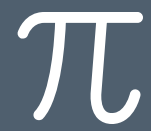
Projet dirigé

SCRUM-1

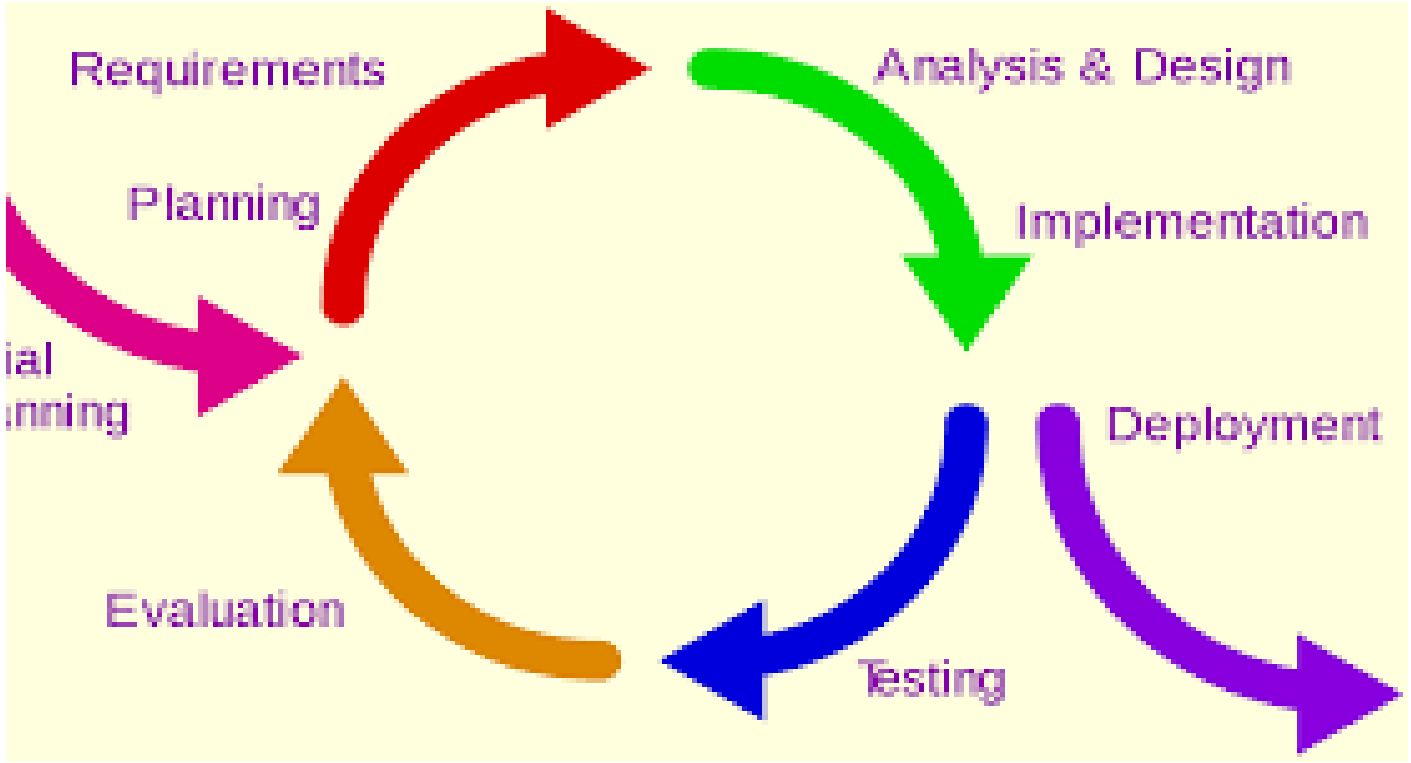


SCRUM,

Plan de la séance

- Retour sur les user-stories
- Rappels
  - Approches agiles
  - Valeurs agiles
  - Manifeste agile.
- L'approche SCRUM
- Cycle de développement selon SCRUM
- L'équipe SCRUM
- Le backlog du produit.

Rappels, l’approche agile



Rappels

Avantages:

- Itératif et incrémental pour vrai pas uniquement dans le principe.
- Livraison tôt et souvent
- Implication du client: les besoins évoluent, feedback tôt.
- Les itérations (sprint) sont courtes.
- La documentation n’est pas lourde.
- Le rythme de travail est constant

Rappels

Manifeste agile: 4 valeurs et 12 principes

Valeurs agiles

1. Les personnes et leurs interactions sont plus importantes que le processus et les outils
2. Un logiciel qui fonctionne prime sur la documentation
3. La collaboration avec les clients est préférable à la négociation contractuelle
4. La réponse au changement passe avant le suivi d'un plan.

