

Projet d'étude.

Sujet :

Étude de l'impact de l'utilisation de différentes méthodologies de développement sur les performances des applications

Table des matières

Sujet :	1
I. Introduction :	2
a. Présentation du sujet :	2
b. Contexte et enjeux :	2
c. Objectif de l'étude :	3
D. L'histoire du développement.....	4
II. Etat de l'art :	5
III. Les méthodologies :	Erreur ! Signet non défini.
a. Agile	5
b. Itérative	7
c. Lean agile	8
d. Cascade.....	9
e. Spirale.....	10
f. En V	11
g. RAD.....	13
IV. Analyse.....	14
a. Les méthodes de développement des GAFAM.	14
b. Etudes menées.	16
V. Conclusion.....	20

Erreur ! Source du renvoi introuvable.Introduction :

a. Présentation du sujet :

Le sujet porte sur l'étude de l'impact de différentes méthodologies de développement sur les performances des applications. Il s'agit d'analyser comment différentes approches de développement logiciel peuvent influencer les performances d'une application, c'est-à-dire sa vitesse, sa réactivité et son efficacité globale.

Dans le domaine du développement logiciel, il existe plusieurs méthodologies telles que le développement en cascade, le développement agile, le développement itératif, etc. Chaque méthode a ses propres caractéristiques et processus spécifiques, qui peuvent avoir un impact significatif sur les performances finales de l'application développée.

b. Contexte et enjeux :

Au cours des dernières décennies, l'utilisation croissante des applications et des logiciels a généré une demande de plus en plus forte. Afin de répondre à cette demande, il est essentiel de développer des logiciels et des applications performants qui répondent aux besoins des utilisateurs. Pour atteindre cet objectif, il est primordial de comprendre comment optimiser le processus de développement. Plusieurs processus clés peuvent avoir un impact significatif sur cette optimisation.

Tout d'abord, la gestion de projet joue un rôle crucial. Une gestion de projet efficace implique la planification, l'organisation et la coordination des ressources et des tâches. En assurant une gestion rigoureuse du projet, il est possible de maximiser l'efficacité du processus de développement.

De plus, la communication transparente et la collaboration entre les membres de l'équipe de développement sont essentielles. Une communication efficace permet de partager des informations pertinentes, de résoudre rapidement les problèmes et de maintenir un alignement clair sur les objectifs du projet.

L'automatisation* est un autre aspect crucial de l'optimisation du processus de développement. En automatisant les tâches répétitives telles que les tests, la compilation et le déploiement, il est possible de réduire les erreurs humaines et d'accélérer les cycles de développement.

Enfin, la méthodologie utilisée tout au long du processus de développement aura un impact significatif sur le produit final. Des approches telles que Agile, Scrum ou Kanban offrent des cadres de travail structurés favorisant la collaboration, l'adaptabilité et la livraison continue, conduisant ainsi à des résultats optimisés.

En combinant une gestion de projet efficace, une communication fluide, une automatisation* appropriée et l'adoption d'une méthodologie adaptée, il est possible de maximiser l'optimisation du processus de développement. Cela

permet de créer des logiciels et des applications de haute qualité, répondant aux besoins croissants des utilisateurs et assurant leur satisfaction.

Lors de cette Etude, nous nous concentrerons sur les méthodologies et leurs impacts sur les performances des applications.

c. Objectif de l'étude :

Alors que les recherches existantes ont exploré les avantages et les inconvénients de différentes méthodologies de développement, moins d'attention a été accordée à la manière dont ces méthodologies influencent la performance des applications. Cette lacune dans la recherche est particulièrement préoccupante compte tenu de l'importance de la performance des applications dans le contexte commercial et opérationnel actuel.

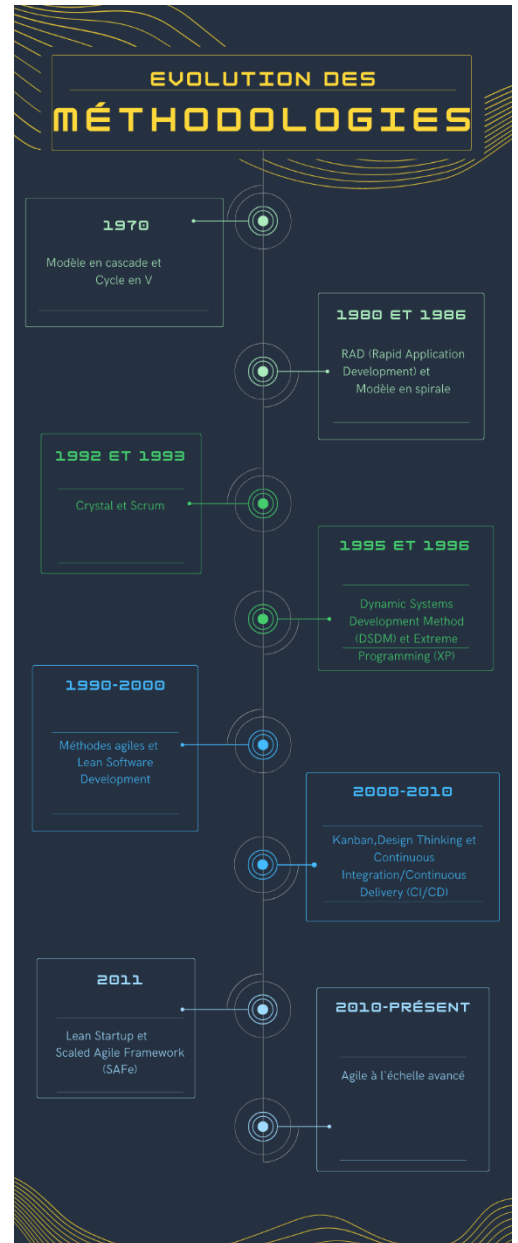
De plus, les organisations déploient souvent de grandes ressources pour adopter et mettre en œuvre de nouvelles méthodologies de développement. Cependant, sans une compréhension claire de l'impact potentiel de ces méthodologies sur la performance des applications, ces investissements pourraient ne pas produire le retour attendu.

