Once and only once). Tous trois sont l'expression d'une même valeur : la simplicité. (Bénard, 2005)

Cette simplicité touche d'abord à la réalisation même du logiciel, dans laquelle un soin particulier est apporté au code pour le débarrasser de toute complexité superflue. En particulier, les mécanismes de généricité le « péché mignon » des développeurs ne sont encouragés que lorsqu'ils servent un besoin concret et immédiat, et en aucun cas un besoin futur plus ou moins imaginaire. Les développeurs sont donc invités à implémenter la chose la plus simple qui puisse marcher et, en ce qui concerne ces fameux mécanismes génériques, ou encore ces outils « magiques » qui en font plus que ce qu'on leur demande, ils n'auront pas besoin ! En revanche, simple ne veut pas dire simpliste : les solutions adoptées doivent être aussi simples que possible, mais toutes les duplications doivent être éliminées de sorte que chaque information, chaque mécanisme ne soit exprimé qu'une fois et une seule ce principe est garant de la facilité de modification du code sur le long terme. (Bénard, 2005)

Cet effort de simplicité s'applique également au client, à qui l'équipe demandera de définir ses besoins et ses priorités avec une grande précision pour éviter d'implémenter des choses inutiles et pour pouvoir se focaliser sur les besoins réellement importants. La simplicité est également recherchée dans le choix des outils (de programmation ou de gestion de projet) et dans la méthode de travail elle-même.

En définitive, XP est fondé sur un pari : « faire simple un jour, avec la possibilité de revenir en arrière le lendemain si un besoin différent apparaît. L'éventuel coût supplémentaire induit sera, d'une part réduite parce que l'application est restée assez simple pour évoluer facilement, et d'autre part sera bien peu de choses au regard des économies réalisées à ne pas faire ce qui n'aurait servi à rien. » C'est au succès de ce pari qu'une équipe XP doit sa vitesse de développement et son ouverture au changement. (Bénard, 2005)

- **Le feedback comme outil de réduction du risque**

Malgré ce que peuvent laisser supposer son nom et ses apparences immédiates, l'eXtreme Programming est avant tout un processus de réduction du risque dans le projet. Le risque est en effet soigneusement contrôlé à tous les niveaux, par la mise en place de boucles de feedback qui permettent à l'équipe de développement, comme à son client, de savoir à tout moment dans quel état se trouve réellement le projet, et de pouvoir rectifier le tir au fur et à mesure pour mener le projet à son terme avec succès. (Bénard, 2005)

Mais le feedback ne se borne pas à l'observation : la pratique de la planification itérative permet de tirer parti des informations recueillies pour, à la fois, améliorer la planification elle-même et faire converger le produit vers une solution mieux adaptée aux besoins réels du client.

L'activité de programmation fait également l'objet de divers mécanismes de feedback, tout d'abord à travers les tests unitaires mis en place, qui donnent aux développeurs des indications immédiates sur le fonctionnement du code qu'ils écrivent. Enfin, les développeurs s'appuient en permanence sur le feedback de leur binôme pour s'assurer de la validité et de la qualité du code qu'ils produisent. (Bénard, 2005)

Ce feedback permanent est un facteur de qualité, puisque les intervenants du projet améliorent sans cesse leur travail à partir de l'expérience qu'ils accumulent.

- **Le courage de prendre les bonnes décisions**

L'expérience fait apparaître que la mise en œuvre des pratiques XP requiert une certaine dose de cran. En effet, il faut du courage pour se lancer dans un projet sans avoir au préalable tout spécifié et conçu dans le détail, même si l'on sait que le processus suivi comporte de nombreux mécanismes de feedback. (Bénard, 2005)

Il faut également du courage pour se borner à réaliser des choses simples, se focaliser uniquement sur les besoins du moment en se disant qu'on pourra adapter l'application à de

nouveaux besoins, le moment venu. Enfin, il faut du courage pour appliquer les principes de communication et de feedback, en particulier lorsqu'il s'agit de maintenir une transparence complète, même lorsque les nouvelles ne sont pas bonnes, ou encore lorsqu'il s'agit de travailler ouvertement avec son binôme en acceptant de lui montrer nos propres limites ou lacunes. (Bénard, 2005)