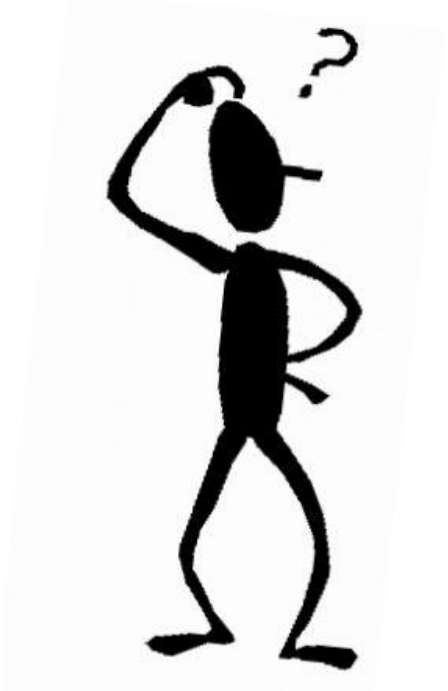
Génie Logiciel

Awa DIATTARA
awa.diattara@ugb.edu.sn



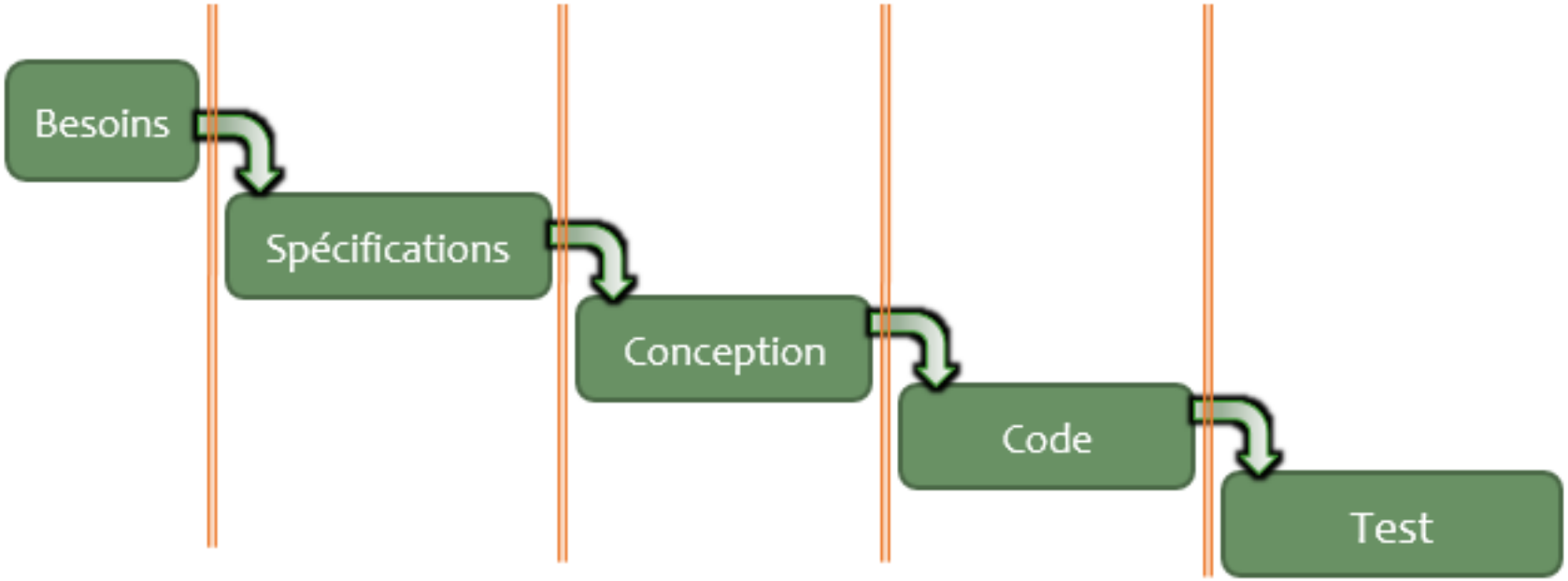
Qu’est ce que l’agilité?

Qu'est ce que l'agilité ?

- Approche réactive et itérative d'organisation de travail.
- Focalisée sur la fonctionnalité et la satisfaction client.
- Construite en adéquation avec les capacités et limites humaines.

Pourquoi agile ?