Ainsi, cette étude vise à combler cette lacune en fournissant une analyse comparative de l'impact de différentes méthodologies de développement sur la performance des applications. Les résultats de cette étude pourraient aider les organisations à faire des choix plus éclairés lors de l'adoption et de la mise en œuvre de méthodologies de développement.

d. L'histoire du développement

Le développement logiciel a évolué au fil des décennies pour répondre aux besoins croissants des utilisateurs. Au départ, les premiers langages de programmation* de bas niveau, tels que l'assembleur, étaient utilisés pour développer des logiciels. Cependant, avec l'introduction des langages de programmation* de haut niveau tels que Fortran, COBOL et C dans les années 1950 et 1960, le développement logiciel a commencé à se démocratiser.

Dans les années 1970, la méthode de développement en cascade a été introduite, où chaque phase du développement était effectuée séquentiellement. Cela a été suivi par d'autres méthodes telles que le développement en spirale et la méthode en "V" dans les années 1980 et 1990.



II. Etat de l'art :

Des nos jours, de nombreuses méthodes de développement ont été élaboré afin de répondre le plus rapidement et justement possible aux attentes des clients. Toutes ont leurs avantages et inconvénients. Adapter la méthodologie de développement au produit est donc indispensable.

a. Agile

Cette méthodologie consiste à mettre le client au centre du projet. Plus le projet avance plus des adaptations sont apportées. Les deux principes de la méthodologie agiles sont la souplesse ainsi que la flexibilité.



Il existe de nombreuses méthodes agiles.

Les plus populaires sont les suivantes :

- Scrum.
- Kanban.
- L'eXtrême Programming (XP).
- Feature Driven Development (FDD).
- Lean Software Development.
- Agile Unified Process (Agile UP ou AUP).
- Crystal.
- Dynamic Systems Development Method (DSDM) .

La méthode Kanban :

La méthode Kanban est une approche de gestion visuelle qui vise à optimiser les flux de travail et à améliorer la productivité. Cette méthodologie est en flux continu et tiré*(pull), elle permet donc une meilleure réactivité grâce à des modifications sans attendre la prochaine itération* contrairement à Scrum. Ainsi que de connaître précisément l'avancée du projet grâce au tableau kanban qui rend visible le processus ainsi que le travail en cours.



En Kanban, l'équipe se concentre sur les tâches à faire et à finir avant de passer à une nouvelle tâche du projet. Le principal est le bon déroulement du travail et l'équipe avancera à son rythme.

La méthode Scrum :

C'est une méthodologie agile en flux poussé*. Il y a plusieurs rôles qui ont chacun leur importance. L'équipe est composée de développeurs, de testeurs, d'architectes un product owner et un scrum master.

La méthode Scrum consiste à mettre en place une "User Story" qui permet de connaître le travail à faire et l'importance de chaque tâche puis un Product Backlog* permet de savoir ce qu'il faut réaliser pour faire la "User Story"* et évolue avec les nouveaux besoins. Ensuite se divise en plusieurs itérations appelées Sprint* qui dure de 2 à 4 semaines ; il y a une planification, du développement, du test de qualité puis une livraison. Il y a une constante amélioration en mettant le client au centre du projet.



b. Itérative

Il s'agit d'une méthode de développement consistant à améliorer constamment un projet.

Cette méthode n'est pas incompatible avec d'autres méthodologies.

