

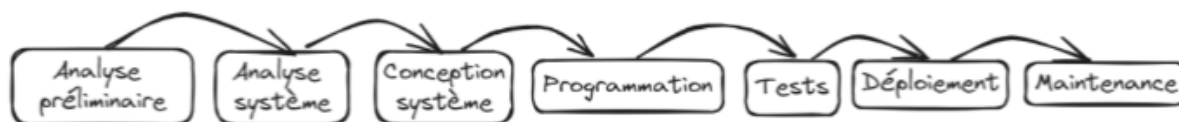
1 - Méthodologies De Développement De Systèmes

Méthodologie	Flux global	Analyse	Conception	Développement	Tests
SDLC	Structuré	Méthodique	Méthodique	Séquentiel	SLD
RAD	Structuré	Méthodique	Interactif	Interactif	CALD
Agile	Incrémental	Itératif	Itératif	Itératif	CALD
Lean	Minimaliste	Itératif	Itératif	Itératif	CALD

SLD = Suivent le développement

CALD = Combinés avec le développement

1-1 Cycle de vie du développement des systèmes SDLC



A- Analyse Préliminaire:

- Évaluation d'une demande de remplacement ou de création d'un nouveau système.
- Étude de faisabilité technique, économique et juridique.
- Réalisation par un analyste des besoins métier ou un analyste commercial.

B- Analyse Système:

- Détermination des exigences spécifiques pour le nouveau système par un ou plusieurs analystes système.
- Documentation des procédures, interviews des acteurs clés, développement des exigences de données.
- Rédaction d'un document d'exigences système.
- Réalisation par un analyste système.

C- Conception système:

- Développement des détails techniques spécifiques nécessaires au système par un concepteur.
- Traduction des exigences métier en exigences techniques spécifiques.

- Développement de la conception de l'interface utilisateur, de la base de données, des entrées et sorties de données, et des rapports.
- Rédaction d'un document de conception système.
- Réalisation par un analyste système, un développeur ou un architecte système.

D- Programmation:

- Écriture du code par les programmeurs en utilisant le document de conception système comme guide.
- Développement d'un programme de travail initial répondant aux exigences spécifiées dans la phase d'analyse système et à la conception développée dans la phase de conception système.
- Réalisation par des développeurs, des ingénieurs logiciels, des programmeurs ou des codeurs

E- Tests:

- Mise en œuvre d'une série de tests structurés pour le programme logiciel développé dans la phase de programmation.
- Test unitaire pour évaluer les différentes parties du code pour détecter les erreurs ou les bogues.
- Test système pour s'assurer que les différents composants du système fonctionnent correctement ensemble.
- Test d'acceptation utilisateur pour permettre aux utilisateurs de tester le système afin de s'assurer qu'il répond à leurs normes.
- Résolution des bugs, des erreurs ou des problèmes détectés lors des tests.
- Réalisation par des testeurs, des analystes de test ou des responsables de l'assurance qualité.

F- Déploiement:

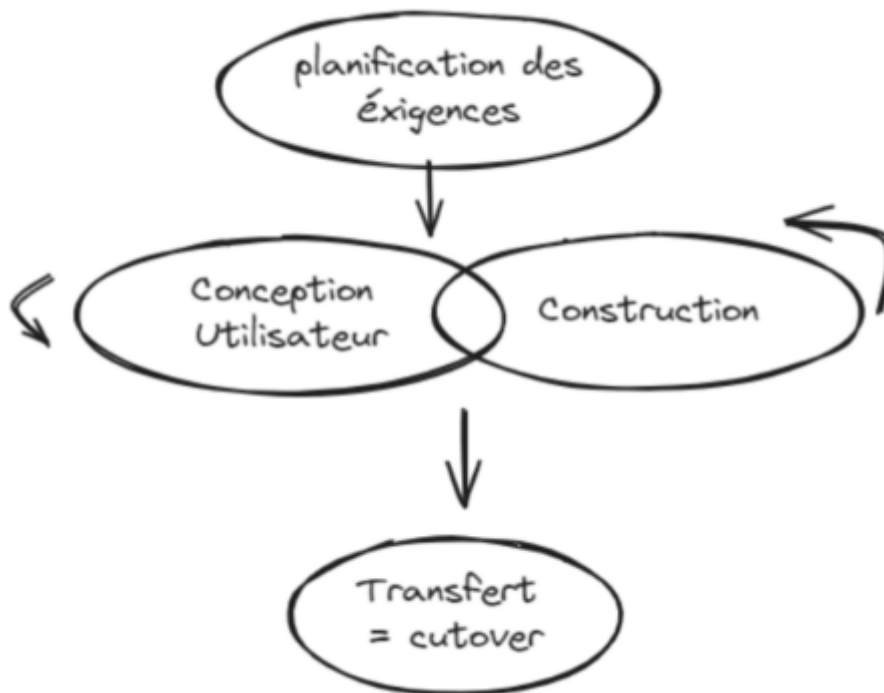
- Implémentation du nouveau système dans l'organisation.
- Formation des utilisateurs.
- Fourniture de la documentation.
- Conversion des données de l'ancien système vers le nouveau système.
- Réalisation par des gestionnaires de projet, des responsables du déploiement ou des responsables de la mise en œuvre.

