

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

GESTION DE LA QUALITÉ DES PROJETS EN GÉNIE LOGICIEL : FACTEURS  
CLÉS ET SOLUTIONS

MÉMOIRE PRÉSENTÉ  
COMME EXIGENCE PARTIELLE DE LA  
MAÎTRISE EN GESTION DE PROJET

PAR  
ISSAKA GMAKOUBA

Février 2024

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire, de cette thèse ou de cet essai a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire, de sa thèse ou de son essai.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire, cette thèse ou cet essai. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire, de cette thèse et de son essai requiert son autorisation.

## SOMMAIRE

Un projet est caractérisé par un objectif bien défini afin de satisfaire des besoins spécifiques. Ces besoins de nature implicite ou explicite sont l'objet même pour lequel le projet a été entrepris. La qualité est devenue un sujet très important pour les entreprises en recherche de moyens pour améliorer leurs produits, services ou livrables. Ils sont en quête des pratiques et méthodologies afin d'accroître leurs performances. La gestion de la qualité est même un domaine de connaissances de la gestion de projet. Il regroupe l'ensemble des concepts et méthodes afin de garantir la conformité du résultat final ou du livrable aux objectifs prédéfinis et satisfaire considérablement les besoins. Bien que ce sujet soit largement abordé partout et dans beaucoup de domaines, il manque encore des recherches en gestion de projets afin de mieux le cerner et accroître considérablement la qualité dans les projets. Ainsi, s'intéresse particulièrement cette recherche à la gestion de projets logiciels et elle a pour objet d'étudier les éléments suivants : les Facteurs Clés de Succès, la Qualité, l'Agilité et l'Assurance Qualité. L'objectif de cette recherche est d'identifier les facteurs clés de succès pouvant conduire à la qualité du logiciel et d'étudier l'agilité et l'assurance qualité comme facteurs modérateurs de la relation entre facteurs clés de succès et la performance.

## TABLE DES MATIÈRES

<b>LISTE DES TABLEAUX .....</b>	<b>4</b>
<b>LISTE DES FIGURES.....</b>	<b>6</b>
<b>REMERCIEMENT .....</b>	<b>7</b>
<b>CHAPITRE 1: PARTIE INTRODUCTIVE .....</b>	<b>8</b>
1.1. CONTEXTE DE L'ÉTUDE .....	8
1.2. PROBLÉMATIQUE GÉNÉRALE.....	11
1.3. PROBLÉMATIQUE SPÉCIFIQUE .....	13
1.4. LOCALISATION DE LA RECHERCHE .....	15
1.5. OBJECTIFS DE LA RECHERCHE ET QUESTIONS DE RECHERCHE .....	16
<b>CHAPITRE 2. REVUE DE LITTÉRATURE .....</b>	<b>19</b>
2.1. LES FACTEURS CLÉS DE SUCCÈS .....	19
2.1.1. <i>Objectifs du projet bien clair</i> .....	20
2.1.2. <i>Des exigences bien claires</i> .....	22
2.1.3. <i>Les risques bien identifiés</i> .....	25
2.1.4. <i>Attentes réalistes</i> .....	26
2.1.5. <i>Responsabilisation</i> .....	27
2.1.6. <i>Personnel compétent</i> .....	29
2.1.7. <i>Soutien de la haute direction</i> .....	30
2.2. L'AGILITÉ.....	31
2.2.1. <i>Qu'est-ce que l'Agilité ?</i> .....	31
2.2.2. <i>Les indicateurs d'agilité</i> .....	35
2.3. LA QUALITÉ .....	48
2.3.1. <i>Définition de la qualité</i> .....	48
2.3.2. <i>Logiciel de qualité</i> .....	50
2.4. L'ASSURANCE QUALITÉ.....	53
2.5. CADRE CONCEPTUEL FINAL.....	55
<b>CHAPITRE 3. APPROCHE MÉTHODOLOGIQUE .....</b>	<b>56</b>
3.1. POSITIONNEMENT DE LA RECHERCHE.....	56

3.2.	PERSPECTIVES DE LA RECHERCHE ET THÉORIES MOBILISÉES.....	56
3.3.	APPROCHE DE LA RECHERCHE.....	57
3.4.	DESIGN DE RECHERCHE ENVISAGÉ : RECHERCHE MIXTE.....	57
3.5.	MÉTHODOLOGIES MOBILISÉES .....	58
3.6.	TECHNIQUE, PLAN ET PROCÉDURES DE COLLECTE ET DE TRAITEMENT DES DONNÉES .....	59
3.6.1.	<i>Techniques de collecte des données</i> .....	59
3.6.2.	<i>Plan de collecte de données</i> .....	60
3.6.3.	<i>Traitements, analyse et interprétation des données</i> .....	61
	<b>CHAPITRE 4. RÉSULTAT ET DISCUSSION.....</b>	<b>63</b>
4.1.	PRÉSENTATION DES RÉSULTATS.....	63
4.1.1.	<i>Présentation des résultats de l'objectif 1</i> .....	64
4.1.2.	<i>Présentation des résultats de l'objectif 2</i> .....	67
4.1.3.	<i>Présentation des résultats de l'objectif 3</i> .....	71
4.2.	DISCUSSION DES RÉSULTATS .....	72
4.2.1.	<i>Interprétation des résultats de l'objectif 1</i> .....	72
4.2.2.	<i>Interprétation des résultats de l'objectif 2</i> .....	79
4.2.3.	<i>Interprétation des résultats de l'objectif 3</i> .....	84
	<b>CHAPITRE 5. CONCLUSION ET CONTRIBUTION.....</b>	<b>86</b>
5.1.	SYNTHÈSE DES RÉSULTATS .....	86
5.2.	CONTRIBUTIONS ET RECOMMANDATIONS .....	87
5.3.	LIMITATIONS .....	88
5.4.	RECHERCHES FUTURES .....	88
	<b>RÉFÉRENCES .....</b>	<b>90</b>
	<b>ANNEXE 1 : QUESTIONNAIRE DE RECHERCHE.....</b>	<b>94</b>

## LISTE DES TABLEAUX

TABLEAU 1: TABLEAU DE CLASSIFICATION DES GRANDES ORGANISATIONS EN DÉVELOPPEMENT LOGICIEL EN FONCTION DE LEUR REVENU EN 2021 .....	9
TABLEAU 2: TABLEAU DES OBJECTIFS ET QUESTIONS RECHERCHES .....	17
TABLEAU 3: TYPE DE PERSONNALITÉS .....	39
TABLEAU 4 : ANALYSE DE L'INFLUENCE DE TYPE PERSONNALITÉ SUR LES MEMBRES D'ÉQUIPE .....	41
TABLEAU 5: DIMENSIONS ET TYPES DE PERSONNALITÉS .....	42
TABLEAU 6 : ÉCHELLE DE RÉPONSES SECTION1 .....	64
TABLEAU 7 : LES DONNÉES SUR LA RELATION ENTRE FACTEURS CLÉS DE SUCCÈS ET L'EFFICACITÉ .....	65
TABLEAU 8: LES DONNÉES SUR LA RELATION ENTRE FACTEURS CLÉS DE SUCCÈS ET L'EXACTITUDE .....	65
TABLEAU 9: LES DONNÉES SUR LA RELATION ENTRE FACTEURS CLÉS DE SUCCÈS ET LA ROBUSTESSE .....	66
TABLEAU 10: LES DONNÉES SUR LA RELATION ENTRE FACTEURS CLÉS DE SUCCÈS ET LA CONVIVIALITÉ .....	66
TABLEAU 11: LES DONNÉES SUR LA RELATION ENTRE FACTEURS CLÉS DE SUCCÈS ET LA MAINTENABILITÉ .....	67
TABLEAU 12: LES DONNÉES SUR LA RELATION ENTRE FACTEURS CLÉS DE SUCCÈS ET LA RÉUTILISABILITÉ .....	67
TABLEAU 13: COMBINAISON DES DONNÉES DE LA SECTION 2 ET 3 .....	68
TABLEAU 14: LES DONNÉES DU DEGRÉ DE MODÉRATION DE LA RELATION FACTEURS CLÉS DE SUCCÈS ET LA PERFORMANCE PAR LA COLLABORATION CLIENT .....	69
TABLEAU 15: LES DONNÉES DU DEGRÉ DE MODÉRATION DE LA RELATION FACTEURS CLÉS DE SUCCÈS ET LA PERFORMANCE PAR UNE ÉQUIPE CONSTITUÉE MAJORITAIREMENT DE PERSONNALITÉS EXTRAVERTIES .....	69
TABLEAU 16: LES DONNÉES DU DEGRÉ DE MODÉRATION DE LA RELATION FACTEURS CLÉS DE SUCCÈS ET LA PERFORMANCE EN PRIVILÉGIANT LE DÉVELOPPEMENT ITÉRATIF .....	70

TABLEAU 17: LES DONNÉES DU DEGRÉ DE MODÉRATION DE LA RELATION FACTEURS CLÉS DE SUCCÈS ET LA PERFORMANCE EN FAVORISANT LES CHANGEMENTS.....	70
TABLEAU 18: LES DONNÉES DU DEGRÉ DE MODÉRATION DE LA RELATION FACTEURS CLÉS DE SUCCÈS ET LA PERFORMANCE EN PRIORISANT LA SATISFACTION DU CLIENT.....	71
TABLEAU 19: LES DONNÉES DU DEGRÉ DE MODÉRATION DE LA RELATION FACTEURS CLÉS DE SUCCÈS PAR L'ASSURANCE QUALITÉ.....	72

## LISTE DES FIGURES

FIGURE 1: CADRE CONCEPTUEL PRÉLIMINAIRE .....	15
FIGURE 2: DIAGRAMME DE VENN SITUANT LE CHAMP DE RECHERCHE .....	16
FIGURE 3: LES VALEURS DU MANIFESTE AGILE .....	32
FIGURE 4 : STRUCTURE DES CONTRATS AVEC LA MÉTHODE TRADITIONNELLE ET LA MÉTHODE AGILE .....	36
FIGURE 5 : GESTION DE CHANGEMENT .....	37
FIGURE 6 : QUALITÉS INTERNES ET QUALITÉS EXTERNES D'UN LOGICIEL .....	51
FIGURE 7: CADRE CONCEPTUEL FINAL .....	55
FIGURE 8 : PROCESSUS DE TRAITEMENT ET D'ANALYSE DES DONNÉES.....	61
FIGURE 9 : L'EFFICACITÉ DES LOGICIELS .....	74
FIGURE 10: EXACTITUDE D'UN LOGICIEL .....	75
FIGURE 11: ROBUSTESSE D'UN LOGICIEL .....	76
FIGURE 12: CONVIVIALITÉ D'UN LOGICIEL .....	77
FIGURE 13: MAINTENABILITÉ D'UN LOGICIEL .....	78
FIGURE 14: RÉUTILISABILITÉ D'UN LOGICIEL .....	79
FIGURE 15: COLLABORATION CLIENT .....	80
FIGURE 16: ÉQUIPE CONSTITUÉE DE PERSONNALITÉS EXTRAVERTIES .....	81
FIGURE 17: DÉVELOPPEMENT ITÉRATIF .....	82
FIGURE 18: FAVORABLE AUX CHANGEMENTS .....	83
FIGURE 19: AXÉ SUR LA SATISFACTION DU CLIENT .....	84
FIGURE 20: ASSURANCE QUALITÉ .....	85

## REMERCIEMENT

C'est avec un réel plaisir que je rends grâce à Dieu Tout Puissant, créateur du Ciel et de la Terre, qui a permis de rendre ce travail de recherche possible. Je remercie avec gratitude toutes les personnes de près et de loin qui m'ont soutenu, encouragé et accompagné du début de mon programme jusqu'à la fin.

Le travail de recherche, nécessite beaucoup du temps, de rigueur, de discipline et surtout du courage. J'avoue que ça n'a pas été une tâche facile, mais grâce à la contribution de tous, par vos prières, encouragements et soutiens, le résultat est là!

J'aimerais remercier premièrement et spécialement mon grand frère bien-aimé Wahabou Gmakouba qui n'a jamais cessé de croire en moi. Par tes soutiens moraux, financiers et de tout genre, et par la bénédiction du Seigneur, j'ai pu commencer et terminer ce merveilleux programme de maîtrise en gestion de projet.

Je remercie toute la famille Gmakouba : mes parents, mon frère Oukpane et sa famille ainsi que ma sœur Nana sans oublier ma tante Bewa. Ils ont aussi été présents et ne cessent de me motiver à aller de l'avant.

J'ai eu la chance d'avoir des amis(es) de proche et de loin qui n'ont cessé de me soutenir durant mes recherches. Merci beaucoup, Chimène, Moïse, Yatou...

Mes remerciements vont également à mes familles Tim Hortons et Loginnove qui ont été présentes quand j'ai eu besoin d'elles. Vous avez écrit des chapitres dans mon livret de vie.

Durant tout le projet de recherche, j'ai été suivi et accompagné par mon directeur de recherche, M. Alencar Bravo. Merci monsieur, pour tout le temps investi dans ce projet de recherche.

En bref, je remercie toutes les personnes qui ont été présentes pour moi pendant toute la durée de mes études. La liste est longue et constituée de belles personnes que je ne peux tout nommer.

MERCI !!!

## CHAPITRE 1: PARTIE INTRODUCTIVE

### 1.1.CONTEXTE DE L'ÉTUDE

La gestion de projets de développement logiciel est la discipline consistant à évaluer les caractéristiques du logiciel à développer, à choisir le cycle de vie de développement logiciel le mieux adapté, l'approche de gestion de projet appropriée pour garantir la satisfaction des besoins du client afin de fournir une valeur commerciale aussi efficace et efficiente que possible (Wysocki, 2006).

L'industrie des technologies de l'information (TI) est en croissance et en expansion constantes, obligeant les personnes qui travaillent dans ce domaine à améliorer leur flux de travail et les méthodes de livraison de logiciels (Fagarasan, Popa, Pisla, & Cristea, 2021).

Le développement de logiciel est utilisé dans tous les domaines et organisations au monde, que ce soit dans le domaine agricole, le secteur manufacturier, éducatif et même en médecine, les logiciels sont devenus incontournables.

Le développement de logiciels est sollicité par les organisations sous deux aspects : les producteurs et les exploitants.

Pour certaines entreprises, les progiciels représentent leur pain et leur beurre (Reifer, 2013) : ce sont les producteurs de logiciels et c'est de ça que vit l'entreprise. Le développement de logiciel constitue leur activité de base. Ils ont pour fonction d'analyser, évaluer les besoins de leurs clients et concevoir des logiciels adaptés pour répondre aux besoins de ces derniers. Leurs revenus proviennent de la vente de leurs produits et services.

Nous pouvons citer par exemple IBM, CISCO, Oracle, SAP, VMware qui sont de grandes entreprises informatiques sur le plan mondial.

*Tableau 1: Tableau de classification des grandes organisations en développement logiciel en fonction de leur revenu en 2021*

<b>Rank</b>	<b>Name</b>	<b>Sales(B\$)</b>	<b>Profit(B\$)</b>	<b>Market Value (B\$)</b>	<b>Country</b>
<b>1</b>	Microsoft	153.3	51.3	1.966.6	United States
<b>2</b>	Dell Technologies	94.3	3.3	77.4	United States
<b>3</b>	IBM	73.6	5.6	119.4	United States
<b>4</b>	Cisco Systems	48	10.1	222.9	United States
<b>5</b>	Oracle	39.7	12.8	227.7	United States
<b>6</b>	SAP	31.7	6.3	164.2	Germany
<b>7</b>	Salesforce	21.3	4.1	213.6	United States
<b>8</b>	Adobe	13.7	5.6	251	United States
<b>9</b>	VMware	11.8	2.1	68.8	United States
<b>10</b>	Intuit	7.7	1.7	113.9	United States

(Global 2000, 2021)

Pour d'autres, les logiciels font partie intégrante de l'infrastructure des technologies de l'information que les institutions industrielles, gouvernementales et universitaires utilisent pour mener leurs activités (Reifer, 2013). Ce sont des exploitants de logiciels qui sont alors des outils de travail pour ces entreprises. Ils permettent d'effectuer des tâches administratives, de faire des traitements automatisés, la synchronisation des données, l'assistance à la clientèle, la gestion des paies, la gestion des ressources humaines, le marketing, la vente et beaucoup d'autres

tâches. Grâce aux logiciels, le travail à distance est devenu possible et a fait ses preuves surtout pendant la période de pandémie de Covid 19 où la distanciation physique est très souhaitée.

Le développement de logiciel apporte une contribution dans le développement de ces organisations. Que ce soit sur le plan économique, organisationnel ou technique, il permet d'accroître leurs performances.

Dans les usines par exemple, il est maintenant possible de créer une usine intelligente où l'Internet, des capteurs sans fil, des logiciels et d'autres technologies de pointe sont utilisés de concert afin d'optimiser la production et d'améliorer la satisfaction des clients cirano (2020). Une révolution où dans les usines on note principalement une implantation des systèmes de gestion ERP, divers logiciels ou encore des robots ou des équipements à commande numérique.

Sur divers plans et manières, les TI en général et les logiciels en particulier apportent de la valeur dans les organisations. Avec leurs fonctions et applications, ils permettent aux entreprises d'effectuer diverses tâches de manière rapide, sécuritaire et automatisée. Grâce aux bases de données, il est plus simple et rapide de stocker, analyser, traiter et partager des informations, ce qui renforce la qualité du système d'information dans les organisations.

Sur le plan Marketing, les logiciels jouent un rôle important dans les organisations. Face à un marché dynamique et plein de concurrents, les projets logiciels permettent aux organisations de s'adapter et de se démarquer. Ces dernières s'imposent sur le marché grâce aux succès et à la qualité de leurs produits issus du développement de logiciels et créent ainsi de l'avantage concurrentiel.

Face à ces avantages, les organisations ne cessent de se lancer dans des projets de développement de logiciel. Mais malheureusement nombreux sont ceux qui échouent.

D'où vient l'intérêt de notre sujet de recherche qui a pour but d'étudier les facteurs clés de succès des projets logiciels et de déterminer la meilleure méthode permettant de garantir ou d'accroître la qualité dans les projets logiciels.

## 1.2.PROBLÉMATIQUE GÉNÉRALE

En projet logiciel, la portée, le calendrier, le budget, la qualité ou les exigences changent fréquemment. L'impact de ces changements sur d'autres tâches et sur le projet dans son ensemble peut aller de négligeable à important Snedaker and Hoenig (2005). Ils sont présentés au début comme de petits changements, en suite s'enchaînent d'autres. Les exigences des projets TI et surtout en développement de Logiciel changent tout au long du projet. Parfois les objectifs mal définis au début du projet, les exigences des clients pas claires ou incomplètes causent des changements en cours du projet, entraînant des complications voire l'échec du projet. L'adaptation des nouvelles exigences ou des modifications au projet devient un grand défi au management du projet. Ces changements perturbent parfois d'autres activités ou encore d'autres fonctionnalités entraînant leurs retards ou leurs complications.

D'après Project Management Institute (2008), la portée est la somme des produits, services et résultats produits dans un projet. Au cours du projet, il arrive que sa portée mentionnant l'étendue du livrable attendu et le travail nécessaire pour l'accomplir, subisse trop de changements non autorisés menant le projet à un niveau de complexité élevée et par conséquent rend sa gestion compliquée : C'est le *Scope Creep* ou dérive de la portée.

Le *Scope Creep* est à la base d'échecs de plusieurs projets qu'elle suscite des questionnements au niveau des organisations. Le Project Management Institute (2008) définit cinq (5) causes de la dérive de la portée des projets. Elles nécessitent la responsabilité des sponsors de projet, l'équipe du projet, le gestionnaire et les autres parties prenantes du projet. Il s'agit de la définition ambiguë ou non raffinée de la portée, l'absence de gestion formelle de la portée ou des exigences, processus incohérent de collecte des exigences en matière de produits, du manque de parrainage

et de participation des parties prenantes et de la durée du projet Project Management Institute (2008).

La portée de haut niveau est mieux documentée dans une charte de projet et élargie dans l'énoncé de la portée (Project Management Institute, 2009). Mais souvent insuffisante, car elle ne permet pas de se faire bien comprendre. Quand l'équipe projet n'a pas suffisamment de détails pour comprendre la portée du projet, des confusions se créent et finalement elle est ignorée ou mal interprétée.

Parfois, plusieurs changements sur le projet se font sans suivre un processus formel de gestion du changement : soit parce que les cadres supérieurs au niveau des sponsors n'accordent pas trop d'importance au projet, ce qui permet à l'équipe projet d'agir sans attendre une décision formelle. Il arrive aussi que l'équipe projet juge certains changements trop petits et se permette de les gérer sans qu'ils soient approuvés par la haute direction. L'équipe travaille alors sur des exigences qui n'étaient pas prévues dans le délai et le budget du projet. Par conséquent on a un dépassement de budget et de temps du projet.

La dérive de portée survient aussi lorsque les exigences sont mal recueillies. Les promoteurs s'attendent à ce que les équipes projet suivent une méthode cohérente de collecte des exigences et ne pensent pas beaucoup à la sollicitation des exigences (Project Management Institute, 2009). Si le processus de collecte des exigences n'est pas bien clarifié, on a une augmentation de parties prenantes. Chaque partie prenante donne ses besoins et si ces derniers ne sont pas inscrits dans la portée du projet, on aboutira alors à une dérive de la portée.

La dérive de la portée dans toutes ses formes est à éviter dans les projets à cause de ses innombrables conséquences. On passe beaucoup de temps à travailler sur des exigences supplémentaires qui ne sont parfois pas adaptées au projet, en plus nécessitent plus de temps et donc des coûts supplémentaires au projet. Par conséquent, on obtient un dépassement en termes de coût et de délai. Ce qui est à la base de nombreux échecs dans les projets de développement de logiciel. La dérive de la portée occasionne plusieurs risques. Il peut s'agir d'une augmentation des coûts

de production ou du développement de logiciels de mauvaise qualité ou d'une incapacité à terminer le projet à temps (Komal et al., 2020).

Outre que la dérive de la portée, dans le développement de logiciels, les risques sont partout. Les systèmes externes représentent un risque. Un code non maintenable est un risque. Avoir un nombre élevé de défauts est un risque. Même le choix de la technologie comporte un risque (Chowdhury & Arefeen, 2011).

À ce niveau il faut préciser que l'approche de gestion des projets logiciels qui se faisait en mode traditionnel (*waterfall*) ne correspondait pas aux livrables attendus des projets de développement logiciel. C'est alors que la *Méthodologie agile* semble une option rationnelle plus adaptée à l'environnement des projets logiciels Fagarasan et al. (2021).

### 1.3.PROBLÉMATIQUE SPÉCIFIQUE

La gestion de la qualité n'est pas un aspect nouveau tant en gestion de projets logiciels que dans tout type de projets. La preuve, de nombreuses recherches ont essayé de trouver des moyens, outils et solutions pour garantir la qualité en projet logiciel. Mais il y a encore des notions qui ne sont toujours pas claires ou des moyens peu efficaces.

La définition même de la qualité est un sujet divergent entre les auteurs. De nombreuses initiatives de la qualité n'atteignent pas toujours leurs objectifs ou ne réussissent que partiellement. Cet échec résulte peut-être de la focalisation étroite de nombreux programmes de qualité et d'un manque conséquent d'étendues dans la compréhension du véritable rôle et de la signification de la qualité dans l'efficacité organisationnelle Beckford (2002). Pour chaque type de projets, nous avons des définitions de la qualité qui diffèrent.