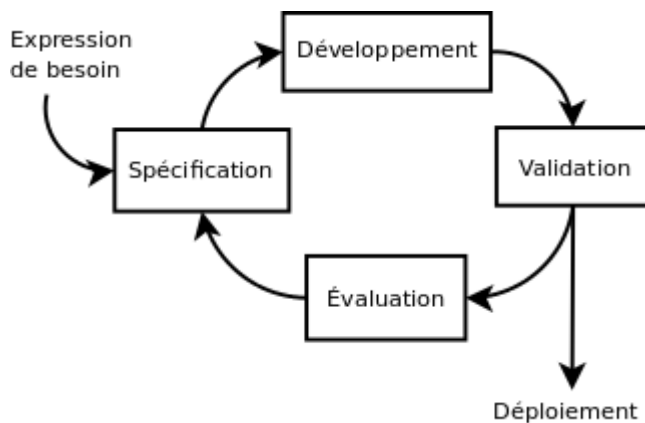
Lors de la création d'un projet en utilisant cette méthode, de nouvelles idées seront tester en cours de route.

Les avantages de la méthodologie itérative :

- Efficacité
- Collaboration optimisée
- Adaptabilité
- Risques limités
- Retours utilisateur fiable
-

Les inconvénients de la méthodologie itérative :

- Rigidité des exigences
- Délais incertains
- Dérive des objectifs



c. Lean agile

La méthode Lean est une approche de gestion axée sur l'efficacité et l'élimination des gaspillages dans les processus.

Les avantages de la méthodologie Lean :

- Efficacité : La méthode Lean vise à éliminer les activités qui n'ajoutent pas de valeur, ce qui permet d'optimiser les processus et d'améliorer l'efficacité globale.
- Réduction des coûts : En éliminant les gaspillages et en optimisant les processus, la méthode Lean permet de réduire les coûts de production.
- Amélioration continue : La philosophie Lean encourage l'amélioration continue à tous les niveaux de l'organisation.
- Satisfaction client : En réduisant les délais de livraison, en améliorant la qualité des produits et en répondant aux besoins des clients de manière plus efficace, la méthode Lean permet d'augmenter la satisfaction client.
- Flexibilité : La méthode Lean favorise la flexibilité en adaptant les processus aux besoins changeants du marché.

Les inconvénients de la méthodologie Lean :

- Complexité de mise en œuvre : La mise en œuvre de la méthode Lean peut être complexe et nécessite souvent un changement culturel au sein de l'organisation.
- Temps de transition : La transition vers une approche Lean peut prendre du temps et nécessiter des investissements initiaux.

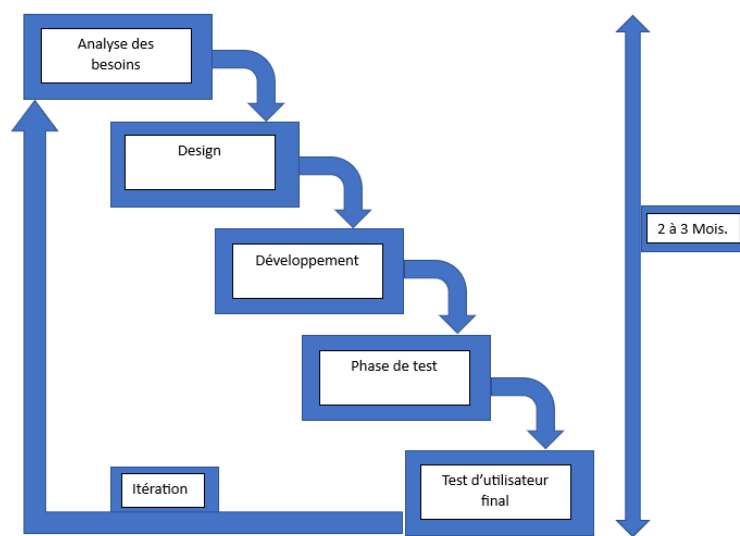
En résumé, la méthode Lean offre des avantages significatifs tels que l'efficacité, la réduction des coûts, l'amélioration continue, la satisfaction client et la flexibilité. Cependant, il est important de prendre en compte les défis potentiels tels que la complexité de mise en œuvre, le temps de transition et la résistance au changement lors de l'adoption de cette approche.

d. Cascade

Il s'agit d'un modèle où les phases de développement suivent rigoureusement un ordre spécifique

Les phases de la méthodologie Cascade sont :

- Cadrage du projet
- Conception générale
- Conception détaillée
- Production
- Tests et corrections
- Livraison



