

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**, la conduite de projet, l'ingénierie de projet, ou encore le management 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.

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, les équipements informatiques, les logiciels...

Livrables de projet :

La conduite d'un projet débouche sur un produit, un service, une nouvelle organisation, etc. Cette finalité, appelée "livrable", est le résultat tangible d'une production réelle, appréhendable, mesurable attendue par le client final. Un projet peut, bien sûr, avoir plusieurs livrables.

- Principaux rôles dans un projet informatique :

La maîtrise d'ouvrage (MOA) : il s'agit du « client » du projet, soit celui qui en attend des résultats concrets. • La maîtrise d'œuvre (MOE) : il s'agit du « fournisseur » du projet, soit celui qui réalise l'ouvrage même. • Le chef de projet, appelé aussi Project Manager, est responsable de l'équipe projet en charge de la préparation, de la réalisation et de la finalisation du projet.

Un cahier des charges : (parfois abrégé en CDC) est un document qui doit être respecté lors de la conception d'un projet. C'est un document qui permet d'expliquer toutes les spécificités, les attentes et les contraintes d'un projet aux freelances ou aux agences.

Le product owner sera ainsi responsable de :

- fournir des items clairs, complets et compris de tous
- d'organiser et de prioriser le contenu de son backlog
- d'optimiser la valeur que pourra délivrer l'équipe de développement
- rendre visible le backlog, la vision du produit et le travail imaginé pour les prochaines itérations.

La Sprint Planning : est la cérémonie que l'équipe scrum réalise en ouverture de Sprint pour cadrer le sprint qui démarre.

Les artefacts Scrum : sont des éléments permettant de faire fonctionner le cadre de travail Scrum, en accord avec les principes et les pratiques s'inscrivant dans l'ADN de l'agilité. • le sprint backlog, • le product backlog, • l'incrément produit.

Le product backlog : représente le regroupement de l'ensemble des choses à réaliser sur le produit pour le faire évoluer ; il est de la responsabilité du product owner.

Le sprint backlog : représente l'ensemble des items qui sont pris en charge par l'équipe de développement lors du sprint en cours ; il contiendra également le plan pour livrer ces items dans un environnement stable.

Jira : est un système de suivi de bugs, de gestion des incidents et de gestion de projets développé par Atlassian et publié pour la première fois en 2002. Il propose des solutions à la fois à destination des développeurs et des intervenants non développeurs.

Un backlog produit : est une liste hiérarchisée de fonctionnalités ou d'éléments de travail ou tâches destinées à l'équipe de développement.

gestion de version : est un outil (logiciel) permettant d'enregistrer, de suivre et de gérer plusieurs versions d'un fichier ou d'un code source.

- Un système de suivi des bugs est un logiciel qui permet d'effectuer un suivi des bugs signalés dans le cadre d'un projet de développement de logiciel. •

L'intégration continue (CI) est un ensemble de pratiques utilisées en génie logiciel consistant à vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régression dans l'application développée. • La livraison continue (CD) est une approche d'ingénierie logicielle dans laquelle les équipes produisent des logiciels dans des cycles courts, ce qui permet de le mettre à disposition à n'importe quel moment. Le but est de construire, tester et diffuser un logiciel plus rapidement. • Un wiki est une application web qui permet la création, la modification et l'illustration collaboratives de pages à l'intérieur d'un site web.

Git : est un outil de gestion de version ou VCS en anglais (version control system) qui permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectuées dessus.

GitLab est un logiciel libre de forge basé sur git proposant les fonctionnalités de wiki, un système de suivi des bugs, l'intégration continue et la livraison continue.

Rôles dans un projet informatique ; a. Chef de projet ; b. La matrice RACI :

1. Responsable : Qui réalise la tâche ; 2. Accountable : Qui approuve la tâche ; 3. Consulted : Celui qui est consulté ; 4. Informed : qui doit être informé.

Acteurs externes : 1. Les clients : les premiers concernés en externe, impactés directement si leur rôle s'inscrit dans l'utilisation du produit ; 2. Les fournisseurs ; 3. Les utilisateurs ; 4. Les organismes privés ; 5. Les investisseurs. **Acteurs internes :** 1. La direction ; 2. L'équipe ; 3. Les utilisateurs (Services impactés) ; 4. Actionnaires ; 5. Le sponsor ; 6. Experts ; 7. Le commanditaire : Client interne ; 8. Syndicats.