Le management de projets étant étroitement lié au développement de projet logiciel, il est alors important d'étudier les facteurs qui peuvent dans la mesure du possible amener les projets logiciels vers le succès et comment garantir leur qualité.

L'approche de gestion de projets agile et les diverses méthodes agiles apparaissent plus adaptées aux innombrables défis de l'environnement de développement de logiciels.

L'intégration de l'assurance qualité (AQ) dans une gestion de projet logiciel au moyen d'une approche de méthodologie agile serait alors une opportunité pour mener efficacement les projets de développement logiciel vers le succès et garantir leurs qualités. Par conséquent, il en résulterait de meilleures performances pour les organisations, les permettant de s'imposer sur le marché concurrentiel.

L'objet de notre recherche est d'abord d'apporter une meilleure compréhension de certaines notions comme la qualité, l'agilité, l'assurance qualité, l'environnement des projets logiciels en mettant en évidence les facteurs clés de succès et trouver les meilleurs méthodologies et moyens pour mettre en relations ces notions afin de les adapter à l'environnement de la gestion de projet logiciel.

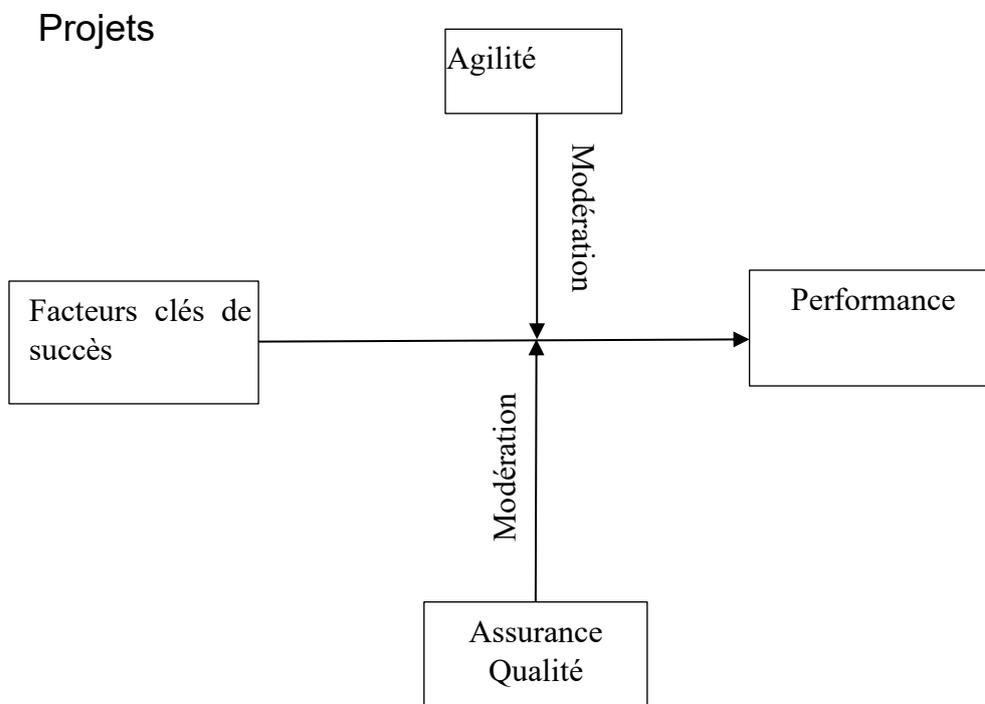
Dans notre revue de littératures, nous identifierons les facteurs clés de succès en gestion de projets logiciel, essayerons de définir la qualité, l'assurance qualité, l'agilité. Nous étudierons en suite les relations entre nos aspects énumérés et leur rôle modérateur dans la relation entre les facteurs clés de succès et la performance.

#### **Problématique spécifique :**

**Répondant à ce qui précède, il est important d'identifier premièrement les facteurs clés de succès en développement de logiciel et étudier les rôles modérateurs de l'agilité et l'assurance qualité sur les facteurs clés de succès pour aboutir à la performance des logiciels.**

Le cadre conceptuel préliminaire de cette recherche ci-dessous permet de visualiser les diverses relations et interactions entre les facteurs de l'étude.

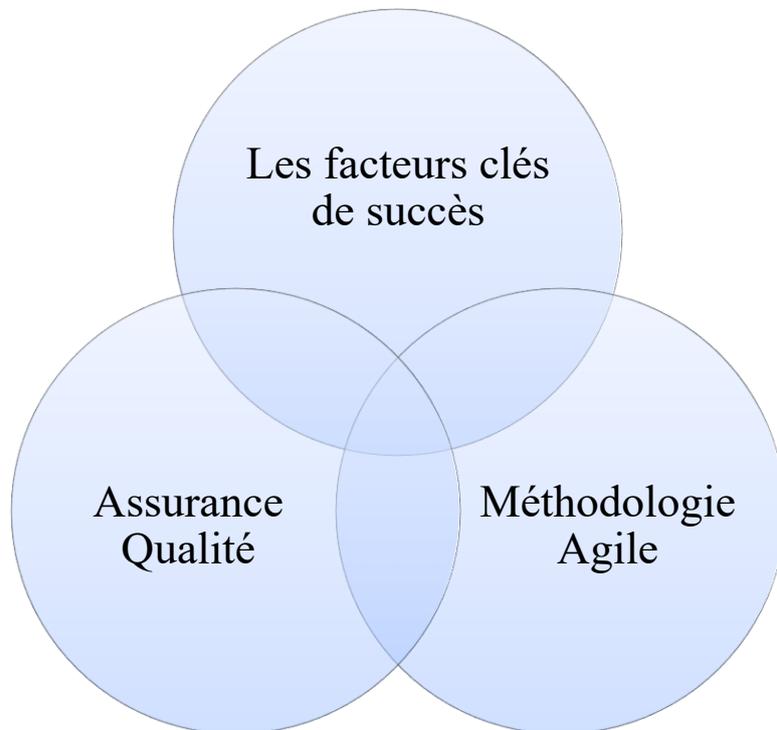
*Figure 1: Cadre conceptuel préliminaire*



#### 1.4.LOCALISATION DE LA RECHERCHE

Pour mener à bien cette étude, il est d'abord intéressant et important de bien comprendre l'environnement des projets logiciels et d'étudier l'interaction des facteurs illustrés par notre diagramme de Venn : l'assurance qualité et la méthodologie agile. Nous analyserons comment l'application de l'assurance qualité dans la gestion de projets logiciels gérée au moyen d'une méthodologie agile peut conduire à un livrable de qualité. Ensuite, nous verrons comment nous pouvons aboutir à une meilleure performance des logiciels.

*Figure 2: Diagramme de Venn situant le champ de recherche*



## 1.5.OBJECTIFS DE LA RECHERCHE ET QUESTIONS DE RECHERCHE

**Tableau 2: Tableau des objectifs et questions recherches**

Objectifs	Questions de recherches
<b>Identifier et définir les facteurs / concepts / notions suivants :</b>	
1. Indiquer les facteurs clés de succès des projets logiciels	QR1.1: Quels sont les facteurs clés de succès ? QR1.2: Quels sont les facteurs clés de succès des projets logiciels
2. Comprendre l'agilité	QR2.1: Qu'est-ce que l'agilité ? QR2.2: En quoi l'agilité est-elle favorable aux projets logiciels ?
3. Identifier l'assurance qualité dans les projets logiciels	QR3.1: Qu'est-ce que la qualité ? QR3.2: Qu'est-ce que l'assurance qualité ? QR3.3: Qu'est-ce que l'assurance qualité en projets logiciels ?
4. Définir un logiciel de qualité	QR4.1: Qu'est-ce qu'un logiciel de qualité ?
5. Identifier la performance dans les projets logiciels	QR6.1: Quels sont les indicateurs de performance ? QR6.2: Quels sont les indicateurs de performance dans les projets logiciels ?

<p><b>Analyser et comprendre les relations entre les facteurs/ concepts / notions suivants :</b></p>	
<p>OB1. Établir les relations entre les facteurs clés de succès des projets logiciels et la performance</p>	<p>QR7: Comment les facteurs clés de succès des projets logiciels peuvent conduire à une performance ?</p>
<p>OB2. Établir l'influence de l'agilité sur la relation entre facteurs clés de succès et performance</p>	<p>QR8: Comment l'agilité modère la relation entre facteurs clés de succès et performance des projets logiciels ?</p>
<p>OB3. Établir l'influence de l'Assurance Qualité sur la relation facteurs clés de succès et performance.</p>	<p>QR9: Comment l'Assurance Qualité modère la relation facteurs clés de succès et performance des projets logiciels ?</p>

## REVUE DE LITTÉRATURE

### 1.6. LES FACTEURS CLES DE SUCCÈS

Le terme de facteurs clés de succès ou FCS est utilisé pour désigner les éléments clés permettant d'atteindre un objectif ou un but déterminé.

Ce sont des éléments pertinents aux projets que l'on doit maîtriser pour réussir son projet et se positionner sur le marché. Ils permettent de mettre toutes les chances de son côté pour atteindre les objectifs fixés et être plus compétitif. C'est vrai que pour réussir à se distinguer de la concurrence sur le marché, il faut proposer des produits ou stratégies plus performants. Avec les facteurs clés de succès bien étudié et bien maîtrisé au début du projet, il est plus facile de détecter des avantages concurrentiels.

Pour chaque type de projet ou chaque domaine, il est important d'étudier, comprendre et maîtriser les facteurs clés de succès avant le début du projet. Cela implique les outils, produits, stratégies, technologies à utiliser tout au long du projet. D'après Chowdhury and Arefeen (2011), pour diverses raisons, y compris l'influence des précédentes directives de gestion de logiciels basées sur les documents, les projets se concentrent sur des activités qui ne sont pas essentielles à leur succès. Ils poursuivent en disant : « La chose la plus importante à faire pour un projet logiciel est de se concentrer sur ses facteurs critiques de réussite ».

En développement de logiciel, les facteurs clés de succès sont aussi importants comme dans tous les domaines. Ils sont aussi nombreux et leur pertinence varie selon le type de livrable. Pour cette recherche nous en étudierons quelques-uns, qualifiés de plus importants et requis pour avoir un logiciel de qualité et accroître leurs performances.

### 1.6.1. Objectifs du projet bien clair

L'objectif du projet correspond à la description du résultat ou de la solution qu'on souhaite obtenir. Il peut s'agir des livrables ou des ressources, mais aussi d'objectifs plus intangibles, par exemple une amélioration d'un processus.

L'établissement d'objectifs pour un projet décrit les effets et les résultats à long terme attendus du projet. Il est important de décrire ces objectifs le plus tôt possible, car ils peuvent éclairer chaque décision dans le cycle de vie du projet. Les produits à succès avaient une définition beaucoup plus précise avant leur développement. Les projets dotés de définitions aussi précises avaient 3,3 fois plus de chance de réussir, avec un taux de réussite de 85,4 % (Cooper & Kleinschmidt, 2011).

Pour bien établir les objectifs d'un projet, l'économiste George Doran (1981), invente la méthode SMART dans son article « *There's a SMART way to write management's goals and objectives* ».

Tout d'abord, la méthode SMART veut dire *Specific, Measurable, Assignable, Realistic and Timely*. En français : spécifique, mesurable, assignable, réaliste et temporel.

D'après l'auteur, l'objectif du projet doit être SMART:

#### •Spécifique

Un objectif spécifique définit clairement le résultat attendu et doit être simple à comprendre. Il doit être précis par rapport à ce qui doit être fait, par qui et comment. L'équipe projet et toutes les autres personnes intervenant sur le projet doivent comprendre facilement.

Un énoncé d'objectif vague est difficile à l'atteindre, car il sera difficile de définir le succès.

- **Mesurable**

Un objectif mesurable est un objectif quantifiable. La quantification de l'objectif permet de savoir si l'objectif est atteint. La mesure de l'objectif peut prendre différentes formes. Le seuil à atteindre peut notamment correspondre à une quantité ou une qualité.

Un objectif mesurable permet aussi de savoir le pourcentage de réussite du projet.

- **Assignable**

Un objectif assignable indique clairement qui doit faire quoi. Cela spécifie la responsabilité de chaque partie.

- **Réaliste**

Compte tenu des ressources disponibles, indiquer les résultats atteignables de manière réaliste.

- **Temporel**

Pour réussir votre objectif, vous devez l'atteindre dans le laps de temps que vous lui avez accordé. Un objectif SMART définit un délai de réalisation. Le délai doit être précis et raisonnable.

Un objectif bien clair depuis le début du projet apporte une valeur ajoutée au projet, aide à répartir la charge de travail plus équitablement et permet d'évaluer le progrès du projet en cours de route. L'un des avantages de l'objectif du projet est aussi la gestion efficace du temps. Ce qui peut réduire le travail et les ressources inutiles pour l'équipe projet. L'objectif du projet permet aussi de clarifier le succès et de l'évaluer à la fin du projet.

### **1.6.2. Des exigences bien claires**

Les exigences du client sont des particularités ou des précisions sur le livrable ou le service qu'on souhaite obtenir. C'est par les exigences du projet que le client clarifie ses points essentiels. En développement de logiciel on peut parler des fonctionnalités du logiciel et ses performances. D'après Project Management Institute (2021), une exigence est « une condition ou une capacité qui est nécessaire pour être présente dans un produit, un service, ou un résultat pour satisfaire un besoin commercial ». Cooper and Kleinschmidt (2011) citent la compréhension des exigences comme premier facteur clé de succès. Parfois, les exigences sont incomplètes. Une exigence incomplète est sujette à des interprétations différentes, tandis qu'une exigence testable doit être complète, cohérente et sans ambiguïté (Blackburn, Busser, & Nauman, 2001).

La complexité des exigences représente les défis liés à la compréhension de ce que nous devrions construire. Nous connaissons tous des situations dans lesquelles les parties prenantes ont du mal à fournir les détails nécessaires sur ce qui est nécessaire ou ne parviennent pas à s'entendre sur les exigences (Durham & Michel, 2021).

Souvent, cette incapacité à fournir les détails est due au fait que le client ou les parties prenantes ont très peu de connaissances en développement de logiciel et ne savent pas quelles sont les informations nécessaires à fournir. Aussi, les parties prenantes ont-elles souvent tendance à croire que l'équipe projet comprend facilement leurs besoins et donc ils n'ont pas besoin de fournir plus de détails, or cela n'est pas toujours le cas. Dans certaines situations, le client ne sait même pas ce qu'il veut ou n'a aucune idée de comment cela peut se faire. Il sait qu'il a un problème, mais n'a aucune idée de la solution appropriée. Pour remédier à ces manques d'informations, il est préférable de collaborer avec le client tout au long du projet : prendre le temps de comprendre le problème du client, lui poser des questions et ne pas se contenter des informations fournies au début. Une collaboration plus étroite avec le client et des discussions plus en profondeur sur ses exigences permettent non

seulement de mieux les comprendre, mais aussi de proposer des solutions plus adaptées. Le client est alors plus impliqué et devient une ressource pour le projet.

Des exigences du client plus claires permettent une meilleure compréhension, une réduction du nombre de nouvelles demandes ou des demandes de changement et d'éviter des erreurs de jugement en prenant de bonnes décisions.

L'International Council On Systems Engineering (INCOSE) explique le processus de gestion des exigences en termes d'activités suivantes :

Identifier et documenter :

- Comprendre pourquoi le produit ou le système final doit exister en termes de valeur fournie à l'utilisateur final.
- Obtenir et documenter les exigences du client et des parties prenantes.
- Obtenir leur accord sur les déclarations écrites et sur les mesures de réussite.
- Veiller à ce que les exigences réglementaires soient comprises par toutes les parties.

Analyser et allouer :

- Déduire des exigences mesurables, explicites, utiles pour la conception qui sont directement traçables pour fournir la valeur du système dans le cadre de contraintes (par exemple, coûts, contraintes réglementaires et sociales).
- Continuer la dérivation et la décomposition de telle sorte que les exigences dérivées puissent être attribuées au système et aux composants.
- Relier les exigences et leurs bases de support pour assurer une traçabilité bidirectionnelle.

- Démontrer la validité des exigences par le biais d'une analyse, d'une autorité ou d'une autre démonstration de besoin.

- Les exigences valides sont celles qui sont nécessaires, claires, réalisables et vérifiables.

#### Changement de contrôle :

- Analyser les modifications apportées aux exigences pour en déterminer l'impact total et approuver les modifications par un processus prescrit et des niveaux d'autorité appropriés.

- Comprendre l'impact avant de s'engager dans le changement. Assurez-vous que les modifications approuvées sont propagées à tous les éléments concernés.

#### Vérifier la conformité :

- Vérifier que les exigences sont respectées et que le système résultant fournit réellement la valeur nécessaire dans l'environnement réel d'utilisation.

- Déterminer les conditions de test et les critères d'acceptation lors de la conception pour permettre la planification des tests.

#### Assurer la surveillance :

- Gérer les informations qui en résultent et les processus mis en œuvre. La profondeur des exigences gérées par un programme doit être définie. Habituellement, il suffit de gérer au niveau des sous-traitances ou des articles achetés.

- Développer des métriques utiles pour montrer la qualité des activités de gestion des exigences. Intégrez les exigences dans tous les systèmes pour garantir une solution équilibrée et des interfaces compatibles.

Ces activités font partie de la pratique d'ingénierie des systèmes basée sur des modèles prônée par INCOSE. (Phillip A. Laplante, 2022).

### 1.6.3. Les risques bien identifiés

Un risque est « L'impact combiné de la probabilité qu'un événement se produise et des conséquences de cet événement lorsqu'il se produit »(IEEE, 2023). En gestion de projets logiciels, les risques que l'on rencontre fréquemment sont généralement liés au calendrier du projet, au coût, et à la qualité. Ces risques compromettent la réussite du projet et aussi la qualité et performance du logiciel développé.

Le risque est un facteur important influençant la réussite du projet et la qualité du livrable. Ainsi est-il important de bien identifier les risques du projet dès la phase de planification. Cela nécessite une bonne pratique pour l'identification et la gestion des risques.

L'identification des risques dès le début du projet, permet de mettre en évidence tous les éventuels risques qui peuvent se produire tout au long du cycle de vie du projet. Après cette étape, une stratégie de gestion des risques doit être mise en place. D'après Weber, Medina-Oliva, Simon, and Iung (2012), l'analyse des risques est un modèle permettant de quantifier et d'évaluer l'occurrence d'un événement critique. Pour chaque risque identifié, il faut connaître la probabilité qu'elle se produise, mettre en place une solution pour soit l'éviter, soit l'atténuer ou prévoir une solution de rechange au cas où il se produirait.

Les risques bien identifiés dès le début du projet diminuent alors la possibilité qu'ils se produisent ou leurs impacts au cas où ils se produiraient. Avec un impact de risque minimisé, le projet a plus de chance d'être réussi avec un livrable plus performant.

L'identification des risques permet d'éviter ou de minimiser les dépassements en termes de délai et de coût.

#### 1.6.4. Attentes réalistes

Les attentes d'un projet diffèrent d'un besoin à un autre ou d'un client à un autre. Ce sont les attentes du client par rapport au projet. Ils sont formulés par le client et donnent une description de manière générale ou spécifique des objectifs du projet. Il est important de développer un logiciel de qualité capable de satisfaire les besoins des utilisateurs sans erreurs (Hussain, Mkpojiogu, & Kamal, 2016).

En mode traditionnel, les attentes sont souvent décrites en détail dans un cahier de charge du projet et constituent un plan à suivre pour la réalisation du projet. Cependant, en mode agile où l'on est plus favorable au changement pendant le cycle de vie du projet, les attentes sont souvent formulées globalement et non en détail. On part d'une vision plus globale de la solution voulue.

Les attentes, que ce soit en mode agile ou traditionnel, sont parfois mal exprimées ou exagérées. Au cours du projet, on se rend compte que certaines attentes ne sont pas réalisables. C'est-à-dire qu'ils ne sont tout simplement pas faisables techniquement. On se retrouve alors avec des bloquants, ou des retards ou pire encore un abandon du projet.

En projet de développement de logiciel, les attentes peuvent être liées aux fonctionnalités du logiciel, au design ou à diverses caractéristiques. Une attente doit être formulée bien sûr comme le client le souhaite, mais dans la limite des réalités. Il faut que ce soit faisable. Alors le chef de projet doit analyser tous les besoins du client en collaboration avec l'équipe des développeurs afin d'identifier les besoins non réalistes et discuter avec le client pour peut-être lui proposer une autre solution.

Les attentes réalistes permettent d'éviter trop de changements non contrôlés au cours du projet, éviter les dépassements de délai et de coûts pour finalement aboutir au succès du projet et à une meilleure qualité du livrable.

### 1.6.5. Responsabilisation

Un projet de développement de logiciel regroupe souvent un certain nombre de personnes que ce soit à l'interne ou à l'externe de l'organisation. Nous avons l'équipe projet constituée des développeurs, analystes, d'un gestionnaire de projet, des chefs d'équipes, des diverses parties prenantes que ce soit du côté client ou de la direction. Tous ces acteurs sont amenés à travailler ensemble pour un but commun qui est la réussite du projet.

La responsabilisation est d'après LAROUSSE (2001) , de 'rendre quelqu'un, un groupe conscient de ses responsabilités'. Elle consiste à responsabiliser chaque acteur ou chaque partie de ses tâches à accomplir tout au long du projet. Par exemple, le client à ses responsabilités, l'équipe projet et la direction aussi. Chaque partie est consciente de ses tâches, du délai alloué pour les remplir et est responsable du résultat.

D'après Chaurasia (2010), les principaux objectifs de la responsabilité sont :

- Organiser le projet avec des responsabilités distinctes et nommer une personne autorisée avec responsabilité.
- Organiser l'objectif et l'organisation des incitations de telle manière qu'ils se renforcent mutuellement pour atteindre cet objectif.
- Lorsque la personne responsable a atteint l'objectif et exécuté le module sans erreur, les défauts du logiciel sont minimisés et la fiabilité de ce dernier est améliorée.
- Après avoir assemblé, exécuté le module et obtenu une erreur, cela signifie une augmentation de la responsabilité dans la même proportion que la fiabilité du logiciel.

La responsabilisation permet d'éviter plusieurs problèmes auxquels font face plusieurs de projets :

- Faible intérêt :

Un groupe de partie prenante ayant un faible intérêt pour le projet réagit moins et n'est pas assez motivé pour travailler efficacement avec les autres. Une responsabilisation renforce l'intérêt des parties pour le projet, car chaque partie se voit importante et travaille efficacement pour accomplir convenablement ses tâches.

- Défaillance logicielle ou logiciel de moindre qualité

Une défaillance logicielle se produit si le comportement du logiciel diffère des performances réelles du logiciel. Ces échecs peuvent être minimisés grâce à la responsabilité de chaque module pour l'équipe de développement du projet et chaque membre impliqué dans le développement (Chaurasia, 2010).

- Retard du projet

Si les équipes internes et externes ne savent pas clairement qui est responsable des résultats globaux et qui est responsable de certaines tâches, l'avancement du projet et les résultats peuvent être gravement affectés. Avec une responsabilisation et une livraison des tâches convenablement, elles sont plus accomplies avec une meilleure qualité et le projet avance comme prévu.