e. Spirale

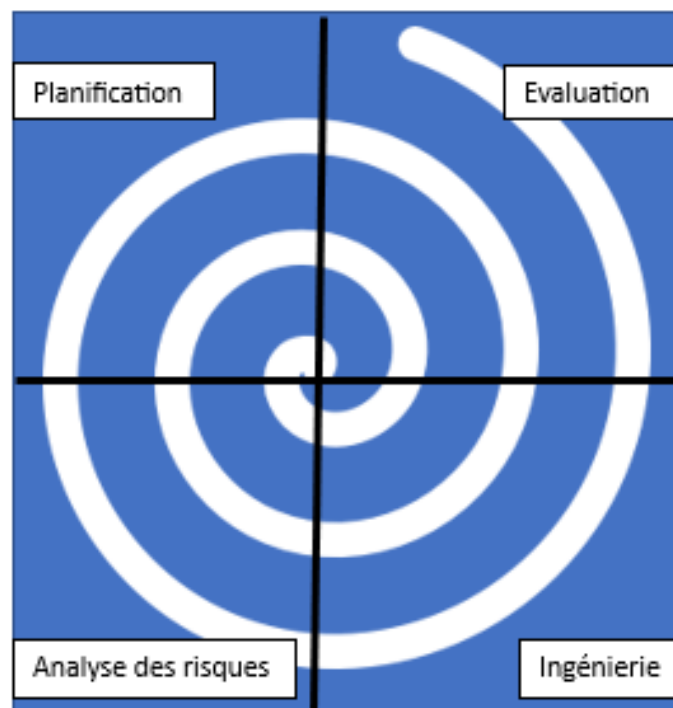
Elle a été définie comme une alternative possible au modèle en cascade, la méthode Spirale ne part pas du principe que les tâches du développement logiciel doivent être organisées de manière linéaire, mais de manière itérative (qui doit être répété jusqu'à ce que le but fixé soit atteint) avec des contrôles réguliers. L'avancement du projet est plus lent que pour la méthode en cascade mais le risque d'échec est beaucoup plus réduit

Avantages du modèle en spirale :

- Évitement des risques car cela implique une grande quantité d'analyse des risques.
- Développement rapide.
- Les modifications des exigences sont facilement adaptées.
- Les exigences peuvent être acquises plus précisément.

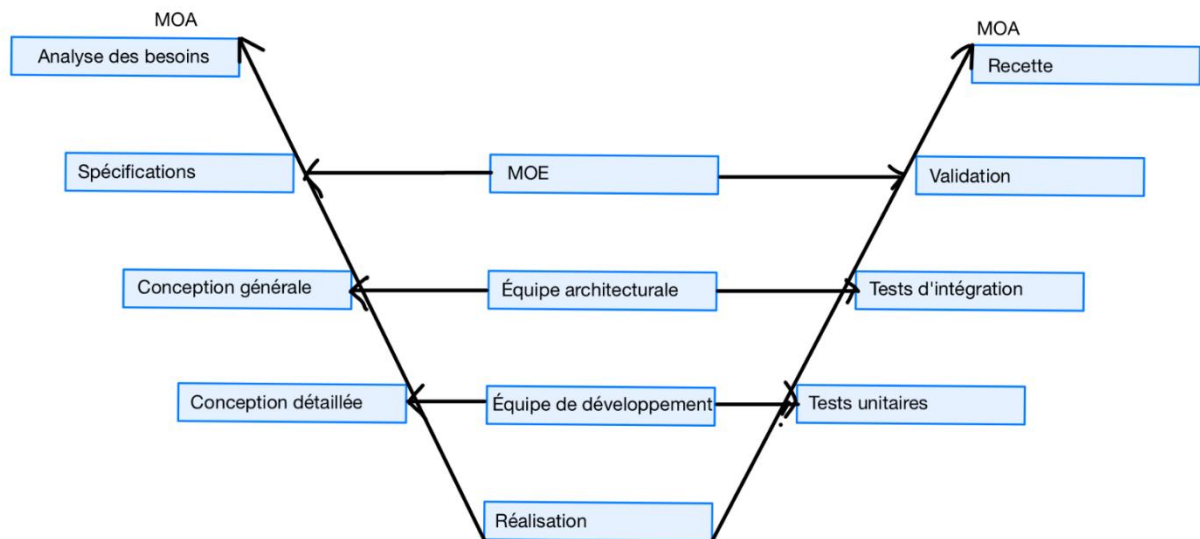
Inconvénients du modèle Spiral :

- Gestion complexe.
- Ne convient pas aux petits projets.
- Peut impliquer un nombre de spirales (indéfinies).
- Onéreux.
- Nécessite une grande quantité d'analyse des risques et d'expertise pour la réussite de leur projet.



f. En V

Cette méthodologie est similaire à la méthode en Cascade. Cependant, celle-ci est d'avantage axée sur la vérification ainsi que la validation du projet tout au long du développement. La méthode en V est souvent utilisée par des équipes dont les produits se doivent d'être très qualitatifs.



g. Secure Development Life

Il s'agit d'une méthode de développement de logiciels qui se concentre sur l'intégration de la sécurité à chaque étape du cycle de vie du développement de logiciels. Il a été introduit par Microsoft en 2004 dans le but de réduire le nombre et la gravité des vulnérabilités dans ses logiciels.

Les Etapes typiques de la méthodologie sont les suivantes :

Formation : Les développeurs reçoivent une formation en sécurité pour les aider à comprendre les principes de base de la sécurité et à éviter les erreurs courantes.

Exigences : Les exigences en matière de sécurité sont définies dès le début du projet.

Conception : Les menaces potentielles sont identifiées et des contre-mesures sont prévues pendant la phase de conception.

