



**WEBFORCE**  
BE THE CHANGE



## RÉSUMÉ THÉORIQUE - FILIÈRE INTELLIGENCE ARTIFICIELLE

### M108 - Appliquer les méthodes de gestion des projets



**20 heures**



# SOMMAIRE

## 1. S'initier à la gestion de projet

- Formaliser des exigences
- Prioriser des "User Stories"

## 2. Appliquer les méthodes Agile

- Appliquer la méthode SCRUM
- Appliquer la méthode SAFE



# MODALITÉS PÉDAGOGIQUES



**WEBFORCE**  
BE THE CHANGE



1

## LE GUIDE DE SOUTIEN

Il contient le résumé théorique et le manuel des travaux pratiques



2

## LA VERSION PDF

Une version PDF est mise en ligne sur l'espace apprenant et formateur de la plateforme WebForce Life



3

## DES CONTENUS TÉLÉCHARGEABLES

Les fiches de résumés ou des exercices sont téléchargeables sur WebForce Life



4

## DU CONTENU INTERACTIF

Vous disposez de contenus interactifs sous forme d'exercices et de cours à utiliser sur WebForce Life



5

## DES RESSOURCES EN LIGNES

Les ressources sont consultables en synchrone et en asynchrone pour s'adapter au rythme de l'apprentissage

# PARTIE 1

## S'INITIER À LA GESTION DE PROJET

Dans ce module, vous allez :

- Formaliser des exigences
- Prioriser des "User Stories"



07 heures

# CHAPITRE 1

## FORMALISER DES EXIGENCES

**Ce que vous allez apprendre dans ce chapitre :**

- Travailler avec des cahiers de charges
- Vous appuyer sur un diagramme de cas d'utilisation



**03 heures**

# CHAPITRE 1

## FORMALISER DES EXIGENCES

1. **Cahier des charges**
2. Diagramme de cas d'utilisation



### Introduction : Qu'est ce qu'un projet ?

- On appelle **projet** l'**ensemble des actions à entreprendre** afin de répondre à un **besoin défini** dans **des délais fixés**. Un **projet** est une action **temporaire** avec un début et une fin, qui mobilise des **ressources** identifiées (humaines, matérielles et financières) durant sa réalisation, qui possède un **coût** et fait donc l'objet d'une budgétisation de moyens et d'un bilan indépendant de celui de l'entreprise
- Les résultats attendus du projet sont appelés « **livrables** »
- Un projet, dès qu'il est suffisamment complexe et a des enjeux importants, consiste en un ensemble d'actions visant un résultat défini, connu et mesurable. Le projet est limité dans le temps et comporte toujours une notion de nouveauté et de changement
- Le projet est un objectif « extraordinaire » (au sens littéral du mot) qui combine quatre aspects :
  - fonctionnel (réponse à un besoin)
  - technique (respect des spécifications)
  - délais (respect des échéances)
  - coûts (respect du budget)
- Le projet peut également faire partie d'un ensemble plus complexe, souvent appelé programme. Le programme recouvre alors l'ensemble des composantes permettant d'aboutir au résultat final dans toutes ses dimensions : juridique, marketing, informatique, technique, formation des personnels, organisation, logistique, communication, etc.

### Gestion de projets

- La gestion de projet est l'utilisation d'un savoir, d'habiletés, d'outils et de techniques dans le cadre des activités d'un projet, en vue de satisfaire ou de dépasser les exigences et les attentes des parties prenantes à l'égard d'un projet. Le gestionnaire de projet, parfois appelé coordonnateur ou chef de projet, en administre les détails, au jour le jour. Il s'agit là d'un défi constant qui demande une compréhension du contexte plus général du projet et la capacité de concilier des exigences contradictoires telles que :
  - les ressources disponibles et les attentes
  - les priorités différentes des parties prenantes
  - les besoins définis et la portée du projet
  - la qualité et la quantité
- La gestion de projet est aussi une démarche visant à structurer, assurer et optimiser le bon déroulement d'un projet suffisamment complexe pour devoir :
  - être planifié dans le temps
  - faire intervenir de nombreuses parties prenantes : c'est l'objet des organisations qui identifient la maîtrise d'œuvre et maîtrise d'ouvrage
  - responsabiliser le chef de projet ou le directeur de projet, mettre en place un comité de pilotage ou de projet
  - suivre des enjeux opérationnels et financiers importants
- L'objectif est d'obtenir un résultat conforme à des normes de qualité et de performances prédéfinies, pour le moindre coût et dans le meilleur délai possible.
- Lorsque la gestion de projet porte sur un ensemble de projets concourant à un même objectif, on parle de gestion de programme.



### Jalon de projet

- Le jalon (appelé aussi "milestone" ou encore "événement") est un point d'arrêt dans le processus permettant le suivi du projet. C'est l'occasion pour l'équipe de faire un bilan intermédiaire, de valider une étape, des documents ou d'autres livrables, puis de reprendre le déroulé des travaux.
- Il faut savoir qu'à ce stade, le retour en arrière n'est pas possible : une fois le jalon franchi, l'équipe passe à la suite sans revenir sur ce qui a été fait ou validé.
- Dans la pratique, les jalons sont utilisés dans les calendriers de planification des tâches. Ces dates sont matérialisées dans les plannings par un repère dont la durée est nulle.