Rappels

Les principes : Le Manifeste annonce douze (12) principes, qui ne définissent pas une méthode agile.

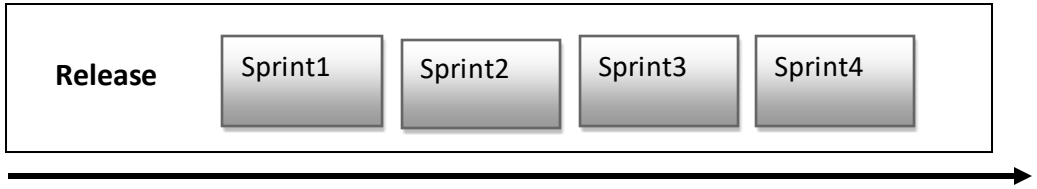
1. Satisfaire le client en livrant tôt et régulièrement des logiciels utiles qui offre une véritable valeur ajoutée.
2. Accepter les changements même tard dans le développement.
3. Livrer fréquemment une application qui fonctionne.
4. Collaborer quotidiennement entre clients et développeurs.
5. Bâtir le projet autour de personnes motivées en leur fournissant environnement et support, et en leur faisant confiance.
6. Communiquer par des conversations face à face.
7. Mesurer la progression avec le logiciel qui fonctionne.
8. Garder un rythme de travail durable.
9. Rechercher l'excellence technique et la qualité de conception
10. Laisser l'équipe s'auto-organiser.
11. Rechercher la simplicité.
12. À intervalle régulier, réfléchir aux moyens de devenir plus efficace.

SCRUM, l'approche

- SCRUM signifie mêlée en rugby. Scrum utilise les valeurs et l'esprit du rugby et les adapte aux projets de développement
- SCRUM est l'approche agile la plus populaire
- Définitions:
  - Un **sprint** est le terme utilisé dans SCRUM pour désigner une itération. Dans le langage SCRUM un sprint est un bloc de temps fixée aboutissant à un incrément de produit potentiellement livrable.
  - Une **version** est produite par une série d'itérations d'un mois, parfois même de 15 jours, appelés **sprint**. Le contenu d'un *sprint* est défini par l'équipe avec le Product Owner, en tenant compte des priorités et de la capacité de l'équipe.
  - **Une release** : (selon le dictionnaire du jargon informatique) : Version d'un système, par exemple un logiciel, effectivement publiée, donc lâchée dans la nature.
  - **Backlog** du produit: liste unique des user-stories **classées par priorité.**

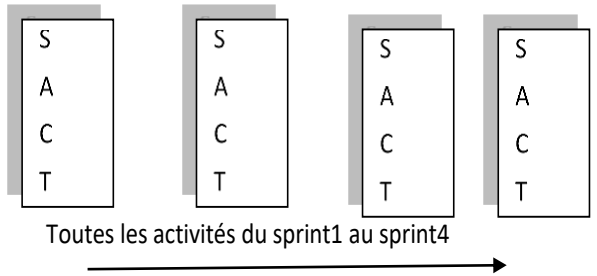
Un sprint trop court risque de stresser l'équipe. Un sprint trop long risque de démotiver l'équipe

SCRUM, l'approche



- Spécification fonctionnelle (Requierements ou étude des besoins fonctionnels)
- Architecture (conception)
- Codage (et test unitaire)
- Test (test d'intégration, test d'acceptation).