- Conflits et rejet des fautes entre les parties :

Responsabiliser chaque partie peut éviter de rejeter des fautes sur les autres, ce qui entraîne des conflits.

Pour éviter les discordances entre les tâches du projet, il est important que les diverses parties prenantes se communiquent entre elles afin d'uniformiser leurs travaux. Ce besoin est surtout important au sein de l'équipe projet. Pour les projets logiciels, la plupart des tâches entre l'équipe développeur sont des fonctionnalités qui seront par la suite assemblées et il faut nécessairement une coordination pour que le logiciel soit fonctionnel. Par exemple, les nomenclatures des variables, fonctions

ou fichiers doivent être les mêmes. Une fonction créée par un groupe A et appelée par un groupe B sous un autre nom, ne fonctionnera pas.

#### **1.6.6. Personnel compétent**

La compétence souvent subdivisée en trois (3) catégories : le savoir, savoir-faire et savoir-être, est un facteur clé influençant la productivité d'une équipe projet. La réussite du projet dépend du travail de l'équipe. Alors, pour constituer une équipe projet, que ce soit le chef d'équipe, les analystes ou les développeurs, il est important de bien choisir les membres. Pour réussir, l'équipe doit disposer des personnes ayant les connaissances nécessaires, l'expérience requise et le savoir-faire nécessaire pour la réussite du projet.

En équipe, l'expérience et le savoir-faire sont importants, cependant le savoir-être l'est aussi. Le savoir-être est nécessaire pour garder une bonne ambiance dans l'équipe et éviter des conflits.

Il n'est pas toujours facile de trouver des personnes avec toutes les compétences requises, mais il est recommandé d'avoir une équipe avec toutes les compétences nécessaires réunies. Chaque membre a ses qualités personnelles qu'il peut partager avec les autres.

Une équipe constituée de personnes compétentes est plus productive. Il est de la responsabilité du chef de projet de bien connaître son équipe, de placer chaque membre à la bonne place et d'assurer une bonne coordination entre les membres.

L'équipe compétente a les ressources humaines nécessaires pour réaliser le projet conformément aux besoins du client et par son expérience, saura s'adapter aux éventuelles situations.

### **1.6.7. Soutien de la haute direction**

Le soutien de la haute direction se traduit par une mise à disposition des ressources matérielles, humaines, et des conseils à l'équipe projet. Une attention particulière au projet et des participations à certaines réunions de l'équipe projet.

Le soutien de la direction n'est pas toujours facile à obtenir. Plusieurs projets échouent parce qu'ils n'ont pas eu de soutien nécessaire de la part de la direction. Il est de la responsabilité du chef de projet d'œuvrer pour obtenir le soutien nécessaire. Vu son importance, le soutien de la haute direction est énuméré comme faisant partie des facteurs clés de succès en gestion de projet logiciel. D'après Young and Jordan (2008), le soutien de la haute direction n'est pas simplement l'un des nombreux facteurs clés de succès nécessaires à la réussite d'un projet, mais qu'il est le plus important.

Le soutien de la direction est un facteur important que souvent, plusieurs gestionnaires de projets ignorent. D'après les études de Keil, Cule, Lyytinen, and Schmidt (1998), un manque d'engagement de la direction envers le projet est l'un des trois facteurs à risque les plus importants.

Le manque de soutien de la part de la direction est parfois dû simplement à un intérêt faible au projet. Le chef de projet doit alors par ses compétences, gérer ce groupe de partie prenante (la haute Direction).

Pour ce faire, il est recommandé d'aligner les projets aux visions, missions, objectifs et valeurs de l'entreprise. Un projet qui s'aligne aux réalités de l'entreprise est plus susceptible d'intéresser la direction et d'obtenir son soutien.

Le chef de projet doit aussi convaincre la direction et les amener à s'intéresser au projet. Pour réussir cette mission, il doit organiser des réunions formelles ou informelles et détailler l'importance du projet pour l'entreprise.

Un projet soutenu par la direction encourage premièrement l'équipe projet. L'équipe aura toute aide matérielle et humaine pour bien mener leur mission. En cas

de difficulté, l'aide de la direction leur permettra de débloquent des situations difficiles et continuer convenablement le projet.

Le soutien de la haute direction est un facteur clé en projet logiciel qui peut conduire à la réussite du projet et à une meilleure qualité du livrable.

L'ensemble de tous ces facteurs de succès cités augmentent la réussite du projet ainsi que la performance du logiciel. Cependant, la méthodologie de gestion de projet utilisée joue aussi un rôle dans la réussite du projet. L'agilité est alors perçue comme un environnement favorable aux facteurs clés de succès et dans la mesure du possible pourrait jouer un rôle de modérateur pour atteindre la performance.

## 1.7.L'AGILITÉ

### 1.7.1. Qu'est-ce que l'Agilité ?

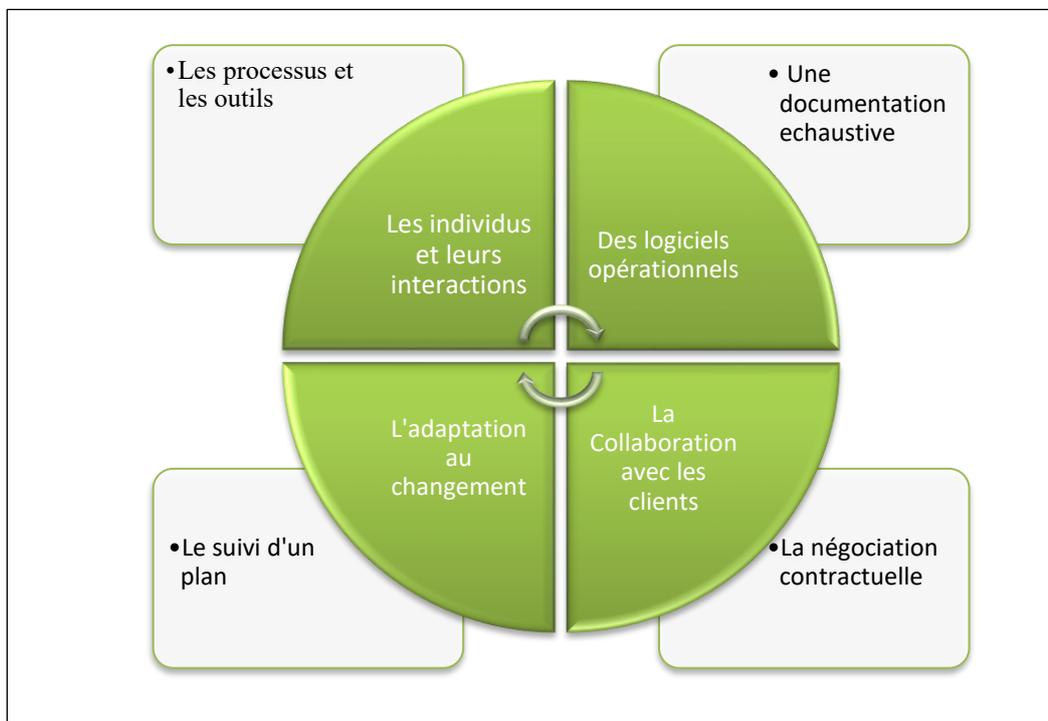
L'agilité est la capacité à la fois de créer et de répondre au changement afin de tirer profit d'un environnement commercial turbulent. L'agilité est la capacité d'équilibrer flexibilité et stabilité (Highsmith 2002).

Le terme agile que l'on emploie aujourd'hui est né d'une évolution successive de cadres d'idées depuis 1940. Measey (2015).

Ce n'est qu'en 2001, lorsque le Manifeste agile a été formulé, qu'un nom générique collectif et des termes de référence prenant en charge tous les cadres ont été définis, et agile en tant que concept est né (Measey, 2015). Œuvre de 17 experts en développement de logiciel, comme son nom l'indique " Manifeste pour le développement agile de logiciels " fut principalement créé pour le développement de logiciel, mais peut être utilisé pour développer plusieurs types de produits.

Le manifeste agile est composé de quatre (4) valeurs et douze (12) principes de base applicables aux projets de développement de logiciels ou d'autres types de produits.

**Figure 3: Les valeurs du manifeste agile**



Il est important de comprendre que les éléments en dehors du cercle ne sont pas à blâmer ou à négliger, mais ils sont seulement moins préférés.

Avec la première valeur par exemple, « les individus et leurs interactions plus que les processus et les outils », Measey (2015) explique qu'il est bien vrai que les outils apportent une valeur significative aux équipes de développement de logiciels et leur permettent d'être agiles, mais les meilleurs processus et outils ne les aideront pas à fournir de la valeur au client sans des personnes habilitées et motivées qui interagissent efficacement en équipe.

Ces quatre valeurs agiles sont soutenues par douze principes :

1. Notre priorité absolue est de satisfaire le client grâce à la livraison rapide et continue de logiciels de valeur.
2. Accueillir positivement les exigences changeantes, même tardivement dans le développement. Les processus agiles exploitent le changement pour l'avantage concurrentiel du client.

3. Fournir fréquemment des logiciels opérationnels, de quelques semaines à quelques mois, avec une préférence pour les délais plus courts.
4. Métiers et développeurs doivent travailler ensemble au quotidien tout au long du projet.
5. Réaliser des projets autour d'individus motivés, leur donner le soutien dont ils ont besoin, et leur faire confiance pour atteindre les objectifs fixés.
6. La méthode la plus efficace et efficiente pour transmettre des informations à une personne et au sein d'une équipe de développement est la communication en face à face.
7. Le logiciel opérationnel est la principale mesure de progrès
8. Les processus agiles favorisent le développement durable. Les sponsors, développeurs et utilisateurs doivent pouvoir maintenir un rythme constant indéfiniment.
9. Une attention continue à l'excellence technique et à une bonne conception améliore l'agilité.
10. L'art de maximiser la quantité de travail non fait - est essentiel.
11. Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées.
12. À intervalles réguliers, l'équipe réfléchit à la manière de devenir plus efficace, puis ajuste son comportement en conséquence.

Après le manifeste agile, plusieurs chercheurs ont aussi créé des outils toujours dans le but d'améliorer la gestion de projet en mode agile et d'avoir un meilleur rendement. C'est ainsi que des dérivés du manifeste agile ont vu le jour. On peut citer Le Manifeste du Software Craftsmanship et le Manifeste agile marketing qui sont aussi populaires.

Il y a eu aussi des propositions d'alternative au manifeste agile qui apparaissent plus actualisées et plus adaptées à l'agilité d'aujourd'hui. C'est l'exemple de Modern Agile créé par Kerievsky Joshua (2016) et qui propose des idées plus globales que le manifeste agile.

L'agile est composé de plusieurs méthodologies qui ont tous les mêmes cadres d'idée agile, mais qui se distinguent par leurs capacités à s'adapter aux divers types de projets. Alors une chose est de faire le choix d'une gestion de projet agile, mais aussi de choisir la méthodologie agile la plus adaptée à son projet. Par rapport à la méthode de gestion de projets traditionnels, les méthodes agiles impliquent au maximum le client et permettent une grande réactivité à ses demandes. Elles reposent sur un cycle de développement itératif, incrémental et adaptatif.

Les méthodes agiles les plus connues et souvent utilisées dans les organisations sont : Scrum, EXtreme Programming, Safe, Kanban et bien d'autres.

Le Scrum est une méthode agile basée sur un processus itératif et incrémental adapté à tous types de projets. Scrum, par rapport aux approches séquentielles traditionnelles, est une approche holistique, dans laquelle les phases se chevauchent fortement et l'ensemble du processus est exécuté par une équipe interfonctionnelle à travers les différentes phases (Dathan & Menon, 2010).

En Scrum, le projet est subdivisé en de périodes au cours desquelles l'équipe travaille sur des tâches assignées pour produire progressivement des livrables fonctionnels. Les rôles principaux dans Scrum sont le *Scrum Master* qui maintient les processus et travaille comme un chef de projet, le *Product Owner* qui représente les parties prenantes et l'équipe qui comprend les développeurs (Dathan & Menon, 2010).

Les Méthodes agiles ont révolutionné et changent l'approche de développement logiciel. Elle a aussi évolué grâce à plusieurs outils intégrés à l'approche de projets de développement logiciel. La démarche AQ garantit en certaines mesures la résolution d'éventuels problèmes.

## **1.7.2. Les indicateurs d'agilité**

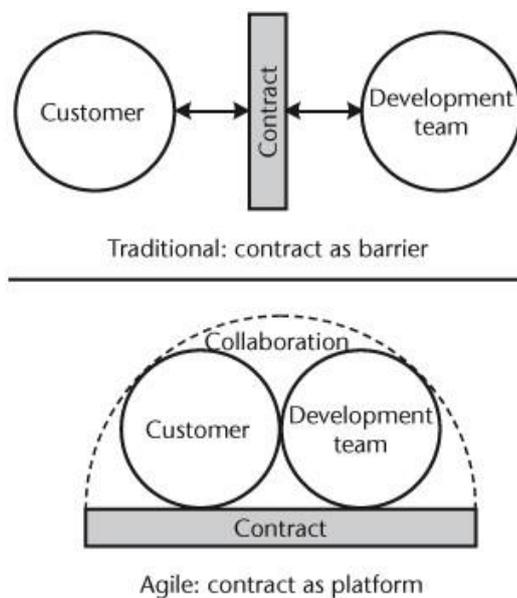
### *1.7.2.1. Collaboration client*

Il est très important d'impliquer le client dès le début du projet pour diverses raisons.

En gestion de projet traditionnel, la relation entre le client et l'équipe projet se traduit généralement par un contrat signé entre les deux parties. Ce contrat détaille les rôles et responsabilités de chaque partie et leur sert de garantie l'une envers l'autre.

Bien qu'elle reconnaisse l'importance du contrat et autre forme d'accord, la gestion de projet agile apporte une solution plus efficace. Il s'agit de la collaboration avec le client comme le montre la figure 4. Le développement logiciel agile ainsi que les exigences processus de gestion du changement (RCM), tous deux nécessitent une communication et une collaboration riches (Akbar, Shameem, Ahmad, Maqbool, & Abbas, 2018; Kamal et al., 2020; Matharu, Mishra, Singh, & Upadhyay, 2015).

**Figure 4 : Structure des contrats avec la méthode traditionnelle et la méthode agile**



*(Koch, 2005)*

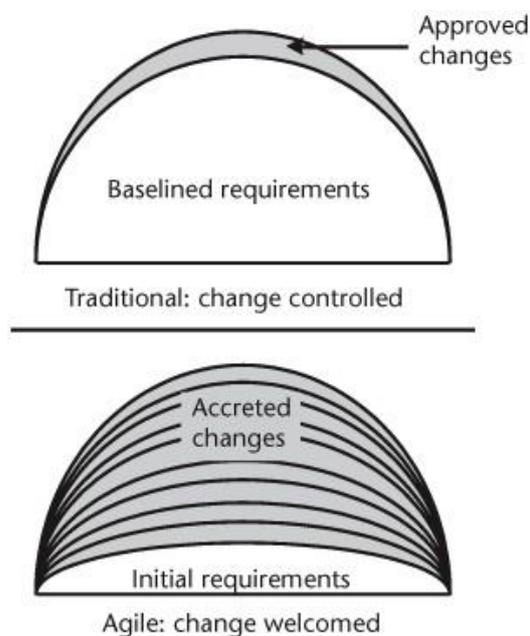
La collaboration étroite avec le client identifiée comme troisième valeur parmi les quatre du Manifeste agile permet de contribuer plus vivement au succès du projet en atténuant les risques et en développant d'autres solutions comme l'explique Koch (2005).

Le client peut être une ressource vraiment déterminante pour le succès du projet. La participation du client au processus est essentiellement conçue comme une question de « gestion des connaissances » (Chang & Taylor, 2016). Dès le début il peut contribuer à une compréhension plus claire du projet et par conséquent permettre aux développeurs d'avoir une idée plus claire et détaillée du livrable voulu. Cette méthode garantit que les utilisateurs prévus participent à la définition et au test de ce que fera le logiciel prévu (Chitchyan, Bird, & Ieee 28th International Requirements Engineering Conference Zurich, 2020).

Il est sans doute vrai qu'en projet informatique et surtout en développement de logiciel, les exigences du projet ne sont pas les mêmes tout au long du cycle de vie du projet. La collaboration étroite entre le client et les développeurs favorise une interaction régulière et importante entre les deux parties (Koch, 2005). Cette

interaction peut être une solution à la gestion du changement. Elle permet d'assimiler et de gérer efficacement les nouvelles exigences en les adaptant continuellement aux projets. D'après Sommerville et al. (1993), les méthodes contrastées pour l'engagement actif des utilisateurs dans le processus d'élicitation des exigences sont les méthodes ethnographiques et d'observation.

**Figure 5 : Gestion de changement**



**(Koch, 2005)**

Cette collaboration permet d'atténuer les risques de dépassement en termes de délai. Avec une collaboration entre les deux parties, il est plus rapide de recevoir les nouvelles exigences de la part du client et aussi d'avoir l'avis du client face à un changement qui survient au cours du projet. On perd alors moins de temps par rapport à la gestion de projet traditionnel qui a tendance à utiliser des contrôleurs d'accès et des spécifications d'exigences pour assurer la médiation entre l'équipe de développement et le client (Koch, 2005), comme le montre la figure 5.

La collaboration entre le client et les développeurs est vraiment un acteur déterminant pour réussir son projet en développement de logiciel grâce à ces

nombreux avantages. Un niveau élevé de coordination entraîne de nombreux avantages tels que des cycles de développement plus courts, des économies de coûts et des produits mieux intégrés (Espinosa, Slaughter, Kraut, & Herbsleb, 2007). Aussi, l'intégration du client dans les approches de développement agiles permet de réagir plus rapidement aux exigences changeantes du marché (Boehm & Turner, 2004).

#### *1.7.2.2. Équipe de personnalités extraverties*

Un projet de développement est souvent réalisé en équipe. Il est sans doute vrai que la composition de l'équipe de développement est un facteur clé dans la réussite du projet et la qualité du livrable. D'après (Gorla & Lam, 2004), la composition de l'équipe est considérée comme un déterminant du succès des projets logiciels, car une collaboration étroite et une bonne communication au sein de l'équipe sont une partie intégrante de la créativité et de la maîtrise des défis technologiques. Des équipes inefficaces peuvent être le produit d'une composition inappropriée. Décider d'utiliser une approche d'équipe n'est que la première étape. Il faut faire preuve d'un grand soin dans la construction de l'équipe pour assurer son efficacité ultime (Bradley & Hebert, 1997).

Pour réussir la composition de l'équipe pour un projet logiciel de qualité, il y a plusieurs aspects qui entrent en jeu. Certains des facteurs liés à l'équipe qui peuvent affecter la performance du projet sont la composition de la personnalité des membres, le leadership de l'équipe, la communication et la coordination intraéquipe (Gorla & Lam, 2004).

Quelques études ont été réalisées sur la composition de la personnalité des membres d'équipe dans le cadre de projet logiciel. Les avantages attendus des équipes qui réussissent comprennent une motivation accrue, un plus grand engagement dans les tâches, des niveaux de performance plus élevés, la capacité à résister au stress, des solutions plus innovantes et une diminution du temps de développement (Bradley & Hebert, 1997).

En 1984 (White) a étudié sur le terrain deux équipes sous deux dimensions de personnalités diverses et constate que les équipes de personnalités diverses réussissent mieux. De même Bradley and Hebert (1997) ont analysé les performances de deux équipes à l'aide de l'indicateur de type Myers-Briggs (MBTI) . Après avoir considéré plusieurs facteurs au début comme l'âge, le QI, la personnalité, le sexe, il s'est avéré qu'il y avait plus de différences significatives avec le facteur personnalité que les autres. Leur étude s'est alors basée sur l'influence de type de personnalité. D'après leur résultat, dans les situations d'équipe, les introvertis ont souvent tendance à garder l'information pour eux-mêmes et sont moins communicatifs lors des réunions.

**Tableau 3: Type de personnalités**

Critères	Dimension de la personnalité	Équipe performante	Écart type	Signification
Chef d'équipe	Collecte d'informations	Intuitive > détection	0.297	.0130
	Prise de décision	Sentiment > pensée	0.297	.0130
Analyste de système	Prise de décision	Pensée > sentiment	0.482	.0038
Programmeur	Interaction avec le monde	Extravertie > introvertie	0.505	.0020
Hétérogénéité Chef membre	Interaction avec le monde	Extravertie-introvertie	0.181	.0612
	La collecte d'informations	intuitives-détection	0.595	.0001
Hétérogénéité membre-membre	Toutes dimensions			Insignifiant

Les personnes extraverties cependant sont des personnes sociables et aiment interagir avec les autres. L'équipe la plus performante a été celle avec plus de personnalités extraverties. Le grand pourcentage d'introvertis de l'équipe la moins performante a peut-être inhibé la réussite d'une communication intraéquipe (Bradley & Hebert, 1997).

Dans la même logique, Yellen, Winniford, and Sanford (1995) ont constaté que les extravertis étaient relativement plus efficaces que les introvertis dans la prise de décision en groupe.

Il est vrai que ces études précédentes pour la plupart sont menées sur de grandes équipes de développement logiciel et sont perspicaces, mais avaient une portée limitée et ne sont pas généralisables (Gorla & Lam, 2004). Elles sont limitées, car ne considérant qu'une ou deux équipes, ou n'examinant qu'une ou deux dimensions de la personnalité (White, 1984).

Dans le même souci de trouver une meilleure formule dans la composition de l'équipe projet en système d'information et accroître la performance au sein de ces équipes, Gorla and Lam (2004) proposent une étude à base d'une méthodologie d'enquête, contrairement aux analyses d'études de cas utilisées dans la plupart des recherches antérieures. Ils considèrent quatre dimensions de la personnalité à savoir : l'interaction sociale, la collecte d'informations, la prise de décision et les relations avec le monde extérieur.

**Tableau 4 : Analyse de l'influence de type personnalité sur les membres d'équipe**

Critères	Dimension de la personnalité	Équipe performante	Écart type	Signification
Chef d'équipe	Collecte d'informations	Intuitive > détection	0.297	.0130
	Prise de décision	Sentiment > pensée	0.297	.0130
Analyste de système	Prise de décision	Pensée > sentiment	0.482	.0038
Programmeur	Interaction avec le monde	Extravertie > introvertie	0.505	.0020
Hétérogénéité Chef membre	Interaction avec le monde	Extravertie-introvertie	0.181	.0612
	La collecte d'informations	Intuitif-détection	0.595	.0001
Hétérogénéité membre-membre	Toutes dimensions	-----	-----	Insignifiant

Chaque dimension étudie deux (2) types de personnalité. Pour chaque type de personnalité, il y a deux (2) résultats possibles. La personnalité globale d'un membre de l'équipe est classée dans une catégorie dans chacune des quatre dimensions, ce qui donne seize (16) personnalités possibles comme explique le tableau suivant.