G- Maintenance:

- Mise en place d'un processus de support structuré pour le système.
- Correction des bugs signalés.

- Évaluation et mise en œuvre des demandes de nouvelles fonctionnalités.
- Réalisation de mises à jour du système et de sauvegardes du logiciel pour chaque nouvelle version du programme
- . Réalisation par des responsables de maintenance, des responsables de la fiabilité ou des responsables du support technique.

1-2 Développement Rapide D'Applications: RAD



- Se concentre sur la création rapide d'un modèle fonctionnel du logiciel, l'obtention de commentaires des utilisateurs et l'utilisation de ces commentaires pour mettre à jour le modèle fonctionnel.

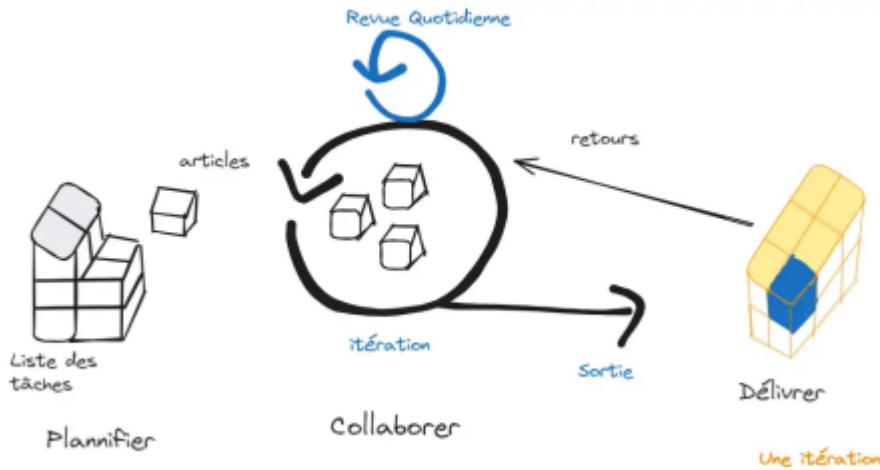
A- Méthodologie RAD :

- **Planification des exigences** : définit les exigences globales du système, identifie une équipe et détermine la faisabilité.
- **Conception utilisateur** : crée de manière interactive la conception du système avec les représentants des utilisateurs.
- **Construction** : construit la prochaine version du système grâce à un processus interactif avec les utilisateurs.
- **Transfert** : implique le passage de l'ancien système au nouveau logiciel.

