

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

La gestion de projet : est l ensemble des activités visant à organiser le bon déroulement d un projet et à en atteindre les objectifs en temps et en heures selon les objectifs visés .

Caractéristiques de base d un projet :

Objectifs / Résultat ou produit attendu : Les projets ont des buts et objectifs clairement définis

Durée/Espace : Les projets se cadencent obligatoirement dans le temps : ils possèdent une date de début et une date de fin.

Activités : Chaque projet est unique et nécessite la mise en place d une organisation et d un mode de pilotage spécifique.

Ressources : Il est indispensable de bien répartir les ressources, qu elles soient humaines, financières ou matérielles.

Les contraintes : De projet sont les limites générales d’un projet, notamment les délais, les coûts et les risques. Il est important d’identifier les contraintes d’un projet, car elles ont des répercussions sur les performances de ce dernier :

Délais : La gestion du temps est essentielle à la réussite du projet.

Coûts : Budget pour réaliser le projet .

Portée : La portée d un projet correspond à son ampleur en termes de qualité, de détails et de livrables.

Ressources de projet : **Les ressources humaines** : Il s agit tout simplement de l ensemble des personnes qui vont intervenir sur le projet.

Les ressources financières : Elles font référence au budget global du projet.

Les ressources matérielles : Elles concernent tout ce dont l’équipe projet va avoir besoin pour mener à bien le projet : les salles, logiciels.

La méthode en cascade : Elle permet de simplifier la gestion du projet au travers d un processus strict et séquencé : **Exigence ->Analyse->Conception->Mise en œuvre ->Validation->mise en service.**

Le cycle en Y : La famille des “ Unified Process ” constitue une trame commune pour intégrer les meilleures pratiques de développement.

Le cycle en V : est un modèle de gestion de projet qui implique toutes les étapes du cycle de vie d’un projet : **Exigences->Analyse-> Conception générale-> Conception détaillée-> mise en œuvre ->Test unitaire, Test d’intégration-> Test système ->Test d’acceptation.**

La méthode Scrum : est un Framework ou cadre de développement et de travail , de produits complexes (transparence) ,(l’inspection), (l’adaptation).

Les rôles d une équipe Scrum :

->**une équipe technique** (développeurs, architectes, designers, testeurs).

->**le product owner**, qui a la vision du produit et s’assure de la bonne traduction des attentes du client à l’équipe projet.

->**le scrum master**, ou maître de mêlée, qui est le chef d’orchestre, le coordinateur de l’équipe agile, dont il fait partie intégrante.

Glossaire: **product owner** (« directeur de produit ») : Personne ayant la responsabilité de produire et de maintenir à jour le carnet de produit.

scrum master (« chef de mêlée») :Membre de l’équipe dont l’objectif principal est de la protéger des perturbations extérieures

product backlog (« carnet du produit ») : Liste des fonctionnalités, des fonctions, des exigences, des améliorations et des correctifs qui sont nécessaires à l’évolution du produit,

daily scrum (« mêlée quotidienne ») :Réunion quotidienne de quinze minutes maximum pour faire le point sur ce qui a été fait depuis la dernière mêlée .

méthode Kanban : repose sur un système à flux tirés, qui démarre dès que le client passe une commande.

Jira (**Jira Software, Jira Service Management, Jira Work Management**) Jira Software est une solution de gestion de projet éditée par l’entreprise Atlassian

Les Tickets de Jira :

->**Une epic** (épopée) :est considérée comme un grand objectif.

->**Une story** : représente une fonctionnalité à réaliser, o Une tâche : est généralement une tâche technique à effectuer

-> **Un bug** : désigne un problème à corriger.

Un Backlog est une liste de fonctionnalités ou d’éléments de travail.

Git : est un logiciel libre de gestion de versions.

Gitlab : est une plateforme open source et collaborative de développement basé sur Git. gitlab permet d'héberger des projets web, du code, et de la documentation .

SonarQube : est un logiciel open source de mesure de la qualité du code source de projets de développement.

Exemples de analyser le code : FindBugs, CodeSonar, Veracode.

A : 0 bug 0 vulnérabilité.

B : Au moins 1 bug mineur Au moins 1 vulnérabilité mineure.

C : Au moins 1 bug majeur Au moins 1 vulnérabilité majeure.

