


Introduction à l'agilité

Aymen Sioud
Département Informatique et Mathématique
Courriel : aymen.sioud@uqac.ca

Tables des matières

- ▶ Limite des approches classiques
 - ▶ L'approche en cascades
 - ▶ L'approche en V
- ▶ L'agilité
- ▶ Scrum
- ▶ Lean et Kanban

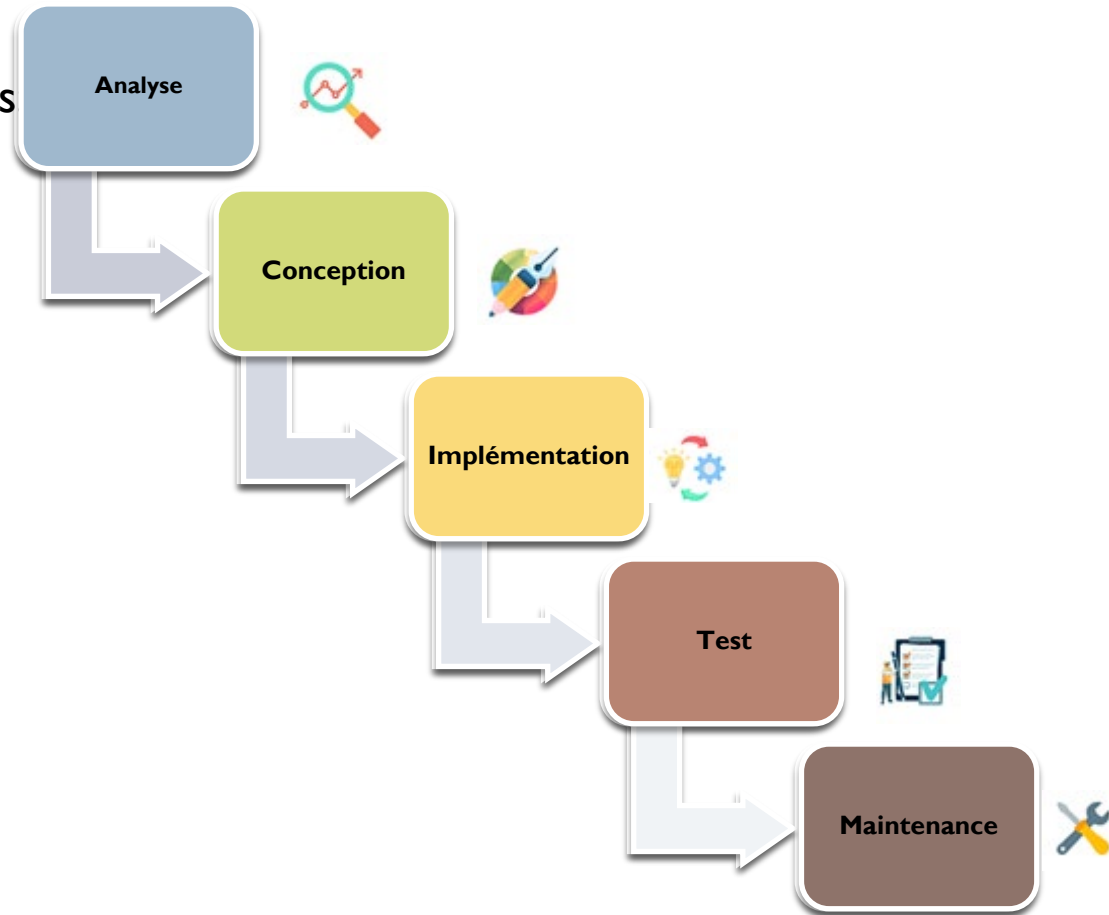


Approche en cascade



Étapes du processus de développement

- Débute à l'analyse de problème et dure tant que le logiciel est maintenu en exploitation
- Se décompose en plusieurs étapes
 - Analyse/Exigence
 - Conception/Design
 - Codage/Implémentation
 - Test
 - Maintenance



Gestion de projet en cascades

- ▶ Modèle dans lequel chaque phase doit être terminée avant de passer à la suivante.
- ▶ Lie les différentes phases telles que l'analyse des besoins, la conception, la mise en œuvre, les tests et la maintenance de manière linéaire.
- ▶ L'une des principales limites de cette méthode est sa rigidité, car une fois qu'une phase est achevée, il devient difficile et coûteux de revenir en arrière pour effectuer des modifications.
- ▶ Cela peut poser problème si des erreurs ou des besoins changeants sont identifiés ultérieurement dans le processus.

D'autres problèmes

- ▶ La méthode exige que tous les besoins et exigences soient définis à l'avance
 - ▶ peut-être problématique dans un environnement en constante évolution où les besoins des clients et du marché peuvent varier.
- ▶ Délais plus longs, car chaque phase doit être terminée avant de passer à la suivante
 - ▶ retarder ainsi la livraison et pouvant faire perdre un avantage concurrentiel.
- ▶ La communication est également limitée dans la structure séquentielle
 - ▶ peut conduire à des malentendus et des erreurs qui ne sont découvertes que tardivement.
 - ▶ risque élevé, car si des problèmes sont découverts tardivement dans le processus, ils peuvent être très coûteux à résoudre.
 - ▶ De plus, si les besoins du client changent, le projet peut même devenir obsolète avant sa livraison.

Encore des enjeux

- ▶ Manque de collaboration avec le client, qui peut souvent être exclu de l'ensemble du processus de développement jusqu'à la phase finale.
 - ▶ peut entraîner un manque d'alignement avec les attentes et les besoins du client.
- ▶ Donc, en résumé, la rigidité et le manque de flexibilité peuvent empêcher l'organisation de réagir aux changements et d'innover.
 - ▶ C'est ici que l'agilité, avec ses principes de flexibilité, de collaboration et d'adaptation, se présente comme une alternative attrayante et essentielle.