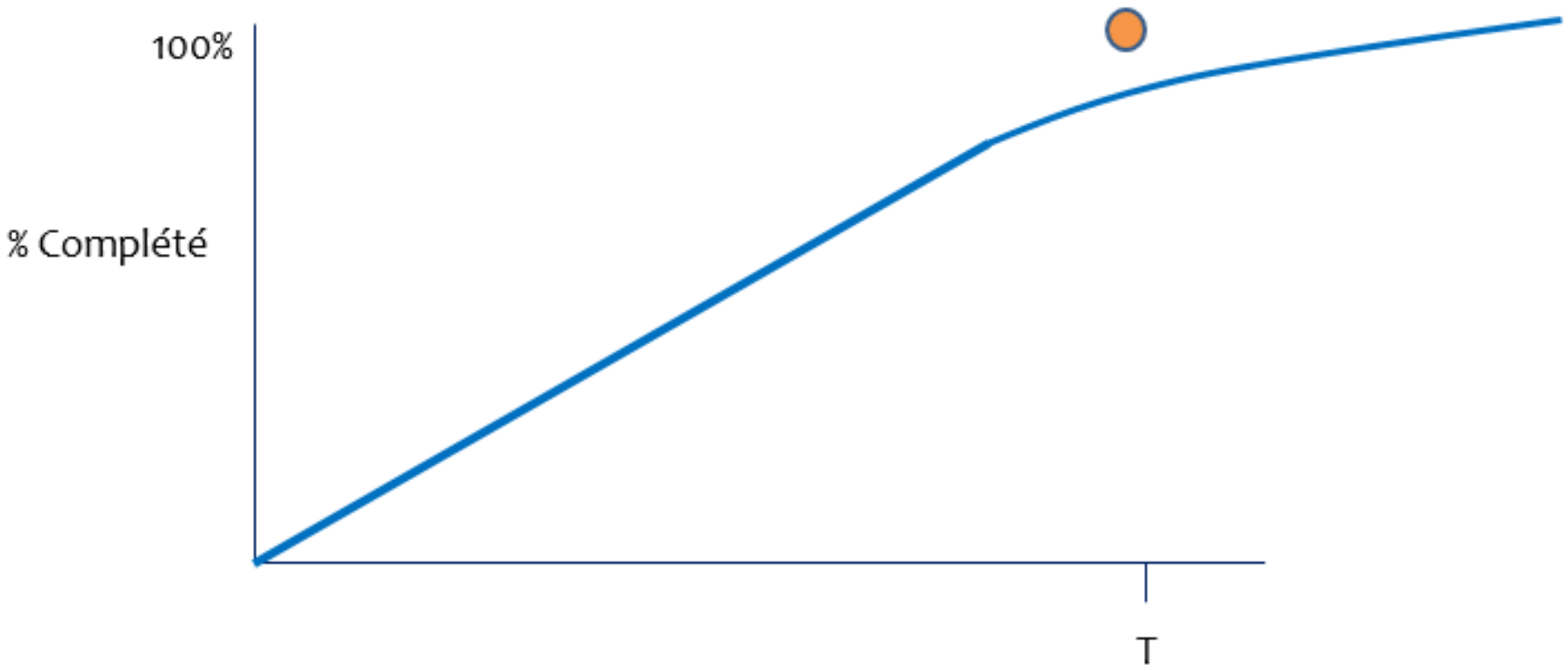
- En réaction aux problèmes des approches traditionnelles qui sont un peu trop rigides.



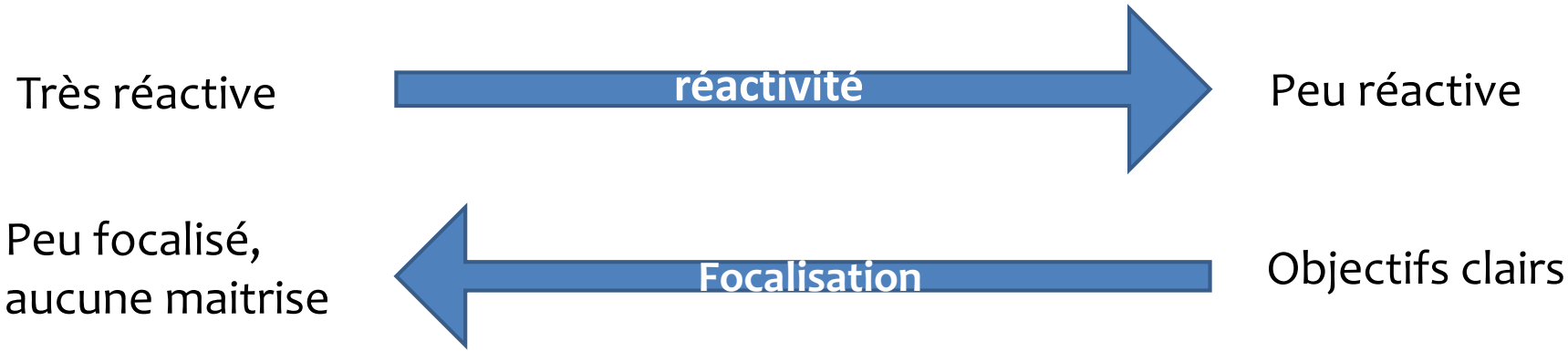
Les constats…

- Les meilleures idées ne viennent pas forcément au début du projet.
  - Il est plus facile de construire par étape que tout imaginer dès le début.
- Les besoins peuvent évoluer pendant le projet.
- Le formalisme n'est pas naturel.
- Les chiffrages et certaines étapes sont difficiles à évaluer.

Un projet informatique… La réalité



Agile : un juste milieu



Absence de méthode

Méthodes prédictives

Agile

- Agile regroupe plusieurs méthodologies :
  - Scrum
  - e**X**trême **P**rogrammain (XP)
  - **D**ynamic **S**ystems **D**evelopment **M**ethod (DSDM)
  - Crystal
  - **A**gile **U**nified **P**rocess (Agile UP ou AUP),
  - **F**eature **D**riven **D**evelopment (FDD) ou développement dirigé par les fonctionnalités
  - Kanban
  - RUP
  - Etc.

- Notion officialisée en 2001 avec le manifeste Agile.

Le manifeste agile

Qu’est ce que le manifeste agile ?

- Le manifeste agile est un **texte fondateur** publié en **2001** par 17 experts en développement logiciel.