**Tableau 5: Dimensions et types de personnalités**

Dimensions	Types de personnalité	Explications
Interaction sociale	Extraverti (E) OU Introverti (I)	Les I sont sociables et aiment interagir avec les autres. Les E préfèrent travailler seuls.
Collecte d'informations	Sensible (S) OU Intuitive (N)	Une personne S pr
Prise de décision	Pensant (T) OU Ressentant (F)	
Traitement avec le monde extérieur	Jugeant (J) OU Percevant (P)	

Dans les dimensions d'interaction sociale et de collecte d'informations, le responsable doit envisager de sélectionner le personnel de sorte qu'il y ait une hétérogénéité de personnalité entre les membres de l'équipe et le chef d'équipe. Ainsi, le chef de projet et les membres doivent être sélectionnés de manière à ce que les quatre catégories de personnalité soient représentées : extraverti-intuitif, extraverti-sensible, introverti-intuitif et introverti-sensible (Gorla & Lam, 2004).

La personnalité suggérée pour le chef d'équipe dans le cadre d'un projet logiciel ou informatique en général est *intuitif* (N) pour ce qui est de la dimension de collecte d'informations et sentiments (F) pour ce qui est de la dimension décisionnelle. Un chef d'équipe intuitif sera en mesure de mieux visualiser les futurs besoins. Il est aussi préféré d'avoir un chef d'équipe de personnalité jugement pour ce qui est de la dimension impliquant les relations avec le monde extérieur. De cette manière les jalons du projet seront mieux établis et il pourra aussi bien faire respecter les délais du projet.

Pour ce qui est de la personnalité optimale pour un analyste de système et dans une dimension décisionnelle, il est moins préféré d'avoir un analyste réfléchi,

car ce dernier utilisera plus une approche scientifique et fondera ses décisions sur un raisonnement logique contrairement à une personne pensée (T). En ce qui concerne la dimension de collecte d'informations, un analyste de système de type détection (S) est optimal.

Pour les programmeurs, il est plutôt recommandé une personnalité extravertie comme le disait Yellen et al. (1995). Un extraverti peut communiquer facilement avec les autres participants au système de développement plutôt qu'un introverti qui a tendance à garder les informations et préfère travailler seul. Comme pour les autres membres de l'équipe, les types de personnalité souhaités pour un programmeur sont la détection (S) sur la dimension de collecte d'informations et le jugement (J) sur la dimension impliquant les relations avec le monde extérieur.

Il est important de préciser que ces études sont menées sur des petites équipes et ne sont pas généralisables avec les grandes équipes.

Sélectionner les membres d'équipe ayant la personnalité appropriée et qui concorde avec les autres est bien une étape importante. Mais en plus de cela il est préférable que les responsables des projets en SI s'assurent que ces membres possèdent aussi de l'expertise diversifiée dans les applications informatiques (Faraj & Sproull, 2000).

#### *1.7.2.3. Privilégier le développement itératif*

Le développement itératif est une approche de développement logiciel qui divise le processus de développement d'une grande application en plus petites parties. L'idée de base derrière cette approche est de développer le système à travers des boucles répétées (itérations), permettant aux développeurs de logiciels de tirer parti de ce qu'ils ont appris pendant le développement, des parties ou des versions antérieures du système. Le processus de développement est divisé en cycles de développement itératifs, appelés sprints (Riesener, Doelle, Perau, Lossie, & Schuh, 2021). L'apprentissage provient du développement et de l'utilisation du système, où les jalons possibles du processus commencent par de simples implémentations de

sous-ensembles d'exigences logicielles et améliorent de manière itérative les versions évolutives jusqu'à ce que le système complet soit implémenté. Des modifications de conception et de nouvelles fonctionnalités sont ajoutées à chaque itération. Chaque partie « itératif » représente l'ensemble du processus de développement et contient les étapes de planification, de conception, de développement et de test. Les projets agiles fournissent des logiciels fonctionnels selon un horaire régulier, généralement hebdomadaire ou bihebdomadaire. Les fonctionnalités fournies dans chaque « itération » sont déterminées par l'entreprise (Anand & Dinakaran, 2016; Campanelli & Parreiras, 2015).

Pendant la première étape qui est la planification, le plan du projet qui doit être cohérent avec les objectifs généraux est défini. C'est à ce stade qu'il faut préciser les exigences à respecter, à savoir les actions qui doivent être absolument traitées pour garantir la réussite du projet. Sans une bonne planification, l'on risque de faire évoluer ses productions sans atteindre les objectifs.

Le processus de conception itérative peut être appliqué tout au long du processus de développement de nouveaux produits. Cependant, il est plus facile et moins coûteux de mettre en œuvre des changements dans les premiers stades de développement. La première étape du processus de conception itératif consiste à développer un prototype. Les prototypes doivent être évalués par des groupes de discussion ou des groupes non liés au produit afin de fournir une opinion impartiale. Les informations des groupes de discussion doivent être synthétisées et incorporées dans la prochaine itération de conception. Ce processus doit être répété jusqu'à ce que les problèmes des utilisateurs soient ramenés à un niveau acceptable.

La conception itérative est un moyen de confronter la réalité des besoins et des comportements imprévisibles des utilisateurs qui peuvent conduire à des changements radicaux et fondamentaux dans une conception.

Ce processus est destiné à améliorer en fin de compte la qualité et la fonctionnalité d'une conception. Dans la conception itérative, l'interaction avec le système conçu est utilisée comme une forme de recherche pour informer et faire

évoluer un projet, au fur et à mesure que des versions successives ou des itérations d'une conception sont mises en œuvre.

La conception itérative des interfaces utilisateurs peut être réalisée de plusieurs façons. Une méthode courante d'utilisation de la conception itérative dans les logiciels est le test de logiciels. Bien que cela inclue le test des fonctionnalités du produit au-delà de l'interface utilisateur, un retour d'informations importantes sur l'interface peut être obtenu en testant des versions antérieures du programme. Les cycles de définition, de construction, de révision et de publication se poursuivront jusqu'à ce que tous les incréments de produit soient livrés et que la portée du projet soit terminée (Fagarasan et al., 2021). Cela permet aux éditeurs de logiciels de fournir un produit de meilleure qualité au public et d'éviter de modifier le produit après sa sortie.

Le test logiciel est une enquête menée pour fournir aux parties prenantes des informations sur la qualité du produit logiciel ou du service testé. Les tests de logiciels peuvent également fournir une vue objective et indépendante du logiciel pour permettre à l'entreprise d'apprécier et de comprendre les risques liés à la mise en œuvre de logiciels. Les techniques de test comprennent le processus d'exécution d'un programme ou d'une application dans le but de trouver des échecs, et de vérifier que le logiciel est apte à être utilisé.

Le travail de test est un processus itératif, car lorsqu'un défaut est corrigé, il peut éclairer d'autres défaillances dues à des défauts plus profonds, ou même en créer de nouveaux.

L'un des principaux objectifs des tests est de détecter les défaillances logicielles afin que les défauts puissent être découverts et corrigés. Les tests logiciels impliquent l'exécution d'un composant logiciel ou d'un composant système pour évaluer une ou plusieurs propriétés d'intérêt. En général, ces propriétés indiquent dans quelle mesure le composant ou le système testé, répond aux exigences qui ont guidé sa conception et son développement, répond correctement à toutes sortes d'entrées, remplit ses fonctions dans un délai acceptable, est suffisamment utilisable,

peut être installé et exécuté dans les environnements prévus, atteint le résultat général souhaité par ses parties prenantes.

Le développement logiciel itératif consiste à construire le produit étape par étape, ce qui permet d'identifier et de corriger les défauts au cours des premières étapes, afin d'éviter leur flux descendant dans le processus ultérieur. Ce processus vise à améliorer en fin de compte la qualité et la fonctionnalité d'une conception.

Le développement itératif est, à la base, une approche basée sur l'équipe pour résoudre des problèmes et développer des solutions (Bittner & Spence, 2006). Le développement itératif permet de gagner du temps sur la documentation qui accompagne souvent le flux de travail en cascade et a concentré davantage ce temps sur la conception du projet. Avec le développement itératif, nous pouvons obtenir des commentaires des utilisateurs, tels que la façon dont ils voient le produit maintenant et à quoi est-ce qu'ils s'attendent à l'avenir. Cela nous permettra d'apporter des améliorations et modifications nécessaires. À cet effet, les changements et risques tout au long du projet sont plus faciles à gérer. La qualité du livrable sera accrue, car toutes les parties ont été développées, examinées et testées individuellement.

#### *1.7.2.4. Favorable aux changements*

Au cours du projet, des demandes de modifications du travail à effectuer ou des caractéristiques du livrable arrivent au fur et à mesure que le projet avance. Akbar, Shafiq, Ahmad, Mateen, and Riaz (2018) précisent que les modifications des exigences peuvent être réclamées à tout moment. Ces modifications sont demandées pour certaines raisons. La correction d'une pensée ou d'une exigence non alignée avec les objectifs attendus, une meilleure option non pensée au début du projet, qui permettrait d'augmenter la qualité du livrable ou de développer plus de fonctionnalités. La découverte d'une solution moins coûteuse ou plus viable ou encore qui nécessite moins d'efforts peut aussi être la cause d'une demande de changement. La considération du client, les politiques organisationnelles ou gouvernementales, l'évolution des requêtes du marché, la concurrence sur le marché

aussi peuvent être des causes des demandes de changement (Nurmuliani, Zowghi, & Powell, 2004).

Plutôt qu'un suivi de plan lié à un contrat, l'agilité est une méthodologie qui a une gestion de changement très remarquable et différente. Contrairement à la gestion de changement standard qui est conçu pour être le plus prévisible possible, l'approche agile conçue pour gérer des changements imprévisibles, est tout d'abord une approche ouverte aux changements tout au long du projet. Le système ou le logiciel fonctionnel est livré aux clients le plus tôt possible et met en œuvre les modifications suggérées même tard dans les cycles de vie du projet (Anand & Dinakaran, 2016). La gestion du changement en mode agile se vit à un changement plus rapide et à rythme plus soutenu Villafiorita (2014).

La gestion de projet agile accueille favorablement les changements au cours du projet et grâce à sa flexibilité, arrive à les intégrer sans difficulté. Le processus de la gestion de changement en agilité est plus contrôlé en minimisant le risque de dérive de portée.

#### *1.7.2.5. Axé sur la satisfaction du client*

Dans toute entreprise, la satisfaction du client est toujours une priorité. Non seulement elle offre des avantages concurrentiels pour l'organisation, mais permet aussi d'accroître la qualité du livrable. La concentration sur les besoins du client reste un principe prioritaire dans le processus de développement de la méthode de développement logiciel dynamique DSDM, un cadre de livraison de projet agile (Anand & Dinakaran, 2016).

En priorisant la satisfaction du client, notre projet se concentre plus sur les besoins de ce dernier. La conformité du logiciel aux exigences du client. Les diverses exigences des parties prenantes ayant des besoins spécifiques du marché doivent être prises en compte dans les produits plus complexes (Schuh, 2017).

Ce facteur joue un rôle important dans l'atteinte de la performance du livrable, car déjà, toutes les exigences du client seront prises en compte et le livrable développé devrait satisfaire aux besoins du client.

Les méthodes agiles sont basées sur l'implication précoce des clients, le développement itératif, l'auto-organisation des équipes et l'adaptabilité au changement, issues d'approches agiles de développement de nouveaux produits remontant au 19<sup>e</sup> siècle (Rico & PMP, 2008).

Le développement de produits agiles répond aux besoins des cycles de développement plus courts en mettant l'accent sur les clients et le développement itératif d'incrément fonctionnels du système (Riesener et al., 2021). Grâce à l'agile, la valeur client est créée en éliminant le gaspillage, en amplifiant l'apprentissage, en responsabilisant l'équipe, en livrant rapidement et en optimisant le produit dans son ensemble (Fagarasan et al., 2021).

## 1.8. LA QUALITÉ

### 1.8.1. Définition de la qualité

Depuis plus de cinquante (50) ans, la qualité est devenue et restée une réflexion dominante qu'on rencontre dans les organisations (Beckford, 2002). Toute organisation au monde qui cherche à améliorer ses produits et ses services afin de mieux se positionner sur le marché et faire face à la concurrence, se lance à la recherche de la qualité. L'amélioration de la qualité et l'innovation sont donc devenues des stratégies établies alors que les entreprises cherchent à créer et à défendre leur position concurrentielle (Bourke & Roper, 2017). La qualité est un facteur déterminant et indispensable pour la réussite d'un projet tellement qu'elle figure sur le triangle de fer de la gestion de projet suivi des autres facteurs tel que le temps et le coût.

Toutefois, de nombreuses initiatives de la qualité n'atteignent pas toujours leurs objectifs ou ne réussissent que partiellement. Cet échec résulte peut-être de la focalisation étroite de nombreux programmes de qualité et d'un manque conséquent

d'étendue dans la compréhension du véritable rôle et de la signification de la qualité dans l'efficacité organisationnelle (Beckford, 2002). Pour chaque type de projets, nous avons des définitions de la qualité qui diffèrent.

En gestion de projets, la qualité n'est pas toujours bien cernée comme le temps et le coût qui sont des facteurs quantitatifs à cause de ces multiples définitions qui diffèrent parfois par leurs multiples dimensions (R. Basu, 2014; Geraldi, Kutsch, & Turner, 2011). D'après Heisler (1990), cette complexité du sujet de la qualité peut être due au fait qu'elle n'est pas quantifiable comme le temps et le coût.

D'après (Ron Basu, 2011; R. Basu, 2014), « la qualité est ce que le client attend comme une expérience durable ». De son côté, Juran (1989) définit la qualité comme les fonctionnalités qui répondent aux besoins des clients. Donc qui garantissent leur satisfaction. D'après ces deux définitions, il est non seulement important que le chef de projet et son équipe s'assurent que le projet réponde aux besoins des clients, mais aussi que cela soit une expérience durable pour celui-ci.

Parlant de la définition de la qualité en gestion de projets, il est important de savoir ce que pensent les grandes références de ce domaine. D'après PRENCE 2, la qualité est «la mesure dans laquelle un ensemble de caractéristiques inhérentes à un produit, service, processus, personne, organisation, système ou ressource répond aux exigences » (David, 2018). Selon lui, il est important de garder à l'esprit que « la qualité consiste à rendre quelque chose adapté à l'usage pour lequel il est finalement nécessaire ». De son côté, le Project Management Institute aussi fait l'accent sur les caractéristiques qui doivent satisfaire aux exigences. En ces mots, « la qualité est le degré auquel un ensemble de caractéristiques intrinsèques satisfait à des exigences » (Project Management, 2017).

Comme pour tous les produits, les qualités logicielles qui intéressent le plus les gens sont celles qui mesurent de quelle façon le système répond à ses exigences (Faella & Safari, 2020)?

Les groupes de normalisation de la qualité logiciel définissent la qualité logiciel comme : la mesure dans laquelle un produit logiciel répond aux exigences établies ; cependant, la qualité dépend de la mesure dans laquelle ces exigences établies représentent avec précision les besoins, désirs et attentes des parties prenantes (Incorporated, 2014). Pour ISO/IEC 25010:2011, la Qualité du logiciel est la conformité aux exigences logicielles établies ; la capacité d'un produit logiciel pour satisfaire les besoins déclarés et implicites lorsqu'il est utilisé dans des conditions spécifiées (Laporte & April, 2018).

Dans notre recherche, il serait intéressant de savoir les caractéristiques importantes qui définissent la qualité en gestion de projet logiciel. Donc ce que c'est qu'un logiciel de qualité.

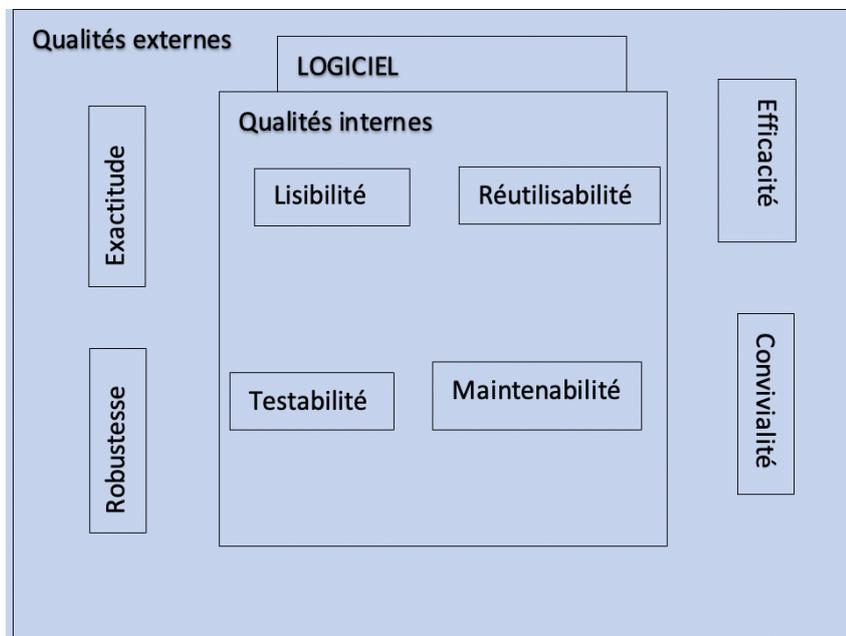
### **1.8.2. Logiciel de qualité**

La qualité d'un logiciel n'est pas chose facile à mesurer. On peut distinguer deux aspects de la qualité d'un logiciel. La qualité interne et la qualité externe.

La qualité interne concerne l'aspect interne du logiciel : le code source pour être précis. Il s'agit de fixer et d'évaluer les normes de l'architecture interne du logiciel.

La qualité externe est l'aspect visible et le plus souvent apprécié d'un logiciel. C'est par cet aspect que l'utilisateur ou le client vérifie si le logiciel répond à ses exigences. Il comprend les interfaces, le design, les fonctionnalités.

**Figure 6 : Qualités internes et Qualités externes d'un logiciel**



(Faella & Safari, 2020)

#### 1.8.2.1. Qualités internes

D'après l'image précédente, nous avons comme qualité interne la lisibilité, Réutilisabilité, Testabilité et Maintenabilité.

La lisibilité caractérise le design du code source de l'application. La manière dont le code est structuré. Il est important de disposer le code pour qu'il soit plus facile à lire et plus simple à comprendre. Les balises, les fonctions, et les requêtes doivent être simples à repérer. Un code source bien lisible permet de repérer facilement les bogues et apporter des modifications. Il pourra être simple à comprendre par un autre développeur favorisant ainsi sa réutilisabilité. Il est aussi souhaité d'ajouter des commentaires afin de mieux expliquer le code.

La réutilisabilité est une qualité souhaitée d'un logiciel, pouvant lui permettre d'être utilisé et réutilisé au besoin. C'est normal qu'un logiciel qui n'est pas utilisable ne serve à rien, car ne pouvant résoudre un problème. Une première

chose est que le logiciel réponde aux exigences donc fonctionnel, selon celles qui sont souhaitées. S'assurer que le code continue de fonctionner est l'une des plus grandes considérations auxquelles les ingénieurs logiciels sont confrontés et doit être pris en compte à toutes les étapes du codage (Long, 2021). Les préférences des consommateurs, les besoins des entreprises et les considérations technologiques peuvent tous changer au cours du temps et il va falloir appliquer des modifications au code. Alors le code doit être conçu pour assurer la continuité du logiciel et faciliter sa modification pour d'éventuels changements.

Notre troisième qualité interne d'un logiciel est la testabilité : elle consiste à écrire le code afin que celui-ci puisse faciliter les tests. Pour les procédures par exemple il est conseillé d'avoir des procédures « Main » qui renvoient une valeur plutôt que des procédures « Void » qui ne renvoient rien. Avec *void*, il n'y a pas de valeur à la fin. En cas de test on ne peut pas savoir si la procédure est correcte ou pas. Ce qui rend difficiles les recherches de bogues.

En fin nous avons la maintenabilité qui est aussi un attribut important pour un bon logiciel. C'est un facteur clé d'un logiciel de qualité qui indique la facilité à laquelle le logiciel peut être maintenu. Un logiciel qui répond à ce critère rend plus simple la localisation des bogues, la facilité à la maintenance.

#### 1.8.2.2. *Qualités externes*

- L'exactitude

Il s'agit ici d'avoir un logiciel qui répond aux besoins du client, c'est-à-dire qui fait exactement pourquoi il est conçu. Un logiciel capable de résoudre les problèmes. Ce serait inutile d'avoir un logiciel qui n'est pas capable de résoudre un problème pour lequel il est conçu. Cette qualité se vérifie souvent par des tests afin de déterminer si les résultats sont exacts ou pas.

- Robustesse

La robustesse du logiciel est sa capacité à résister aux éventuels problèmes. Par exemple, face à un problème de coupure de courant, un logiciel robuste serait capable de sauvegarder les données. C'est aussi le cas d'un logiciel qui résiste face aux piratages.

- Convivialité

La convivialité du logiciel est liée à la conception de son interface utilisateur (UI) et est abordée par des disciplines telles que l'interaction homme-machine et la conception de l'expérience utilisateur (UX) (Faella & Safari, 2020). Il est intéressant d'avoir un logiciel qui fonctionne, mais aussi qui est simple à utiliser. Le logiciel sera utilisé par un client ou un utilisateur final qui n'est souvent pas le développeur. Il est dans ce cas important de considérer la capacité de l'utilisateur à comprendre et utiliser le logiciel afin d'atteindre ses objectifs. Bien décrire les champs, les liens, les boutons et toute l'interface par des mots, phrases ou icônes simples à comprendre.

- Efficacité

L'efficacité d'un logiciel est son adéquation par rapport à ses objectifs et à certains critères propres à lui-même. Il s'agit du temps d'exécution, de la consommation des données (en mémoire) et la consommation de la bande passante. Un logiciel rapide (temps d'exécution faible) permet de résoudre plus rapidement les problèmes et de gagner du temps. En plus de ce critère, un logiciel qui consomme moins de ressource est beaucoup plus préféré. Le plus souvent, ces deux critères sont liés, car un logiciel qui consomme moins de ressources est plus rapide.

## 1.9.L'ASSURANCE QUALITÉ

L'assurance qualité est un programme couvrant les activités nécessaires pour assurer la qualité du travail afin de répondre aux exigences du produit ou projet. Le QA implique l'établissement de politiques, de procédures, de normes, de formations, de lignes directrices et d'un système lié au projet, tous nécessaires pour produire de

la qualité (S. L. Tang & Poon, 2005). En projet logiciel, on parle d'assurance qualité logicielle.

D'après Laporte and April (2018), l'Assurance Qualité Logiciel (SQA ou AQL) est un ensemble d'activités qui définissent et évaluent l'adéquation des processus logiciels pour fournir des preuves qui établissent la confiance que les processus logiciels sont appropriés et produisent des produits logiciels de qualité appropriée pour leurs objectifs. Ils soutiennent que le terme AQL qui est censé garantir une qualité absolue pourrait être un peu trompeur, car ne permet que de minimiser les risques. En leur mot, « l'assurance qualité est mise en œuvre pour réduire les risques de développer un logiciel qui ne répond pas aux désirs, besoins et attentes des parties prenantes dans les limites du budget et du calendrier ».