Exemple

► Système de gestion des réservations

- Une compagnie aérienne qui souhaite développer un nouveau système de gestion des réservations.
- Elle décide d'adopter la méthode de gestion de projets en cascade.
 - Après des mois de travail acharné, l'analyse des besoins est terminée, et l'équipe passe à la phase de conception.
 - Six mois plus tard, la conception est terminée et l'équipe commence le développement.
 - Tout semble se dérouler comme prévu, jusqu'à ce qu'ils approchent la phase de test, près de dix-huit mois après le début du projet.
- Lors des tests, ils découvrent que de nombreuses fonctionnalités essentielles ne répondent pas aux besoins réels des utilisateurs.
 - Par exemple, le système ne prend pas en compte les différentes classes de sièges ni les remises pour les enfants ou les personnes âgées.
 - De plus, avec l'évolution rapide de la technologie et du marché pendant ces dix-huit mois, de nouvelles plateformes mobiles ont émergé, et le système n'est pas optimisé pour ces nouveaux appareils.

Exemple (suite)

- ▶ La direction est face à un dilemme
 - ▶ revenir en arrière et réviser le système, ce qui signifie d'énormes coûts supplémentaires et des retards, ou lancer un produit qui ne répond pas pleinement aux besoins des clients.
- ▶ De plus, pendant la phase de développement, une autre compagnie aérienne a lancé un système similaire, mais beaucoup plus intuitif et adapté aux nouvelles tendances technologiques.
 - ▶ Notre compagnie aérienne est désormais en retard sur le marché, avec un produit moins compétitif.
- ▶ Avec une approche agile, la compagnie aurait pu identifier ces problèmes plus tôt, adapter le produit en conséquence, et probablement éviter ces coûts supplémentaires et ces retards.

Limites approches en cascade

- ▶ Les limites de l'approche en cascade dans la gestion de projet sont nombreuses et ont conduit beaucoup d'organisations à chercher des alternatives plus flexibles et adaptatives. Voici un aperçu détaillé de ces limites :
 - ▶ Rigidité et manque de flexibilité
 - ▶ Communication limitée
 - ▶ Longs délais de livraison
 - ▶ Manque de collaboration avec le client
 - ▶ Risque élevé
 - ▶ Exigences initiales rigides
 - ▶ Inadaptée aux projets complexes et changeants

Exemple 2

- ▶ L'équipe de développement de TechCorp a été chargée de créer un nouveau logiciel de gestion de projets. Ils ont choisi d'utiliser la méthode en cascade pour ce projet, avec les phases clairement définies de la conception à la mise en œuvre.
- ▶ Phase de conception : l'équipe a passé des semaines à recueillir les exigences du client et à planifier le projet dans ses moindres détails. Tout semblait clair et prêt pour le développement.
- ▶ Phase de développement : après avoir terminé la conception, l'équipe a commencé à coder. Cependant, à mi-chemin du développement, le client a demandé des changements majeurs.
- ▶ Difficultés rencontrées : la rigidité de la méthode en cascade signifiait que ces changements en cours de route étaient difficiles à intégrer. De plus, les tests en fin de cycle ont révélé des bugs qui étaient coûteux et chronophages à corriger.
- ▶ Résultat final : le projet a été livré en retard et au-dessus du budget. Le client n'était pas entièrement satisfait du produit, car il ne répondait pas aux besoins changeants de son entreprise.

Exemple

Transition vers les méthodes agiles

- ▶ **Collaboration et communication améliorées**
 - ▶ Les méthodes agiles ont encouragé une communication constante entre l'équipe et le client. Cela a permis de comprendre rapidement les besoins changeants et de les intégrer dans le développement.
- ▶ **Développement itératif**
 - ▶ En travaillant en sprints, l'équipe a pu tester et réviser le produit régulièrement. Les problèmes étaient identifiés et corrigés plus tôt, réduisant les coûts et les délais.
- ▶ **Réponse flexible aux changements**
 - ▶ L'agilité a permis à l'équipe de s'adapter rapidement aux demandes changeantes du client, sans perturber tout le projet.
- ▶ **Satisfaction du client**
 - ▶ En impliquant le client tout au long du processus, le produit final était plus en phase avec ce qu'il voulait et ce dont il avait besoin. Cela a conduit à une plus grande satisfaction du client et à un produit davantage réussi.