La Matrice des Responsabilités : (RACI) La matrice des responsabilités est un outil qui vous permet de visualiser les 4 niveaux de responsabilité

- (R) Responsable : il est garant de la réalisation du lot de travaux. Il peut déléguer les tâches.
- (A) Acteur : il réalise le lot de travaux.
- (C) Consulté : c'est un expert, consulté pour la réalisation d'une tâche ou une validation technique.
- (I) Informé : c'est un coéquipier à qui l'on diffuse des documents, que l'on tient au courant de l'avancement, convie aux réunions, parce que le lot de travaux a un impact sur sa mission.

Estimation de la durée de réalisation de chaque tâche

Calcul de la durée (Méthode 1) : La durée moyenne de la tâche est alors estimée à (Optimiste, + 4 x Plus Probable + Pessimiste) / 6

Calcul de la durée (Méthode 2) : Durée = (Travail / Capacité) + temps non travaillé

L'ordonnancement se déroule en trois étapes :

- La planification : qui vise à déterminer les différentes opérations à réaliser, les dates correspondantes, et les moyens matériels et humains à y affecter.
- L'exécution : qui consiste à la mise en œuvre des différentes opérations définies dans la phase de planification.
- Le contrôle : qui consiste à effectuer une comparaison entre planification et exécution, soit au niveau des coûts, soit au niveau des dates de réalisation.

Scrum est un framework, cadre de travail de type organisationnelle le plus répandu à ce jour créée par Ken Schwaber. Elle se base sur 3 piliers essentiels : la transparence, l'inspection et l'adaptation. Elle se mélange très bien à d'autres méthodes agiles ou apparentées agiles comme kanban, extreme programming, lean startup, devops...

La transparence : Le Scrum impose de partager tous les aspects liés au processus à l'ensemble des décideurs et la vision globale des développements à l'ensemble des observateurs.

L'inspection : Il est essentiel de bien suivre l'avancement des développements et des objectifs de l'équipe. **Adaptation :** Le Scrum préconise d'adapter les processus et l'environnement de travail afin de proposer un contexte optimal à l'équipe. La cérémonie de la rétrospective en fin de sprint permet de définir des axes d'amélioration en équipe afin d'être dans une réelle démarche d'amélioration continue.

L'équipe de développement : se compose de professionnels qui fournissent un incrément « Fini » potentiellement publiable (Releasable) à la fin de chaque Sprint.

Le Scrum Master : est chargé de promouvoir et supporter Scrum tel que défini dans le Guide Scrum. Les Scrum Masters remplissent leur rôle en aidant tout le monde à comprendre la théorie, les pratiques, les règles et les valeurs de Scrum. • l'organisation du product backlog • rendre transparent ce product backlog • la clarté des items de celui-ci pour l'équipe de développement • l'aider à ordonner les items pour maximiser la valeur livrée .

Les étapes pour effectuer un ajout d'un fichier "fichier1" à la branche master sont les suivantes :

Cloner le repository sur votre ordinateur local

Ajouter le fichier "fichier1" au répertoire local

Utiliser la commande "git add" pour ajouter le fichier "fichier1" au repository

Utiliser la commande "git commit" pour enregistrer les changements

Utiliser la commande "git push" pour envoyer les changements vers la branche master

Les avantages d'utiliser GIT sont:

La possibilité de suivre et de gérer les versions des fichiers de code

La possibilité de travailler en collaboration sur un projet en utilisant des branches et des merge

La possibilité de retrouver facilement les anciennes versions des fichiers de code en cas de besoin

La possibilité de gérer les conflits de manière efficace.

Le rôle d'un bilan de projet est de fournir une évaluation complète et objective de la performance d'un projet, en comparant les résultats obtenus avec les objectifs initiaux fixés. Il permet également de déterminer les réussites et les difficultés rencontrées pendant la réalisation du projet, et d'identifier les leçons à tirer pour les projets futurs.

Caractéristiques de base d'un projet :

1. Objectifs ; 2. Limite dans le temps ; 3. Activité qui transforme les ressources (main d'œuvre, connaissances, l'équipement, les matières premières, le temps) en résultats attendus dans un délai de temps spécifié ; 4. Une série d'activités forme un processus ; 5. Les ressources (les intrants) sont les éléments qui se transforment en résultats (tangibles). 6. Les résultats attendus se créent à la suite des activités du projet

Les types de risques : 1. Financiers : coût supérieur à l'estimation, manque de budget, etc. 2. Humains : manque de compétences, absentéisme, démission au cours du projet, conflits au sein de l'équipe, etc. 3. Temporels : retards des sous-traitants ou des fournisseurs, mauvaise estimation des délais, etc. 4. Techniques : logiciel inadapté, pannes, matériel obsolète, etc. 5. Juridiques : réglementations et lois à respecter, faillite d'un fournisseur, etc. 6. Environnementaux : impacts négatifs du projet sur l'environnement, ou environnement ayant un impact sur le projet (inondation, sécheresse, tempête...). 7. Organisationnels : changement dans la politique de l'entreprise, changements économiques, etc.

