



Cycle de vie du GL et principes de l'agilité du développement

Module : Outils RAD

Niveau : S4 du L2/ISIL

2020-2021

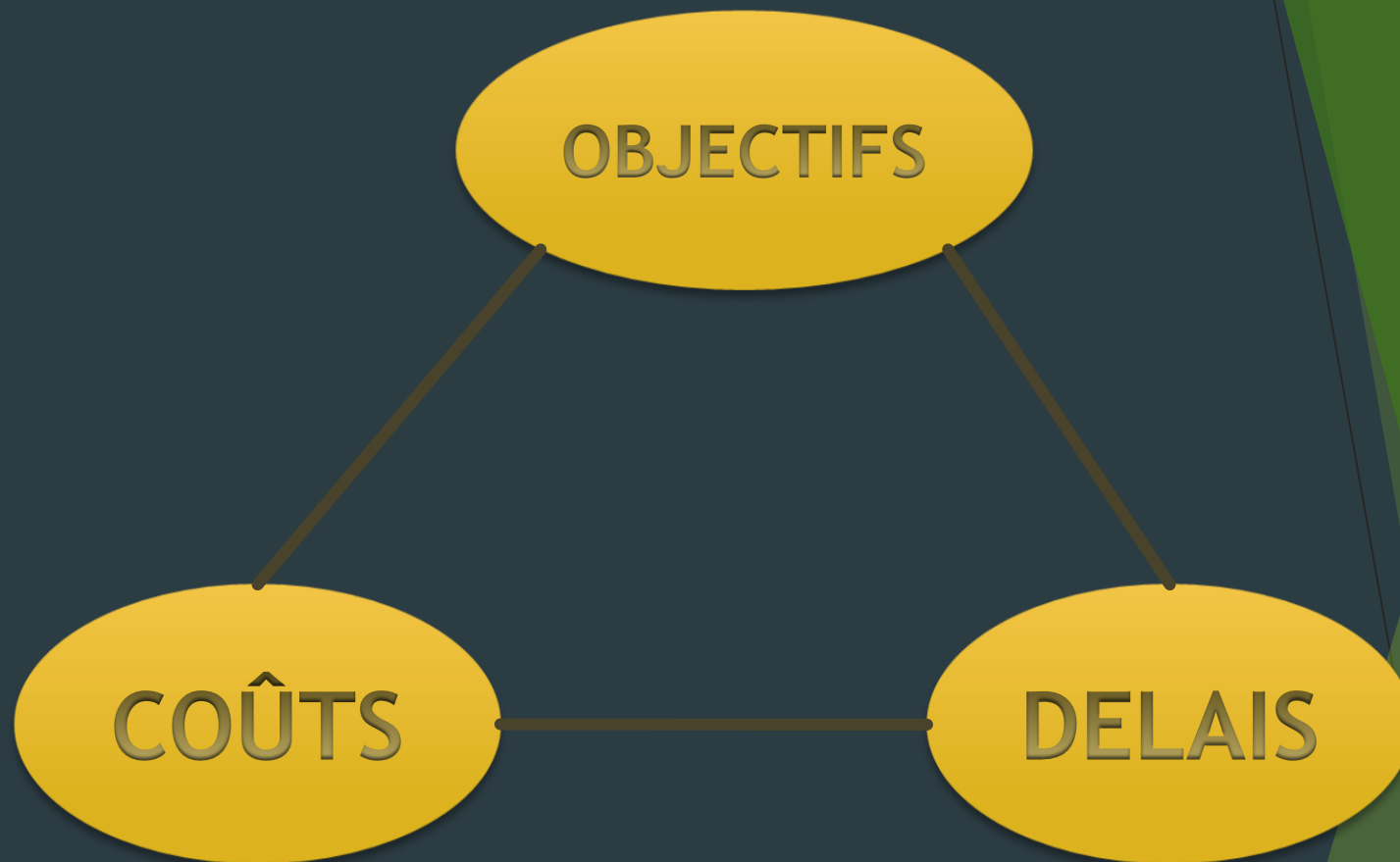


Le terme **génie logiciel** (en anglais *software engineering*) désigne **l'ensemble des méthodes, des techniques et outils** concourant à la production d'un logiciel, au-delà de la seule activité de programmation.