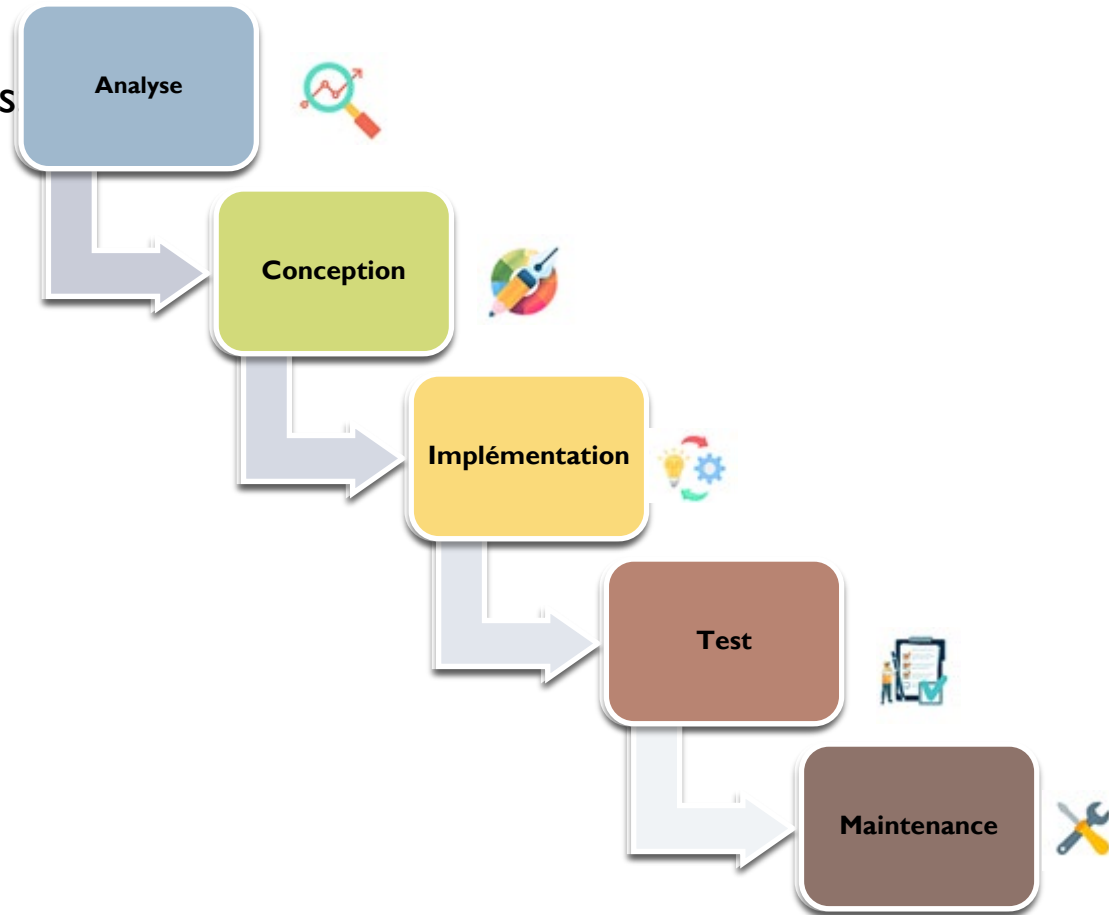


Approche en V



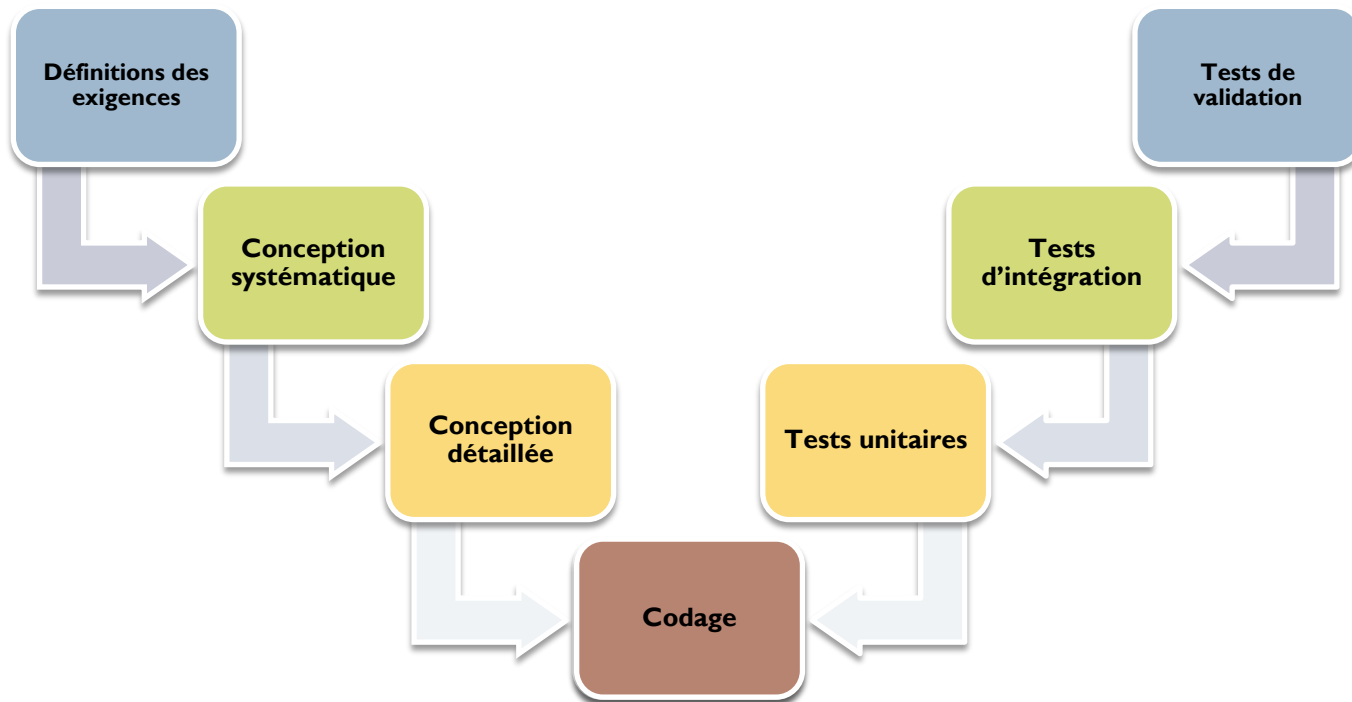
Étapes du processus de développement

- Débute à l'analyse de problème et dure tant que le logiciel est maintenu en exploitation
- Se décompose en plusieurs étapes
 - Analyse/Exigence
 - Conception/Design
 - Codage/Implémentation
 - Test
 - Maintenance

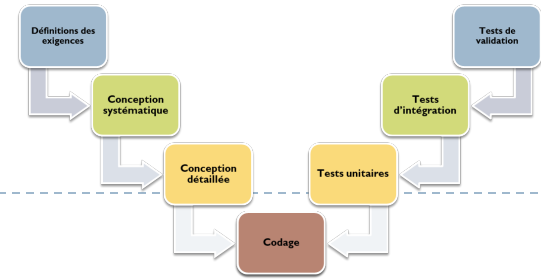


Cycle en V

- Le cycle en V est une méthodologie de gestion de projet qui améliore celle en cascade où la précision et la qualité sont cruciales.