Une **tâche** désigne une seule et même unité de travail : Action ou Etape. Elle doit être accomplie dans un certain délai et doit contribuer à la réalisation des objectifs professionnels

Le diagramme de Gantt, couramment utilisé en gestion de projet, est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet. La colonne de gauche du diagramme énumère toutes les tâches à effectuer, tandis que la ligne d'en-tête représente les unités de temps les plus adaptées au projet (jours, semaines, mois etc.). Chaque tâche est matérialisée par une barre horizontale, dont la position et la longueur représentent la date de début, la durée et la date de fin.

Le product owner est la personne responsable de la vision et de la stratégie de produit dans un environnement de développement de produit agile. Il est responsable de définir les fonctionnalités du produit, de prioriser les exigences et de communiquer avec les parties prenantes pour s'assurer que le produit répond aux besoins des utilisateurs et des clients. Il travaille étroitement avec l'équipe de développement pour s'assurer que le produit est livré de manière efficace et en temps opportun

Les 3 principaux artefacts de la méthode Scrum sont :

Product Backlog: Il s'agit d'une liste de toutes les fonctionnalités, tâches et bugs à effectuer pour le projet. Il est géré par le Product Owner et est révisé et trié régulièrement pour s'assurer que les éléments les plus importants sont priorisés.

Sprint Backlog: Il s'agit d'une liste des tâches qui seront effectuées pendant le prochain sprint. Il est généralement créé à partir de l'Product Backlog lors de la réunion de planification de sprint et est géré par l'équipe de développement.

Increment: Il s'agit d'un produit ou d'une partie du produit qui a été développé pendant le sprint. Il est généralement démontré à l'équipe de développement et au Product Owner lors de la réunion de revue de sprint. Il doit être complètement fonctionnel et utilisable

Les histoires utilisateur (ou "**user stories**") sont un outil couramment utilisé dans la méthodologie Agile pour décrire les fonctionnalités d'un produit ou les tâches à effectuer. Les histoires utilisateur sont écrites par les utilisateurs finaux ou le Product Owner (responsable du produit) et décrivent les besoins et les souhaits des utilisateurs.

Les épics sont un type spécifique d'histoires utilisateur utilisées pour décrire les grandes fonctionnalités d'un produit. Les épics sont généralement plus vastes et plus complexes que les histoires utilisateur standard et nécessitent généralement plusieurs sprints pour être complétés. Les épics peuvent être décomposés en plusieurs histoires utilisateur plus petites qui peuvent être complétées dans un sprint.

Les épics sont généralement écrits par le Product Owner, en collaboration avec les utilisateurs finaux et l'équipe de développement, pour décrire les fonctionnalités clés et les besoins des utilisateurs. Les épics sont utilisés pour établir les objectifs à long terme pour le produit et pour planifier les sprints à venir.

Les 4 valeurs de Scrum sont :

Emphasise sur la collaboration: Tous les membres de l'équipe travaillent ensemble pour atteindre les objectifs communs.

Emphasise sur l'adaptation: Scrum est une méthodologie itérative et incrémentale, ce qui signifie que l'équipe s'adapte aux changements et aux incertitudes tout au long du projet.

Emphasise sur la réalisation de tâches: L'équipe se concentre sur l'accomplissement de tâches concrètes pour atteindre les objectifs du projet.

Emphasise sur la transparence: Toutes les informations concernant le projet sont partagées avec tous les membres de l'équipe.

Proposition des solutions possibles. 1. Elaborer le diagramme de GANTT : Planification, Marges, Retards... 2. Dès le lancement, définissez clairement et précisément les enjeux et les objectifs du projet. 3. Il est essentiel de préparer et d'évaluer votre planning avec précision. 4. Tenir compte de la disponibilité des membres de votre équipe. 5. Utiliser les feuilles de temps pour connaître les disponibilités des membres de votre équipe, et ainsi mieux répartir la charge de travail. On doit connaître la quantité de temps déjà passée sur une tâche, ainsi que le temps restant avant son échéance. 6. Entretenir un dialogue constant et constructif entre l'équipe et le chef de projet, mais aussi avec les autres acteurs impliqués. Cela améliore le travail en équipe et permet à chacun d'être informé de l'évolution du projet en temps réel pour avoir une équipe productive.