Le choix du terme « génie » fait directement référence à celui du génie civil, désignant l'art de la construction.



Le domaine du **génie logiciel** fut donc largement inspiré du domaine de la construction mais aussi celui de la gestion de projet où le souci permanent consiste à réaliser l'équilibre entre « **Coûts** », « **Délais** » et « **Objectifs** » tout en assurant la « **Qualité** ».



En effet, dans un **environnement complexe**, de surcroît, contraint par le **time to market**, il doit (faire) développer **un produit** au **moindre coût** dans des **délais de plus en plus courts** avec une **qualité irréprochable** tout en atteignant les **objectifs fixés**.

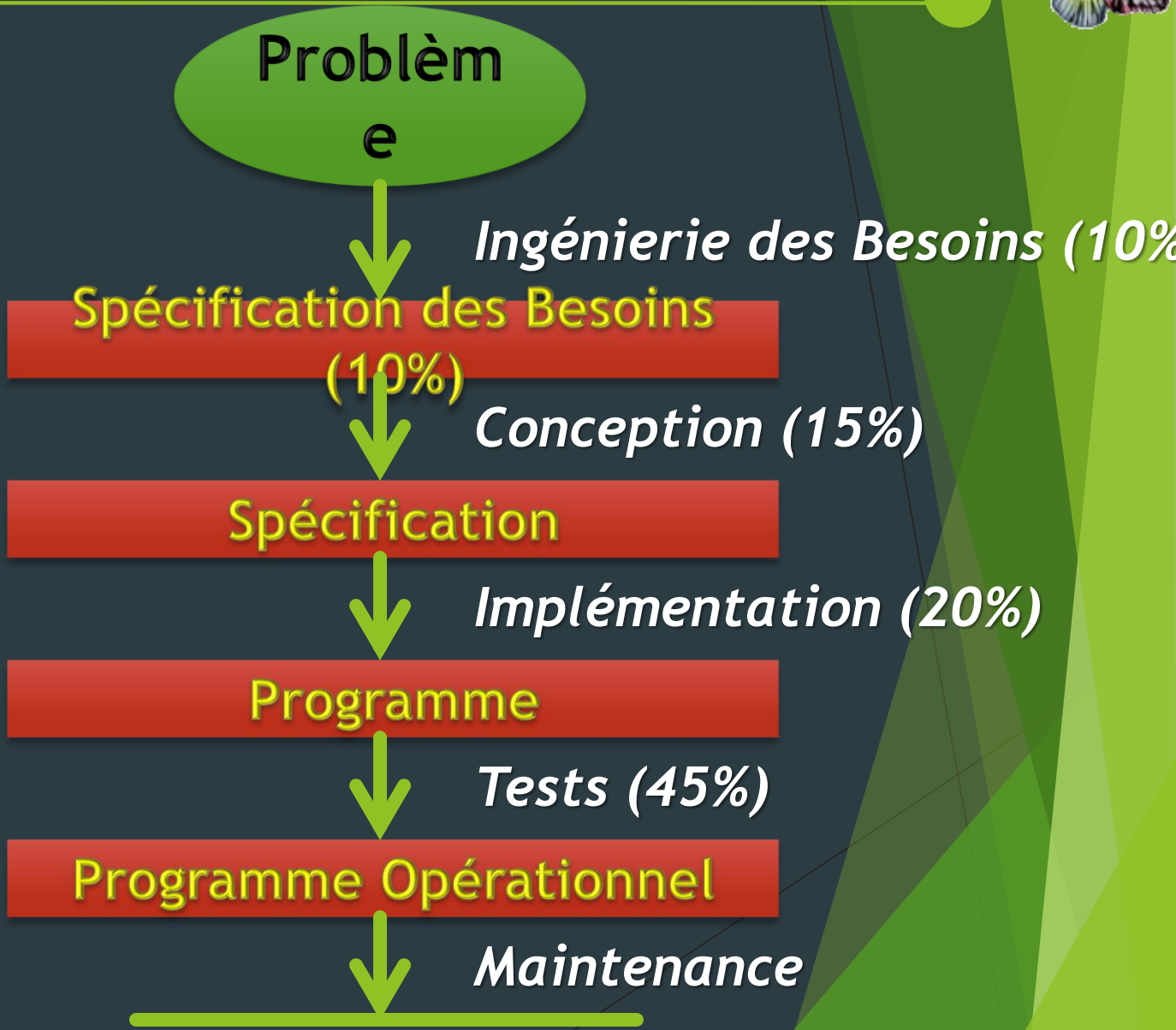


Le « **cycle de vie d'un logiciel** » (en anglais *software lifecycle*), **désigne toutes les étapes du développement d'un logiciel**, de sa conception à sa disparition.