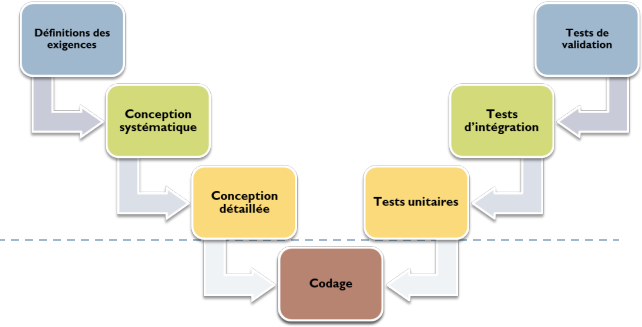
Cycle en V



- ▶ Le fondement théorique du cycle en V repose sur l'idée que chaque étape du développement doit être associée à une phase de test correspondante.
- ▶ Sur le côté descendant du "V", le projet commence par la définition des exigences, suivie par la conception systématique, la conception détaillée (architecture), le développement de modules et le codage. Chaque étape doit être validée avant de passer à la suivante, similaire à la méthode en cascade.

Cycle en V

Les tests



- ▶ Le côté ascendant du "V" représente les étapes de test et de validation.
 - ▶ Chaque phase de développement est couplée à une phase de test correspondante. Par exemple, le codage est lié au test unitaire.
- ▶ La forme en "V" illustre que chaque étape de développement est intimement liée à une étape de validation.
- ▶ Ceci garantit que les problèmes sont détectés et corrigés au fur et à mesure qu'ils se produisent, plutôt qu'à la fin du projet.
 - ▶ Amélioration de cascade 😊

Cycle en V

Le client

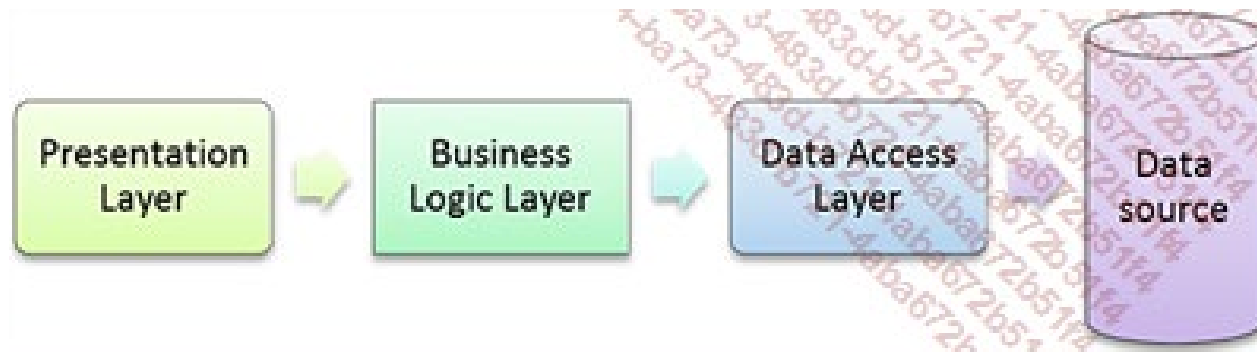
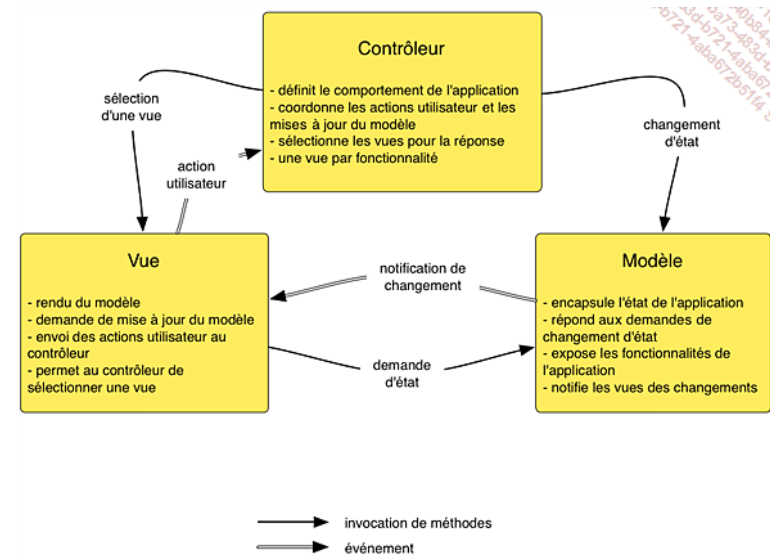
- ▶ Le cycle en V met également l'accent sur l'engagement du client ou de l'utilisateur tout au long du processus, ce qui assure que le produit final répond aux besoins et aux attentes de l'utilisateur.
- ▶ Bien que moins flexible que certaines méthodes agiles, le cycle en V permet néanmoins des ajustements en cours de route.
- ▶ Si un problème est détecté lors de la phase de test, il peut être remonté au niveau de conception correspondant pour être corrigé.
- ▶ En somme, le cycle en V est une approche méthodique et rigoureuse qui équilibre le développement et les tests.
- ▶ C'est un modèle linéaire qui peut ne pas être adapté à tous les projets, notamment ceux qui nécessitent une adaptabilité et une réactivité élevées.

Cycle en V

Application

- ▶ L'application du cycle en V dans la gestion de projet est un processus structuré qui suit une séquence linéaire. Voici comment cela fonctionne dans la pratique :
 - ▶ Définition des exigences
 - ▶ Conception systématique (architecture)
 - ▶ Conception détaillée
 - ▶ Codage
 - ▶ Tests
 - ▶ Validation
 - ▶ Maintenance
- ▶ L'approche utile dans les environnements où la qualité et la précision sont essentielles, comme dans les industries réglementées.
 - ▶ Pour les projets nécessitant une adaptabilité rapide et une collaboration étroite entre les équipes, des méthodes plus agiles peuvent être davantage appropriées.