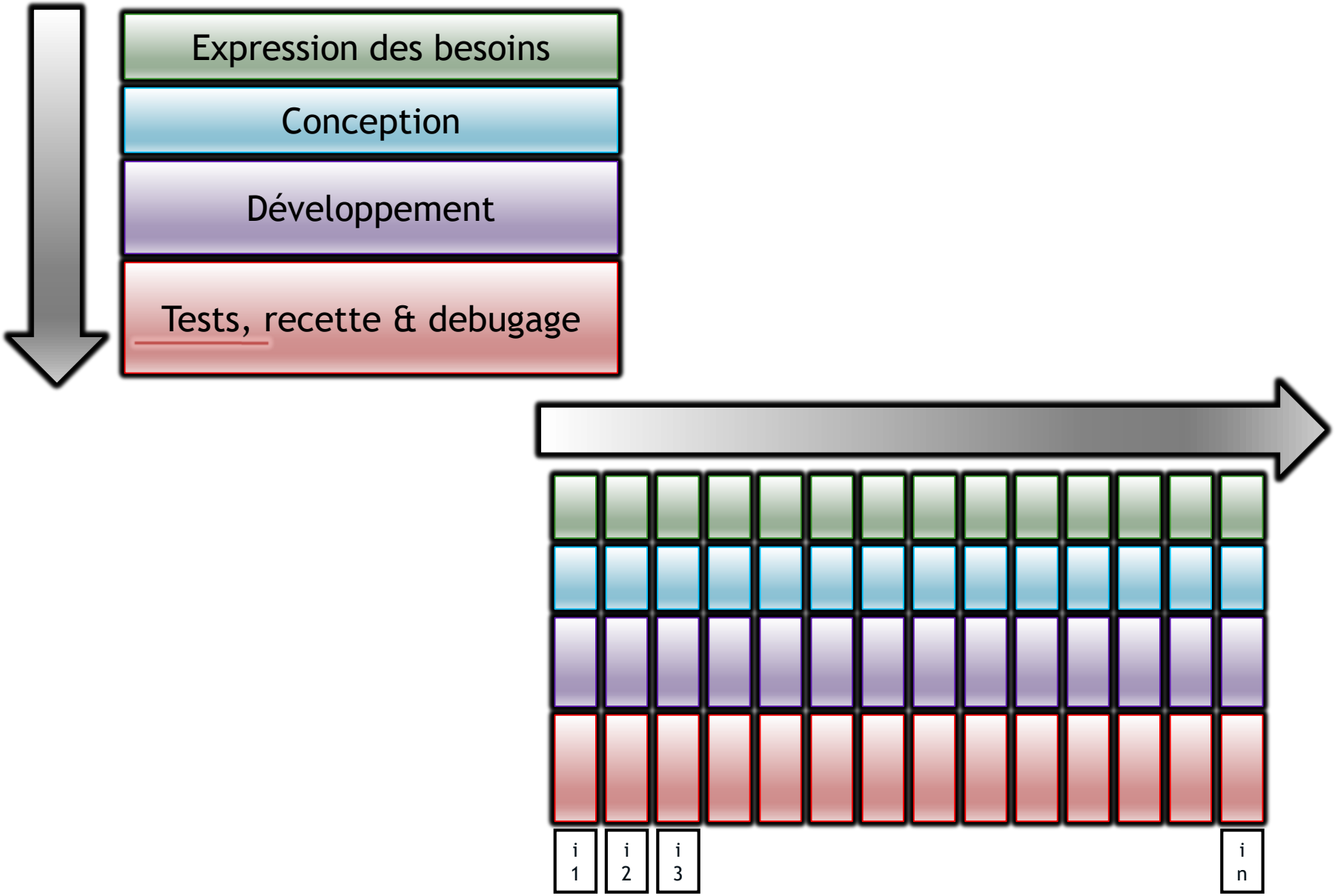
- Il propose une **nouvelle approche de gestion de projet**, plus **souple**, **collaborative** et **centrée sur la valeur pour le client**, en réaction aux méthodes traditionnelles qualifiées souvent de rigides et lourdes.

Le manifeste agile

- Le manifeste agile repose sur 4 valeurs :

Personnes et interactions	Plutôt que	Processus et outils
Un produit opérationnel	Plutôt que	Documentation exhaustive
Collaboration avec le client	Plutôt que	Négociation d'un contrat
Adaptation au changement	Plutôt que	Suivi d'un plan

Le manisfeste agile



Les solutions agiles

- Libérer le génie humain.

  - Pour l'auto-organisation dans un contexte qu'il peut maîtriser :
    - La taille de l'équipe est limitée
    - Le domaine du problème est limité

    - Petites équipes autogérées
    - Portée fonctionnelle restreinte à un moment donné
    - Garder un rythme de travail soutenable
    - Avancement par itération

Les solutions agiles

❑ Collaboration avec le client

- Pas de contrat pour :
  - Instaurer la confiance autrement
  - Éviter les effets pervers d'un contrat

Le manisfeste agile

- Toujours focalisé sur le produit final

  - Une vision commune pour l'équipe
    - La satisfaction du client
  - Découper le projet autrement
    - Par fonctionnement
  - Organiser en cycles de développement réduits
    - Itérations

Les solutions agiles

❑ Adaptables

- Réactives aux nouveaux besoins
- Réceptives aux nouvelles solutions

- ➢ Prendre les décisions définitives le plus tard possible.
- ➢ De courtes itérations permettent de changer de direction sans laisser des éléments à moitié fait.
- ➢ Éviter les effets pervers d'un contrat.

Agile : planification

- L'estimation de charge est difficile, mais les courtes itérations aident :
  - Plus de précisions avec les petites tâches
  - Feedback très rapide
  - Plus facile à s'adapter face aux dérives, surprises, …

Les 12 principes des méthodes agiles

1. Prioriser la satisfaction du client.
2. Accepter les changements.
3. Livrer en permanence des versions opérationnelles de l'application.
4. Assurer le plus souvent possible une coopération entre l'équipe du projet et les gens du métier.
5. Construire les projets autour de personnes motivées.
6. Favoriser le dialogue direct.
7. Mesurer l'avancement du projet en fonction de l'opérationnalité du produit.