L'objectif d'un tel découpage est de permettre de définir des **jalons intermédiaires** permettant la **validation** du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la **vérification** du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre.



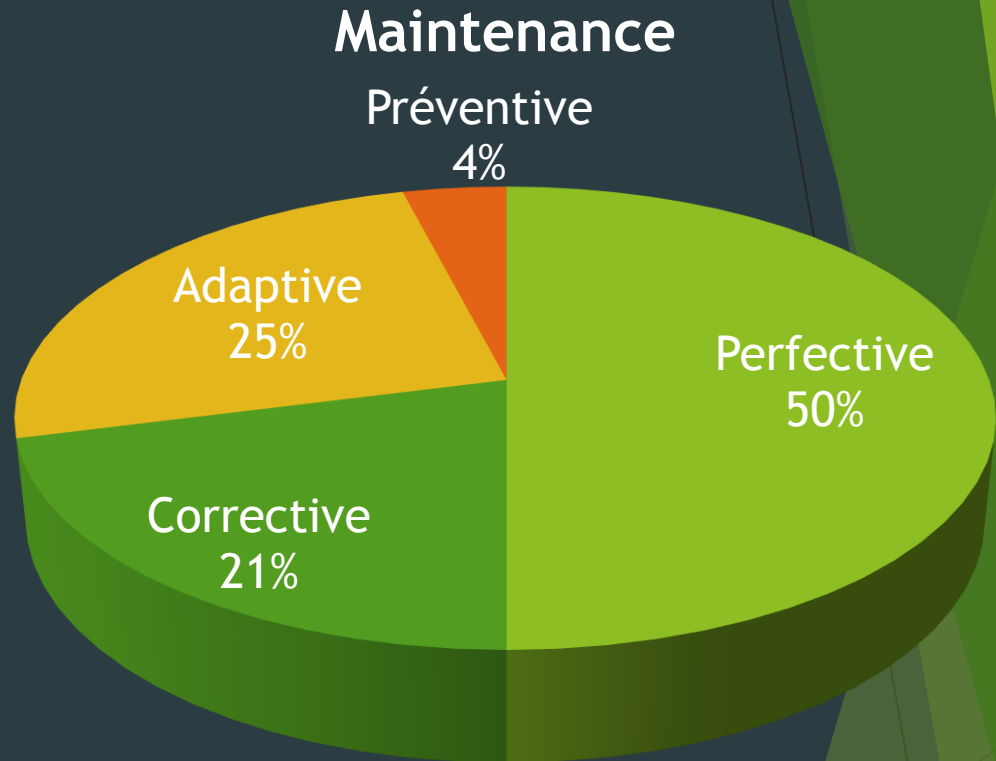
Observons le schéma suivant qui décrit de manière assez simple les phases de développement de logiciel.





La maintenance couvre un certain nombre d'activités dont :

- ✓ La Perfective
- ✓ La Corrective
- ✓ L'adaptative
- ✓ La Préventive





- ✓ L'origine de ce découpage provient du constat que les erreurs ont un coût d'autant plus élevé qu'elles sont détectées tardivement dans le processus de réalisation.
- ✓ Le cycle de vie permet de détecter les erreurs au plus tôt et ainsi de maîtriser la qualité du logiciel, les délais de sa réalisation et les coûts associés.
- ✓ Le cycle de vie du logiciel comprend généralement au minimum les activités suivantes :



Etape	Explication
Définition des objectifs	consistant à définir la finalité du projet et son inscription dans une stratégie globale.
Analyse des besoins et faisabilité	c'est-à-dire l'expression, le recueil et la formalisation des besoins du demandeur (le client) et de l'ensemble des contraintes.
Conception générale	il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel.
Conception détaillée	consistant à définir précisément chaque sous-ensemble du logiciel.
Codage	implémentation ou programmation, i-e, la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception.