Exemple architecture



Cycle en V

Les acteurs

- ▶ Dans le processus du cycle en V, divers acteurs jouent des rôles essentiels à différentes étapes du projet. Voici une description des principaux acteurs et de leurs responsabilités :

- ▶ Le client ou l'utilisateur final
- ▶ Le gestionnaire de projet
- ▶ L'équipe de conception
- ▶ Les développeurs
- ▶ Les testeurs
- ▶ Les analystes d'affaire
- ▶ Les parties prenantes
- ▶ L'équipe de support et de maintenance



- ▶ Chaque acteur apporte une expertise et une perspective uniques au projet, et une communication claire et une collaboration efficace entre eux sont essentielles pour le succès du projet.
- ▶ Le modèle du cycle en V exige une définition claire des rôles et des responsabilités, ainsi qu'une coordination rigoureuse à chaque étape du processus.

Effet tunnel

- ▶ Métaphore utilisée dans la gestion de projet, en particulier dans les méthodologies traditionnelles comme le cycle en V, pour décrire une situation où l'avancement du projet se fait dans l'obscurité ou sans visibilité claire de la progression jusqu'à ce qu'il approche de sa fin.
- ▶ Dans un projet géré selon le cycle en V, chaque étape doit être complétée avant que la suivante puisse commencer.
 - ▶ Cela peut créer une situation où les clients et les autres parties prenantes n'ont pas une image claire de ce qui se passe à l'intérieur du "tunnel" jusqu'à ce que le projet soit presque terminé.
- ▶ Cet effet tunnel peut conduire à plusieurs problèmes :
 - ▶ Manque de flexibilité
 - ▶ Communication
 - ▶ Risques cachés
 - ▶ Insatisfaction du client
- ▶ L'effet tunnel vs les méthodes agiles
 - ▶ Sprints, transparence, flexibilité meilleure collaboration et une adaptation aux changements



L'agilité



Introduction

Approches agiles

- ▶ Origine début des années 1990 avec la méthode en spirale.
 - ▶ Nouvelle approche de gestion de projet est née du désir de s'éloigner des méthodologies traditionnelles, souvent rigides et hiérarchiques, qui ne pouvaient pas répondre efficacement aux besoins changeants et aux environnements de projet dynamiques.
- ▶ Formalisée en 2001 avec la publication du Manifeste agile
 - ▶ Un document rédigé par un groupe de dix-sept professionnels de l'informatique.
 - ▶ Identifie les valeurs (4) et principes (12) communs qui sous-tendent une méthode de travail plus souple, adaptative et centrée sur l'humain.

AGILE

un acronyme d'abord 😊

- ▶ **Adaptive**

- ▶ L'apprentissage prime la prédiction

- ▶ **Group effort**

- ▶ Collaboration au sein de l'équipe)

- ▶ **Iterative**

- ▶ Développement rapide, intégration continue, tests et livraison fréquente).

- ▶ **Lean**

- ▶ Processus simples et souples avec le minimum de bureaucratie

- ▶ **Empowered**

- ▶ Bien équipé pour réussir le projet

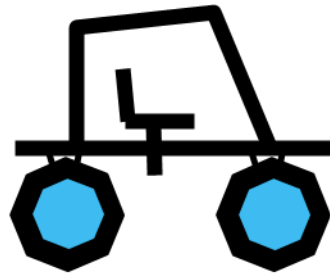
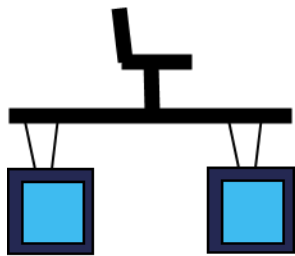
Développement agile

Agile = Itératif + Incrémental + Adaptatif

Développement Itératif

► Itératif

- Produire fréquemment une version du système qui fonctionne, mais n'ayant qu'une partie des fonctionnalités demandées qui doivent être intégrées et testées aussi soigneusement qu'une livraison finale
- Série de mini-projets courts, de durée fixe, nommée itérations. Le résultat de chacune des itérations est un système testé, intégré et exécutable



Itératif



Développement incrémental

► Incrémental

- Le logiciel évolue par incrément et chaque itération correspond à un incrément.
- Incrément = résultat de chaque itération



Incrémental



Développement adaptatif

▶ Adaptatif

- ▶ Capacité à accepter les changements
- ▶ Deux aspects
 - ▶ Adapter un logiciel fréquemment pour tenir compte des changements des besoins des utilisateurs
 - ▶ Processus qui change au fur et à mesure du temps

C'est quoi une méthode agile ?

- ▶ Méthodes itératives à planification souple
- ▶ Itérations très courtes (2 semaines à 2 mois)
- ▶ S'opposent à la procédure et à la spécification à outrance
- ▶ Exemples de méthodes :
 - ▶ eXtreme Programming
 - ▶ DSDM
 - ▶ ASD
 - ▶ CRYSTAL
 - ▶ SCRUM
 - ▶ FDD
 - ▶ ...
- ▶ Création de l'Agile Alliance en 2001 (agilealliance.org)

Les 4 valeurs

- ▶ Le Manifeste met en avant quatre valeurs fondamentales :
 - ▶ les individus et leurs interactions plutôt que les processus et les outils ;
 - ▶ le fonctionnement du logiciel plutôt que la documentation exhaustive ;
 - ▶ la collaboration avec le client plutôt que la négociation contractuelle ;
 - ▶ l'adaptation au changement plutôt que le suivi d'un plan.

Valeur I

- ▶ Les individus et leurs interactions plutôt que les processus et les outils
 - ▶ Met l'accent sur la communication humaine et la collaboration.
 - ▶ Suggère que les relations personnelles et la manière dont les équipes interagissent sont plus importantes que les processus rigides et les outils complexes.
- ▶ **Pas de documentation = c'est faux**
 - ▶ Disposer d'abord d'individus motivés à collaborer, ensuite il faut leur donner un bon guide des objectifs précis et les processus ainsi que les outils pour les soutenir.