Implémentation : Les meilleures pratiques de codage sécurisé sont suivies lors de l'écriture du code.

Vérification : Le code est testé pour s'assurer qu'il est exempt de vulnérabilités connues.

Release : Les mises à jour de sécurité sont publiées de manière responsable

Réponse : Lorsqu'une vulnérabilité est découverte, une réponse rapide est assurée.

h. RAD

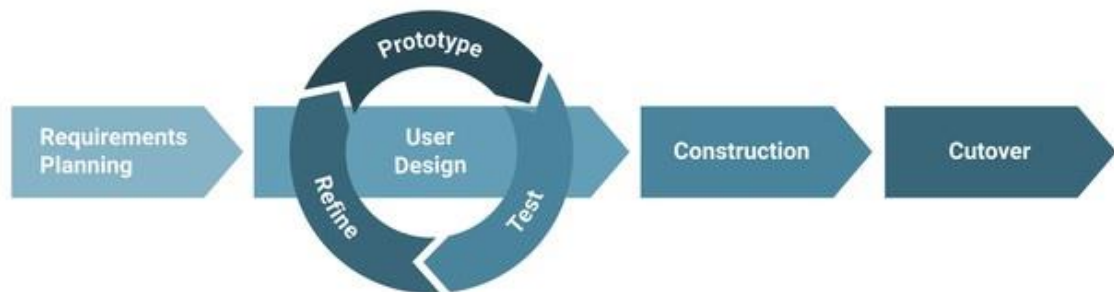
Cette méthode est axée sur la rapidité et l'efficacité, en utilisant des prototypes* et des tests rapides pour itérer rapidement sur la conception et le développement.

Avantages du modèle RAD :

- Les progrès peuvent être mesurés.
- Réduit le temps de développement.
- Une réutilisabilité accrue.
- Révisions initiales rapides.
- Améliore les commentaires des clients.

Inconvénients du modèle RAD :

- Nécessite des ressources hautement qualifiées.
- Estimation coûteuse.
- Non applicable pour les projets moins chers.
- Seul un système modulaire peut être construit à l'aide de cette méthode.



Rapid Application Development (RAD)

IV. Analyse.

a. Les méthodes de développement des GAFAM.

Google :

Google est connu pour être une entreprise axée sur l'innovation et l'efficacité. Lorsqu'il s'agit de choisir une méthodologie de développement, Google opte généralement pour des approches agiles telles que Scrum. Cette décision repose sur plusieurs facteurs clés. Tout d'abord, l'approche Agile permet à Google de répondre rapidement aux demandes changeantes du marché et des utilisateurs. Google peut livrer des produits et des fonctionnalités de manière itérative, permettant ainsi d'obtenir rapidement des retours des utilisateurs et de les intégrer dans le processus de développement. De plus, les méthodologies de développement agiles, telles que Scrum, offrent à Google une agilité, une collaboration et une qualité élevées. Ces approches permettent à Google de s'adapter rapidement, d'innover efficacement et de fournir des produits de haute qualité, répondant ainsi aux attentes des utilisateurs et contribuant à son succès continu dans l'industrie technologique.

Apple :

Apple a traditionnellement adopté une approche itérative et centrée sur le design, connue sous le nom de "Design Loop". Cette méthodologie met l'accent sur la conception centrée sur l'utilisateur, la qualité et la performance des produits. Apple accorde une grande importance à l'expérience utilisateur, à l'esthétique et à l'ergonomie de ses produits. En utilisant le "Design Loop", Apple assure une attention méticuleuse aux détails et un contrôle rigoureux tout au long du processus de développement. Cette approche permet à Apple de créer des produits technologiques de haute qualité, innovants et esthétiquement plaisants. Il est à noter que les méthodologies de développement peuvent évoluer en fonction des besoins et des projets spécifiques de l'entreprise.

Facebook :

Facebook, tout comme Google, opte pour la méthode Agile Scrum. Cette approche permet à Facebook de répondre rapidement aux demandes changeantes du marché et des utilisateurs, de livrer des fonctionnalités itérativement et d'assurer une collaboration efficace dans le processus de développement. En utilisant Scrum, Facebook favorise l'agilité, la qualité et l'innovation, contribuant ainsi à son succès en tant que plateforme sociale populaire et innovante. Amazon :

Amazon utilise une méthodologie Agile appelée Kanban, qui se concentre sur la gestion visuelle des tâches et des flux de travail. Cette approche favorise la transparence, la flexibilité et l'efficacité. Les équipes d'Amazon peuvent ainsi

répondre rapidement aux demandes changeantes du marché, ajuster les priorités et fournir des solutions de haute qualité. En utilisant Kanban, Amazon maintient son rythme de développement soutenu et s'adapte aux besoins évolutifs des clients, renforçant ainsi sa position en tant que leader de l'industrie.