Etape	Explication
Tests unitaires	permettant de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux spécifications.
Intégration	dont l'objectif est de s'assurer de l'interfaçage des différents éléments (modules) du logiciel. Elle fait l'objet de <i>tests d'intégration</i> consignés dans un document.
Qualification (ou <i>recette</i>)	c'est-à-dire la vérification de la conformité du logiciel aux spécifications initiales.
Documentation	visant à produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs.



Etape	Explication
Mise en production	vise la production du logiciel.
Maintenance	comprenant toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.



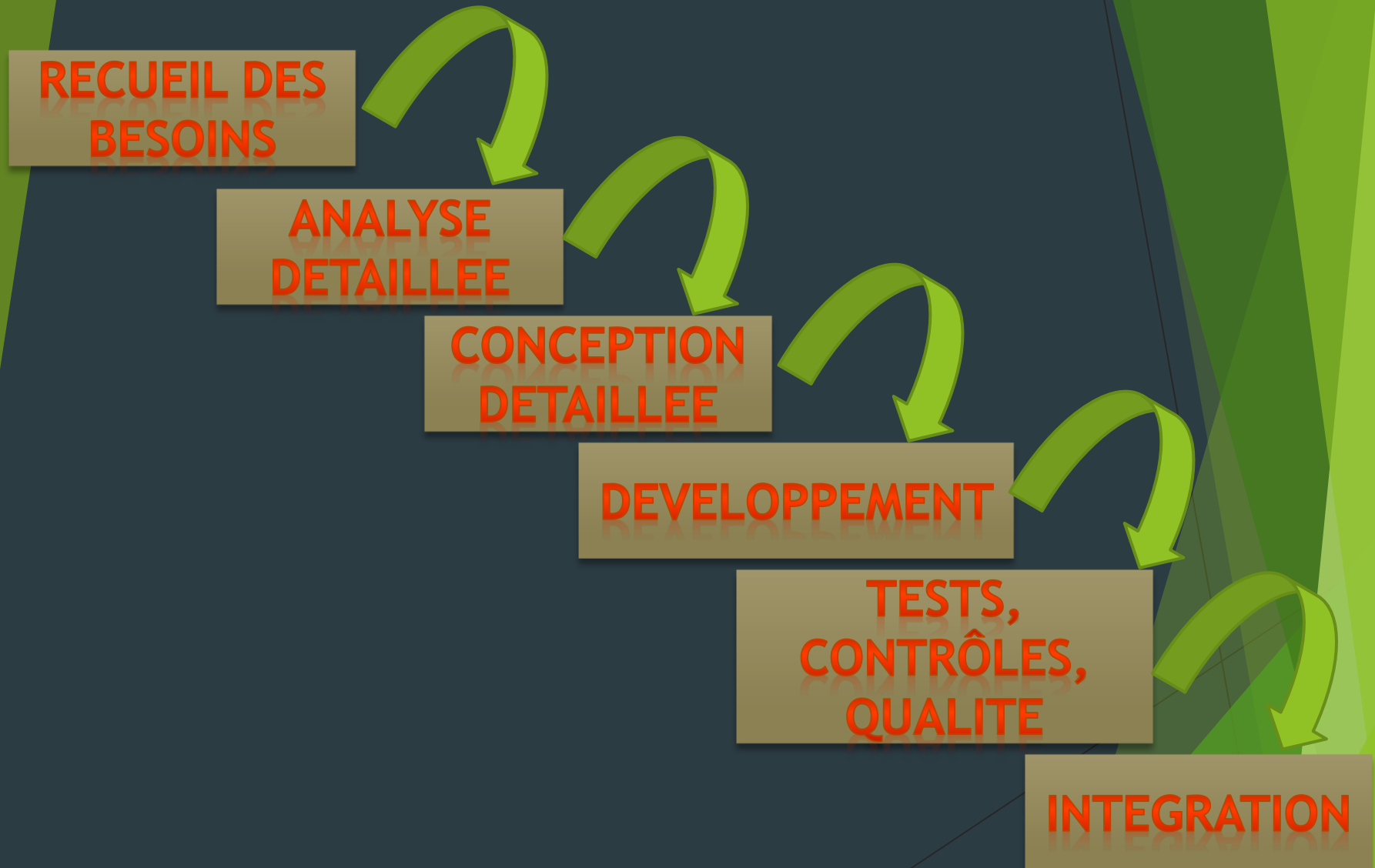
La littérature du domaine abonde de modèles et de démarches visant à améliorer le ou les processus associés aux développement de logiciels.