Valeur II

- ▶ Le fonctionnement du logiciel plutôt que la documentation exhaustive
- ▶ Le développement de logiciels fonctionnels doit primer sur la documentation exhaustive.
- ▶ Le meilleur transfert de connaissances s'effectue par le travail en équipe
- ▶ Les développeurs agiles doivent participer à toutes les étapes du développement et travailler ensemble.
- ▶ Documentation concise, ne décrivant que les grandes lignes de l'architecture du produit et régulièrement mise à jour, et à la documentation permanente du code lui-même

Valeur III

- ▶ La collaboration avec le client plutôt que la négociation contractuelle
- ▶ La collaboration continue avec le client est essentielle pour comprendre ses besoins et ses exigences.
- ▶ Au lieu de s'en tenir strictement aux termes d'un contrat, les équipes agiles cherchent à travailler en étroite collaboration avec les clients pour s'assurer que le produit final répond à leurs attentes.

Valeur IV

- ▶ L'adaptation au changement plutôt que le suivi d'un plan
 - ▶ Les équipes agiles doivent être flexibles et capables de s'adapter rapidement aux changements.
 - ▶ Atteindre un compromis entre les spécifications initiales présentées aux développeurs et le résultat final
 - ▶ Utilisation d'outils de suivi et une attitude constante de coopération avec le client
 - ▶ Capacité à accepter le changement
 - ▶ Voir les changements comme une opportunité plutôt qu'un obstacle

Objectifs des 4 valeurs

- ▶ Encourager une culture où la flexibilité, la communication ouverte et la réponse rapide aux changements sont privilégiées.
- ▶ Soutenir une approche de la gestion de projet plus humaine et centrée sur le client, ce qui a été un facteur clé du succès de la méthodologie agile dans de nombreux secteurs et organisations.

Principes 1 - 3

1. Notre plus haute priorité est de satisfaire le client en lui livrant rapidement, et de façon continue, un logiciel de qualité
 - ▶ Affinement successif et le feed-back du client
2. Accueillir les changements aux exigences, même tardifs. Les processus agiles mettent à profit les changements pour augmenter les avantages compétitifs du client
 - ▶ Intégrer les changements aux exigences même s'ils arrivent tard
3. Livrer fréquemment un logiciel fonctionnel, de quelques semaines à quelques mois, en privilégiant les courts délais.
 - ▶ Permettre aux utilisateurs de fournir leur rétroaction sur le produit plus tôt dans le projet.

Principes 4 - 6

4. Le client et les développeurs doivent travailler de concert quotidiennement durant tout le projet
 - ▶ Interaction permanente
5. Bâtir des projets autour d'individus motivés. Donnez-leur l'environnement et le support requis, et ayez confiance qu'ils feront le travail.
 - ▶ Les individus sont remplaçables, les rôles ne le sont pas
6. La méthode la plus efficace de transmettre l'information à l'équipe de développement et à l'intérieur de celle-ci est par conversation de personne-à-personne.
 - ▶ Le mode de communication par défaut au sein d'une équipe agile est la conversation face à face

Principes 7 - 9

7. Un logiciel fonctionnel est la mesure primaire de l'avancement du projet.
 - ▶ Disposer de logiciels fonctionnels à la fin de chaque itération est le premier indicateur d'avancement du projet

8. Les processus agiles favorisent le développement durable. Les responsables, les développeurs et les usagers devraient pouvoir maintenir un rythme constant jusqu'à la fin du projet.
 - ▶ Adapter son rythme pour préserver la qualité de son travail durant toute la durée du projet.

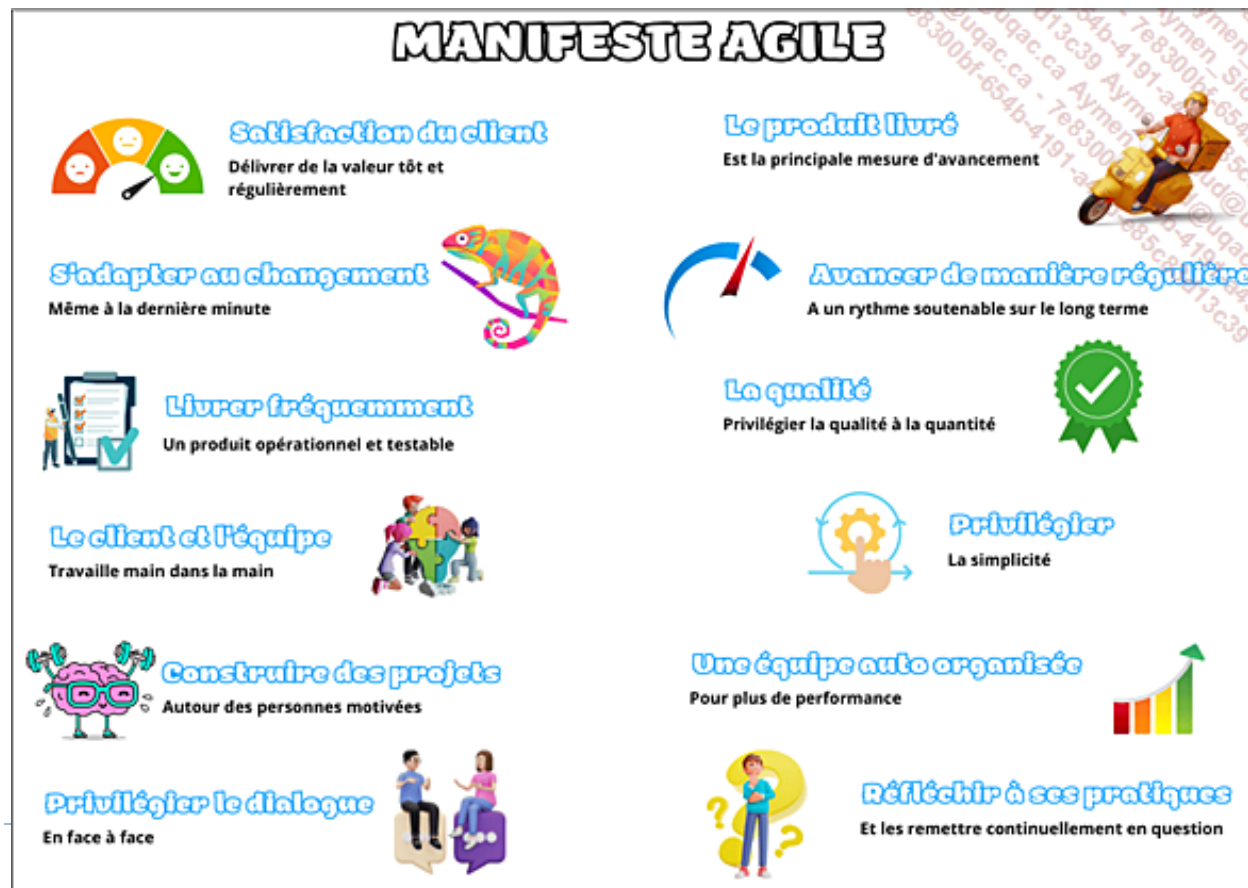
9. Une attention constante à l'excellence technique et un bon design augmentent l'agilité.
 - ▶ Maintenir le code propre, le nettoyer et le réécrire (Refactoring)

