Résumer - Agile.

+ Définition d'un projet :

est une série d'activités conduites par une effort complexe, non répétitif et unique, limité par des contraintes, de temps, de budget et de ressources.

+ Gestion de projets :

→ La gestion de projet est l'ensemble des activités visant à organiser le bon déroulement d'un projet et pour atteindre les objectifs ~~en~~ en temps et en heures selon les objectifs visées.

+ Une ressource :

les éléments nécessaires à la réalisation d'une tâche ou d'un projet (humaines, financières, matérielles)

+ livrable :

les résultats tangibles attendus par un projet (un projet peut avoir plus d'un livrable)

+ charte de projet :

un document décrivant les objectifs, la manière de réaliser le projet et les parties prenantes.

+ chef de projet : (CPI).

+ Expert en informatique garant l'état d'avancement d'un projet informatique il - se doit posséder des compétences techniques et managériales à la fois.

+ Rôles de chef de projet :

+ Intégrer un logiciel ou une solution informatique spécifique

+ couvrent l'ensemble de toutes les étapes du projet depuis le cahier de charge jusqu'à la recepti par le client et maintenance.

+ il coordonne les équipes, gère les ressources et communique avec les parties prenantes.

+ Cahier de charge :

est un document qui permet de formaliser les besoins, les objectifs, les contraintes les délais et budget d'un projet.

→ MOA : maîtrise d'ouvrage → le client qui connait les besoins

→ MOE : maîtrise d'oeuvre → fournisseur, le réalisateur.

+ Matrice RACI :

est un outil de gestion de projet utilisé pour clarifier les rôles et donner à chaque membre de l'équipe un niveau de responsabilité.

R : Responsable ⇒ personne qui execute la tâche.

CamScanner

A : accountable : personne qui responsable de décisions finales vis-à-vis le travail.

C : Consulted : consulté avant qu'un décision ou une action soit prise.

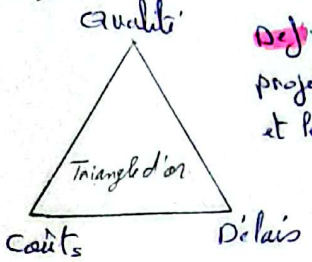
I : informed : celui qui doit être informées des résultats, décisions prises.

+ les caractéristiques de base d'un projet :

- limité dans le temps : y a un début un fin.
- une activité : qui transforme les ressources en resultats dans un délai précis.
- les ressources : sont les choses qui se transforment en résultat tangibles
- les resultats attendus :

+ les contraintes dans la gestion d'un projet :

- contraintes de délais
- contraintes due aux clients
- contraintes de coûts
- contraintes de qualité



Déf : sont les limites générales d'un projet, notamment les délais, les coûts et les risques.

+ phases de la gestion du projet :

Lancement - planification - exécution - clôture

- phase de lancement :
  1. objectifs :
  2. budget :
  3. rôle et responsabilité : chef de projet - équipe de développement - client consommateur
  4. cahier de charges :
     1. contexe de problème : projet
     2. objectif de projet
     3. périmètre (limites)
     4. Ressources (financières, humain ...)
     5. Délais.
- phase planification :

1. Définir le contenu du projet :

   La charte de projet →
   - le cahier des charges
   - le plan de management de projet (PMP)
   - Work Break down structure (WBS)
   - Resource Breakdown structure (RBS)
   - le plan de communication.
2. Découper le projet (WBS) :

   Gantproject - microsoft project - serena - TeamGantt
3. Déterminer les ressources projet :
4. estimer les durée et ordonnancer les tâches.
5. estimer les coûts des tâches
6. identifier les risques : financiers - humains - temporels - techniques - juridiques - organisationnels

② • La phase d'exécution :

La phase d'exécution sert à crées les livrables de projet selon ce qui ~~été~~ a été défini.

• La phase de clôture :

⇒ Le bilan de projet :

- est le document que le chef de projet doit réaliser et soumettre à la fin du projet.
  + Assurer la transition vers l'exploitation
  + s'assurer que les objectifs du projet sont atteints
  + Documenter l'approbation et la réception du projet
  + finaliser les contrats des prestataires.