Les 12 principes des méthodes agiles

8. Adopter un rythme constant et soutenable par tous les intervenants du projet.
9. Auto-organiser et responsabiliser les équipes.
10. Contrôler continuellement l'excellence de la conception et la bonne qualité technique.
11. Privilégier la simplicité en évitant le travail inutile.
12. Améliorer régulièrement l'efficacité de l'équipe en ajustant son comportement.

La méthode SCRUM

Méthode Scrum

- Elle définit un cadre de travail permettant la réalisation de projets complexes.

- C'est une méthode agile de management qui permet de gérer l'aspect humain d'un projet, principalement la question de ressources humaines et son allocation. Elle a été proposée par Ken Schwaber, en 1996.

- Permet de produire le maximum de valeur pour le minimum de coût.

- Les projets qui suivent la méthode agile SCRUM sont divisés en plusieurs cycles de travail relativement courts que l'on appelle « sprints ».

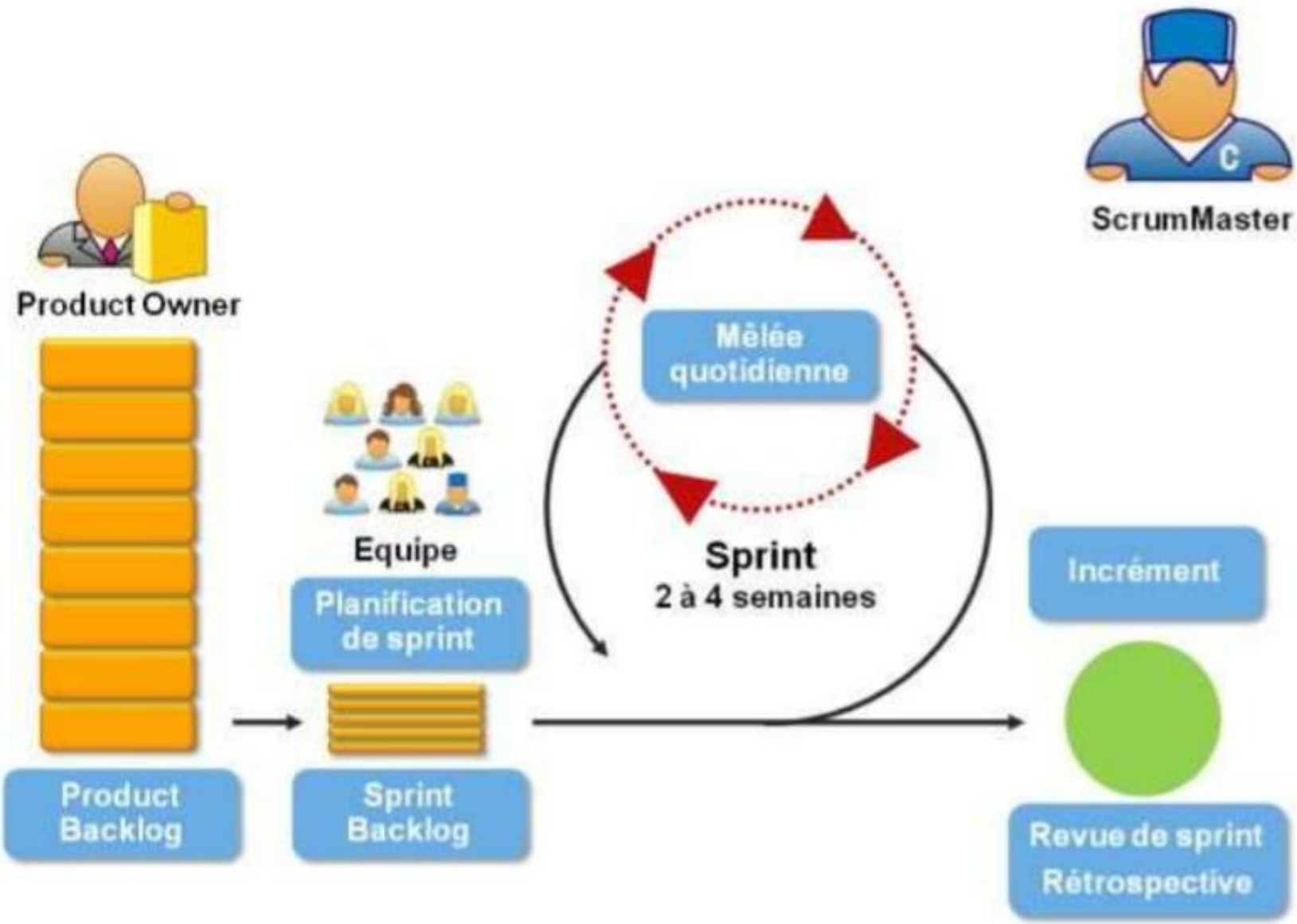
Méthode Scrum

- Les sprints permettent aux membres de l'équipe de mieux planifier les prochaines étapes de développement du projet mais aussi d'évaluer régulièrement les progrès liés au projet.

- Les sprints permettent également de réajuster ou réorienter la direction prise par le projet si besoin.

- Les sprints peuvent durer 1 à 4 semaines.

Méthode Scrum



Répartition des rôles

- Les projets qui utilisent SCRUM se forment autour d'une équipe auto organisée et multifonctionnelle.
  - L'équipe est auto organisée : dans la mesure où il n'y a pas de chef d'équipe qui décide des rôles de chacun ou de la manière dont un problème est résolu, puisque les problématiques sont traitées par l'équipe dans son ensemble.
  - L'équipe est multifonctionnelle car chaque membre est partie prenante dans le développement de chaque fonctionnalité, de l'idée à l'implémentation finale.