Principes 10 - 12

10. La simplicité - l'art de minimiser la quantité de travail fait inutilement – est essentielle.
 - ▶ Construire le système le plus simple répondant aux besoins actuels.
 - ▶ Rien ne sert d'essayer d'anticiper les besoins futurs.
11. Les meilleures architectures, exigences et conception émergent d'équipes auto-organisées
 - ▶ La définition des responsabilités est confiée à toute l'équipe et les tâches sont partagées selon le principe du volontariat.
12. À intervalles réguliers, l'équipe doit réfléchir à la façon de devenir plus efficace, puis adapter et ajuster son comportement en conséquence.
 - ▶ Essayer de s'ajuster continuellement tout en demeurant dans un cadre organisationnel et, par la suite, ce principe peut être aussi lié à la notion d'apprentissage organisationnel (réorientation des compétences, systèmes d'information, motivation et promotion de l'initiative personnelle).

4 valeurs et 12 principes

- ▶ Ces principes soutiennent et amplifient les quatre valeurs agiles, guidant les équipes dans leur application pratique de l'agilité.
- ▶ Permettent de mettre l'accent sur le client, la collaboration, l'adaptabilité, la qualité et la réflexion continue, permettant une livraison de logiciel plus efficace et alignée avec les besoins du client.



Agilité et structure

- ▶ L'agilité est souvent perçue à tort comme une approche non structurée ou anarchique.
 - ▶ En réalité, elle est très structurée, mais de manière différente des méthodes traditionnelles.
 - ▶ Cadre bien défini
 - ▶ Itérations courtes
 - ▶ Inspection et adaptation
 - ▶ Collaboration étroite avec le client
 - ▶ Auto-organisation
 - ▶ Accent sur la qualité
 - ▶ Transparence



Scrum



Introduction Scrum

- ▶ Méthodologie agile axée sur la collaboration, la transparence et la livraison continue de valeur.
- ▶ Utilisée principalement dans le développement de logiciels, mais elle est aussi adaptable à divers types de projets.
- ▶ Scrum divise le développement en cycles appelés sprints, qui durent généralement de deux à quatre semaines.
 - ▶ Chaque sprint vise à produire un élément fonctionnel du produit.

Les rôles

- ▶ Au sein de Scrum, nous distinguons trois rôles principaux :
 - ▶ le Product Owner, responsable de définir les priorités du produit selon les besoins des clients et des parties prenantes ;
 - ▶ le Scrum Master, qui s'assure du bon déroulement du processus Scrum ;
 - ▶ et l'équipe de développement, qui met en œuvre les éléments choisis du produit.

Déroulement

- ▶ Le processus commence avec la planification du Sprint, où l'équipe détermine ce qui sera développé pendant le sprint.
- ▶ Chaque jour, l'équipe tient une mêlée quotidienne, une réunion rapide pour synchroniser les activités et planifier les 24 heures à venir.

Artefacts

- ▶ **Scrum utilise trois principaux artefacts :**
 - ▶ le backlog du produit, une liste de toutes les fonctionnalités, fonctions et exigences du produit final ;
 - ▶ le backlog du Sprint, une liste de tâches à accomplir lors du sprint en cours ;
 - ▶ et l'incrément, l'ensemble des éléments du produit terminés à la fin du sprint.

La revue et la retrospective sprint

- ▶ À la fin de chaque sprint, l'équipe organise la revue du sprint pour présenter l'incrément aux parties prenantes et recueillir leurs retours.
- ▶ Cela est suivi par la rétrospective du sprint, où l'équipe réfléchit à ce qui s'est bien passé et à ce qui pourrait être amélioré.

Les valeurs

- ▶ Les valeurs fondamentales de Scrum incluent
 - ▶ le courage,
 - ▶ la concentration,
 - ▶ l'ouverture,
 - ▶ le respect
 - ▶ l'engagement.
- ▶ L'un des avantages de Scrum est sa flexibilité.



Lean et Kanban



Méthodes agiles, différentes pratiques...

(<https://www.agilealliance.org/agile101/subway-map-to-agile-practices/>)

Subway Map to Agile Practices



The colored "subway" lines represent practices from the various Agile approaches or areas of concern.



C'est quoi ? 😊

- ▶ D'autres mouvements parallèles émergent et enrichissent la philosophie agile.
- ▶ Parmi eux, les approches Lean et Kanban se démarquent particulièrement, offrant des perspectives et des outils qui complètent et élargissent la vision agile.