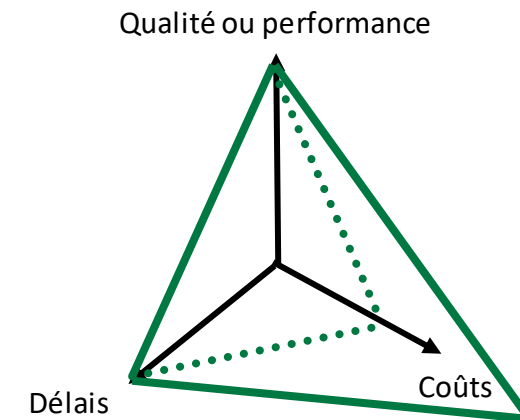
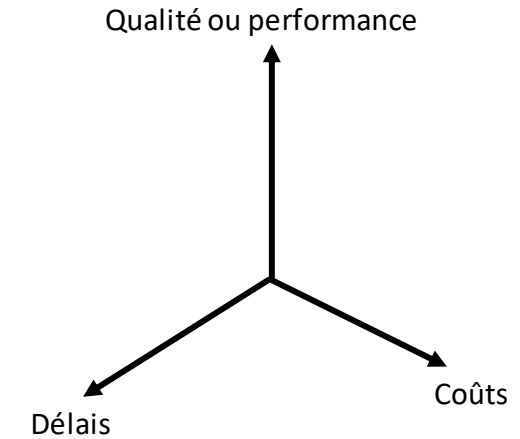
L'objectif d'un jalon est de s'assurer à un moment donné :

- **du bon déroulement de la phase précédente** - notamment pour capitaliser sur ce qui a fonctionné ou pas et revoir potentiellement les pratiques pour la suite
- **de la validation des livrables attendus** : obtention et conformité. Un livrable peut être le point de départ d'une nouvelle étape : par exemple, la réception d'une livraison d'un fournisseur déclenche une étape de conception
- **de la décision du lancement de l'étape suivante** - pouvant donner lieu à une re planification

### Contraintes: le triangle d'or

#### Triptyque QCD :

- **Coût :**
  - respecter le budget définis lors du lancement du projet ;
  - identifier les risques de déviation ;
  - mettre en œuvre les mesures de correction appropriées.
- **Délais :**
  - réaliser le projet dans le respect des délais annoncés au client, (en respectant les principaux jalons) ;
  - identifier les risques de dérapage de planning ;
  - mettre en œuvre les mesures de correction appropriées.
- **Qualité :**
  - réaliser le projet dans le respect de l'état de l'art ;
  - valider chaque étape de la réalisation avec les différents intervenants.



*Exemple : Non respect des coûts*

### Cahier des charges

- Le cahier des charges (CDC) est un document contractuel à respecter lors d'un projet. Le cahier des charges permet au maître d'ouvrage de faire savoir au maître d'œuvre ce qu'il attend de lui lors de la réalisation du projet, entraînant des pénalités en cas de non-respect. Il décrit précisément les besoins auxquels le prestataire ou le soumissionnaire doit répondre, et organise la relation entre les différents acteurs tout au long du projet
- Selon le Guide PMBOK, « Le cahier des charges du projet est la description de la portée du projet, des principaux éléments livrables, des hypothèses et des contraintes »
- Ce document fournit des détails sur le projet, l'étendue du produit et le livrable du projet. Il donne une description détaillée des tâches à exécuter, des hypothèses, des contraintes, des jalons, et plus encore. L'énoncé de la portée du projet aide les parties prenantes à avoir une compréhension commune du projet et de son produit livrable
- Il contient l'inclusion et l'exclusion du projet qui fixe les attentes des parties prenantes du projet, de sorte que ne pas rencontrer plus d'obstacles pour mener à bien le projet

# 01 - Formaliser des exigences

## Cahier des charges



### Importance du cahier des charges

- Il aide les parties prenantes à comprendre la portée du projet et du produit
- Elle réduit les risques de dérapage à des stades ultérieurs
- Il aide le chef de projet à développer le plan du projet sans effort
- Elle aide à développer les lignes de base du projet
- Le cahier des charges réduit l'incertitude en limitant la portée du projet et en donnant à toutes les parties prenantes un aperçu clair du projet

### Portée de projet ou cahier des charges

- Bien que portée de projet et énoncé des travaux (ou cahier des charges) soient souvent regroupés sous l'abréviation SoW -Statement of Work (ou EDT - Énoncé des travaux, en français), il s'agit de deux documents distincts. La portée est une sous-partie du cahier des charges qui décrit de quelle manière les objectifs du projet seront atteints.