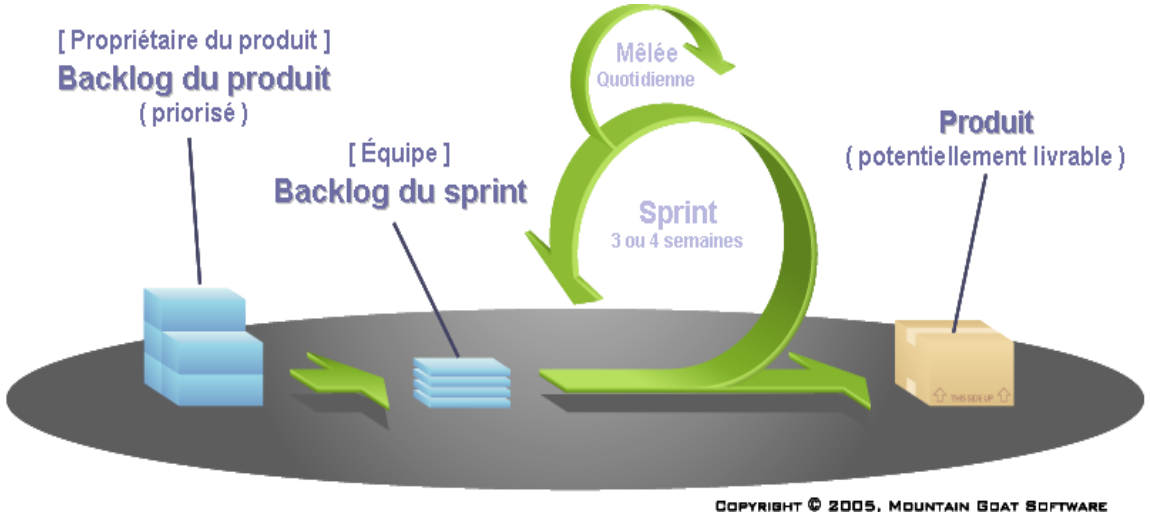
Cycle SCRUM : les sprints et leurs phases se déroulent en parallèle.



SCRUM, l'approche

- Il n'y a pas de chevauchement. Les *sprints* s'enchaînent sans délais. Un *sprint* n'est commencé que si le précédent est terminé. Le nouveau *sprint* démarre immédiatement après le précédent
- À la fin de chaque *sprint,* l'équipe obtient un **produit partiel**, (qui s'enrichit d'un nouveau incrément à chaque *sprint*) **qui fonctionne**. Il est potentiellement livrable. L' évaluation et le *feedback* récolté permettent d'ajuster le *backlog* pour le *sprint* suivant.
- La date de fin d'un *sprint* est fixée au début de celui-ci. Elle ne change pas même si l'équipe ne réalise pas tout ce qu'elle pensait faire
- Arrêter un sprint plutôt que de l'étendre.

SCRUM, l'approche



Source de l'image : https://fr.wikipedia.org/wiki/Scrum_(d%C3%A9veloppement)

SCRUM, l'approche

- Pour une équipe, une release dure environs 3 mois avec des **sprints de deux à trois semaines**. Ce qui permet d'avoir de quatre à 6 sprints dans une release.
- Il n'y a pas de chevauchement entre les sprints. Ils s'enchaînent sans délais.
- La fin d'un sprint peu être un produit potentiellement livrable.
- Le résultat d'une *release* est le produit livrable fourni à ses utilisateurs. La façon dont il est fourni dépend de son déploiement.
- Souvent, le jalon majeur que représente la *release* correspond à une annonce marketing.

SCRUM, l'approche

SCRUM…. Développement léger

- **Création du *backlog*** de produit (un To-Do list) de toutes les fonctionnalités d'un projet. Les éléments du backlog doivent être estimés (temps de réalisation) et priorisés.
- **Création d'un sprint backlog** : fonctionnalités à compléter durant la durée du sprint (15 jours ou un mois)
- **Effectuer des rencontres quotidiennes durant le sprint** : des mêlées quotidiennes (Daily *scrum*).
- Finalisation du sprint avec **démonstration et évaluation**

SCRUM, rôles et responsabilités

Rôles et responsabilités:

Le Product Owner : (PO) : Le PO est l'expert du domaine (niveau métier). En tant que représentant des clients et utilisateurs, il est responsable de définir les caractéristiques du produit développé par l'équipe, en termes de :

- Fonctionnalités offertes. Plus précisément, il identifie chaque exigence que doit satisfaire le produit et la collecte comme élément du backlog de produit. Il est souhaitable d'inclure les tests d'acceptation.
- Priorité. C'est lui qui définit l'ordre dans lequel ces éléments seront développés en fonction de la valeur qu'ils apportent aux clients et utilisateurs. Cela permet d'alimenter l'équipe avec un backlog de produit prêt pour la planification des sprints
- But. C'est lui qui définit l'objectif d'une release et qui prend les décisions concernant le planning de la release.
- Son implication dans le projet est capitale pour la réussite de celui-ci.