Méthode Scrum

- Il existe 3 principaux rôles :
  - Le product owner (responsable produit)
  - Le SCRUM Master
  - Les membres de l'équipe

Méthode Scrum

- Le product owner
  - Définit les spécifications fonctionnelles et communique la vision globale du produit à l'équipe.
  - Établit les priorités des fonctionnalités à développer ou à corriger et valide les fonctionnalités développées.
  - Il doit jouer le rôle de client final, se mettre à sa place, et donc prioriser ses besoins.
  - Ce rôle est celui qui a le plus de responsabilité et d'autorité.
  - Il est en première ligne lorsque quelque chose ne passe mal, ce qui nécessite donc de trouver le juste équilibre entre autorité – responsabilité et engagement.

- Le SCRUM Master
  - Facilitateur entre le product owner et l'équipe.
  - Élimine tous les obstacles qui peuvent empêcher l'équipe d'atteindre les objectifs visés pour chaque sprint de travail.
  - Il s'assure que les principes et les valeurs SCRUM sont bien respectées.
  - Il facilite la communication au sein de l'équipe et cherche à améliorer la productivité et le savoir-faire de son équipe.
  - Il conseille également le product owner sur la façon de maximiser le Return On Investment (retour sur investissement) général de l'équipe.
  - Il anime l'équipe mais ne la dirige pas.

Méthode Scrum

- Les membres de l'équipe
  - Responsables de la réalisation opérationnelle des tâches.
  - Généralement au nombre de 6 à 10 personnes mais peuvent aller jusqu'à 200 personnes.
  - C'est toute l'équipe qui est responsable du résultat final de chaque sprint.
  - La manière dont sont exécutées les tâches est très libre mais cette liberté doit être néanmoins cadrée par l'obligation de répondre aux objectifs du sprint.

Les piliers de la méthode SCRUM

- Les 3 piliers de la méthode SCRUM sont :
  - La transparence. Les aspects importants du processus doivent être visibles à tous. C'est pourquoi SCRUM insiste sur le fait d'avoir un langage commun entre les membres.
  - L'inspection. Un bilan régulier sur les résultats produits est réalisé afin de détecter les écarts indésirables.
  - L'adaptation. Lorsqu'un écart est constaté pendant l'inspection, le processus devra être adapté grâce à des actions visant à améliorer la situation.

Comment ça marche ?

- Le référentiel des exigences initiales est dressé et hiérarchisé avec le client. Il constitue le « product backlog ». Il ne doit pas nécessairement contenir toutes les fonctionnalités attendues dès le début du projet, il va évoluer durant le projet.
- Les fonctionnalités décrites portent le nom de « User Stories » et sont décrites en employant la terminologie utilisée par le client.
- Une User Story contient généralement :
  - ID. un identifiant unique.
  - Nom. Un nom court, descriptif de la fonctionnalité attendue par le client (sans ambiguïtés).
  - Importance. Un nombre entier qui fixe la priorité des stories. Elle peut changer en cours de réalisation du projet.
  - Estimation. La quantité de travail nécessaire pour développer, tester et valider cette fonctionnalité (estimation en nombre de jours idéaux).
  - Démo. Un test de validation relativement simple.
  - Notes. Toute autre information utile : clarifications, références documentation…

Comment ça marche ?

- La méthode SCRUM implique que le projet progresse à travers la mise en place de séries de « sprints ».

- A chaque lancement d'un sprint, une réunion de planification (sprint planning meeting) est organisée afin que chaque membre de l'équipe puisse s'engager sur le nombre de tâches qu'il pourra exécuter, ainsi que sur la création du « sprint backlog » qui est la liste globale des tâches à réaliser lors du sprint. Elles sont développées, testées et livrées au client à la fin du sprint.

Comment ça marche ?

- Chaque jour de sprint, tous les membres de l'équipe (ainsi que le product owner et le SCRUM Master) doivent assister à la réunion SCRUM (ou Mêlée quotidienne (daily scrum).
  - Elle ne dure pas plus de 15 minutes (partage de travail de la veille, identification de tout problème, synchronisation avec tous les membres)
  - A la fin de chaque réunion d'avancement, le SCRUM Master maintient un graphique appelé « sprint burndown chart ». Ce graphique donne une très bonne vision de ce qui a été réalisé et du rythme de travail de l'équipe. Il permet également d'anticiper si toutes les stories du Sprint Backlog seront terminées à la fin de l'itération ou non.

- La fin d'un sprint est marqué par une session de débriefing permettant de présenter le travail achevé au responsable produit et de partager des informations pouvant influer sur le sprint suivant.

Scrum : exemples

La méthode SCRUM
User story

User stories

- Un user story ou récit utilisateur est une **explication non formelle**, générale d'une fonctionnalité logicielle écrite du point de vue de l'utilisateur final.

- Son but est d'expliquer comment une fonctionnalité logicielle apportera de la **valeur ajoutée à l'utilisateur final.** Autrement dit, quelle est l'incidence de la fonctionnalité logicielle sur ce dernier ?

User stories

- Pour rédiger une user story il faut s'intéresser à trois aspects :
  - Utilisateur type : qui sera l'utilisateur type ?
    - Pour qui créons-nous cette fonctionnalité logicielle
    - Quel sont les profils démographique et psychographique de l'utilisateur final ?
  - Besoin : quel est le besoin de l'utilisateur type ?
    - Que cherche à accomplir l'utilisateur type ?
    - Comment votre fonctionnalité aidera-t-elle l'utilisateur final à accomplir ses objectifs ?
  - Fonctionnalité : quel est l'objectif de la fonctionnalité logicielle pour l'expérience de l'utilisateur final ?
    - Quels sont les avantages de la fonctionnalité logicielle ?
    - Quel problème cette fonctionnalité vient-elle résoudre ?
    - Comment cette fonctionnalité s'intègre-t-elle aux objectifs généraux ?