+ Partie prenante de projet :

→ L'ensemble des personnes et des organisations qui sont quelque chose à voir avec le projet

+ Acteur externe :

• fournisseurs - les organismes privés - les investisseurs et partenaires.

+ Acteur interne :

• le sponsor du projet - la direction - l'équipe de projet.

+ les différentes méthodes de gestion de projet :

① méthode prévisibles (cascade, V, Y) :

• Méthodes prévisibles :

+ regroupent des méthodes de gestion de projet qui suivent une organisation stricte et avancent étape par étape sans revenir en arrière.

① Le modèle en cascade (waterfall) :

⇒ est une méthode où les différentes phases d'un projet se suivent de manière linéaire et ordonné on termine une phase avant de passer à la suivante.

+ exigence → Analyse → conception → mise en oeuvre → Validation mise en service.

→ Avantages :

- cette méthode offre une planification claire dès le départ
- et des étapes bien définies et organisées.

→ Inconvénients :

- il est moins adaptée aux projets complexes
- manque de flexibilité pour les changements en cours
- les erreurs découverts à la fin sont très difficiles à corriger.

② Le cycle en Y :

- est une méthode développement basée sur deux axes.

→ l'axe fonctionnel : se concentre sur les besoins

→ l'axe technique : traite la manière dont ce fonctionnalités sont réalisées.

CamScanner

- Avantages :
- combine organisation et flexibilité grâce à des itérations qui permettent d'améliorer le projet progressivement.
- inconvénients :
- il peut être complexe.
- nécessitent des équipes qualifiés.
- et une bonne maîtrise des outils comme UML.

③ Le cycle en V :

- modèle de gestion de projet dérivé du modèle en cascade, il suit une progression linéaire en deux phases la conception et la validation par testes.

- Avantages :
- clarté et simplicité
- Adapté aux structures multi-sites.

- inconvénients :
- moins flexible que des méthodes plus modernes comme les méthodes agiles.

② méthodes imprévisibles (Agile) : (scrum - Kanban - lean Development)

→ les méthodes Agiles se concentrent sur un développement flexible et adaptatif.

+ Principe de l'agilité : (12)

+ Livrer de la valeur au client
+ Intégrer les demandes de changement
+ Livraisons fréquentes
+ collaboration continue
+ communication face à face
+ Réaliser les projets avec des personnes motivées
+ Minimiser la quantité de travail inutile.
+ avancer à un rythme soutenable et constant.
+ Autogestion
+ Amélioration continue
+ objectifs à court terme
+ Flexibilité.

4 - Valeurs agiles fondamentales :

① les individus et les interactions plus que les processus et les outils
② des logiciels opérationnels plus qu'une documentation exhaustive.
③ la collaboration avec les clients plus que la négociation contractuelle
④ l'adaptation au changement plus que le suivi d'un plan.

Scrum → Développe le projet en cycles courts appelés 'sprints'

Kanban → Visualise les tâches et leur état d'avancement à l'aide d'un tableau avec des colonnes représentant les différentes étapes du processus

lean Development → similaire à Kanban mais avec une amélioration continue des processus.

CamScanner

③ + Répartition de l'ensemble des fonctionnalités en tâches :

① approche descendante
- Commencez par une vue géninal par la suite. Décomposez ce projet en sous-projets.

② Approche ascendante
Identifier les petites tâches et les regroupez en sous-projets.

③ Approche combinée
c'est un mélange de l'approche descendante et ascendante.

+ Périmètre de projet :
• Représente tous les éléments nécessaires à la réalisation d'un projet, notamment les tâches, les délais et les ressources.

La méthode MPM : (méthode des potentiels et des antécédents Mélka)
• est une technique utilisée en gestion de projet pour planifier et organiser les tâches d'un projet

+ Chemin critique :
• Il permet d'identifier la séquence de tâches qui détermine la durée total du projet.

• Avantage MPM :
① Visualisation claire
② Détection rapide
③ optimisation des ressources

+ Le diagramme de Gantt :
est un graphique qui consiste à placer les tâches chronologiquement en fonction des contraintes techniques de succession.

+ Réseaux pert :
- Le reseau pert est constitué des étapes et des taches.

+ Tâche :
c'est le déroulement dans le temps d'une opération.

+ Marge Libre :
Le retard maximal qu'on peut faire dans une tâche sans retarder le début de les tâches suivantes.