Microsoft :

Comme Google, Amazon et Facebook, Microsoft opte pour une approche agile. Cette approche permet de livrer rapidement des solutions logicielles de qualité, de favoriser la collaboration et l'innovation, de s'adapter aux changements du marché et de mettre l'accent sur la satisfaction des clients. Cette méthodologie agile contribue à maintenir la position de Microsoft en tant que leader technologique et à offrir des produits et des services qui répondent aux besoins en constante évolution de l'industrie.

b. Etudes menées.

L'étude de l'impact des méthodologies de développement sur les performances des applications étant essentielle à l'amélioration de l'efficacité, la qualité et la satisfaction des utilisateurs, de nombreuses études sur le sujet ont donc déjà été menées.

Parmi les études déjà menées, nous pouvons citer :

- "Comparing the Effectiveness of Agile and Waterfall Software Development Methodologie".

Les résultats de cette étude ont montré que l'approche Agile était généralement plus efficace que l'approche en cascade.

Les méthodologies Agile ont montré de meilleures performances en termes de respect des délais et de livraison des fonctionnalités.

Elles ont également été associées à une meilleure qualité du logiciel, grâce à une approche orientée vers les tests et une rétroaction continue. Les utilisateurs ont exprimé une plus grande satisfaction avec les produits développés selon des méthodologies Agile, en raison de l'implication continue des utilisateurs et de la possibilité de fournir des commentaires fréquents.

De plus, les méthodologies Agile se sont avérées plus efficaces dans la gestion des risques, grâce à leur flexibilité et à leur capacité à s'adapter rapidement aux changements.

- « L'étude comparative intitulée "A Comparative Study of Scrum and Kanban Approaches in Software Development"

Cette Etude s'est penchée sur la comparaison des approches Scrum et Kanban dans le développement logiciel. Les résultats ont révélé des différences significatives entre les deux méthodologies.

L'approche Scrum a montré des avantages en termes de planification, de collaboration d'équipe et de visibilité du projet. Les équipes Scrum ont pu définir clairement les objectifs à court terme et maintenir un rythme de travail soutenu grâce aux sprints*. De plus, la structure des rôles et des réunions dans Scrum a favorisé la communication et la prise de décision collective.

D'autre part, l'approche Kanban a mis l'accent sur la visualisation du flux de travail et la gestion du travail en cours (WIP). Les équipes Kanban ont bénéficié d'une plus grande flexibilité pour gérer les priorités et adapter leur flux de travail en fonction de l'évolutions des besoins. L'utilisation d'un tableau Kanban a permis une meilleure transparence et une meilleure coordination des tâches.

En termes de flexibilité et d'adaptabilité, Kanban a montré des avantages par rapport à Scrum. La capacité à gérer efficacement les demandes changeantes et à ajuster le flux de travail a été soulignée dans cette étude.

Cependant, Scrum a été considéré comme plus approprié pour les projets où la planification et la prévisibilité sont essentielles.

Le choix entre ces deux méthodes dépend donc des objectifs du projet et des préférences de l'équipe de développement.

- "Impact of Agile Methodologies on Software Testing"
- "Lean Software Development: An Empirical Study"
- "Comparative Study of Spiral and Iterative Model in Software Development"

V. Projection et Proposition.

Dans le monde en constante évolution du développement de logiciels, il est essentiel de choisir la bonne méthodologie pour maximiser les performances des applications. Après avoir examiné l'état de l'art, nous proposons une approche hybride qui combine les meilleures pratiques de plusieurs méthodologies pour répondre aux besoins spécifiques de différents types d'applications. Pour illustrer cette proposition, nous examinerons deux exemples : une application de gestion de système ferroviaire et une application de réseau social basée sur le Web 3.0.

A. Application de gestion de système ferroviaire

La tâche d'orchestration d'un réseau ferroviaire est sans conteste délicate et nécessite une synchronisation méticuleuse et une fiabilité de premier ordre. Les outils qui gèrent ces réseaux sont chargés de l'harmonisation de multiples éléments : de la cartographie des trajets à l'organisation des horaires, en passant par le suivi des trains en temps réel, l'entretien des infrastructures et la résolution des imprévus. Il est donc indispensable que ces outils soient fiables, efficaces et sécurisés.

Dans ce cadre, la méthodologie de développement en cascade peut s'avérer être une option adaptée. Cette méthode prône une planification détaillée, une documentation exhaustive et une conception rigoureuse, des éléments clés pour ce type d'outils où chaque détail a son importance.

Prenons par exemple la phase de conception de cette méthode en cascade : elle permettrait de déterminer précisément comment l'outil va gérer diverses tâches, comme l'établissement des trajets ou l'organisation des horaires. Elle définirait également l'interaction entre l'outil et les autres systèmes en jeu, comme le contrôle des trains et les bases de données d'entretien des infrastructures.