De manière générale, on distingue deux grandes catégories de méthodes :

- A. Les méthodes classiques ou basées sur une démarche classique.
- B. Les méthodes modernes visant la rapidité, la flexibilité, ..., etc, communément appelées « Méthodes Agiles ».



Passons en revue quelques principaux modèles de l'approche classique dont le **modèle en cascade** et le **modèle itératif ou incrémental**.





DES
BESOINS



?

BOITE NOIRE



UN
PRODUIT

👉 L'effet tunnel est une des caractéristiques de l'approche « en cascade » : un projet **dure un an**, la phase de recueil des besoins **dure deux mois** et le **client ne voit le résultat que neuf mois plus tard !**

👉 **Que s'est-il passé entre-temps ?** « On ne sait pas trop ce qu'ils font ces informaticiens ! », « Que va-t-il sortir de la « boîte » ? », « Mais, ce n'est pas ce que l'on attendait ! » ou bien « C'est ce que nous voulions mais notre besoin a un peu évolué depuis ! »



👎 D'une part, la non-transparence des équipes de développement suscite des sarcasmes sur leur capacité à coopérer.

👎 D'autre part, la longueur des phases techniques auxquelles le client n'est pas associé rend celui-ci dubitatif sur le résultat à venir.

👎 Ce qui ne favorise pas la collaboration efficace entre informaticiens et utilisateurs !

👎 D'autant plus si le résultat **livré n'est pas conforme à ce qui est attendu.**



Sont alors apparues des démarches qualifiées d'itératives ou incrémentales où le principe repose sur :

- 👉 Le découpage en phases
- 👉 La prise en compte de niveau d'incréments
- 👉 Le prototypage



Résistance ou Peur du Changement :

« Quand on trouve une **recette qui marche bien**, on a du mal à la quitter même si l'on constate que **son efficacité semble diminuer** ; il existe une inertie due à la **peur du changement**, à la **recherche de facilité** ou à l'**ivresse du succès** (ce qui marchait hier doit marcher demain...). Eh bien non !!! »



Fort du constat que les **plans initiaux** sont finalement **toujours modifiés** et que les besoins évoluent en permanence pour répondre aux changements du marché, ces **approches prédictives** se sont révélées **trop « rigides »** parfois, exposant les organisations à trop peu de réactivité dans le contexte de nouveaux projets stratégiques.



Sont alors apparues, dans les années 1990, des **méthodes moins prédictives**, plus **souples** face aux besoins d'adaptation, facilitant ainsi l'agilité des organisations face aux contraintes du marché.

☞ Ce sont les méthodes dites « agiles ».



Qu'est-ce qu'une méthode agile ?

- ☞ Une méthode agile est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif, avec juste ce qu'il faut de formalisme.
- ☞ Elle génère un produit de haute qualité tout en prenant en compte l'évolution des besoins des clients.



Des Méthodes Itératives et Incrémentales

Cependant :

Le résultat d'une itération n'est pas un prototype ou une « proof of concept », mais bien une version intermédiaire du produit final.