+ Marge Total :
Le retard maximal qu'on peut faire dans une tâche sans retarder le l'ensemble du projet.

CamScanner

+ Définit Scrum :

est une méthode agile de gestion de projet, Vise à améliorer la productivité et optimiser le produit Via des feedbacks réguliers.

+ Rôles et responsabilités :

- stakeholders : sont des personnes externes comme de clients
- le scrum master :

  Le guide de l'avancement du projet, il assure le respect des valeurs du scrum par l'équipe.

- product owner :

  c'est l'expert qui collabore avec le client et maîtrise ses besoins.

- l'équipe de développement : (équipe scrum) :

  • Se compose de professionnels qui fournissent un incrément potentiellement publiable à la fin de chaque sprint.

+ Evénements Scrum :

• **Sprint planing :**

- la réunion que l'équipe scrum réalisent en ouverture de sprint pour cadrer le sprint qui démarre.

• **Sprint :**

- pendant sprint l'équipe se concentre sur le développement d'une fonctionalité à livrer à la fin de cycle. en général la durée d'un sprint presque deux semaines ou trois.

• **Daily Scrum :**

- est une réunion quotidienne de 5-10 min où l'équipe discute des tâches réalisées et les obstacles rencontrées.

• **Sprint review :**

- est une réunion en fin de sprint où l'équipe fait un démo de la fonctionnalité crée, et le product owner ou le client valide le travail.

• **Sprint retrospective :**

- est une réunion où l'équipe évalue les réussites et difficultés, pour améliorer les pratiques dans les prochaines sprints.

CamScanner

④ + Artefacts scrum :

• Product backlog :
- représente l'ensemble des choses à réaliser sur le produit pour le faire évoluer.

• Sprint backlog :
- représente l'ensemble des ~~items~~ tâches qui sont pris en charge par l'équipe de développement lors du sprint en cours.

• L'incrément :
- La ~~dernière~~ dernière version du produit. Il est utilisable et potentiellement publiable en production.

+ **phase pre-game :** (planification + architecture)
c'est la phase de la planification et la conception du projet. On crée une liste de tâches où en décomposant les grandes fonctionalité en petit user stories on décide après comment réaliser les tâches selon les besoins.

+ **Phase game :** (sprint + réunion scrum)
→ c'est la phase où le travail est réalisé.
• 1- développer le backlog de sprint (sprint backlog)
• 2- organiser des réunions chaque jour (Daily scrum)
• 3- participer à la réunion de revue de sprint (sprint review) pour présenter le travail
• 4- participer à la réunion rétrospective.

+ **phase final (Post-game)** → (démo + clôture).
c'est où le développement est terminer et le produit partiel livré.

+ **Présentation Jira :**
+ est une solution de gestion de projet édité par l'entreprise Atlassian.
Elle permet aux équipes de s'organiser efficacement, d'établir une communication durable et de visualiser le projet en un coup d'oeil grâce à ses tableaux de bord personnalisées

+ **Jira permet de :**
• travailler en méthode agile grâce aux tableaux kanban, scrum
• Accélérer la livraison des projets.
• Améliorer en continu des projets.
• Faciliter le travail des équipes

+ **Quelles équipes utilisent Jira :**
• les équipe métier (Finance, RH)
• les équipe support (DevOps)
• les équipe de développement logiciel.

CamScanner

- **Pour quoi utiliser jira :**
  - pour pouvoir l'intégrer avec un dépôt git (git hub)
  - pour avoir un suivi visuel des tâches (backlog, tableau de bord, Tableau agil)
  - pour gérer l'avancé du projet à l'aide d'un reporting
  - pour travailler à distance grâce à l'application mobile (jira software cloud)

- **Qu'est ce qu'un ticket Dans jira :**

Un ticket c'est une tâche à effectuer, c'est un élément de travail

- epic : est considéré comme un grand objectif ou une grosse fonctionnalité, divisé en plusieurs tâches.
- une story :

représente une fonctionnalité à réaliser.

- bug :

désigne un problème à corriger

- tâche :

est généralement une tâche technique à effectuer.

+ un projet jira :

- regroupe des tickets afin de réaliser un objectif

+ backlog :

- est une liste de fonctionnalités ou d'éléments de travail.