Alors tout développeur ou gestionnaire de projet dans le souci de garantir une qualité d'un projet logiciel devrait mettre en place le processus d'assurance qualité logiciel et conformer tout le cycle de vie du projet aux normes des grandes organisations de normalisation de l'assurance qualité. Nous avons par exemple :

- IEEE Std. 730-2014 – Software quality assurance
- ISO/IEC/IEEE 12207:2008 – Systems and software engineering Soft-
- IEEE Std. 1012-2012 – Systems and software verification and validationISO

L'évaluation des sujets de conformité comprend : plans de projet, Produits logiciels de projet, produits du projet pour acceptabilité par le client, produits de la phase d'exploitation du projet et l'évaluation du produit logiciel par des mesures (Galin, 2018). Pour se faire, l'auteur cite trois activités requises pour le processus d'évaluation de l'AQL :

1. Identifier les processus du cycle de vie requis par les exigences contractuelles, les réglementations et les conventions.

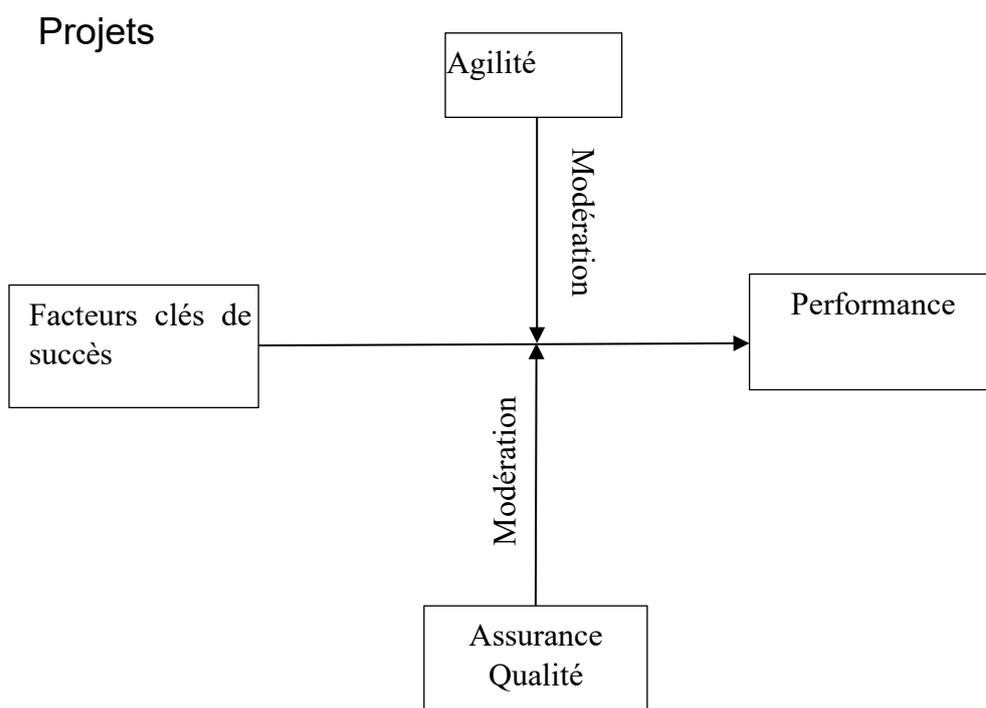
2. Examiner les processus de cycle de vie planifiés pour s'assurer qu'ils sont conformes aux exigences de processus établies pertinentes.

3. Examiner les processus en cours d'exécution pour s'assurer qu'ils sont conformes aux exigences de processus pertinentes établies. L'examen devrait produire des listes de non-conformité.

### 1.10. CADRE CONCEPTUEL FINAL

À la suite de la revue de littérature, la figure ci-dessous illustre les relations entre les objectifs et les questions de recherche.

*Figure 7: Cadre conceptuel final*



## APPROCHE MÉTHODOLOGIQUE

Ce chapitre présente la méthodologie utilisée pour répondre à la problématique de cette recherche. Il discute de la nature de la recherche envisagée, les méthodes de collecte des données proposées ainsi que les outils d'analyse utilisés.

### 1.11. POSITIONNEMENT DE LA RECHERCHE

Cette recherche se présente autour de quatre principaux domaines (la gestion de projet, le développement des logiciels, l'agilité et la qualité). Pour cette étude, notre premier objectif se base sur les facteurs clés de succès des projets de développement de logiciel en vue d'une meilleure qualité. Pour ce premier objectif, la compréhension des notions de la gestion de projet logiciel et de la qualité logiciel sont essentielles. Ensuite vient l'analyse des pratiques en agilité pour une modération ainsi que l'utilité de la pratique de l'assurance qualité logicielle.

### 1.12. PERSPECTIVES DE LA RECHERCHE ET THÉORIES MOBILISÉES

L'objet de cette étude a été motivé principalement par un taux d'échec alarmant dans la gestion des projets logiciels et une faible qualité des livrables. Une approche inductive est utilisée afin d'explorer la nature du problème et mieux comprendre les concepts.

Elle nous permet dans un premier temps d'expliquer les facteurs qui interviennent et les pratiques importantes pour arriver à l'excellence en gestion de projets de développement de logiciels. L'analyse des relations entre nos variables identifiées pour cette étude nous permet de proposer des recommandations aux futurs chercheurs.

### 1.13. APPROCHE DE LA RECHERCHE

Cette étude vise à identifier les facteurs influençant positivement la gestion de projets logiciels et l'impact de l'agilité ainsi que l'assurance qualité sur ces facteurs en vue d'obtenir un meilleur résultat.

L'approche inductive utilisée pour cette étude nous permet d'analyser les données qualitatives et quantitatives collectées grâce aux études antérieures, à la revue de littérature, et à nos enquêtes et entretiens. Nous arrivons aussi à tester la plausibilité de nos hypothèses grâce à l'analyse des données et par ailleurs comprendre l'impact de l'agilité et l'assurance sur la relation facteur clés de succès et performance.

Pour y arriver, un questionnaire a été préalablement conçu de manière à cerner le périmètre de notre étude. Les données sont ensuite collectées, traitées, analysées et interprétées. À la fin de cette recherche, les données traitées devraient permettre de combler au moins partiellement les manquants de la littérature afin d'apporter une réelle contribution au développement de la connaissance actuelle sur le sujet.

### 1.14. DESIGN DE RECHERCHE ENVISAGE : RECHERCHE MIXTE

Pour l'analyse des données, nous avons le choix entre les méthodes quantitatives, qualitatives et mixtes (Vogt, 2014). La méthode mixte fut le choix, car les données qualitatives et quantitatives ont toutes été traitées et analysées. Dans un premier temps, une méthode qualitative nous a permis d'explorer et analyser les données de la revue de littérature et une méthode quantitative nous a permis de quantifier nos données recueillies à travers une enquête et d'expliquer les résultats statistiques. Ces deux méthodes ont permis d'exploiter plus en profondeur nos données tout en donnant un sens et une signification aux chiffres.

Une méthode qualitative implique des données qualitatives alors qu'une méthode quantitative utilise des données quantifiables. L'adoption de ces deux méthodes dans une ou plusieurs étapes du processus de recherche constitue la méthode mixte.

Cette recherche basée sur une enquête grâce aux questionnaires nous a aidées à tester la plausibilité de nos propositions. Tout un processus a été suivi tout au long de cette recherche : la collecte, le traitement et l'analyse des données. Ces données sont de deux différents types : les données secondaires (issues de la revue de littérature) et les données primaires (issues d'un travail plus pratique). Les données primaires sont collectées à l'aide d'enquête et d'entrevue conduites auprès des praticiens.

Cette démarche nous permet, à travers nos données collectées, d'expliquer l'influence des facteurs clés de succès des projets logiciels sur la performance et l'impact de l'agilité et de l'assurance qualité comme facteurs modérateurs. Pour finir, un lien entre les données collectées et les objectifs de notre recherche est établi.

### 1.15. MÉTHODOLOGIES MOBILISÉES

La méthodologie adoptée permet d'étudier les différents facteurs qui influencent la gestion de la qualité en génie logiciel, d'étudier individuellement l'impact de l'agilité et de l'assurance qualité sur ces facteurs dans le but d'obtenir une meilleure performance. Pour accéder aux données primaires de notre recherche, nous avons procédé par des questionnaires d'enquête :

Questionnaire d'enquête : il est composé de 12 questions subdivisées en trois sections. Ce questionnaire élaboré au moyen de *Google form*. Le lien du formulaire est ensuite envoyé aux répondants par courriel, SMS ou vis l'application Teams. Les réponses anonymes sont ensuite collectées directement sur *Google form* sans discussion.

La suite de cette partie décrit et explique les techniques mobilisées pour aboutir aux objectifs de cette étude.

## 1.16. TECHNIQUE, PLAN ET PROCÉDURES DE COLLECTE ET DE TRAITEMENT DES DONNÉES

Dans cette partie, les méthodes de collecte et de traitement des données seront expliquées ainsi que la raison du choix.

### 1.16.1. Techniques de collecte des données

Deux types de données ont été collectés : les données secondaires issues de la revue de littérature et les données primaires issues des questionnaires d'enquête.

#### 1.16.1.1. *Revue documentaire*

Afin d'approfondir le sujet de la recherche et de mieux comprendre les notions de la gestion de projet logiciel, l'agilité et l'assurance qualité, une revue de littérature est nécessaire. Cette partie représente une documentation des notions qui entourent notre sujet et des idées ou théories développées par des chercheurs. Le cadre conceptuel initial et le cadre conceptuel final ont été élaborés et détaillés grâce à cette revue documentaire.

#### 1.16.1.2. *Enquête par questionnaire*

L'enquête vise principalement la collecte d'informations sur les éléments précis auprès des personnes concernées. Cette technique permet de recueillir les informations concrètes auprès des praticiens. La population cible de notre recherche est composée de 17 personnes, dont les gestionnaires de projets, développeur de logiciels, coordonnateurs de projets sans préférence de sexe, ni préférence géographique. Notons que la majorité habite au Canada.

Le questionnaire de collecte de données est composé de 12 questions subdivisées en trois grandes parties, toutes des questions fermées. Avant toute question, les divers concepts du sujet de notre recherche ont été définis en vue d'une meilleure compréhension et des résultats cohérents. Les réponses sont évaluées à travers une échelle de Likert (pas du tout d'accord, en désaccord, neutre, en accord et fortement en accord).

Les questions posées reflètent le but de la recherche et sont conçues pour apporter une réponse plus ou moins précise de chaque partie de la recherche.

### **1.16.2. Plan de collecte de données**

Cette partie explique le plan adopté pour recueillir les données primaires.

#### *1.16.2.1. Choix de la population*

Étant donné que notre recherche est basée sur le domaine informatique, principalement le développement de logiciel, il était important que les gestionnaires répondants proviennent du domaine. Notre but étant de comprendre les facteurs favorisant le succès et la performance des projets logiciels, et l'impact de l'agilité et l'assurance qualité, notre population est composée des gestionnaires de projets logiciels, les coordonnateurs et les développeurs. Nous avons eu aussi des réponses des membres de la direction. Notons que tout ce profil traite de la gestion de projet logiciel.

À la fin de cette enquête, nous comptons 17 réponses dont la plupart résident au Canada. Nous avons aussi des répondants depuis les États-Unis et la France.

La diversité des répondants que ce soit par leurs expériences ou leur pays de résidence, enrichit cette étude et permet d'avoir une vue d'ensemble différent sur la gestion de projet logiciel.

### 1.16.3. Traitements, analyse et interprétation des données

Cette partie explique le processus suivi pour le traitement et l'analyse des données.

*Figure 8 : Processus de traitement et d'analyse des données*



Organisation : après le partage des liens des questionnaires avec les répondants, les données sont collectées directement sur *Google Form* après chaque réponse. L'avantage est la rapidité de collecte des données sans aucune altération.

Exportation des données : après toutes les réponses, les données sont téléchargées en fichier csv et exportées vers le logiciel Excel pour codage.

Codage des données : la codification consiste à organiser les données quantitatives recueillies sous forme de tableau et diagramme exploitables au moyen du logiciel Excel. Les résultats du fichier csv ont été réorganisés. Chaque colonne représentant une question de recherche a été organisée en tableau dynamique exploitable pour faciliter l'analyse.

Analyse : cette étape a pour but de rechercher des réponses aux questions de cette recherche à travers les données collectées. Dans le but de prendre connaissance de l'image des variables étudiées, une analyse descriptive du contenu des données collectées a été premièrement faite.

Par la suite, il y a eu une analyse relationnelle pour compléter le premier niveau de traitement effectué. Cette analyse relationnelle a permis de décrire les divers objets et leurs attributs et les relations entre ces objets.

Le chapitre suivi présente les résultats de cette recherche et leurs discussions.

## RÉSULTAT ET DISCUSSION

Ce chapitre présente les principaux résultats de cette recherche recueillis par enquête auprès des gestionnaires de projets. Le chapitre est structuré en deux sections : une première section qui présente une description des résultats récoltés, et une deuxième section qui fait une analyse des résultats.

### 1.17. PRÉSENTATION DES RÉSULTATS

Les données récoltées sont regroupées en trois catégories :

- Les données sur les facteurs clés de succès et la performance des logiciels. L'objectif est de déterminer l'influence des facteurs clés de succès sur la performance des logiciels. Les réponses de la section 1 du questionnaire de recherche ont permis de mieux comprendre cette partie.
- Les données sur la modération de la relation facteurs clés de succès et performance grâce à l'agilité. Pour répondre à l'objectif 2 (OB2), les questions sont destinées à mesurer le degré de modération de la relation entre les facteurs clés de succès et performance grâce à l'agilité.
- Les données sur la modération de la relation facteurs clés de succès et performance grâce à l'assurance qualité. L'objectif est de comprendre selon les répondants, le degré de modération de la relation facteur clé de succès et performance grâce à l'assurance qualité.

Pour leurs meilleure présentation visuelle et interprétation, les résultats de chaque section du questionnaire ont été regroupés en un tableau par section. Ces résultats sont combinés d'une échelle de 1 à 5 à une échelle de 1 à 3.

Exemple de la combinaison des données de la section 1.

**Tableau 6 : Échelle de réponses section1**

Échelle originale			Échelle simplifiée	
1	Fortement désaccord	en	1	Pas d'accord
2	En désaccord			
3	Neutre		2	Neutre
4	En accord		3	En accord
5	Fortement accord	en		

La présentation des résultats est faite à base de l'échelle originale et la discussion avec l'échelle simplifiée.

### 1.17.1. Présentation des résultats de l'objectif 1

L'objectif est d'identifier les facteurs clés de succès des projets de développement de logiciel. Pour ce faire, une liste de sept facteurs de la gestion de projets logiciels et de 6 indicateurs de performances ont été identifiés grâce à la revue de littérature. En suite la première section des questionnaires de notre recherche permet d'évaluer l'influence de ces facteurs sur la performance. La première section des questionnaires est composée de 6 questions conçues comme suit : « Selon vous, chaque facteur clé de succès a-t-il une influence sur 'chaque indicateur de performance' d'un logiciel ? »

Les données sont présentées en valeur et en pourcentage.

Nombre de participants : 17

○ **Efficacité d'un logiciel**

**Tableau 7 : Les données sur la relation entre facteurs clés de succès et l'efficacité**

	Efficacité				
	Fortement en désaccord	En désaccord	Neutre	En accord	Fortement en accord
Objectif du projet bien clair	0	0	1	4	12
	0,00%	0,00%	5,88%	23,53%	70,59%
Des exigences du client bien clair	0	0	1	2	14
	0,00%	0,00%	5,88%	11,76%	82,35%
Des risques bien identifiés	0	0	2	8	7
	0,00%	0,00%	11,76%	47,06%	41,18%
Attentes réalistes	0	0	3	7	7
	0,00%	0,00%	17,65%	41,18%	41,18%
Responsabilisation	0	1	0	6	10
	0,00%	5,88%	0,00%	35,29%	58,82%
Personnel compétent	0	0	1	5	11
	0,00%	0,00%	5,88%	29,41%	64,71%
Soutien de la haute direction	0	0	3	6	8
	0,00%	0,00%	17,65%	35,29%	47,06%

○ **Exactitude d'un logiciel**

**Tableau 8: Les données sur la relation entre facteurs clés de succès et l'exactitude**

	Exactitude				
	Fortement en désaccord	En désaccord	Neutre	En accord	Fortement en accord
Objectif du projet bien clair	0	1	0	1	15
	0,00%	5,88%	0,00%	5,88%	88,24%
Des exigences du client bien clair	0	0	0	5	12
	0,00%	0,00%	0,00%	29,41%	70,59%
Des risques bien identifiés	0	0	2	7	8
	0,00%	0,00%	11,76%	41,18%	47,06%
Attentes réalistes	0	0	1	5	11
	0,00%	0,00%	5,88%	29,41%	64,71%
Responsabilisation	0	0	1	7	9
	0,00%	0,00%	5,88%	41,18%	52,94%
Personnel compétent	0	0	1	7	9
	0,00%	0,00%	5,88%	41,18%	52,94%
Soutien de la haute direction	0	1	3	6	7
	0,00%	5,88%	17,65%	35,29%	41,18%

○ **Robustesse d'un logiciel**

**Tableau 9: Les données sur la relation entre facteurs clés de succès et la robustesse**

	Robustesse				
	Fortement en désaccord	En désaccord	Neutre	En accord	Fortement en accord
	2	0	4	2	9
Objectif du projet bien clair	11,76%	0,00%	23,53%	11,76%	52,94%
	0	0	3	6	8
Des exigences du client bien clair	0,00%	0,00%	17,65%	35,29%	47,06%
	0	1	2	3	11
Des risques bien identifiés	0,00%	5,88%	11,76%	17,65%	64,71%
	1	1	4	7	4
Attentes réalistes	5,88%	5,88%	23,53%	41,18%	23,53%
	0	0	3	8	6
Responsabilisation	0,00%	0,00%	17,65%	47,06%	35,29%
	0	0	0	6	11
Personnel compétent	0,00%	0,00%	0,00%	35,29%	64,71%
	2	0	2	10	3
Soutien de la haute direction	11,76%	0,00%	11,76%	58,82%	17,65%

○ **Convivialité d'un logiciel**

**Tableau 10: Les données sur la relation entre facteurs clés de succès et la convivialité**

	Convivialité				
	Fortement en désaccord	En désaccord	Neutre	En accord	Fortement en accord
	0	2	2	2	11
Objectif du projet bien clair	0,00%	11,76%	11,76%	11,76%	64,71%
	0	1	3	2	11
Des exigences du client bien clair	0,00%	5,88%	17,65%	11,76%	64,71%
	0	2	5	4	6
Des risques bien identifiés	0,00%	11,76%	29,41%	23,53%	35,29%
	0	2	2	10	3
Attentes réalistes	0,00%	11,76%	11,76%	58,82%	17,65%
	0	1	2	7	7
Responsabilisation	0,00%	5,88%	11,76%	41,18%	41,18%
	0	0	2	5	10
Personnel compétent	0,00%	0,00%	11,76%	29,41%	58,82%
	1	2	3	7	4
Soutien de la haute direction	5,88%	11,76%	17,65%	41,18%	23,53%

○ **Maintenabilité d'un logiciel**

**Tableau 11: Les données sur la relation entre facteurs clés de succès et la maintenabilité**

	Maintenabilité				
	Fortement en désaccord	En désaccord	Neutre	En accord	Fortement en accord
Objectif du projet bien clair	0	0	2	5	10
	0,00%	0,00%	11,76%	29,41%	58,82%
Des exigences du client bien clair	0	0	3	6	8
	0,00%	0,00%	17,65%	35,29%	47,06%
Des risques bien identifiés	0	0	1	7	9
	0,00%	0,00%	5,88%	41,18%	52,94%
Attentes réalistes	0	0	4	6	7
	0,00%	0,00%	23,53%	35,29%	41,18%
Responsabilisation	0	0	4	6	7
	0,00%	0,00%	23,53%	35,29%	41,18%
Personnel compétent	0	0	1	6	10
	0,00%	0,00%	5,88%	35,29%	58,82%
Soutien de la haute direction	0	1	4	8	4
	0,00%	5,88%	23,53%	47,06%	23,53%

**Réutilisabilité**

**Tableau 12: Les données sur la relation entre facteurs clés de succès et la réutilisabilité**

	Réutilisabilité				
	Fortement en désaccord	En désaccord	Neutre	En accord	Fortement en accord
Objectif du projet bien clair	0	1	2	3	11
	0,00%	5,88%	11,76%	17,65%	64,71%
Des exigences du client bien clair	0	1	2	6	8
	0,00%	5,88%	11,76%	35,29%	47,06%
Des risques bien identifiés	0	1	4	3	9
	0,00%	5,88%	23,53%	17,65%	52,94%
Attentes réalistes	0	1	5	4	7
	0,00%	5,88%	29,41%	23,53%	41,18%
Responsabilisation	0	1	5	7	4
	0,00%	5,88%	29,41%	41,18%	23,53%
Personnel compétent	0	0	0	6	11
	0,00%	0,00%	0,00%	35,29%	64,71%
Soutien de la haute direction	0	1	1	11	4
	0,00%	5,88%	5,88%	64,71%	23,53%

**1.17.2. Présentation des résultats de l'objectif 2**

Cette section porte sur l'agilité comme facteur de modération de la relation entre les facteurs clés de succès et la performance. Les variables principales sont

les facteurs clés de succès à la colonne gauche, les indicateurs d'agilité définis dans la revue de littérature. Les questionnaires de la section 2 sont alors posés pour évaluer le degré de modération de la relation entre les facteurs clés de succès et chaque indicateur d'agilité : « D'après vous, la relation facteurs clés de succès et la performance est-elle modérée avec 'chaque indicateur d'agilité'? ». Donc un total de 5 questions.

Les données de la section 2 et 3 collectées sont combinées d'une échelle de 1 à 5 à une échelle de 1 à 3. Contrairement à la section 1, l'échelle originale des données de la section 2 et 3 part de 'Très négativement' à 'Très positivement'.

**Tableau 13: Combinaison des données de la section 2 et 3**

Échelle originale		Échelle simplifiée	
1	Très négativement	1	Négativement
2	Négativement		
3	Neutre	2	Neutre
4	Positivement	3	Positivement
5	Très positivement		

Cette section compte 17 répondants pour les 6 questions.

Le résultat de chaque question est organisé dans un tableau et représenté en valeur et en pourcentage.