D : Au moins 1 bug critique Au moins 1 vulnérabilité mine critique.

DevOps : est un ensemble de pratiques qui met l’accent sur la collaboration et la communication entre les développeurs de logiciels et les professionnels des opérations informatiques.

les outils de CI/CD : permettent la modernisation des applications en réduisant le temps nécessaire pour créer de nouvelles fonctions

Principes de Gitlab CI/CD :

Les pipelines : sont le composant de niveau supérieur de l’intégration.

Le Runner : • Est une application qui fonctionne avec GitLab CI / CD pour exécuter des tâches (Jobs) dans un pipeline.

Les tâches : est un ensemble d’instructions qu’un runner doit exécuter.

Les étapes : déterminent quand exécuter les tâches.

Les artefacts : sont des fichiers créés par GitLab après la fin d'un pipeline.

Les tags : Utilisez tags pour sélectionner un Runner spécifique.

Un pipeline CI/CD : est une série d'étapes à réaliser en vue de distribuer une nouvelle version d'un logiciel.

CI/CD : CONTINUOS Integration/ CONTINUOUS Deployment

\$ git status
Liste tous les nouveaux fichiers et les fichiers modifiés à commiter

\$ git add [fichier]
Ajoute un instantané du fichier, en préparation pour le suivi de version

\$ git reset [fichier]
Enleve le fichier de l'index, mais conserve son contenu

\$ git diff
Montre les modifications de fichier qui ne sont pas encore indexées

\$ git diff --staged
Montre les différences de fichier entre la version indexée et la dernière version

\$ git commit -m "[message descriptif]"
Enregistre des instantanés de fichiers de façon permanente dans l'historique des versions

\$ git log
Montre l'historique des versions pour la branche courante

\$ git log --follow [fichier]
Montre l'historique des versions, y compris les actions de renommage, pour le fichier spécifié

\$ git diff [premiere-branche]...[deuxieme-branche]
Montre les différences de contenu entre deux branches

\$ git show [commit]
Montre les modifications de métadonnées et de contenu incluses dans le commit spécifié

\$ git fetch [nom-de-depot]
Récupère tout l'historique du dépôt nommé

\$ git merge [nom-de-depot]/[branche]
Fusionne la branche du dépôt dans la branche locale courante

\$ git push [alias] [branche]
Envoie tous les commits de la branche locale vers GitHub

\$ git pull
Récupère tout l'historique du dépôt nommé et incorpore les modifications

\$ git config --global user.name "[nom]"
Définit le nom que vous voulez associer à toutes vos opérations de commit

\$ git config --global user.email "[adresse email]"
Définit l'email que vous voulez associer à toutes vos opérations de commit

\$ git config --global color.ui auto
Active la colorisation de la sortie en ligne de commande.

git branch
Liste toutes les branches locales dans le dépôt courant

\$ git branch [nom-de-brancher]
Crée une nouvelle branche

\$ git checkout [nom-de-brancher]
Bascule sur la branche spécifiée et met à jour le répertoire de travail

\$ git merge [nom-de-brancher]
Combine dans la branche courante l'historique de la branche spécifiée

\$ git branch -d [nom-de-brancher]
Supprime la branche spécifiée

\$ git init [nom-du-projet]
Crée un dépôt local à partir du nom spécifié

\$ git clone [url]
Télécharge un projet et tout son historique de versions

\$ git rm [fichier]
Supprime le fichier du répertoire de travail et met à jour l'index

\$ git mv [fichier-nom] [fichier-nouveau-nom]
Renomme le fichier et prépare le changement pour un commit

\$ git stash
Enregistre de manière temporaire tous les fichiers sous suivi de version qui ont été modifiés ("remiser son travail")

\$ git stash list
Liste toutes les remises

\$ git stash pop
Applique une remise et la supprime immédiatement

\$ git stash drop
Supprime la remise la plus récente

\$ git rm [fichier]
Supprime le fichier du répertoire de travail et met à jour l'index

\$ git mv [fichier-nom] [fichier-nouveau-nom]
Renomme le fichier et prépare le changement pour un commit

\$ git reset [commit]
Annule tous les commits après `[commit]`, en conservant les modifications localement

\$ git reset --hard [commit]
Supprime tout l'historique et les modifications effectuées après le commit spécifié