+ Rapports Agile :

- permettent une vue globale des projets, des tendances et de la performance de l'équipe

+ Tableau de bord :

est un outil puissant et personnalisé permettant aux utilisateurs de visualiser rapidement et simplement les données des projets et des tickets.

+ Type de rapport :

- Graphiques cumulatif
- graphiques d'avancements des sprints (sprint Burndown)
- rapports de Demandes créées récemment.

(5)

+ Work flow Sonarqube :

• SonarQube => analyse le code statiquement avec son scanner pour détecter bugs, vulnérabilité

Nb • les résultats sont sauvegardés dans le serveur Sonarqube et accessibles via une interface web.

+ les types de métriques avec SonarQube :

+ Densité des commentaires
+ Taux de couverture des tests unitaires
+ conventions de nommage
+ Détecte de bogues.

+ Git : ⚠

+ SonarQube classe les problèmes 3 types (de code)

• les bugs :
des erreurs dans le code.

• les vulnérabilités :
des failles dans le code affectent la sécurité

• le code smells :
=> code mal écrits

• la gestion de version :

est un système qui permet de suivre les modifications apportées à des fichiers ou à un projet au fil du temps.

+ Gestion de Version locales :

• le dépôt avec changement se trouve ~~dans~~ physiquement sur la même machines qui stocke les fichiers tracés.

+ Gestion de Version centralisées :

le dépot des changements se trouve dans un serveur centralisé, tandis que les différents ordinateurs individuels des personnes qui travaille sur le projet ne gardent que la dernière version des fichiers de travail.

+ Gestion de Version distribuée : (décentralisé)

combine la gestion de version local et centralisé.

+ Git :

est un logiciel de gestion de version. c'est un outil qui permet d'archiver et maintenir les différentes versions d'un ensemble de fichiers constituant le code source d'un projet. logiciel.

• les avantages :

+ Tous les dépots git contient l'intégralité des données, code source, ...
+ chaque participant de projet travail sur son dépot local
+ offre des mécanismes permettant de synchroniser les modifications entre les dépots.

+ GitLab : (wiki, CVS)

• une plateforme web qui utilise Git pour héberger des projets, tout en offrant des outils supplémentaires comme la gestion de projet, le suivi des bugs.

+ GitHub :

plateforme basé sur le Git. utilisé pour hébergement et collaboration sur des projet

CamScanner

+ fork :

est une copie d'un dépôt.

- pourquoi utilisé :

+ contribution open source
+ Expérimentation
+ collaboration décentralisés.

+ merge request :

permet de vérifier et collaborer sur des modifications de code avant leur fusion dans une branche.

+ conflit de fusion

un conflit de fusion arrive quand deux branches modifient la même partie d'un fichier, et git te demande de choisir.

GitHub Vs	Git Lab
→ Répository privé payant	→ répository privé Gratuit
→ pas d'hébergement sur serveur privé	→ hébergement sur serveur privé
→ aucune plateforme déploiement intégrée	→ Déploiement logiciel avec kubernete
→ interface graphique compliquée	→ interface claire et facile.

+ Noti des métriques...

+ Déf : une métrique : est une caractéristique d'une application.

+ métrique logicielle :

- mesure d'une propriété d'un logiciel (Exc : nbr de ligne de code)

+ Exemples de métriques :

- lignes de codes
- Nbr de méthode par classe
- Niveau d'abstraction
- indic instabilité.

+ Qualité de code :

d'un logiciel reflète sa clarté, sa fiabilité, sa facilité de modification et son adaptabilité aux évolutions.

CamScanner

6. Nb < > — enlever les <> du texte

+ **Branche :**

- une ligne de développement séparée qui vous permet de faire des modifications, d'ajouter des fonctionalités ou de corriger des bogues indépendamment.