Cela dit, cette méthode peut être trop rigide pour faire face à l'imprévisible. C'est là que l'introduction de certains aspects de la méthodologie Agile pourrait s'avérer utile. L'approche Agile, avec sa souplesse, facilite les ajustements tout au long du processus de développement. L'implémentation de nouvelles fonctionnalités peut se faire au cours de sprints* de développement définis sur une durée limitée. De plus, les feedbacks réguliers permettent d'affiner le produit en fonction des retours d'information et des changements dans les besoins.

En mêlant le meilleur de la méthode en cascade et de la méthode Agile, nous pouvons créer un outil de gestion de réseau ferroviaire qui combine fiabilité et adaptabilité, capable de répondre aux besoins spécifiques de ce domaine tout en s'ajustant aux changements et aux évolutions.

Réseau social Basée sur le Web 3.0

Le Web 3.0, parfois appelé web sémantique, marque l'évolution des plateformes en ligne vers une capacité d'interprétation du contenu, de compréhension des utilisateurs et de fourniture d'informations plus précises et personnalisées. En ce qui concerne une plateforme de médias sociaux, cela se traduit par une interaction plus détaillée et personnalisée pour les utilisateurs, avec des fonctionnalités comme des suggestions de contenu basées sur l'intelligence artificielle, des interactions en temps réel et la prise en charge de transactions cryptographiques.

Pour créer une telle plateforme, il est nécessaire d'adopter une approche de développement flexible et axée sur l'utilisateur. La méthodologie Agile, avec ses cycles de développement rapides et ses retours fréquents des utilisateurs, est parfaitement adaptée à ce genre de plateformes. Elle permet une réaction rapide aux besoins changeants des utilisateurs et un ajustement de la plateforme en fonction de l'évolution du marché.

Par exemple, un sprint de développement Agile pourrait se concentrer sur l'implémentation d'une nouvelle fonctionnalité de suggestion de contenu basée sur l'IA. Les utilisateurs seraient ensuite invités à tester cette fonctionnalité et à donner leurs impressions, qui seraient utilisées pour améliorer la fonctionnalité dans les sprints suivants.

Cependant, le Web 3.0 introduit également de nouveaux défis en matière de sécurité et de confidentialité. Les plateformes de médias sociaux basées sur le Web 3.0 doivent non seulement protéger les données des utilisateurs, mais aussi adhérer aux principes de décentralisation et de contrôle des utilisateurs sur leurs propres données. Pour cela, l'incorporation de la méthodologie Secure Development Lifecycle (SDL) est cruciale. En intégrant la SDL au processus de développement Agile, on peut s'assurer que les questions de sécurité sont prises en compte tout au long du développement de la plateforme.

Par exemple, lors de la création d'une nouvelle fonctionnalité, une analyse des menaces pourrait être effectuée pour identifier les risques potentiels pour la sécurité et la confidentialité des utilisateurs. Ils seraient ensuite pris en compte lors de la conception et de la mise en œuvre de la fonctionnalité, et des tests de sécurité seraient effectués pour s'assurer que la fonctionnalité est sécurisée avant sa mise en service.

En combinant la méthodologie Agile avec la SDL, nous pouvons créer une plateforme de médias sociaux basée sur le Web 3.0 qui est non seulement réactive et axée sur l'utilisateur, mais aussi sécurisée et respectueuse de la confidentialité des utilisateurs.

VI. Conclusion.

En conclusion, l'étude de l'impact de l'utilisation de différentes méthodologies de développement sur les performances des applications met en évidence l'importance de choisir la méthodologie appropriée en fonction des besoins et des objectifs du projet.

Les résultats des différentes études examinées ont démontré que les méthodologies Agile, telles que Scrum, ont tendance à offrir une meilleure performance en termes de respect des délais, de qualité du logiciel et de satisfaction des utilisateurs. L'approche Agile favorise la flexibilité, la collaboration, la rétroaction continue et la gestion efficace des risques.

D'autre part, les méthodologies en cascade, bien qu'elles puissent être adaptées à certains projets spécifiques, peuvent présenter des limitations en termes d'adaptabilité aux changements et de réactivité.

Il est également important de noter que d'autres méthodologies, telles que Kanban, ont été étudiées et ont démontré leurs avantages en termes de visualisation du flux de travail, de gestion du travail en cours et de flexibilité.

En somme, le choix de la méthodologie de développement aura un impact significatif sur les performances des applications. Il est donc essentiel d'évaluer attentivement les besoins du projet, la nature du travail à effectuer et les caractéristiques de l'équipe de développement afin de sélectionner la méthodologie la plus appropriée.

L'adoption de bonnes pratiques de gestion de projet, de communication efficace, d'automatisation* et d'une méthodologie adaptée permettra d'optimiser le processus de développement et de maximiser les performances des applications. Une analyse continue et une amélioration continue sont également essentielles pour s'adapter aux évolutions du projet et aux attentes des utilisateurs.

En fin de compte, une approche réfléchie et adaptée en matière de méthodologie de développement contribuera à la réalisation de produits logiciels performants, répondant aux besoins des utilisateurs et offrant une expérience utilisateur satisfaisante.