Scrum : exemples

EN TANT QUE {X}
JE VEUX {Y}
AFIN DE {Z}

Y EST UNE FONCTIONNALITÉ
Z EST LE BÉNÉFICE DE LA FONCTIONNALITÉ
X EST LA PERSONNE (OU RÔLE) QUI VA EN BÉNÉFICIER

VOUS IDENTIFIEZ AINSI LA VALEUR APPORTÉE PAR L'HISTOIRE AU MOMENT OÙ ELLE EST DÉFINIE.

User stories : exemples

- En tant que client du CA,
  je veux retirer du liquide du distributeur
  Afin de ne plus faire la queue à l'agence

- En tant que client récurrent,
  je souhaite que mes informations soient sauvegardées
  afin que mon expérience de paiement soit plus fluide

La méthode SCRUM
Burndown chart

Scrum : burndown chart

- Le Burndown Chart est un outil visuel essentiel dans Scrum et d'autres méthodologies agiles, utilisé pour suivre l'avancement du travail et la quantité de travail restant à réaliser au cours d'un sprint.

- Il permet de visualiser si l'équipe est en avance, en retard ou en ligne avec le plan idéal, facilitant ainsi les ajustements et la prise de décision

- Il existe deux types de burndown chart
  - burndown chart d'itération (ou de sprint)
  - burndown chart de release (relatif à la livraison d'une version)

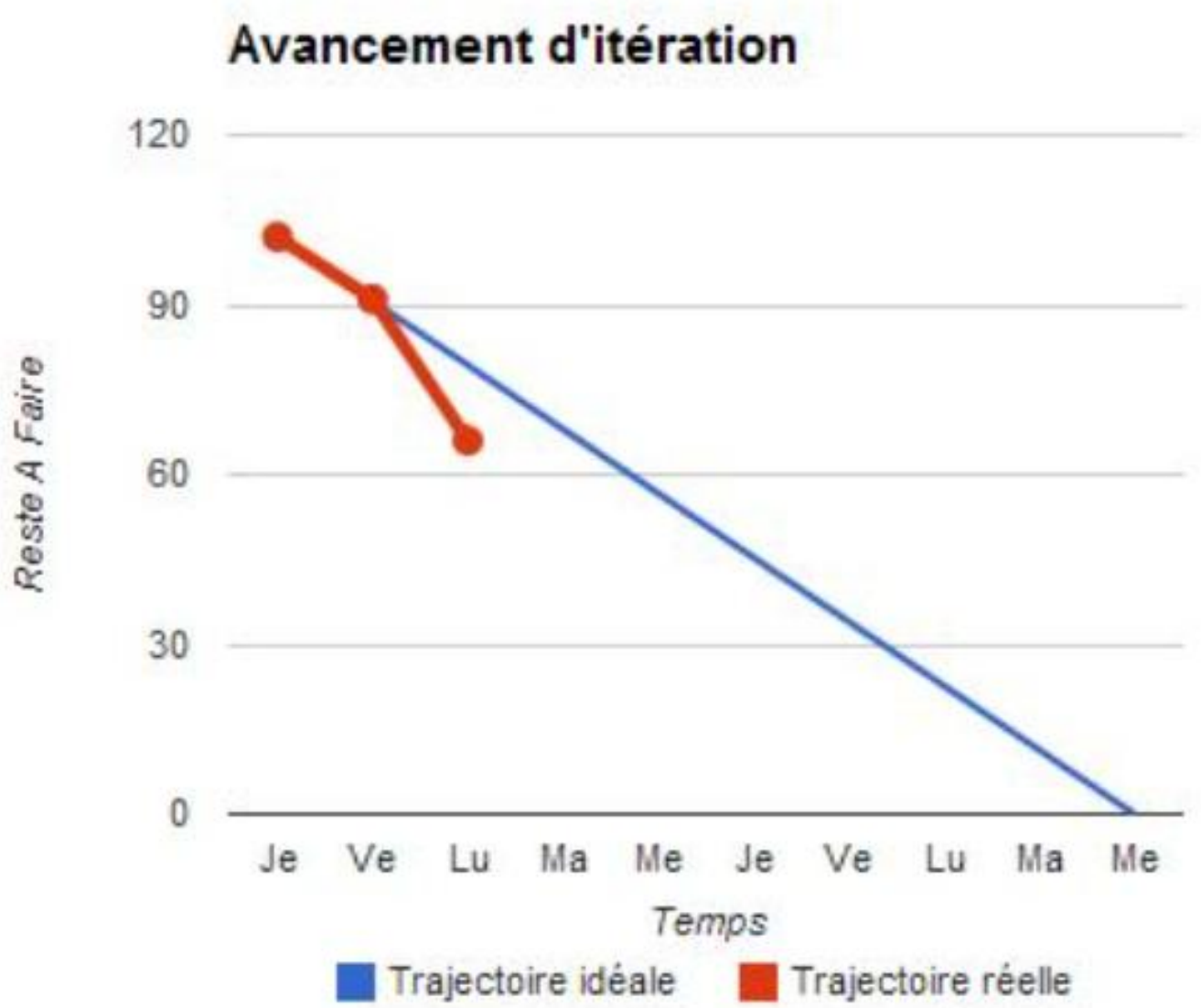
burndown chart : comment le tracer ?