+ **Commandes :**

1. initialisation de dossier :
   ⇒ git init
2. Créez une copie de votre dépot local :
   ⇒ git clone url.git
3. utilisez un serveur distant :
   ⇒ git clone username@host:/path/to/repository
4. propose un changement :
   ⇒ git add <fichier>
   ⇒ git add .
5. valider un changements :
   ⇒ git commit -m 'message'
6. lier le dépot local à un dépot distant :
   ⇒ git remote add origin <url de depot>
7. Envoyer à votre dépôt distant :
   ⇒ git push origin main
8. Afficher la liste des branches locales dans votre dépôt :
   ⇒ git branch
9. Afficher status :
   ⇒ git status
10. Configurer les info... :
    ⇒ git config --global user.name 'votre nom'
    ⇒ git config --global user.email 'votre email'
11. Créez branche :
    ⇒ git branch <nom branch>
12. Delete branche :
    ⇒ git branch -d <nom branch>

CamScanner

13. Crée et passe en même temps à une branch :

⇒ git checkout -b <nom_br>

14. Seulement passe au branch déjà existé :

⇒ git checkout <nom_branch>

15. fusionner une branch dans branch active :

⇒ git merge <nom_branch>

16. Récupère les dernières modification :

⇒ git pull origin main

17. Crée un dossier :

⇒ mkdir nom_dossier
⇒ cd nom_dossier
⇒ git init

} new dépot

18. Crée un fichier avec contenue ... :

echo > Readme.md
echo " texte ..... " > README.md Nb ⇒ le fichier peut être fichier.txt

19. l'historique de commit :

⇒ git log

20. Différence entre deux branche :

⇒ git diff <source-branch> <target-branch>

21. Annuler modification locaux :

⇒ git checkout -- <nom_fichier> (restor <nom_fichier>)

22. Réinitialiser votre branche principale au dernier commit du dépôt distant :

⇒ git fetch origin
⇒ git reset --hard origin/master

+ Introduction la chaîne DevOps

+ Déf. DevOps :

- est une approche qui facilite la collaboration entre développeurs et équipes IT en automatisant le déploiement des logiciels et la gestion des infrastructures.

+ Avantage :

- la collaboration
- la vitesse
- l'agilité
- la satisfaction du client
- l'innovation
- la sécurité.

+ Outils DevOps :

- outils de gestion de code source, les test CI/CD
- conteneurs
- cloud providers
- automatisation de gestion de configuration
- Monitoring et alerting
- outils de gestion de projet
- gestion de secrets.

CamScanner

(7). + Lien entre l'agilité et DevOps:

→ l'agilité permet une livraison rapide, tandis que DevOps automatise et optimise la collaboration, facilitant un déploiement continu.

+ L'intégration continue CI:

→ consiste à intégrer régulièrement les modifications de code dans un référentiel central, suivie d'une création et des tests automatiques.

Nb ⇒ en limitant le nombre de tâches manuelles

+ Le déploiement continue CD:

→ Le déploiement continu consiste à diffuser automatiquement les modifications de code en production.

+ Livraison continue Vs Déploiement continue:

→ La livraison continue assure que le logiciel est prêt à être publié avec une validation manuelle, tandis que CD automatise la publication en production après des tests réussis.

+ Les pipelines:

⇒ Sont des processes automatisés qui gèrent le développement, les testes et le déploiement du code.

+ Les runners:

⇒ Sont des applications ou services qui exécutent des jobs.

+ Les tâches:

⇒ Sont des instructions spécifiques exécutées séquentiellement ou en parallèle.

+ Les stages:

⇒ définissent quand les jobs doivent être exécutés.

+ Artefacts de pipeline:

⇒ Sont des fichiers crée par gitlab après la fin d'une pipeline.

+ Les tags:

⇒ utilisez ~~un~~ tags pour sélectionner un runner spécifique dans la liste de tous les runners disponibles pour le projet.

+ Pipeline CI/CD:

une série d'étapes qui doivent être effectuées pour fournir une nouvelle version.

Nb → elements pipeline: (Création, test, Lancement, Déploiement, Validation et conformité)

CamScanner

→ Quel est l'avantage du cycle en Y par rapport au cycle en V ?

• Le cycle en Y permet une meilleure intégration et validation continue, réduisant ainsi les risques d'erreur tardive.

→ Quel est l'avantage du cycle en V par rapport au modèle en cascade ?

• Le cycle en V permet une meilleure structuration des phases de test, avec une validation plus rigoureuse à chaque étape, contrairement au modèle en cascade qui ne prévoit les tests qu'en fin de projet.

⇒ Gantt

Est un outil de gestion de projet permet de représenter visuellement les tâches à réaliser sous forme des barres horizontales dans un calendrier pour voir les tâches à réaliser leur durée et l'avancement du projet