**Tableau 14: Les données du degré de modération de la relation entre facteurs clés de succès et la performance par la collaboration client**

	Collaboration client				
	Très négativement	Négativement	Neutre	Positivement	Très positivement
Objectifs du projet bien clair	0	0	1	1	15
	0,00%	0,00%	5,88%	5,88%	88,24%
Des exigences du client bien clair	0	0	1	4	12
	0,00%	0,00%	5,88%	23,53%	70,59%
Des risques bien identifiés	0	1	1	5	10
	0,00%	5,88%	5,88%	29,41%	58,82%
Attentes réalistes	0	1	1	5	10
	0,00%	5,88%	5,88%	29,41%	58,82%
Responsabilisation	0	0	3	8	6
	0,00%	0,00%	17,65%	47,06%	35,29%
Personnel compétent	0	1	2	7	7
	0,00%	5,88%	11,76%	41,18%	41,18%
Soutien de la haute direction	0	1	3	7	6
	0,00%	5,88%	17,65%	41,18%	35,29%

**Tableau 15: Les données du degré de modération de la relation entre facteurs clés de succès et la performance par une équipe constituée majoritairement de personnalités extraverties**

	Équipe de personnalités extraverties				
	Très négativement	Négativement	Neutre	Positivement	Très positivement
Objectifs du projet bien clair	0	1	6	2	8
	0,00%	5,88%	35,29%	11,76%	47,06%
Des exigences du client bien clair	0	4	4	3	6
	0,00%	23,53%	23,53%	17,65%	35,29%
Des risques bien identifiés	0	2	5	4	6
	0,00%	11,76%	29,41%	23,53%	35,29%
Attentes réalistes	0	1	7	7	2
	0,00%	5,88%	41,18%	41,18%	11,76%
Responsabilisation	0	2	5	4	6
	0,00%	11,76%	29,41%	23,53%	35,29%
Personnel compétent	0	1	5	7	4
	0,00%	5,88%	29,41%	41,18%	23,53%
Soutien de la haute direction	0	3	5	6	3
	0,00%	17,65%	29,41%	35,29%	17,65%

**Tableau 16: Les données du degré de modération de la relation entre facteurs clés de succès et la performance en privilégiant le développement itératif**

	Privilégier le développement itératif				
	Très négativement	Négativement	Neutre	Positivement	Très positivement
Objectifs du projet bien clair	0	0	1	6	10
	0,00%	0,00%	5,88%	35,29%	58,82%
Des exigences du client bien clair	0	0	1	5	11
	0,00%	0,00%	5,88%	29,41%	64,71%
Des risques bien identifiés	0	0	1	6	10
	0,00%	0,00%	5,88%	35,29%	58,82%
Attentes réalistes	0	0	1	6	10
	0,00%	0,00%	5,88%	35,29%	58,82%
Responsabilisation	0	1	1	7	8
	0,00%	5,88%	5,88%	41,18%	47,06%
Personnel compétent	0	0	1	9	7
	0,00%	0,00%	5,88%	52,94%	41,18%
Soutien de la haute direction	0	1	4	7	5
	0,00%	5,88%	23,53%	41,18%	29,41%

**Tableau 17: Les données du degré de modération de la relation entre facteurs clés de succès et la performance en favorisant les changements**

	Favorable aux changements				
	Très négativement	Négativement	Neutre	Positivement	Très positivement
Objectifs du projet bien clair	1	0	0	6	10
	5,88%	0,00%	0,00%	35,29%	58,82%
Des exigences du client bien clair	0	1	1	5	10
	0,00%	5,88%	5,88%	29,41%	58,82%
Des risques bien identifiés	0	1	3	8	5
	0,00%	5,88%	17,65%	47,06%	29,41%
Attentes réalistes	0	0	0	8	9
	0,00%	0,00%	0,00%	47,06%	52,94%
Responsabilisation	0	0	4	8	5
	0,00%	0,00%	23,53%	47,06%	29,41%
Personnel compétent	0	0	5	6	6
	0,00%	0,00%	29,41%	35,29%	35,29%
Soutien de la haute direction	0	2	4	8	3
	0,00%	11,76%	23,53%	47,06%	17,65%

**Tableau 18: Les données du degré de modération de la relation entre facteurs clés de succès et la performance en priorisant la satisfaction du client**

	Axé sur la satisfaction du client				
	Très négativement	Négativement	Neutre	Positivement	Très positivement
	1	0	1	3	12
Objectifs du projet bien clair	5,88%	0,00%	5,88%	17,65%	70,59%
	1	0	1	5	10
Des exigences du client bien clair	5,88%	0,00%	5,88%	29,41%	58,82%
	0	0	2	5	10
Des risques bien identifiés	0,00%	0,00%	11,76%	29,41%	58,82%
	1	0	0	8	8
Attentes réalistes	5,88%	0,00%	0,00%	47,06%	47,06%
	0	0	2	7	8
Responsabilisation	0,00%	0,00%	11,76%	41,18%	47,06%
	0	0	1	5	11
Personnel compétent	0,00%	0,00%	5,88%	29,41%	64,71%
	0	0	1	6	10
Soutien de la haute direction	0,00%	0,00%	5,88%	35,29%	58,82%

### 1.17.3. Présentation des résultats de l'objectif 3

Pour analyser l'assurance qualité comme facteur de modération de la relation entre facteurs clés de succès et performance, la question suivante a été posée : d'après vous, la relation entre facteurs clés de succès et performance est-elle modérée avec l'assurance qualité? Le résultat est organisé dans un tableau statistiquement exploitable. Les données sont représentées en valeur (nombre de réponses) et en pourcentages.

Pour cette section, 17 questionnaires de projets en développement de logiciel ont répondu.

**Tableau 19: Les données du degré de modération de la relation entre facteurs clés de succès par l'Assurance Qualité**

	Assurance qualité				
	Très négativement	Négativement	Neutre	Positivement	Très positivement
	0	0	1	4	12
Objectif du projet bien clair	0,00%	0,00%	5,88%	23,53%	70,59%
	0	0	1	3	13
Des exigences du client bien claires	0,00%	0,00%	5,88%	17,65%	76,47%
	0	0	2	1	14
Des risques bien identifiés	0,00%	0,00%	11,76%	5,88%	82,35%
	0	0	3	4	10
Attentes réalistes	0,00%	0,00%	17,65%	23,53%	58,82%
	0	0	2	6	9
Responsabilisation	0,00%	0,00%	11,76%	35,29%	52,94%
	0	1	1	4	11
Personnel compétent	0,00%	5,88%	5,88%	23,53%	64,71%
	0	1	5	5	6
Soutien de la haute direction	0,00%	5,88%	29,41%	29,41%	35,29%

## 1.18. DISCUSSION DES RÉSULTATS

Cette partie correspond à la discussion des résultats présentés dans la section précédente en les combinant avec les résultats des données qualitatives recueillies.

La discussion des résultats porte sur trois (3) objectifs faisant partie du cadre conceptuel. Dans le but de trouver des réponses aux questionnaires de cette recherche, les résultats sont interprétés en suivant l'ordre logique des objectifs énumérés.

Pour que les résultats soient considérés pour atteindre les objectifs, le degré d'accord doit dépasser la moitié plus 1. C'est-à-dire 50 % d'accord + 1. La formule suivante a été adoptée pour déterminer le degré d'accord suffisant pour valider une hypothèse formulée :

- En accord = +51 % (c'est-à-dire 50 + 1)
- Pas d'accord = -49 % (c'est-à-dire 50 - 1)

### 1.18.1. Interprétation des résultats de l'objectif 1.

Les facteurs clés de succès en projet logiciel sont nombreux et la pertinence de chaque facteur diffère d'un type de projet à un autre. La revue de littérature nous

a permis d'énumérer sept facteurs clés de succès importants pour la gestion de projet en développement de logiciel. La première section des questionnaires permet de savoir si les répondants confirment une influence des facteurs clés de succès énumérés sur chaque indicateur de qualité du logiciel.

Les résultats issus de la collecte de données sur la question posée permettent de constater ce qui suit pour chaque indicateur de qualité :

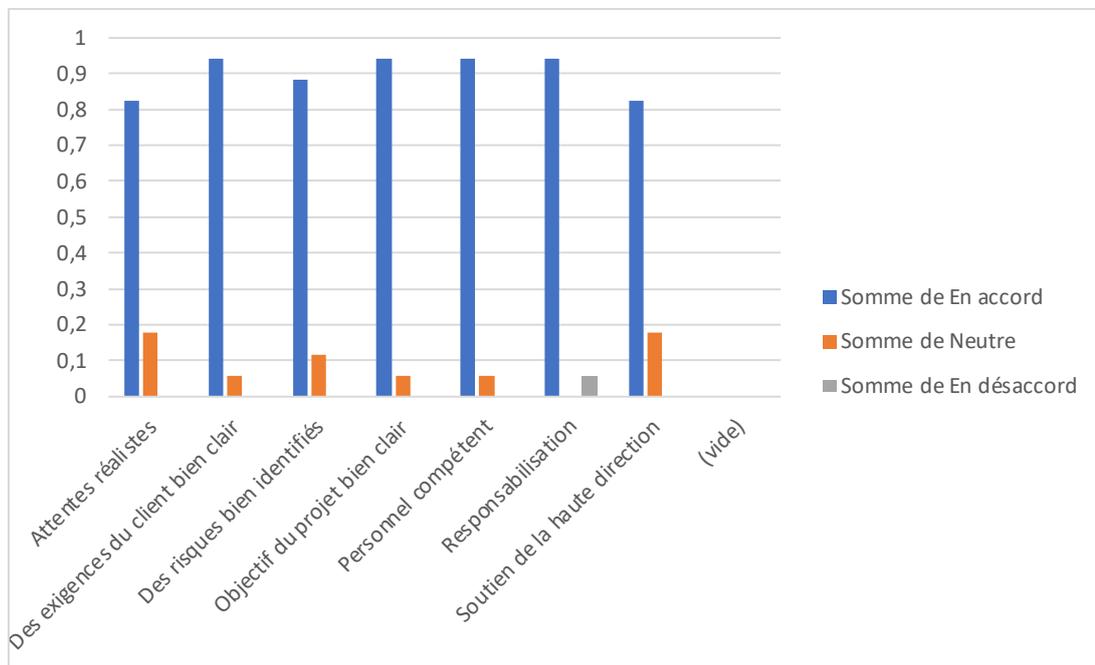
#### *1.18.1.1. L'Efficacité d'un logiciel*

Les 7 facteurs énumérés sont tous retenus :

- Objectifs bien clairs
- Des exigences du client bien claires
- Des risques bien identifiés
- Attentes réalistes
- Responsabilisation
- Personnel compétent
- Soutien de la haute direction

Pour ces sept facteurs, la mesure d'opinion montre que globalement les répondants confirment une influence de ces facteurs sur l'efficacité d'un logiciel. Ceci s'explique par un nombre supérieur des réponses « En accord » : entre 82,35 % et 94,12 % contre 0 à 5,88% manifestant un désaccord.

**Figure 9 : L'efficacité des logiciels**



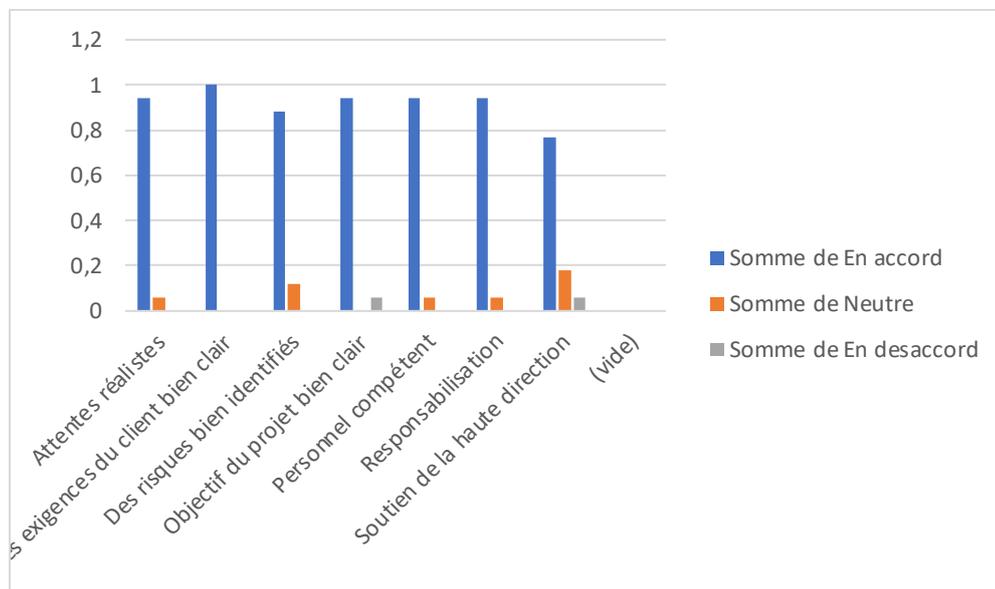
Pour cet indicateur de qualité, les facteurs ayant reçu plus de réponses « En accord » sont : Objectifs bien clairs, des exigences du client bien claires, Responsabilisation et Personnel compétent avec chacun 94,12 %. Le résultat avec un pourcentage le moins élevé est celui du facteur ‘Attentes réalistes’ et ‘Soutien de la haute direction’ avec chacun 82,35 % de réponse « En accord ». Bien que ces facteurs aient un résultat moins élevé par rapport aux autres pour la section « En accord », leur pourcentage est quand même très élevé.

#### 1.18.1.2. L'exactitude d'un logiciel

Pour cet indicateur de qualité, tous les sept facteurs clés de succès énumérés dans la revue de littérature sont également tous retenus. Les données montrent un pourcentage élevé de résultat pour la mention « En accord » : entre 76,47% et 100%. 5,88% de répondants manifestent un désaccord.

Avec ce résultat, on peut donc dire que nos sept facteurs clés de succès exercent une influence positive sur l'exactitude d'un logiciel.

**Figure 10: Exactitude d'un logiciel**



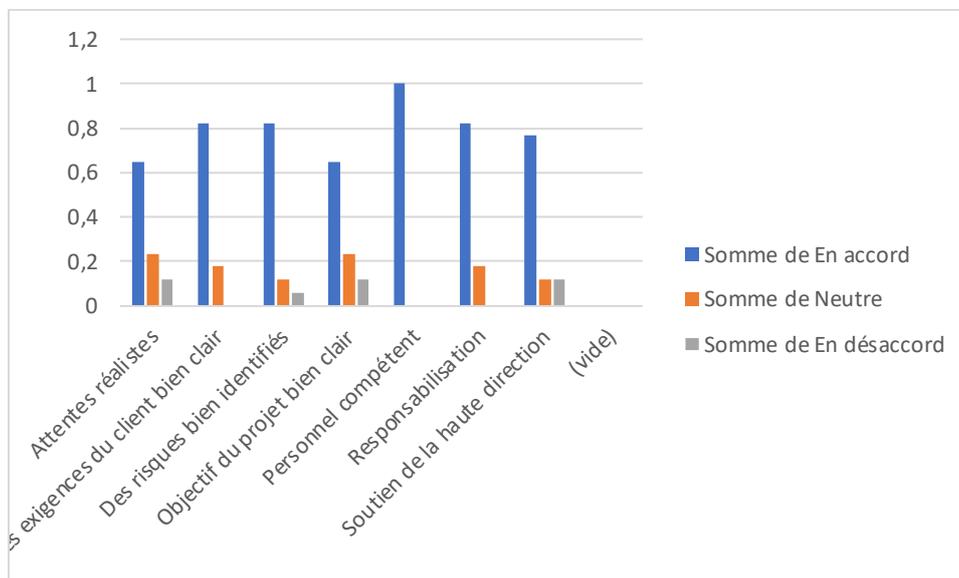
Pour cet indicateur de qualité, 100% des répondants ont répondu favorable à l'influence du facteur lié à la clarté des exigences du client : 'Des exigences du client bien claires'. La réponse ayant récolté moins d'avis 'En accord' est celle liée au facteur 'Soutien de la haute direction' avec un pourcentage de réponse favorable de 76,4 %. Largement supérieur à 51 % nécessaires pour valider un facteur clé de succès.

#### 1.18.1.3. Robustesse d'un logiciel

La question 3 de la première section des questionnaires a pour but de confirmer ou d'infirmer une influence positive des sept facteurs clés de succès sur la robustesse d'un logiciel.

Les résultats permettent de retenir tous les sept facteurs clés de succès avec un pourcentage de réponses favorable plus élevé. Entre 64,71 % et 100%. La majorité des répondants est « En accord » pour une influence de chaque facteur énuméré sur la robustesse d'un logiciel. Entre 5,88% et 11,67% seulement est 'en désaccord'.

**Figure 11: Robustesse d'un logiciel**

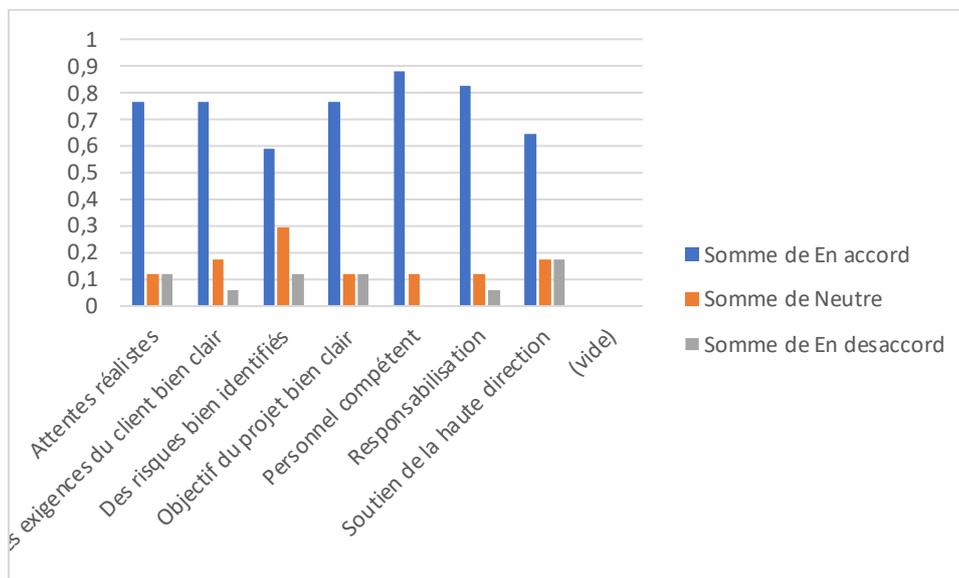


Le facteur qui a récolté le plus d'avis « En accord » est : Personnel compétent avec 100% de réponses favorables. Les sept facteurs clés de succès sont alors toujours valables pour influencer positivement la robustesse d'un logiciel.

#### 1.18.1.4. Convivialité d'un logiciel

Pour les 7 facteurs de la revue de littérature, le nombre de réponses 'En accord' supérieur : Entre 58,82% et 88,24%, permet de retenir également tous les 7 facteurs pour cet indicateur de qualité. Or 5,88 % à 17,65 % seulement des répondants manifestent un désaccord. Ces résultats permettent alors de confirmer que : Objectifs bien clairs ; des exigences du client bien claires ; des risques bien identifiés ; attentes réalistes ; personnel compétent et Soutien de la haute direction influencent positivement la convivialité d'un logiciel pour une meilleure performance.

**Figure 12: Convivialité d'un logiciel**

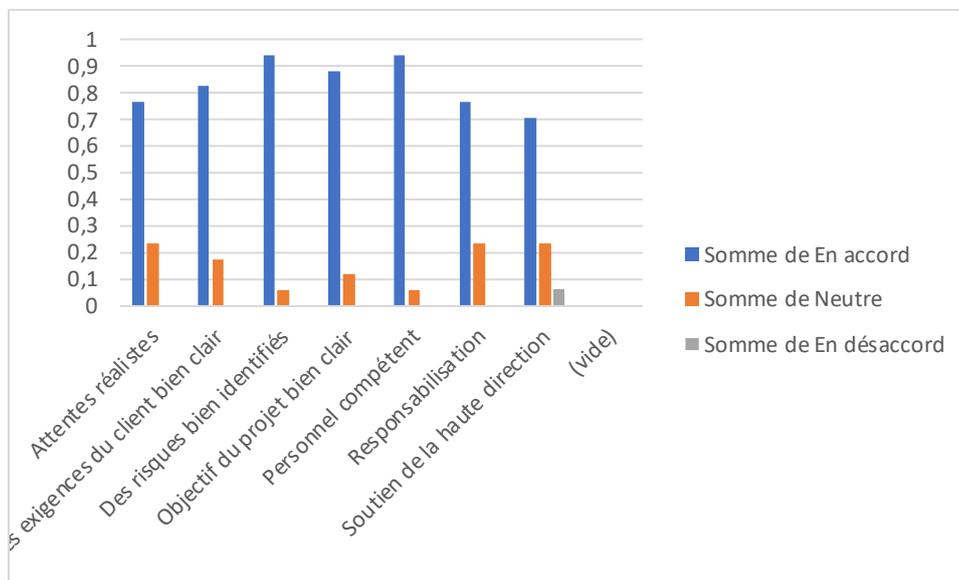


‘Personnel compétent’ est le facteur ayant reçu plus de réponses ‘En accord’ (88,24%). Le facteur ayant reçu le moins de réponses ‘En accord’ est le facteur lié à une bonne identification des risques (58,82%).

#### 1.18.1.5. *Maintenabilité*

Pour la maintenabilité, un nombre élevé de répondants sont en accord avec l’influence positive de nos sept facteurs. Seulement pour cet indicateur de qualité, les résultants des réponses en accord vont de 70,59% à 94,12 % pour les 7 facteurs. Ces réponses étant largement élevées par rapport aux réponses en désaccord (5,88%), nous pouvons donc confirmer que nos sept facteurs influencent positivement la maintenabilité d’un logiciel.

**Figure 13: Maintenabilité d'un logiciel**



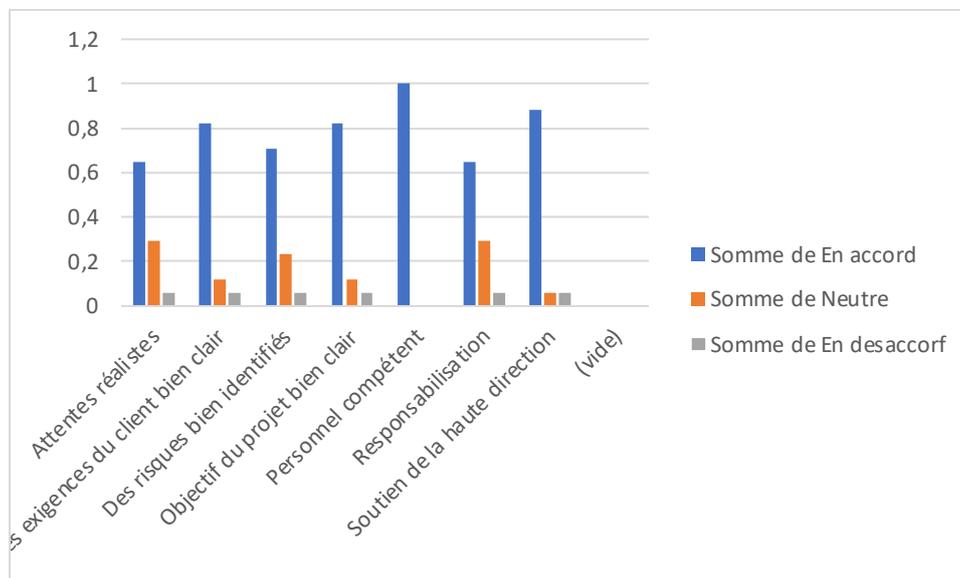
Les facteurs clés de succès avec plus de réponses ‘En accord’ sont : ‘Des risques bien identifiés’ et ‘Personnel compétent’. Tous ces deux facteurs ont reçu 94,12% de réponses ‘En accord’ chacun. ‘Soutien de la haute direction’ est le facteur ayant eu moins de réponses ‘En accord’ (70,59%).

#### 1.18.1.6. Réutilisabilité d'un logiciel

Les facteurs de l'étude sont également les mêmes que ceux énumérés dans la revue de littérature. Les résultats collectés permettent de retenir tous les sept facteurs comme des facteurs clés de succès capables d'influencer positivement la réutilisabilité d'un logiciel.

Entre 64,71 % et 100% de répondants sont en accord pour une influence positive de chacun des sept facteurs sur la réutilisabilité d'un logiciel. 5,88 sont en désaccord pour tous les facteurs sauf ‘Personnel compétent’ pour lequel personne n'est en désaccord. Ce facteur récolte 100% de réponses en accord.