Avantages Des Méthodes Itératives et Incrémentales

Avantage	Les +
La communication est de meilleure qualité.	<p>Les malentendus, incompréhensions, incohérences sont mis en évidence tôt dans le projet ; il est donc encore possible de les corriger.</p> <p>L'utilisateur a la possibilité de clarifier ses exigences au fur et à mesure.</p> <p>Le client reçoit des « preuves » tangibles de l'avancement du projet.</p>
La visibilité est meilleure.	<p>Le client peut ainsi visualiser les travaux plus régulièrement, au fil de l'eau, sans attendre la fin du projet, puisqu'à la fin de chaque itération, les fonctionnalités retenues sont développées, testées, documentées et validées, prêtes pour l'exploitation.</p>
La qualité est évaluée en continu.	<p>Les tests sont effectués à chaque itération, les anomalies détectées sont corrigées au fur et à mesure.</p>
Les risques sont détectés très tôt.	<p>Grâce aux activités de développement précoces, les risques sont détectés tôt et résolus rapidement.</p>
L'équipe prend confiance.	<p>L'itération donne une occasion d'apprendre, donc de capitaliser ou d'adapter les pratiques pour la suite du projet.</p> <p>Les premières itérations fiabilisent les prévisions.</p> <p>Le changement n'est plus une menace, mais au contraire, l'opportunité de mieux faire et de mieux satisfaire le client.</p>
Les coûts sont contrôlés.	<p>Les coûts sont limités, en termes de risques, au périmètre de l'itération ; s'il faut reprendre une itération, on ne perd que les efforts de cette itération et non la valeur du produit dans sa globalité.</p> <p>On peut aussi arrêter le projet à l'issue de quelques itérations si l'on n'a plus de budget.</p>



Parmi les Douze Principes du Manifeste Agile, nous avons :

Principe	Description
Notre priorité est de satisfaire le client en lui livrant très tôt et régulièrement des versions opérationnelles de l'application, source de valeur.	Grâce au développement itératif, chaque livraison intermédiaire donne lieu à une validation par le client ; son feedback est essentiel pour garantir la conformité de la livraison avec ses attentes, pour prendre en compte ses remarques et (re)prioriser.
Accepter le changement dans les exigences, même tard dans le cycle de vie, pour garantir la compétitivité du client.	Cet état d'esprit caractérise une équipe agile qui démontre ainsi sa capacité à comprendre et apprendre comment satisfaire encore mieux la demande.
Livrer le plus souvent possible des versions opérationnelles de l'application, à une fréquence allant de deux semaines à deux mois.	Une version intermédiaire du produit final, visible et testable, satisfait davantage le client qu'une documentation à valider. Il a, de ce fait, la preuve tangible que le projet avance et peut exprimer son point de vue sur le résultat présenté.
Client et développeurs doivent coopérer quotidiennement tout au long du projet.	Les relations conflictuelles ne font pas partie de l'esprit agile ; on préférera des relations collaboratives et de partenariat basées sur la confiance et le consensus. Le client (ou son représentant) est accessible et disponible, totalement impliqué dans toutes les phases du projet.
Construire des projets autour d'individus motivés. Leur donner l'environnement et le support dont ils ont besoin et leur faire confiance pour remplir leur mission.	Le facteur clé du succès d'un projet est l'équipe. Tout obstacle à son bon fonctionnement devra être levé ; un changement, s'il s'avère nécessaire, sera apporté aux processus, aux outils, à l'environnement, à la composition de l'équipe.



Parmi les Douze Principes du Manifeste Agile, nous avons (Suite) :

La méthode la plus efficace de communiquer des informations à une équipe et au sein de celle-ci reste la conversation en face à face.

Par défaut, on privilégie l'échange oral à l'écrit, pour lever toute ambiguïté et favoriser la rapidité de la compréhension. Tout ne peut être formalisé par écrit, notamment la « connaissance tacite », c'est-à-dire l'information « informelle », la culture du projet, détenues par chacun au sein d'une équipe.

Le fonctionnement de l'application est le premier indicateur d'avancement du projet.

Il n'existe pas d'autre indicateur plus pertinent que le pourcentage ou le nombre d'exigences satisfaites ; on ne mesure pas un projet à la quantité de documents produits ou au nombre de lignes de code, non significatifs pour le client.

Les méthodes agiles recommandent que le projet avance à un rythme soutenable.

La qualité du travail fourni dépend du rythme de travail qui doit être adapté en fonction des spécificités du projet. Le rythme doit être soutenu et soutenable sur la durée du projet.

Sponsors, développeurs et utilisateurs devraient pouvoir maintenir un rythme constant indéfiniment.

Ce rythme de travail est à déterminer par l'ensemble des membres de l'équipe et par le client, en fonction de la productivité de l'équipe et des priorités du client. Mais travailler en heures supplémentaires, surtout pour corriger des bogues ou des régressions, n'apporte aucune valeur ajoutée.