VII. Glossaire :

Méthodologies : les méthodologies de développement définissent les processus utilisés pour un projet de logiciel ou d'un produit informatique de manière efficace, grâce à des approches différentes en fonction de chaque méthodologie.

Applications : Les applications sont des programmes informatiques conçus pour développées pour répondre aux besoins des utilisateurs et effectuer des tâches spécifiques sur un ordinateur, un appareil mobile ou un système informatique. Les applications peuvent prendre différentes formes, telles que des applications de bureau, des applications Web ou des applications mobiles.

Logiciel : Un logiciel est un programme informatique, qui permet à un ordinateur d'accomplir des tâches spécifiques. Il peut s'agir de systèmes d'exploitation, de logiciels d'application, de jeux vidéo, de bases de données, etc.

Utilisateurs : Les utilisateurs sont les personnes ou les entités qui interagissent avec un logiciel, une application ou un système informatique.

Processus de développement : Les processus de développement sont les étapes et les tâches impliquées dans la création d'un logiciel ou d'une application.

Automatisation : L'automatisation est la mise en place de technologies et de techniques pour exécuter des tâches ou des processus sans intervention humaine directe.

Optimisation : L'optimisation est l'amélioration des performances, de l'efficacité ou de la qualité d'un système, d'une application ou d'un processus.

Compilation : La compilation est le processus de traduction du code source écrit dans un langage de programmation en un code exécutable compréhensible par un ordinateur.

Déploiement : Le déploiement correspond à la mise en place et à l'installation d'un logiciel ou d'une application dans un environnement de production, prêt à être utilisé par les utilisateurs.

Cycles de développement : Les cycles de développement désignent les itérations ou les étapes du développement à travers lesquelles passe un projet. Ils permettent de gérer et d'organiser le travail.

Vitesse d'exécution : La vitesse d'exécution est à la vitesse à laquelle un logiciel ou un programme s'exécute ou répond aux requêtes.

Consommation de ressources : La consommation de ressources, c'est la quantité de ressources matérielles ou logicielles utilisées par un logiciel ou un système.

Capacité de montée en charge : La capacité de montée en charge fait référence à la capacité d'un logiciel ou d'un système à gérer une charge de travail croissante sans perte de performances ou de fiabilité. Elle peut permettre au logiciel de s'adapter à une augmentation du nombre d'utilisateurs sans compromettre ses performances.

Langages de programmation : Les langages de programmation sont des langages formels utilisés pour écrire des instructions ou des programmes informatiques. Ils

permettent aux développeurs de communiquer avec l'ordinateur et de lui donner des actions à effectuer.

Flux continu et tiré : Le flux continu et tiré est un concept utilisé dans la gestion de projet et de développement, qui vise à optimiser la livraison continue de fonctionnalités ou de tâches.

Itération : Une itération peut être une boucle ou un cycle de travail répété dans le processus de développement.

Flux poussé : Le flux poussé est une approche où les fonctionnalités ou les tâches sont planifiées et assignées aux développeurs avant le début de leur réalisation.

User Story : Une User Story est une technique de spécification des besoins utilisée dans les méthodes Agiles. Elle décrit une fonctionnalité ou un besoin du point de vue de l'utilisateur.

Product Backlog : Le Product Backlog est une liste ordonnée de toutes les fonctionnalités, les améliorations et les tâches à réaliser dans un projet de développement. Il s'agit d'un document vivant qui évolue tout au long du projet.

Sprint : Un Sprint est une itération fixe et courte dans la méthodologie Scrum, qui dure généralement d'une à quatre semaines.

Prototypes : Les prototypes sont des versions préliminaires ou partielles d'un logiciel ou d'une application ou d'un autre type de projet, créées dans le but de tester et de valider des idées, des fonctionnalités ou des concepts.

Conception : La conception est la phase de planification et de création d'un produit qui suit le reste du projet.

Ergonomie : L'ergonomie se réfère à la conception et à l'organisation des systèmes, des produits et des interfaces utilisateur de manière à optimiser l'efficacité, la convivialité et la satisfaction des utilisateurs. L'ergonomie vise à créer des produits qui sont adaptés aux besoins et aux capacités des utilisateurs.

Rétroaction :

VIII. Sources

- <https://www.celge.fr/article-conseil/6-methodologies-cycle-vie-developpement-logiciel>
- <https://www.nutcache.com/fr/blog/les-methodes-agiles/>
- <https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1443838-methode-agile-definition-comparatif-et-avantages/>
- <https://www.eurotechconseil.com/blog/quelles-sont-les-methodologies-de-developpement-logiciels/>
- https://fr.wikipedia.org/wiki/Cycle_en_V
- <https://asana.com/fr/resources/iterative-process>
- <https://www.eurotechconseil.com/blog/quelles-sont-les-methodologies-de-developpement-logiciels/>
- <https://www.hbrfrance.fr/chroniques-experts/2020/05/30140-quest-ce-quune-entreprise-agile/>