Marge libre : C'est le retard que l'on peut prendre dans la réalisation d'une tâche sans retarder la date de début de toute autre tâche qui suit. **Marge totale** : indique le retard maximum que pourrait prendre la tâche sans retarder la fin du projet.

Types de tâches : 1. Tâche critique : C'est la tâche qui ne supporte pas le retard et qui a une marge nulle. 2. Tâche récapitulative : permet le regroupement d'un ensemble des tâches du projet afin de connaître : Le coût du projet, Le nombre d'heures travaillées ou toute autre valeur cumulée au niveau du projet. 3. Tâche jalon : Ce sont des points de référence qui indiquent des événements importants, sa durée est nulle. 4. Tâche périodique : Une tâche qui se produit à intervalles réguliers au cours d'un même projet.

Le Daily Scrum est une réunion quotidienne qui se tient dans le cadre de la méthodologie Agile Scrum. Cette réunion est également appelée "stand-up meeting", car les participants se tiennent debout pour la tenir (afin de la maintenir courte). Les membres de l'équipe de développement se réunissent chaque jour pour discuter de l'avancement du projet et planifier les tâches à effectuer pour le reste de la journée.

Les participants au Daily Scrum répondent chacun à trois questions:

Qu'ai-je fait hier pour aider l'équipe à atteindre son objectif ?
Que vais-je faire aujourd'hui pour aider l'équipe à atteindre son objectif ?
Y a-t-il des obstacles ou des blocages pour moi ou pour l'équipe ?
Le but de cette réunion est de s'assurer que tous les membres de l'équipe sont sur la même longueur d'onde et travaillent ensemble efficacement pour atteindre les objectifs de sprint.

Voici un aperçu général de l'enchaînement de la méthodologie Scrum A à Z :

Product Backlog: Le Product Owner crée une liste des fonctionnalités, tâches et bugs pour le projet.

Sprint Planning Meeting: L'équipe de développement se réunit pour planifier le prochain sprint et déterminer les tâches à effectuer.

Daily Scrum: L'équipe de développement se réunit chaque jour pour discuter de l'avancement du projet et planifier les tâches à effectuer pour le reste de la journée.

Travail sur les tâches: Les membres de l'équipe de développement travaillent sur les tâches définies lors de la réunion de planification de sprint.

Sprint Review Meeting: L'équipe de développement démontre les fonctionnalités développées lors du sprint au Product Owner et à d'autres parties prenantes.

Sprint Retrospective Meeting: L'équipe de développement se réunit pour discuter de ce qui s'est bien passé et de ce qui peut être amélioré pour les sprints futurs.

Fin du sprint: Le sprint actuel est terminé et un nouveau sprint est planifié.

Livraison du produit: Le produit est mis en production ou livré au client une fois que toutes les fonctionnalités ont été développées et approuvées

Les 12 principes de Scrum sont :

- Les clients ou les utilisateurs finaux peuvent changer les exigences à tout moment.
- Le plus tôt une fonctionnalité est terminée, plus tôt les clients ou les utilisateurs finaux peuvent l'utiliser.
- Le projet est décomposé en sprints courts pour maximiser la flexibilité.
- L'équipe est auto-gérée et auto-organisée.
- Les développeurs et les utilisateurs finaux travaillent ensemble quotidiennement.
- Le processus de développement est transparent pour tous les membres de l'équipe.
- Les membres de l'équipe sont choisis pour leur compétence et leur engagement.
- Le développement est effectué par une équipe multidisciplinaire.
- Le développement est effectué par des processus de développement itératifs et incrémentaux.
- La qualité est gérée par l'équipe.
- La capacité de travailler sur le projet est mesurée par la capacité de l'équipe à terminer les tâches.
- Les fonctionnalités sont priorisées pour maximiser le retour sur investissement.

\$ 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]

Enlève 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-branch]

Crée une nouvelle branche

\$ git checkout [nom-de-branch]

Bascule sur la branche spécifiée et met à jour le répertoire de travail

\$ git merge [nom-de-branch]

Combine dans la branche courante l'historique de la branche spécifiée

\$ git branch -d [nom-de-branch]

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