**Figure 14: Réutilisabilité d'un logiciel**



Les résultats de l'analyse des données de la section 1 nous amènent à retenir nos 7 facteurs comme facteurs clés de succès de la qualité d'un logiciel. Nous pouvons donc tirer les conclusions suivantes :

Les Objectifs bien clairs ; des exigences du client bien claires ; des risques bien identifiés ; attentes réalistes ; responsabilisation personnel compétent et soutien de la haute direction influencent positivement la qualité d'un logiciel. (OB1).

### 1.18.2. Interprétation des résultats de l'objectif 2

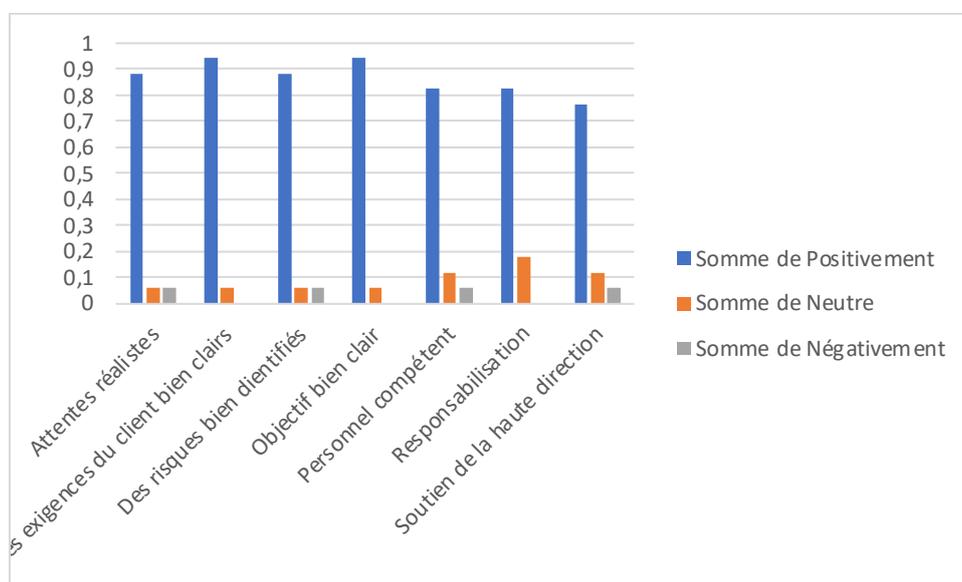
Chaque indicateur d'agilité est analysé par rapport aux sept facteurs clés de succès d'un logiciel. L'analyse de cette partie qui vise à identifier les indicateurs d'agilité capables de modérer les facteurs clés de succès pour une meilleure performance nous permet d'obtenir des résultats permettant d'infirmer ou de confirmer une quelconque modération.

Pour les sections 2 et 3, l'échelle simplifiée est utilisée pour l'interprétation: négativement, neutre ou positivement. L'indicateur d'agilité ayant obtenu au moins 51% de réponses positives sera considéré comme modérateur.

### 1.18.2.1. Collaboration client

Les résultats de l'analyse pour cet indicateur d'agilité montrent que la collaboration client modère tous les 7 facteurs clés de succès. Pour les 7 facteurs, nous avons des réponses positives entre 76,47% et 94,12%. Seulement 0 à 5,88% des répondants pensent que la collaboration client peut affecter négativement les relations entre les facteurs clés de succès et la qualité du logiciel.

**Figure 15: Collaboration client**

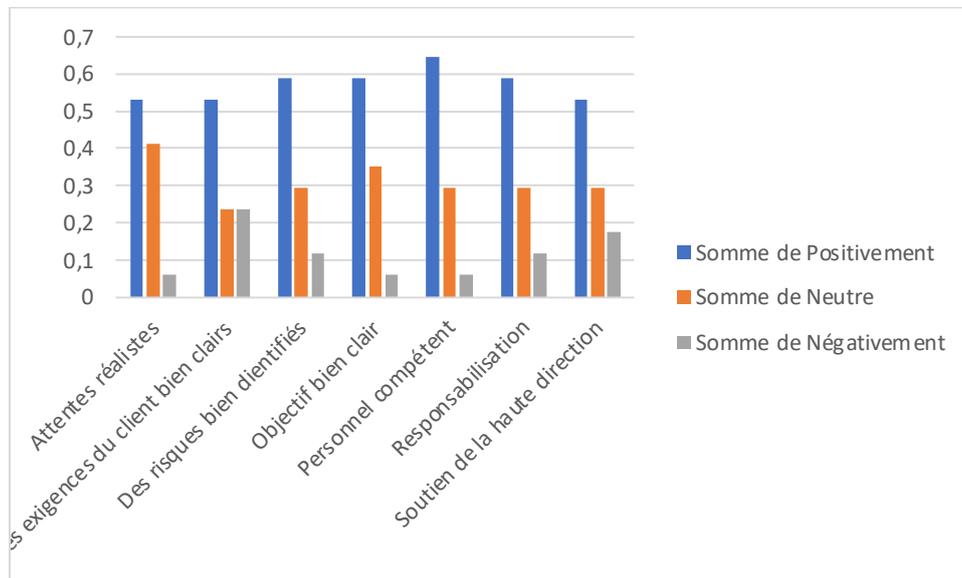


### 1.18.2.2. Équipe constituée de personnalités extraverties

Pour cet indicateur d'agilité, la majorité des répondants affirment une modération positive pour tous les sept facteurs clés de succès énumérés : entre 52,94% et 64,71%.

Nous pouvons donc confirmer une modération de nos sept facteurs clés de succès par une 'équipe constituée de personnalités extraverties'.

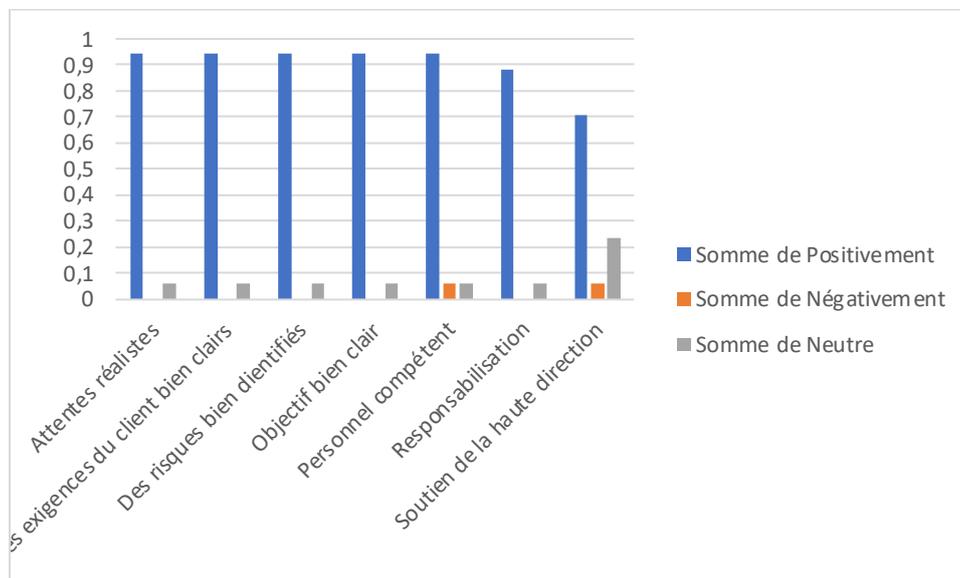
**Figure 16: Équipe constituée de personnalités extraverties**



### 1.18.2.3. Développement itératif

L'analyse des résultats pour cet indicateur d'agilité nous permet de retenir le développement itératif comme facteur modérateur pour les sept facteurs clés de succès. Entre 70,59% et 94,12% de répondants ont répondu positivement pour une modération. Seulement 5,88% sont contre cet avis.

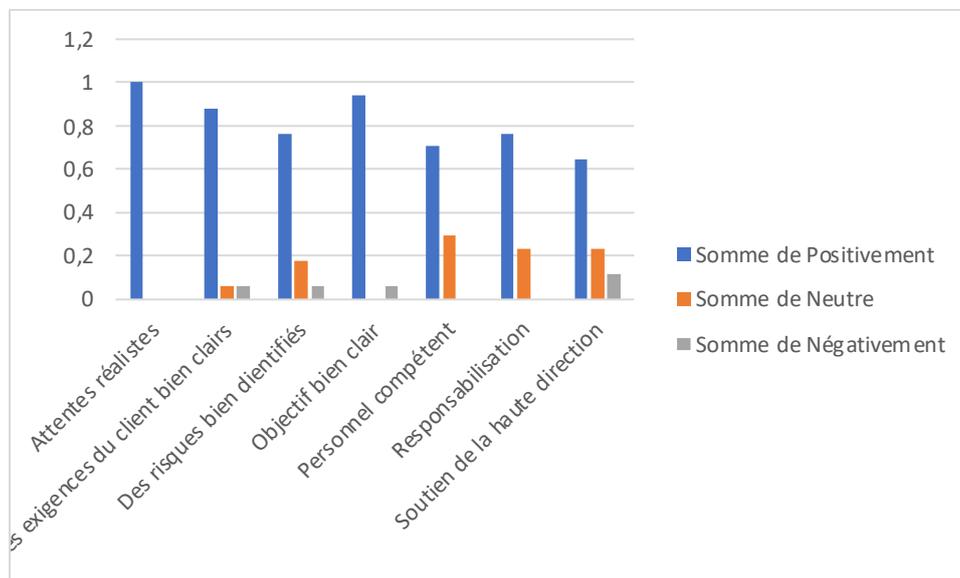
**Figure 17: Développement itératif**



#### 1.18.2.4. Favorable aux changements

L'analyse des résultats de la question 2.4 de notre questionnaire nous permet de valider 'favorable aux changements' comme modérateur pour nos sept facteurs clés de succès cités dans la revue de littérature. La majorité des répondants sont d'accord pour une modulation positive de cet indicateur d'agilité sur les sept facteurs : entre 64,71% et 100%. 100% de répondants affirment une modulation positive sur la relation entre le facteur 'attentes réalistes' et la qualité d'un logiciel.

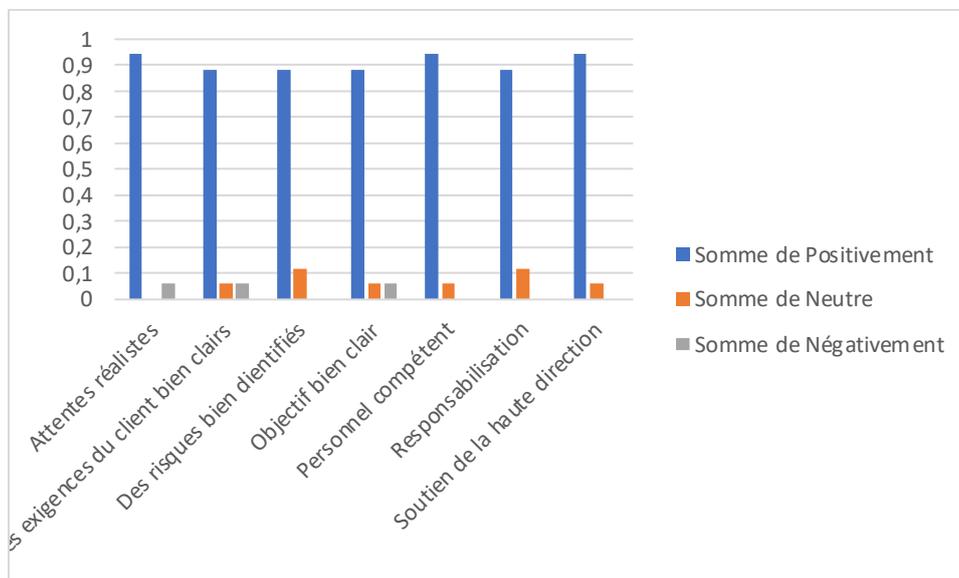
**Figure 18: Favorable aux changements**



#### 1.18.2.5. Axé sur la satisfaction du client

Les résultats remarquables faits sur cet indicateur, à l'aide de la formule présentée à la section 4.2, montrent que l'indicateur d'agilité 'axé sur la satisfaction du client' modère nos 7 facteurs clés de succès dans le but d'atteindre la performance d'un logiciel. Entre 88,24% et 94,12% de réponses sont pour une modération positive.

**Figure 19: Axé sur la satisfaction du client**



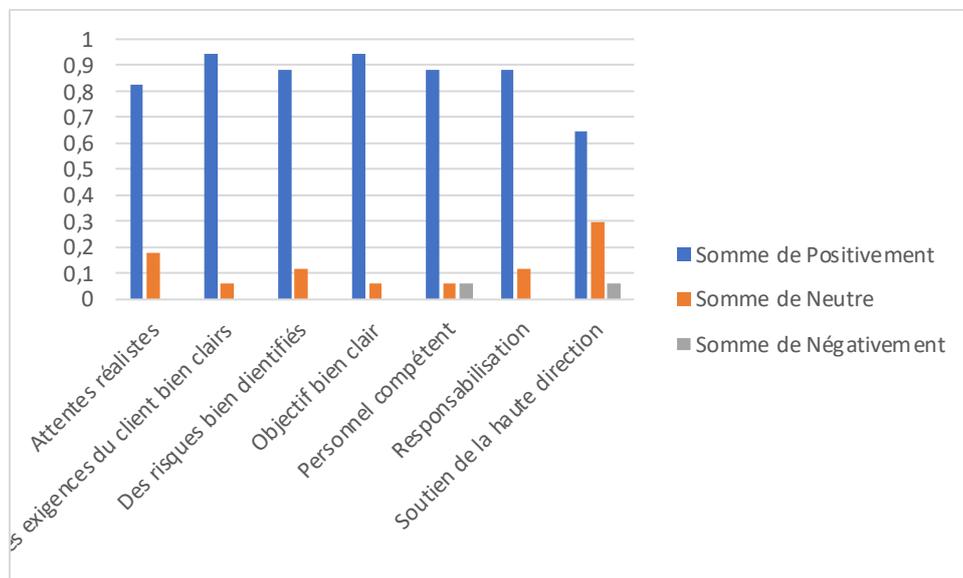
On conclut qu'il y a une modération de la relation entre facteurs clés de succès et performance par l'agilité. Tous les indicateurs d'agilité sont validés pour tous les sept facteurs clés de succès cités dans la revue de littérature. L'objectif pour cette section est donc atteint.

### 1.18.3. Interprétation des résultats de l'objectif 3

L'Assurance Qualité est étudiée dans cette section comme facteur modérateur de la relation entre facteurs clés de succès et performance. La formule énoncée dans la section 4.2 est toujours utilisée pour valider une certaine modération. Les résultats nous permettront de confirmer ou d'infirmer une modération de l'assurance qualité sur tous les sept facteurs clés de succès.

Les résultats de cette étude nous permettent de confirmer l'Assurance Qualité comme modérateur des sept facteurs clés de succès pour une performance des logiciels. Pour tous les sept facteurs, nous avons au moins 64,71% de répondants qui confirment une modération positive. En général, entre 64,71% et 94,12%.

**Figure 20: Assurance Qualité**



## CONCLUSION ET CONTRIBUTION

### 1.19. SYNTHÈSE DES RÉSULTATS

L'analyse des données collectées à l'aide de notre questionnaire de recherche permet de dresser une liste de facteurs clés de succès à privilégier pendant le processus de gestion de projet logiciel pour augmenter le plus possible la qualité du livrable. Aussi, cette analyse nous permet d'énumérer les facteurs pouvant modérer la relation entre facteurs clés de succès et performance.

Ce travail de recherche a été un meilleur moyen pour atteindre nos objectifs qui sont de trouver les facteurs clés de succès pouvant influencer positivement la gestion de projet logiciel pour assurer et accroître la qualité des livrables, et trouver les facteurs modérateurs pour atteindre la performance. Les réponses collectées des 17 répondants sont en général cohérentes et l'analyse des résultats nous permet de tirer des conclusions suivantes :

#### **Facteurs clés de succès :**

Les facteurs clés de succès de la gestion de projet en développement de logiciel sont :

- Objectif du projet bien clair
- Des exigences du client bien claires
- Des risques bien identifiés
- Attentes réalistes
- Responsabilisation
- Personnel compétent
- Soutien de la haute direction.

Bien que d'autres facteurs pourraient aussi favoriser une meilleure qualité, seulement ces sept facteurs jugés pertinents sont étudiés. Il est nécessaire de prendre en considération cette méconnaissance pour réfléchir sur d'autres solutions aussi possibles.

#### **Influence de l'agilité sur la relation facteur clés de succès et performance :**

L'analyse de l'influence de l'agilité sur la relation entre facteurs clés de succès et la performance montre que l'agilité modère positivement la relation entre les 7 facteurs clés de succès énumérés et la performance. Nous pouvons donc conclure que l'agilité modère positivement la relation entre facteurs clés de succès et performance en gestion de projet logiciel.

### **Influence de l'Assurance Qualité sur la relation facteurs clés de succès et performance :**

L'analyse de l'influence de l'Assurance qualité sur la relation entre facteurs clés de succès et la performance nous permet de tirer la conclusion suivante : L'Assurance Qualité modère positivement la relation entre facteurs clés de succès et la performance.

Les résultats de cette recherche montrent que les objectifs sont atteints avec succès. Ce qui prouve que la population étudiée est fiable et le terrain exploité est favorable aux recherches futures sur les trois domaines exploités : Gestion de projet, Génie logiciel et Qualité.

## **1.20. CONTRIBUTIONS ET RECOMMANDATIONS**

Cette étude de recherche est tout d'abord utile à un gestionnaire de projet, car elle contribue à l'amélioration de la gestion de projet de développement de logiciel. Elle s'adresse aussi aux programmeurs ou développeurs des logiciels tout en mettant l'accent sur les facteurs importants à prendre en compte lors de la programmation afin de produire un livrable de qualité, respectant toutes les exigences.

Dans le but d'accroître la qualité des logiciels et d'après l'analyse des résultats, les gestionnaires de projets doivent prendre en considération certaines recommandations :

- Bien comprendre les objectifs du projet au début afin de bien cerner le périmètre du projet
- Bien comprendre les exigences du client pour développer un logiciel adapté à ses besoins.

- Identifier les éventuels risques du projet et mettre en place des solutions pour les éviter, les minimiser ou les atténuer.
- S'assurer que les attentes du projet sont réalistes et éviter les attentes impossibles à réaliser.
- S'assurer que chaque tâche est responsabilisée
- Le personnel joue un rôle important dans la réalisation du projet. Raison pour laquelle il est très recommandé de choisir des personnes ayant les connaissances et l'expérience requise.
- Les résultats de cette étude montrent aussi que la direction est une ressource remarquable pour le projet. Il est alors recommandé aux gestionnaires de projets de l'impliquer au projet et obtenir son soutien.
- Les résultats de la recherche montrent une modération de la relation entre les facteurs clés de succès et la performance par l'agilité et l'assurance qualité. En se basant sur la revue de littérature et des résultats obtenus à travers les questionnaires de recherche, la recommandation est d'opter pour une méthodologie agile qui est très favorable aux facteurs clés de succès et capable d'accroître la qualité du livrable. En s'assurant de respecter les recommandations de l'assurance qualité, on augmente la qualité du logiciel et sa performance.

### 1.21. LIMITATIONS

Malgré ses contributions, cette étude présente certaines limitations qui doivent être prises en compte. Premièrement, notre échantillon est limité à trois pays : le Canada pour la plupart, la France et les États-Unis, ce qui pourrait limiter la généralisation.

De plus, bien que cette étude étudie les facteurs clés de succès en projet logiciel, il est important de noter que certains facteurs autres que ceux qui sont énumérés pourraient influencer positivement la qualité du livrable.

### 1.22. RECHERCHES FUTURES

Les résultats de cette étude ouvrent la voie à des pistes de recherches futures prometteuses dans le domaine de la gestion des projets de développement de logiciel.

En se basant sur les contributions et les limitations de cette recherche, plusieurs domaines pourraient être approfondis pour améliorer la compréhension et la performance des projets.

Les futures études pourraient premièrement se baser sur des échantillons plus larges en touchant plus de pays. De plus, elles pourraient approfondir les facteurs clés de succès pour chaque méthodologie agile. Et elles pourraient s'intéresser davantage à d'autres facteurs pouvant accroître la qualité dans les projets logiciels et augmenter la chance d'atteindre la performance.

## RÉFÉRENCES

- 2000, G. (2021, May 13,2021). How The World's Biggest Public Companies Endured The Pandemic.
- Akbar, M. A., Shafiq, M., Ahmad, J., Mateen, M., & Riaz, M. T. (2018). *AZ-Model of software requirements change management in global software development*. Paper presented at the 2018 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube).
- Akbar, M. A., Shameem, M., Ahmad, J., Maqbool, A., & Abbas, K. (2018). *Investigation of Project Administration related challenging factors of Requirements Change Management in global software development: A systematic literature review*. Paper presented at the 2018 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube).
- Anand, R. V., & Dinakaran, M. (2016). Popular agile methods in software development: Review and analysis. *International Journal of Applied Engineering Research*, 11(5), 3433-3437.
- Basu, R. (2011). *Fit sigma : a lean approach to building sustainable quality beyond Six Sigma*. Chichester: Wiley.
- Basu, R. (2014). Managing quality in projects: An empirical study. *International Journal of Project Management*, 32(1), 178-187. doi:10.1016/j.ijproman.2013.02.003
- Beckford, J. (2002). *Quality* [1 ressource en ligne (xxi, 328 pages) : ill](2nd ed.).
- Bittner, K., & Spence, I. (2006). *Managing iterative software development projects*: Addison-Wesley Professional.
- Blackburn, M. R., Busser, R., & Nauman, A. (2001). Removing Requirement Defects and Automating Test. *STAREAST, May*.
- Boehm, B. W., & Turner, R. (2004). *Balancing agility and discipline: A guide for the perplexed*: Addison-Wesley Professional.
- Bourke, J., & Roper, S. (2017). Innovation, quality management and learning: Short-term and longer-term effects. *Research Policy*, 46(8), 1505-1518. doi:10.1016/j.respol.2017.07.005
- Bradley, J. H., & Hebert, F. J. (1997). The effect of personality type on team performance. *Journal of Management Development*, 16(5), 337-353.
- Campanelli, A. S., & Parreiras, F. S. (2015). Agile methods tailoring – A systematic literature review. *Journal of Systems and Software*, 110, 85-100.
- Chang, W., & Taylor, S. A. (2016). The effectiveness of customer participation in new product development: A meta-analysis. *Journal of Marketing*, 80(1), 47-64.
- Chaurasia, P. K. (2010). How Accountability Improves Software Reliability. *International Journal of Computer Science and Technology (IJCSET)*, 5(9), 868-871.
- Chitchyan, R., Bird, C., & Ieee 28th International Requirements Engineering Conference Zurich, S. A. S. (2020). Theory as a Source of Software

- Requirements. In *2020 IEEE 28th International Requirements Engineering Conference (RE)* (pp. 227-237): IEEE.
- Chowdhury, A. A. M., & Arefeen, S. (2011). Software risk management: importance and practices. *IJCIT, ISSN, 2078-5828*.
- cirano. (2020). *Le Québec économique 9 - Perspectives et défis de la transformation numérique*.
- Cooper, R. G., & Kleinschmidt, E. J. (2011). *New products: The key factors in success*: Marketing Classics Press.
- Dathan, R., & Menon, R. (2010). Managing Testing Projects -- the SCRUM way. *Journal of the Quality Assurance Institute, 24(2)*, 18-19.
- David, H. (2018). *PRINCE2 Study Guide : 2017 Update* (Vol. Second edition -- 2017 update). Indianapolis: Sybex.
- Doran, G. T. (1981). There's a SMART way to write management's goals and objectives. *Management review, 70(11)*, 35-36.
- Durham, D., & Michel, C. (2021). *Lean software systems engineering for developers : managing requirements, complexity, teams, and change like a champ* [1 online resource (214 pages) : illustrations]. doi:10.1007/978-1-4842-6933-6
- Espinosa, J. A., Slaughter, S. A., Kraut, R. E., & Herbsleb, J. D. (2007). Team knowledge and coordination in geographically distributed software development. *Journal of management information systems, 24(1)*, 135-169.
- Faella, M., & Safari, a. O. R. M. C. (2020). *Seriously Good Software* [1 online resource (328 pages)](1st edition. ed.). Retrieved from
- Fagarasan, C., Popa, O., Pislă, A., & Cristea, C. (2021). *Agile, waterfall and iterative approach in information technology projects*. Paper presented at the IOP Conference Series: Materials Science and Engineering.
- Faraj, S., & Sproull, L. (2000). Coordinating Expertise in Software Development Teams. *Management Science, 46(12)*, 1554-1568. Retrieved from
- Galin, D. (2018). *Software quality : concepts and practice* [1 online resource (xxviii, 680 pages)].
- Geraldi, J. G., Kutsch, E., & Turner, N. (2011). Towards a conceptualisation of quality in information technology projects. *International Journal of Project Management, 29(5)*, 557-567. doi:10.1016/j.ijproman.2010.06.004
- Gorla, N., & Lam, Y. W. (2004). Who Should Work with Whom? Building Effective Software Project Teams. *Communications of the ACM., 47(6)*, 79.
- Heisler, S. I. (1990). Project quality and the project manager. *International Journal of Project Management, 8(3)*, 133-137.
- Hussain, A., Mkpojiogu, E. O., & Kamal, F. M. (2016). The role of requirements in the success or failure of software projects. *International Review of Management and Marketing, 6(7)*, 306-311.
- IEEE. (2023). *2749-2023 - IEEE Recommended Practice for Risk Identification and Evaluation of Smart Power Distribution System* [1 online resource].
- Incorporated, I. o. E. a. E. E. (2014). *730-2014 IEEE Standard for Software Quality Assurance Processes* [1 online resource].
- Juran, J. M. (1989). *Juran on leadership for quality : an executive handbook*. New York: Free Press.