B- RAD Vs. SDLC

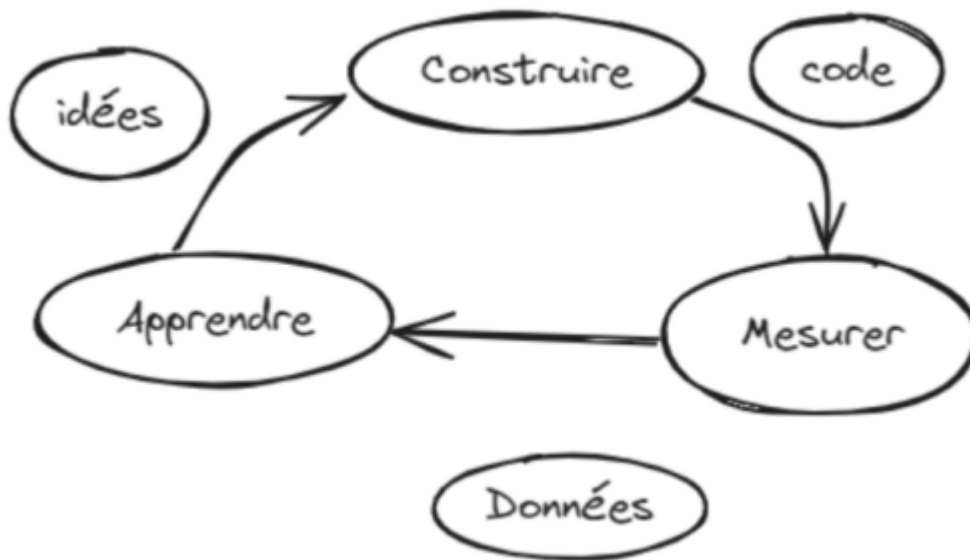
- RAD est mieux adapté aux petits projets et SDLC est mieux adapté aux projets volumineux et gourmands en ressources.

1-3 Développement Rapide D'Applications: RAD



- Un ensemble de méthodologies qui utilisent des changements incrémentiels avec un accent sur la qualité et l'attention aux détails.
- Chaque incrément est publié dans une période déterminée (appelée boîte à temps), créant un calendrier de publication régulier avec des objectifs très spécifiques.
- Bien qu'elle soit considérée comme une méthodologie distincte de RAD, les deux méthodologies partagent certains des mêmes principes tels que le développement itératif, l'interaction avec l'utilisateur et la flexibilité pour le changement.
- Les méthodologies agiles sont basées sur le «Manifeste agile», publié pour la première fois en 2001.
- Développement Agile est Itératif
- Les caractéristiques de la méthodologie agile incluent:
 - De petites équipes inter fonctionnelles qui comprennent des membres de l'équipe de développement et des utilisateurs; -
 - Des réunions quotidiennes d'état pour discuter de l'état actuel du projet;
 - Des incréments de courtes durées (de quelques jours à une ou deux semaines) pour que chaque changement soit terminé;
 - Un projet à la fin de chaque itération qui démontre les progrès aux parties prenantes.
- L'objectif des méthodologies agiles est de fournir la flexibilité d'une approche itérative tout en garantissant un produit de qualité.

1-4 Lean



- Une nouvelle méthodologie issue du livre à succès The Lean Startup d'Eric Reis.
- La méthodologie se concentre sur la création d'un produit minimum viable MVP, une application logicielle fonctionnelle avec juste assez de fonctionnalités pour démontrer l'idée derrière le projet.
- L'équipe de développement donne le MVP à des utilisateurs potentiels pour évaluation.
- Les commentaires sur le MVP sont générés sous deux formes :
 - l'observation directe et la discussion avec les utilisateurs, l
 - es statistiques d'utilisation recueillies à partir du logiciel lui-même.
- À l'aide de ces deux formes de commentaires, l'équipe détermine si elle doit continuer dans la même direction ou repenser l'idée de base du projet, changer les fonctions et créer un nouveau MVP.
- Ce changement de stratégie s'appelle un pivot.
- Plusieurs itérations du MVP sont développées, avec de nouvelles fonctions ajoutées à chaque fois en fonction des commentaires, jusqu'à ce qu'un produit final soit terminé.
- La plus grande différence entre les méthodologies itératives et non itératives est que l'ensemble complet des exigences du système n'est pas connu au lancement du projet.
- Au fur et à mesure de la publication de chaque itération du projet, les statistiques et les commentaires recueillis sont utilisés pour déterminer les exigences.
- La méthodologie Lean fonctionne au mieux dans un environnement entrepreneurial où une entreprise souhaite savoir si le développement du logiciel est nécessaire

2- Outils De Programmation

2-1 Ide: Environnement De Développement Intégré

- Un outil logiciel qui fournit un ensemble d'outils pour aider les programmeurs à développer des logiciels
- Les fonctionnalités clés d'un IDE incluent :
 - Un éditeur de code pour écrire et modifier le code source.