1. Définir l'objectif
   - Identifier un objectif et les actions qui seront nécessaires pour assurer sa réalisation.
2. Découper le projet en tâches et évaluer leur charge
   - Subdivisez les actions que vous jugez nécessaires en tâches.
   - Faire une estimation de l'effort requis pour chaque tâche et à l'évaluation des capacités de chaque personne de l'équipe, vous pouvez affecter un responsable pour chaque tâche.
3. Préparer le graphique
   - Créer un graphique à deux axes :
     - **horizontal,** indiquant le temps (jours, heures, itérations, etc.),
     - **vertical,** indiquant l'effort (story points).
   - Pour évaluer l'évolution du travail et du reste à faire le point de départ n'est évidemment pas le zéro mais le haut du graphique, l'objectif étant d'atteindre le 0 à la fin avec une courbe
4. Compléter et analyser le graphique
   - Chaque jour, le nombre de tâches restant à réaliser est passé en revue et réactualisé.

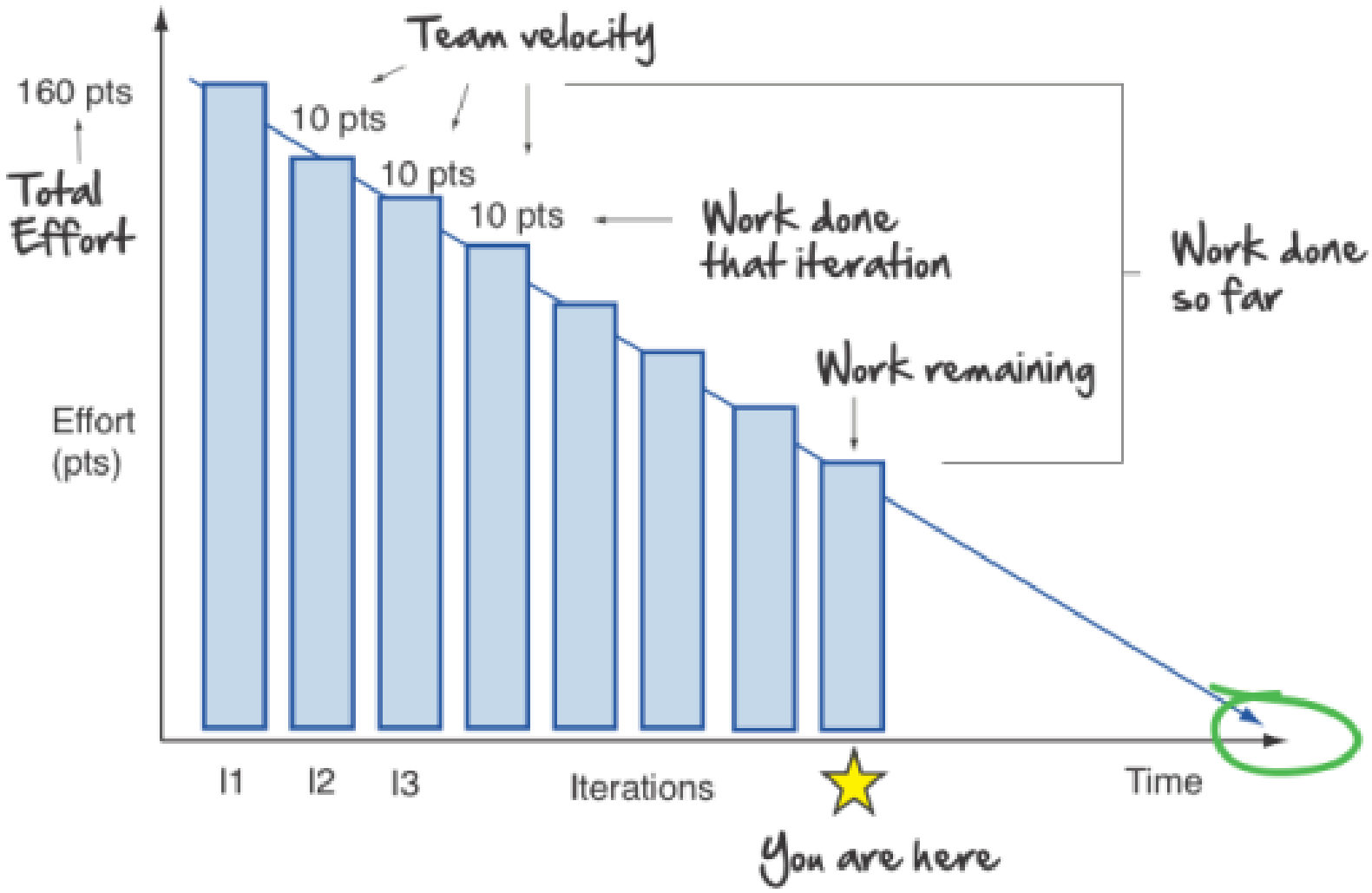
burndown chart : comment le tracer ?

- Pour visualiser l'état d'avancement réel par rapport à l'état d'avancement idéal :
  - Commencez par tracer la droite d'évolution idéale en partant du haut de l'axe vertical jusqu'au bout de l'axe horizontal ;
  - Comparez avec la courbe de la progression réelle dans une autre couleur.

Scrum : exemple de burndown chart d'tération



Scrum : exemple de burndown chart de release



La méthode eXtreme Programming (XP)

Extreme Programaming : caractéristiques

- Méthodologie de développement basée sur des valeurs, principes et pratiques.
- Adaptée aux équipes réduites avec des besoins changeants.
- Son but principal est de réduire les coûts du changement. XP s´attache à rendre le projet plus flexible et ouvert au changement en introduisant des valeurs de base, des principes et des pratiques.
- Elle pousse à l'extrême des principes simples.
  - Les principes de cette méthode ne sont pas nouveaux. L'originalité de la méthode est de les pousser à l'extrême.