- Kamal, T., Zhang, Q., Akbar, M. A., Shafiq, M., Gumaiei, A., & Alsanad, A. (2020). Identification and prioritization of agile requirements change management success factors in the domain of global software development. *IEEE Access*, 8, 44714-44726.
- Keil, M., Cule, P. E., Lyytinen, K., & Schmidt, R. C. (1998). A framework for identifying software project risks. *Communications of the ACM*, 41(11), 76-83.
- Kerievsky Joshua. (2016). An Introduction to Modern Agile. *InfoQ. Web*.
- Koch, A. S. (2005). *Agile software development : evaluating the methods for your organization* [1 online resource (xxviii, 272 pages) : illustrations].
- Komal, B., Janjua, U. I., Anwar, F., Madni, T. M., Cheema, M. F., Malik, M. N., & Shahid, A. R. (2020). The Impact of Scope Creep on Project Success: An Empirical Investigation. *IEEE Access*, 8, 125755-125775. doi:10.1109/ACCESS.2020.3007098
- Laporte, C. Y., & April, A. (2018). *Software quality assurance* [1 online resource](1. ed.). doi:10.1002/9781119312451
- LAROUSSE (2001). Larousse. In: VUEF.
- Long, T. (2021). *Good code, bad code : think like a software engineer* [1 online resource (xx, 353 pages) : illustrations].
- Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). Empirical study of agile software development methodologies: A comparative analysis. *ACM SIGSOFT Software Engineering Notes*, 40(1), 1-6.
- Measey, P. (2015). *Agile foundation : principles, practices and frameworks* [1 online resource (xviii, 179 pages) : illustrations].
- Nurmuliani, N., Zowghi, D., & Powell, S. (2004). *Analysis of requirements volatility during software development life cycle*. Paper presented at the 2004 Australian Software Engineering Conference. Proceedings.
- Phillip A. Laplante, M. H. K. (2022). Requirements Engineering for Software and Systems. *Auerbach Publications*, 428.
- Project Management, I. (2017). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)–Sixth Edition* (Vol. Sixth edition). Newtown Square, PA: Project Management Institute.
- Project Management Institute. (2008). *A guide to the project management body of knowledge (PMBOK Guide)* [1 online resource (xxvi, 467 pages) : illustrations](4th ed.). Retrieved from
- Project Management Institute. (2009). Les cinq principales causes du glissement de la portée ... et que faire à leur sujet. *PMBOK Guide*. Retrieved from
- Project Management Institute. (2021). *The standard for project management and a guide to the project management body of knowledge (PMBOK guide)* [1 online resource (xxvi, 274 pages) : illustrations (some color)](Seventh edition. ed.). Retrieved from
- Reifer, D. J. (2013). *Software war stories : case studies in software management* [1 online resource]. Retrieved from

- Rico, D. F., & PMP, C. (2008). What is the ROI of Agile vs. Traditional Methods? *Tickit international*, 10(4), 9-18.
- Riesener, M., Doelle, C., Perau, S., Lossie, P., & Schuh, G. (2021). Methodology for iterative system modeling in agile product development. *Procedia CIRP*, 100, 439-444.
- S. L. Tang, S. M. A. R. T. A., & Poon, S. W. (2005). *Construction Quality Management*. Hong Kong: Hong Kong University Press.
- Schuh, G. (2017). *Produktkomplexität managen: Strategien-Methoden-Tools*: Carl Hanser Verlag GmbH Co KG.
- Snedaker, S., & Hoenig, N. (2005). *How to cheat at IT project management* [1 online resource (xxvi, 565 pages) : illustrations]. Retrieved from
- Sommerville, I., Rodden, T., Sawyer, P., Bentley, R., Twidale, M., & Proceedings of the Ieee International Symposium on Requirements Engineering San Diego, C. A. U. S. A. J. J. (1993). Integrating ethnography into the requirements engineering process. In *Proceedings of the IEEE International Symposium on Requirements Engineering* (pp. 165-173).
- Villafiorita, A. (2014). *Introduction to software project management* [1 online resource (xix, 268 pages) : illustrations]. Retrieved from
- Vogt, W. P. (2014). *Selecting the right analyses for your data : quantitative, qualitative, and mixed methods* [1 online resource (xix, 500 pages)]. Retrieved from
- Weber, P., Medina-Oliva, G., Simon, C., & Iung, B. (2012). Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas. *Engineering Applications of Artificial Intelligence*, 25(4), 671-682.
- White, K. B. (1984). MIS Project Teams: An Investigation of Cognitive Style Implications. *MIS Q.*, 8, 95-101.
- Wysocki, R. K. (2006). *Effective software project management* [1 online resource (xlix, 618 pages) : illustrations]. Retrieved from
- Yellen, R. E., Winniford, M., & Sanford, C. C. (1995). Extraversion and introversion in electronically-supported meetings. *Information & Management*, 28(1), 63-74.
- Young, R., & Jordan, E. (2008). Top management support: Mantra or necessity? *International Journal of Project Management*, 26(7), 713-725.

## **ANNEXE 1 : QUESTIONNAIRE DE RECHERCHE**

### **QUESTIONS DE RECHERCHES**

Thème : Gestion de la qualité des projets en Génie logiciel : facteurs clés et solutions

Étudiant : Issaka Gmakouba

## Partie A- Objectifs et consignes

### OBJECTIFS :

Pour appréhender les relations existantes entre les différents facteurs de notre cadre conceptuel, nous avons élaboré une série de questionnaires. Le but est de mieux cerner les variables des projets de développement de logiciel afin d'augmenter le succès et la performance de ces derniers.

### CONSIGNES :

Ce questionnaire est strictement anonyme et destiné aux gestionnaires et parties prenantes des projets de développement de logiciel. Afin d'obtenir des résultats exploitables, nous requerrons votre collaboration, en vous basant sur vos expériences et connaissances professionnelles.

Il n'engage en aucun cas ni l'institution ni le répondant. Les données brutes collectées à l'aide de ce questionnaire seront placées sous le sceau de confidentialité et les données individuelles seront conservées et ne seront en aucun cas distribuées ou publiées. S'il vous plaît essayez de répondre aux questions tout en gardant à l'esprit qu'il n'y a ni de réponse vraie ni fausse, c'est votre expérience et vos connaissances qui comptent. Vous pouvez renoncer librement à ce questionnaire sans préjudice.

Nous vous remercions pour votre compréhension et votre collaboration!

**Partie B-** Les données démographiques sur l'organisation et sur le répondant

1.1- Secteur d'activité :

- Petite enfance
- Emploi
- Insertion sociale, intégration sociale et professionnelle
- Environnement
- Médias et communication,
- Promotion et défense des droits sociaux
- Habitation et logement
- Action bénévole
- Santé et services sociaux immigration et intégration sociale des communautés
- Culturelles
- Agriculture
- Lutte contre la pauvreté
- Marketing
- Informatique
- Enseignement
- Restauration
- Condition féminine
- Dépendances
- Personnes handicapées
- Prévention de la violence
- Santé mentale
- Santé physique
- Autre (à préciser)
- Je préfère ne pas répondre

1.2- Durée moyenne d'un projet

- Inférieur à 1an
  - Entre 1 et 2 ans
  - Plus de 3 ans
  - Je préfère ne pas répondre
- 1.3- Budget moyen d'un projet
- Moins de 50 k\$
  - Entre 50 k\$ et 199k\$
  - Entre 200 k\$ et 499 k\$
  - Entre 500k\$ et 999k\$
  - Plus de 1 000 000 \$
  - Je préfère ne pas répondre
- 1.4- Nombre d'employés dans l'organisation :
- 0 à 4 employés
  - 5 à 19 employés
  - 20 à 49 employés
  - 50 à 99 employés
  - 100 à 299 employés
  - 300 à 499 employés
  - 500 employés et plus
  - Je préfère ne pas répondre
- 1.5- Taille de votre équipe projet
- 1 à 3 personnes
  - 3 à 5 personnes
  - 5 à 9 personnes
  - 10 à 19 personnes
  - 20 personnes et plus
  - Je préfère ne pas répondre
- 1.6- Durée moyenne de votre expérience comme membre d'une équipe de projet logiciel
- Moins d'1an
  - 1 à 3ans

- 4 à 7ans
  - 7 ans et plus
  - Je préfère ne pas répondre
- 1.7- Compétences technologiques
- Mauvaise
  - Moyenne
  - Bonne
  - Très bonne
  - Excellente
  - Je préfère ne pas répondre
- 1.8- Poste souvent occupé au sein de l'équipe projet (sélectionnez toutes les réponses qui s'appliquent à vous)
- Gestionnaire de projet
  - Coordonnateur
  - Membre de la direction
  - Développeur/ programmeur
  - Responsable assurance qualité
  - Chef d'équipe
  - Informaticien (autre qu'un développeur/programmeur)
  - Autre (à préciser)
  - Je préfère ne pas répondre

Partie C- Définition de quelques concepts utilisés dans nos questionnaires  
(nos facteurs clés de succès de projet logiciel)

#### OBJECTIFS DU PROJET BIEN CLAIRS

L'objectif du projet correspond à la description du résultat ou de la solution qu'on souhaite obtenir. L'établissement d'objectifs pour un projet décrit les effets et les résultats à long terme attendus du projet. Il est important de décrire ces objectifs le plus tôt possible, car ils peuvent éclairer chaque décision dans le cycle de vie du projet.

#### DES EXIGENCES DU CLIENT BIEN CLAIRES

Les exigences du client sont des particularités ou des précisions sur le livrable qu'on souhaite obtenir. Il est important de respecter ces exigences afin de pouvoir satisfaire aux besoins du client.

#### DES RISQUES BIEN IDENTIFIÉS

Dans tout projet il est important d'identifier les éventuels risques, et leurs apparitions afin de prendre des dispositions pour soit les éviter, soit les atténuer.

#### ATTENTES RÉALISTES

Les attentes du projet sont des idées que nous pouvons avoir sur le produit attendu, sur sa qualité et dans quelle mesure nous espérons le réaliser. Pour ne pas décevoir le client ou soi-même ou encore pour éviter trop de changements au cours du projet, il est important de définir des attentes réalistes et pas d'exagérations.

#### PERSONNEL COMPÉTENT

L'équipe projet est un facteur déterminant dans la réussite du projet. Des personnes compétentes sont des personnes ayant des connaissances et des expériences suffisantes. Avoir des personnes compétentes augmente les chances de réussite du projet.

#### SOUTIEN DE LA HAUTE DIRECTION

Le soutien de la direction est tout support ou tous les actes que la direction peut faire pour vous donner l'aide nécessaire à la réussite du projet. Cela comprend les participations à quelques réunions, des motivations, participation à l'établissement du plan stratégique, une mise à disposition de tout moyen matériel, humain, stratégique nécessaire pour la réussite du projet.

Partie D- Les Questionnaires de recherches

**SECTION 1 : L'influence des facteurs clés de succès sur la performance**

OBJECTIF DE LA SECTION:

Comprendre l'influence des facteurs clés de succès des projets logiciels sur la performance. Nous souhaitons mesurer les facteurs clés de succès pour chaque indicateur de performance.

Voici nos indicateurs de performance pour les projets logiciels : Efficacité, Exactitude, Robustesse, Convivialité, Maintenabilité, Réutilisabilité.

L'EFFICACITÉ :

L'efficacité d'un logiciel est son adéquation par rapport à ses objectifs et à certains critères qui lui sont propres. Il s'agit du temps d'exécution, de la consommation des données (en mémoire) et la consommation de la bande passante. Un logiciel rapide (temps d'exécution court) permet de résoudre plus rapidement les problèmes et de gagner du temps.

**1.1- Selon vous, chaque facteur clé de succès a-t-il d'influence sur l'efficacité d'un logiciel ?**

FCS	Fortement en désaccord	En désaccord	Pas d'influence	En accord	Fortement en accord
Objectifs du projet bien clair					
Des exigences du client bien clair					
Des risques bien identifiés					
Attentes réalistes					
Responsabilisation					
Personnel compétent					
Soutien de la haute direction					

### L'EXACTITUDE

L'exactitude d'un logiciel est sa capacité à répondre aux besoins du client, c'est-à-dire un logiciel capable de résoudre les problèmes pour lesquels il est conçu. Cette qualité se vérifie souvent par des tests afin de déterminer si les résultats sont exacts ou pas.

**1.2- Selon vous, chaque facteur clé de succès a-t-il d'influence sur l'exactitude d'un logiciel ?**

FCS	Fortement en désaccord	En désaccord	Pas d'influence	En accord	Fortement en accord
Objectifs du projet bien clair					
Des exigences du client bien clair					
Des risques bien identifiés					
Attentes réalistes					
Responsabilisation					
Personnel compétent					
Soutien de la haute direction					

### ROBUSTESSE

La robustesse du logiciel est sa capacité à résister aux éventuels problèmes. Par exemple face à un problème de coupure de courant, il doit être capable de sauvegarder les données. C'est aussi le cas d'un logiciel qui résiste face aux piratages.

**1.3- Selon vous, chaque facteur clé de succès a-t-il d'influence sur la robustesse d'un logiciel ?**

FCS	Fortement en désaccord	En désaccord	Pas d'influence	En accord	Fortement en accord
Objectifs du projet bien clair					
Des exigences du client bien clair					
Des risques bien identifiés					
Attentes réalistes					
Responsabilisation					
Personnel compétent					
Soutien de la haute direction					

### CONVIVIALITÉ

La convivialité du logiciel est liée à la conception de son interface utilisateur (UI) et est abordée par des disciplines telles que l'interaction homme-machine et la conception de l'expérience utilisateur (UX). Il est intéressant d'avoir un logiciel qui fonctionne, mais aussi qui est simple à comprendre et à utiliser.

**1.4- Selon vous, chaque facteur clé de succès a-t-il d'influence sur la convivialité d'un logiciel ?**

FCS	Fortement en désaccord	En désaccord	Pas d'influence	En accord	Fortement en accord
Objectifs du projet bien clair					
Des exigences du client bien clair					
Des risques bien identifiés					
Attentes réalistes					
Responsabilisation					
Personnel compétent					
Soutien de la haute direction					

#### MAINTENABILITÉ

C'est un facteur clé d'un logiciel de qualité qui indique la facilité à laquelle le logiciel peut être maintenu. Un logiciel qui répond à ce critère rend plus simples la localisation des bogues et la facilité de la maintenance.

**1.5- Selon vous, chaque facteur clé de succès a-t-il d'influence sur la maintenabilité d'un logiciel ?**

FCS	Fortement en désaccord	En désaccord	Pas d'influence	En accord	Fortement en accord
Objectifs du projet bien clair					
Des exigences du client bien clair					
Des risques bien identifiés					
Attentes réalistes					
Responsabilisation					
Personnel compétent					
Soutien de la haute direction					

### RÉUTILISABILITÉ

La réutilisabilité est une qualité d'un logiciel, pouvant lui permettre d'être utilisé et réutilisé au besoin. C'est normal qu'un logiciel qui n'est pas utilisable ne serve à rien, car ne pouvant résoudre un problème. S'assurer que le code continue de fonctionner est l'une des plus grandes considérations auxquelles les ingénieurs logiciels sont confrontés et il doit être pris en compte à toutes les étapes du codage.

**1.6- Selon vous, chaque facteur clé de succès a-t-il d'influence sur la réutilisabilité d'un logiciel ?**

FCS	Fortement en désaccord	En désaccord	Pas d'influence	En accord	Fortement en accord
Objectifs du projet bien clair					
Des exigences du client bien clair					
Des risques bien identifiés					
Attentes réalistes					
Responsabilisation					
Personnel compétent					
Soutien de la haute direction					

<b>SECTION 2 : La relation Facteurs clés de succès et Performance est plus modérée dans un environnement agile</b>
--

## AGILITÉ

L'agilité est la capacité à la fois de créer et de répondre aux changements afin de tirer profit d'un environnement commercial turbulent. L'agilité est une méthodologie de la gestion de projets avec la capacité d'équilibrer flexibilité et stabilité.

Notre objectif pour cette section est de connaître le degré de modération de l'agilité sur la relation entre facteurs clés de succès et meilleure performance.

## COLLABORATION CLIENT

La collaboration client est le fait de faire du client une ressource vraiment déterminante pour le succès du projet. Il y a une forte collaboration plutôt qu'un suivi de plan ou du contrat. Dès le début du projet, le client peut contribuer à une compréhension plus claire du projet et par conséquent permettre aux développeurs d'avoir une idée plus claire et détaillée du livrable voulu.

**2.1- D'après vous, la relation entre facteurs clés de succès  
et la performance est-elle plus modérée par la collaboration client ?**

FCS	Très négativement	Négativement	Neutre	Positivement	Très positivement
Objectifs du projet bien clair					
Des exigences du client bien clair					
Des risques bien identifiés					
Attentes réalistes					
Responsabilisation					
Personnel compétent					
Soutien de la haute direction					

**1.22.1. ÉQUIPE DE PERSONNALITÉS EXTRAVERTIES**

Les personnes extraverties sont des personnes sociables et aiment interagir avec les autres contrairement aux personnes introverties qui ont souvent tendance à garder l'information pour elles-mêmes et sont moins communicatives lors des réunions.

**2.2- D'après vous, la relation entre facteurs clés de succès et la performance est-elle plus modérée avec équipe projet constituée majoritairement de personnalités extraverties ?**

FCS	Très négativement	Négativement	Neutre	Positivement	Très positivement
Objectifs du projet bien clair					
Des exigences du client bien clair					
Des risques bien identifiés					
Attentes réalistes					
Responsabilisation					
Personnel compétent					
Soutien de la haute direction					

**PRIVILÉGIER LE DÉVELOPPEMENT ITÉRATIF**

Le développement itératif est une approche de développement logiciel qui divise le processus de développement d'une grande application en partie plus petite. Des modifications de conception et de nouvelles fonctionnalités sont ajoutées à chaque itération. Chaque partie « itératif » représente l'ensemble du processus de développement et contient les étapes de planification, de conception, de développement et de test.

**2.3- D'après vous, la relation entre facteurs clés de succès et la performance est-elle plus modérée en privilégiant un développement itératif ?**

FCS	Très négativement	Négativement	Neutre	Positivement	Très positivement
Objectifs du projet bien clair					
Des exigences du client bien clair					
Des risques bien identifiés					
Attentes réalistes					
Responsabilisation					
Personnel compétent					
Soutien de la haute direction					

**FAVORABLE AUX CHANGEMENTS**

Contrairement à la méthodologie traditionnelle, l'agilité accueille positivement les exigences changeantes même tardivement dans le développement. Les processus agiles exploitent le changement pour l'avantage concurrentiel du client.

**2.4- D'après vous, la relation entre facteurs clés de succès et la performance est-elle plus modérée en accueillant favorablement les changements au cours du projet?**

FCS	Très négativement	Négativement	Neutre	Positivement	Très positivement
Objectifs du projet bien clair					
Des exigences du client bien clair					
Des risques bien identifiés					
Attentes réalistes					
Responsabilisation					
Personnel compétent					
Soutien de la haute direction					

**AXÉ SUR LA SATISFACTION DU CLIENT**

La satisfaction du client est la priorité absolue en agilité. C'est le premier principe des 12 du manifeste agile. Il s'agit de la considération des besoins et exigences du client afin de lui fournir un livrable comme voulu.

**2.5- D'après vous, les facteurs clés de succès sont-ils plus modérés en priorisant la satisfaction du client ?**

FCS	Très négativement	Négativement	Neutre	Positivement	Très positivement
Objectifs du projet bien clair					
Des exigences du client bien clair					
Des risques bien identifiés					
Attentes réalistes					
Responsabilisation					
Personnel compétent					
Soutien de la haute direction					

**SECTION 3 : L'Assurance Qualité modère la relation entre facteurs clés de succès et la performance**

Notre objectif pour cette section est de connaître le degré de modération de la relation entre facteurs clés de succès et performance grâce à l'assurance qualité

**ASSURANCE QUALITÉ LOGICIEL**

L'Assurance Qualité Logiciel est un ensemble d'activités qui définissent et évaluent l'adéquation des processus logiciels pour fournir des preuves qui établissent la confiance que les processus logiciels sont appropriés et produisent des produits logiciels de qualité appropriée pour leurs objectifs.

**3- D'après vous, la relation entre facteurs clés de succès et la performance est-elle plus modérée avec l'assurance qualité ?**

FCS	Très négativement	Négativement	Neutre	Positivement	Très positivement
Objectifs du projet bien clair					
Des exigences du client bien clair					
Des risques bien identifiés					
Attentes réalistes					
Responsabilisation					
Personnel compétent					
Soutien de la haute direction					