- Un système d'aide pour accéder à la documentation sur le langage de programmation.
- Un compilateur ou interpréteur pour convertir le code source en langage machine exécutable.
- Un outil de débogage pour identifier et corriger les erreurs dans le code.
- Un mécanisme d'enregistrement/d'extraction permettant aux équipes de programmeurs de travailler simultanément sur un projet sans écraser le code des autres.
- Exemples d'IDE
 - Visual Studio de Microsoft : prend en charge divers langages de programmation Microsoft, notamment Visual Basic, Visual C et Visual C#.
 - Eclipse d'Oracle : prend en charge une variété de langages de programmation, notamment Java, C, C++, Perl, Python et R.

2-2 Outils Case

- Alors qu'un IDE fournit plusieurs outils pour aider le programmeur à écrire le programme, le code doit encore être écrit.
- Les outils de génie logiciel assisté par ordinateur CASE permettent à un concepteur de développer un logiciel avec peu ou pas de programmation.
- Au lieu de cela, l'outil CASE écrit le code pour le concepteur. Les outils CASE existent dans de nombreuses variétés.
- Leur objectif est de générer un code de qualité basé sur les entrées créées par le concepteur.

3- Méthodologies De Mise En Œuvre

- Une fois qu'un nouveau système est développé ou acheté, l'organisation doit déterminer la meilleure méthode d'implémentation.
- Convaincre un groupe de personnes d'apprendre et d'utiliser un nouveau système peut être un processus très difficile.
- Demander aux employés d'utiliser un nouveau logiciel et de suivre un nouveau processus métier peut avoir des effets très profonds au sein de l'organisation.
- Il existe plusieurs méthodologies différentes qu'une organisation peut adopter pour mettre en œuvre un nouveau système.
- Quatre des plus populaires sont répertoriés ci- dessous.

3-1 Basculement Direct:

- l'organisation sélectionne une date particulière à laquelle l'ancien système ne sera plus utilisé.

- À cette date, les utilisateurs commencent à utiliser le nouveau système et l'ancien système n'est pas disponible.
- Les avantages de l'utilisation de cette méthodologie sont qu'elle est rapide et la moins coûteuse.

3-2 Pilote:

- Le nouveau système est mis en œuvre dans un département ou une division pilote avant d'être déployé dans l'ensemble de l'organisation.
- Cela permet à l'organisation d'identifier et de résoudre les problèmes potentiels avant que le nouveau système ne soit déployé à un plus grand nombre d'utilisateurs.
- Les avantages de l'utilisation de cette méthodologie sont qu'elle réduit les risques et permet à l'organisation de tester le nouveau système dans un environnement réel.

3-3 Conversion Parallèle:

- l'ancien système et le nouveau système sont exécutés en parallèle pendant une période de transition.
- Cela permet aux utilisateurs de se familiariser avec le nouveau système avant qu'il ne devienne le système de production
- Les avantages de l'utilisation de cette méthodologie sont qu'elle réduit le risque de défaillance et qu'elle permet aux utilisateurs de se former au nouveau système à leur propre rythme.

3-4 Mise En Œuvre Progressive

- Les fonctionnalités du nouveau système sont implémentées (utilisées) graduellement tandis que les fonctionnalités correspondantes dans l'ancien système sont désactivées.
- Cette approche permet à l'organisation de passer lentement d'un système à un autre

4- Gestion Du Changement Et Maintenance

4-1 Gestion Du Changement

- Lors de la mise en ligne de nouveaux systèmes et de l'élimination progressive des anciens systèmes, il devient important de gérer la manière dont le changement est mis en œuvre dans l'organisation.
- Le changement ne doit jamais être introduit dans le vide. L'organisation doit s'assurer de communiquer les changements proposés avant qu'ils ne se produisent et de planifier pour minimiser l'impact du changement qui se produira après la mise en œuvre.
- La gestion du changement est une composante critique de la supervision informatique.

4-2 Maintenance

- Une fois qu'un nouveau système a été introduit, il entre dans la phase de maintenance.
- Le système est en production et est utilisé par l'organisation.
- Bien que le système ne soit plus activement développé, des modifications doivent être apportées lorsque des bugs sont détectés ou que de nouvelles fonctionnalités sont demandées.
- Pendant la phase de maintenance, la gestion informatique doit s'assurer que le système continue de rester aligné sur les priorités commerciales et continue de fonctionner correctement.

Références Bibliographiques

Bourgeois, David T., James L. Smith, Shouhong Wang, and Joseph Mortati. *Information systems for business and beyond*. Saylor Academy, 2019