Extreme Programaming : caractéristiques

- Communication
  - Entre les membres de l'équipe
  - Verbale
  - Facilitée par la localisation de l'équipe

- Simplicité
  - Cherche la solution la plus simple qui convient au problème du jour.
    - Le refactoring n'est pas un échec, mais une étape normale
    - ✓ Refactorer consiste à modifier un code source de façon à en améliorer la structure, sans que cela modifie son comportement fonctionnel.

Extreme Programaming : valeurs

- Feedback
  - Des tests unitaires, fonctionnels
  - Du client : revue avec le client toutes les 2 à 3 semaines
  - De l'équipe : facilitée par la localisation de l'équipe
    - Par la communication continuelle

- Courage
  - De s'attaquer aux problèmes tout de suite
  - D'appliquer les valeurs XP
  - De jeter du code lorsque c'est nécessaire

- Respect
  - Tous les membres de l'équipe apportent quelque chose peu importe leurs années d'expérience

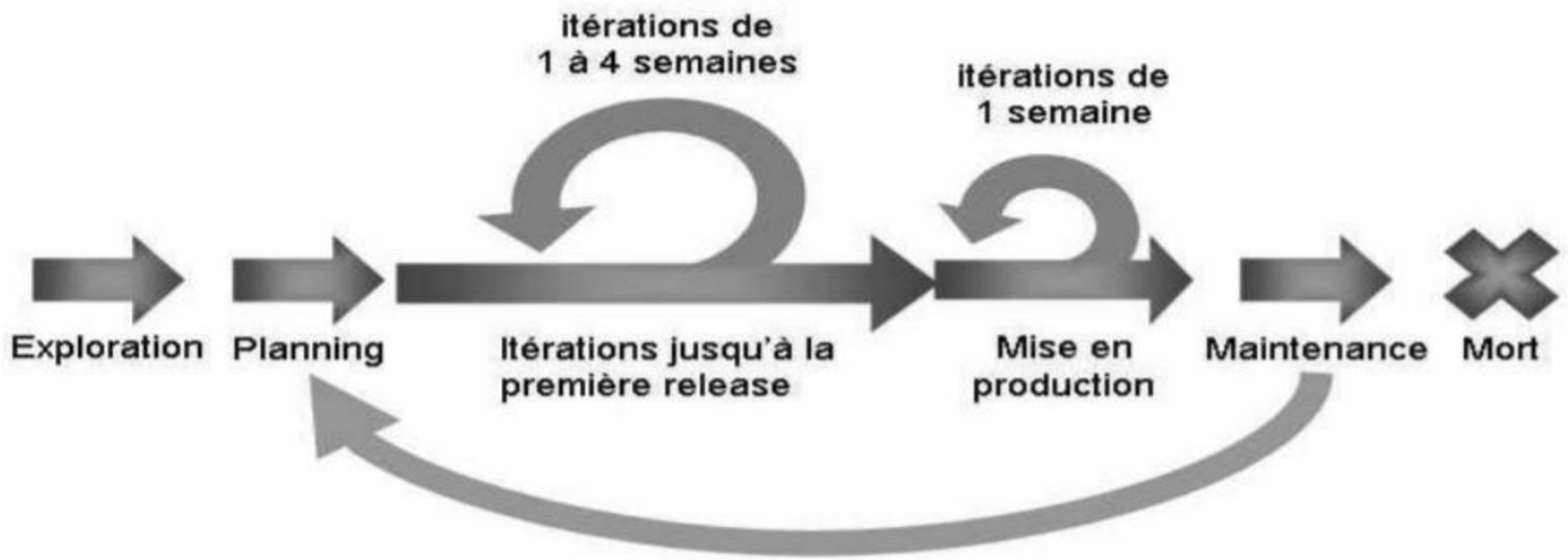
Extreme Programaming : pratiques

- Pair programming (binômage)
  - Partage des idées, bonnes pratiques
  - Partage des expériences
  - Partage des compétences
- Le code appartient à tout le monde
  - Règles de codage
  - Utilisation de pattern, métaphores
- Tests
  - Unitaires
  - Intégration continue
  - TDD (Test Driven Development) ou developpement dirigé par les tests
- Conception incrémentale

Extreme Programaming : pratiques

- Planning game :
  - phase d'exploration : écriture des besoins sous forme de "user stories" et estimation de leur durée
  - phase d'engagement : hiérarchiser les user stories
  - phase de direction : mise à jour du planning
- Petites releases
- Utilisation de métaphores pour décrire l'architecture du système.
- Client sur site : présence sur site d'une personne minimum à temps plein pendant toute la durée du projet.

Extreme Programaming : caractéristiques



maintenance = ajouter des fonctionnalités → nouvelles releases
utilise le même processus que pour la release 1

Extreme Programaming : l'équipe

- 6 principaux rôles
  - Programmeur. C'est le développeur.
  - Client. Définit les priorités, fait les tests de recette (tests d'acceptation, créés par le client), fournit un feedback aux développeurs, etc.
  - Testeur. Il aide le client à faire les tests, installe les outils nécessaires aux tests de recette et tests unitaires.
  - Tracker. Suit l'avancement des tâches au sein d'une itération pour détecter au plus tôt les éventuelles difficultés.
  - Manager. Supérieur hiérarchique des développeurs. Il fournit les moyens humains et matériels pour l'avancement du projet.
  - Coach. Il est garant du respect des règles eXtreme programming.