Lean

- ▶ Système de production de Toyota dans les années 1950.
- ▶ Optimiser l'efficacité en éliminant tout ce qui est superflu ou qui n'ajoute pas de valeur au produit final.
 - ▶ Se concentre sur la minimisation des déchets.
- ▶ Il existe sept catégories classiques de déchets dans le Lean :
 - ▶ Surproduction : produire plus que nécessaire.
 - ▶ Attente : temps mort où une ressource n'est pas utilisée.
 - ▶ Transfert : mouvements inutiles de produits ou d'informations.
 - ▶ Surtraitement : faire plus que ce qui est nécessaire du point de vue du client.
 - ▶ Mouvement : déplacements inutiles des travailleurs.
 - ▶ Inventaire : stocker plus que ce qui est immédiatement nécessaire.
 - ▶ Défauts : travail qui contient des erreurs ou des défauts.
- ▶ **Principes pouvant être adaptés à tout type d'organisation ou de projet, y compris le développement logiciel.**

Kanban

- ▶ Système de visualisation qui permet de gérer le travail en se concentrant sur le débit et la capacité (Toyota)
- ▶ « carte » ou « panneau » en japonais.
 - ▶ Chaque élément de travail est représenté par une carte qui se déplace sur un tableau, passant d'une colonne à l'autre à mesure qu'il avance dans le processus.
- ▶ Le Kanban met l'accent sur :
 - ▶ la visualisation du travail : tout est visible, ce qui facilite la communication et la collaboration ;
 - ▶ la limitation du travail en cours : pour éviter la surcharge et se concentrer sur la finition ;
 - ▶ la gestion du flux : s'assurer que le travail avance de manière fluide et constante ;
 - ▶ l'amélioration continue : toujours chercher des façons d'optimiser le processus.
- ▶ Principes parfaitement alignés avec la mentalité agile, faisant du Kanban un excellent complément ou même une alternative à d'autres méthodes agiles.

Toujours dans un esprit d'amélioration

- ▶ **Améliorer**

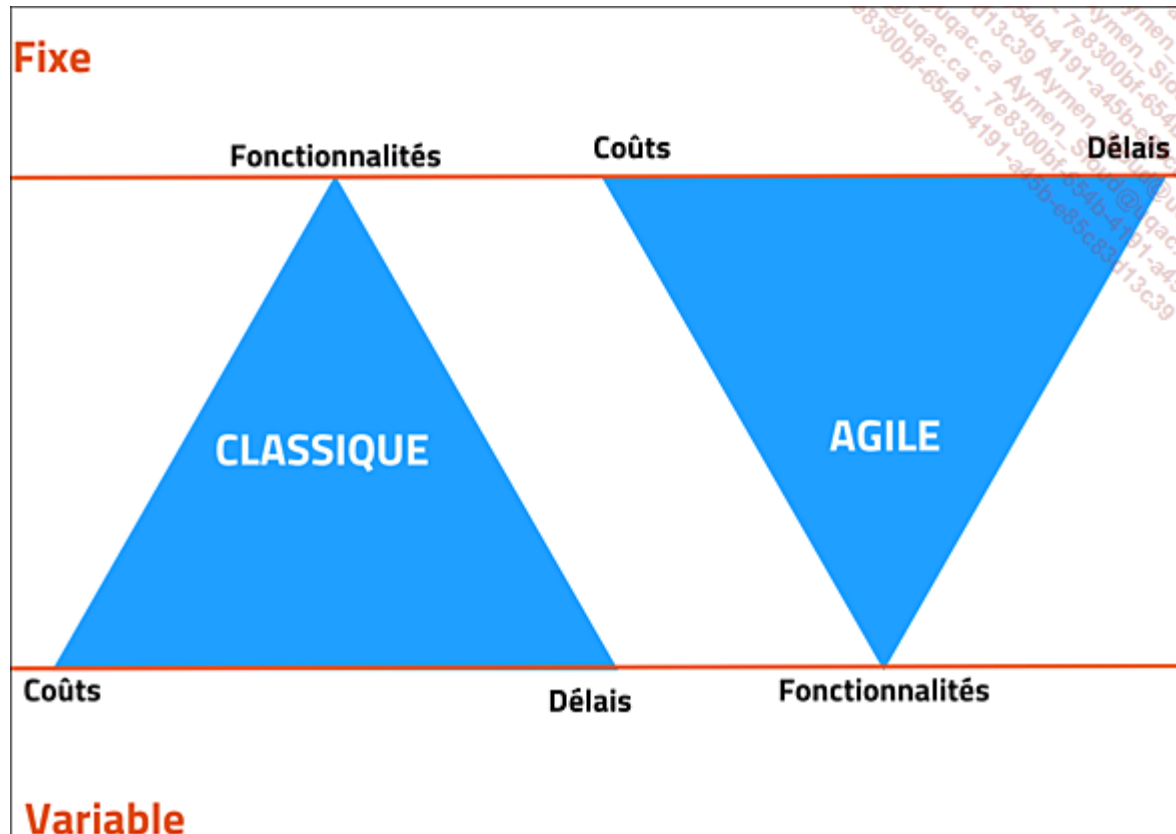
- ▶ L'efficacité
- ▶ La flexibilité
- ▶ La satisfaction client sans précédent.



Comparaison



Approches distinctes



Synthèse

Aspect	Classique	Agile
Approche projet	Linéaire et séquentielle	Itérative et incrémentale
Flexibilité	Difficiles à changer (\$\$\$)	Adaptation et acceptation
Livraison produit	Fin du cycle	Livraisons partielles validées
Rôle des clients	Impliqués au début et à la fin	Engagement continu
Gestion de l'équipe	Contrôle au chef de projet	Équipe collaborative
Évaluation du progrès	À la fin de chaque phase	Évaluation continue
Risques et incertitudes	Analysés au début	Gestion constante