SCRUM, rôles et responsabilités

Rôles et responsabilités:

Le SCRUM Master :C'est le coach de l'équipe (ancien chef de projet). Il a pour rôle:

- Dans le cadre du développement d'un produit, d'aider l'équipe à travailler de façon autonome et à s'améliorer constamment. Il est le garant de l'application du processus, Scrum en l'occurrence.
- S'assurer que l'équipe bénéficie des meilleures conditions pour accomplir les tâches
- Éliminer les obstacles : prendre en compte les problèmes qui surviennent à tout moment sur un projet pour les éliminer au plus vite, en évitant qu'ils ralentissent l'équipe. Il protège l'équipe des interférences extérieures.
- Faire en sorte que l'équipe reste concentrée sur le véritable objectif du projet, qui est de réaliser les éléments du Backlog en collaboration étroite avec le Product Owner , et soit productive. Il s'assure que chacun participe pleinement aux travaux de l'équipe.
- Organise et anime les réunions qui constituent le cérémonial.

SCRUM, le backlog

Le backlog … un peu plus:

- Le cœur de SCRUM est le *Product Backlog ou* ***backlog du produit*** qui représente la liste des requis **priorisés. C'est un tableau de user-stories**
- SCRUM débute avec un produit backlog priorisé**.**
- Chaque entrée du backlog représente une user-story décrite dans le langage et la terminologie du client. Chaque entrée sert à finaliser ce que le client désire obtenir.
- Le Backlog doit être un document partagé, détenu par le PO.
- Le backlog du produit est vivant.
- Garder le **produi**t **backlog** niveau métier. Il doit focaliser sur les buts métier et non les technologies.

SCRUM, le backlog

- **Ce qu'il faut faire :**
  - Cultiver le backlog : le backlog évolue dans le temps, il faudra le mettre à jour.
  - Partager le backlog avec toute l'équipe
  - Surveiller la taille du backlog : ne pas avoir plus de 150 éléments à faire dans le backlog

- **À Éviter :**
  - D'avoir plusieurs backlog pour le même de produit
  - De ne pas avoir de backlog
  - De confondre le backlog de produit avec le backlog de sprint

SCRUM, prioriser les éléments du backlog

- **Prioriser les éléments du backlog de produit → Questions à se poser:**
  - Quel est le bénéfice financier ou la valeur ajoutée à développer cette fonctionnalité ?
  - Quel est le coût de développement ? De maintenance …
  - L'implémentation de la fonctionnalité nous permet-elle d'apprendre ?de développer de nouvelles compétences ?
  - La fonctionnalité va-t-elle améliorer la productivité de mon personnel ?
  - Quel est le préjudice si cette fonctionnalité n'est pas implémentée ?

- **Tenir compte de:**
  - Le bénéfice financier attendu ou valeur ajoutée, c'est l'élément le plus important à considérer.
  - Le coût de développement
  - L'opportunité d'apprentissage pour l'équipe
  - Le risque de développement, le développement de cette fonctionnalité nous expose t-elle à d'avantage de risques.

SCRUM, priority Pocker

Prioriser les éléments du backlog de produit: priority Poker

- Chaque participant reçoit un lot de neuf cartes numérotées de 1 à 9
- Chaque story est étudiée successivement
- Le premier vote porte sur l'intérêt d'avoir le feature. Chaque participant vote avec une carte. On fait le total des points.
- Le deuxième vote, porte sur la pénalité de ne pas avoir le feature dans le produit. On vote également de 1 à 9.
- On définit l'importance des deux votes. On peut donner 4 au premier et 1 au deuxième.
- En faisant la somme des deux votes pondérés on obtient l'utilité de l'élément. Les stories les plus utiles sont les plus prioritaires.

SCRUM, MoScoW

Prioriser les éléments du backlog de produit: MoSCoW

La technique utilisée pour prioriser les besoins dans un contexte itératif est celle de MoSCoW. L'avantage de la méthode *MoSCoW* réside dans la signification de l'acronyme, qui est plus compréhensible que d'autres techniques de priorisation comme élevé/moyen/faible

- **M** pour **Must Have** : ***DOIT*** être fait. L'exigence est essentielle. Si elle n'est pas faîte le projet échoue. On peut dire également priorité haute.
- **S** pour Should **Have** : Il s'agit d'une exigence essentielle, qu'il faut faire dans la mesure du possible (***DEVRAIT***). Mais si elle n'est pas faîte, on peut la contourner et la livrer plus tard.
- **C** pour **Could Have**: Il s'agit d'une exigence souhaitable. Elle ***POURRAIT être*** faîte dans la mesure où elle n'a pas d'impact sur les autres tâches
- **W** pour **Won't Have** Il s'agit d'une exigence «Luxe». ***NE SERA PAS*** faîte cette fois mais plus tard, mais intéressante et à garder pour la prochaine version

SCRUM

CONCLUSION

QUESTIONS ??