<https://asana.com/>

### Les éléments de cahier des charges

Le cahier des charges du projet permet de visualiser l'ensemble du projet et ses résultats. Il comprend les éléments suivants :

- **Introduction** : Une brève introduction qui permet à tous ceux qui consultent le document de comprendre en quoi consiste le projet et qui y participe ;
- **But du projet** : Définit la raison d'être du projet. Il s'agit généralement d'un aperçu des buts et objectifs du projet, qui doit permettre au lecteur de saisir toute son importance ;
- **Description de la portée du produit** : Une description détaillée du produit à réaliser ;
- **Lieu** : Le lieu ou l'environnement dans lequel l'équipe va accomplir son travail : le projet va-t-il se dérouler à distance ou travailler avec les collaborateurs dans les locaux de l'entreprise ?
- **Critères d'acceptation du produit** : Il s'agit des critères selon lesquels les consommateurs accepteront facilement le produit car il répond à leurs attentes ;
- **Produits livrables du projet** : Ce sont les résultats du projet, y compris des éléments tels que des logiciels ou des documents ;
- **Exclusions du projet** : Ces éléments sont en dehors des limites du projet ;
- **Contraintes du projet** : Il s'agit des ressources, de l'organisation, de la politique, de la technologie et de toute autre contrainte qui affecte la performance ;
- **Hypothèses du projet** : Il s'agit de conditions acceptées comme étant vraies pour le bon déroulement du projet - par exemple, la disponibilité des employés au moment voulu ;
- **Jalons** : Étapes importantes de la chronologie d'un projet où se produisent des événements cruciaux ;
- **Période** : Préciser le temps pour produire chaque livrable, le temps consacré à chaque tâche et les heures facturables.

## Développer une description de la portée d'un projet

### 1. Identifier les exigences

- Identifier les différents éléments d'une exigence qui sont essentiels à l'élaboration d'un cahier des charges de projet ;
- Recueillir les exigences des parties prenantes, des utilisateurs et des sponsors par le biais de questionnaires, d'ateliers, d'entretiens ou de méthodes d'observation. Ensuite, traduire les exigences complexes en produits livrables simplifiés ;
- Les exigences comprennent les exigences métier, les exigences des parties prenantes, les exigences de la solution, les exigences du projet, les exigences de transition et les hypothèses. Ainsi, elles permettent de répondre aux désirs et aux attentes.

### 2. Définir les produits

- Documenter et détailler les exigences du produit en utilisant une matrice de traçabilité des exigences (RTM). Cela peut également aider à suivre l'état et les révisions des produits pendant le développement.

### 3. Définir les principaux produits livrables

- Le cahier des charges du projet doit résumer les produits livrables qui permettent aux parties prenantes d'avoir une vue d'ensemble des attentes du produit final.

### 4. Définir les jalons

- Détailler tous les jalons du projet afin que les parties prenantes puissent voir la progression et savoir quand le projet sera terminé.

### 5. Critères d'acceptation

- Il s'agit de ce que doit être le produit pour que le client accepte le livrable. Si l'une de ces conditions n'est pas remplie, le client rejettera le produit.

### 6. Exclusion du projet

- Il est parfois nécessaire d'exclure certains éléments, afin que les parties prenantes comprennent explicitement. Dans les étapes ultérieures, cela permet de réduire la dérive du projet.

### Méthodes de collecte des besoins

- L'analyse des besoins vous permet d'obtenir des données déterminantes relatives aux processus existants, aidant ainsi les équipes à créer des systèmes plus efficaces. Abordons brièvement les méthodes les plus répandues en matière de collecte de données dans le cadre de ce type d'analyse.

#### Questionnaires

- Parmi les méthodes de collecte de données les plus populaires, on retrouve les questionnaires et les entretiens. Un questionnaire est un formulaire contenant des questions fermées d'ordre général. Il vous permet donc de recueillir des informations rapidement, par exemple pour évaluer l'efficacité de votre identité de marque.

#### Enquêtes

- De nombreuses équipes font appel à des enquêtes pour obtenir des informations externes concernant tout ce qui a trait à l'expérience client. Ces enquêtes comportent le plus souvent des questions ouvertes, puisque l'on cherche à obtenir des informations plus détaillées que sur un questionnaire. Elles vous permettent donc de recueillir des données précises rapidement, par exemple pour évaluer la qualité de l'expérience client après achat du point de vue des concernés.

#### Groupes de discussion

- Ces entretiens incluant un nombre réduit de participants qui partagent des caractéristiques ou expériences communes prennent bien plus de temps à mettre en place que les deux autres méthodes. Toutefois, ils permettent de recueillir des informations exhaustives en rapport avec les besoins et l'expérience client. C'est donc l'outil idéal si vous cherchez à obtenir des informations détaillées, par exemple pour vérifier la façon dont vos clients perçoivent votre marque et recueillir leurs retours en matière d'amélioration.



## Méthodes de collecte des besoins

### La méthodologie QQQCCP

#### Présentation

- Outil de questionnement systématique ou d'aide à la construction d'une réflexion. Cette méthodologie est utilisée comme technique de recherche d'informations pour faire émerger des idées, en d'autres termes, pour débroussailler une situation.

#### Principe

- Répondre aux sept questions posées :
  - Qui ? Question en rapport avec les personnes, les responsables
  - Quoi ? Question en rapport avec les objets, les aspects matériels
  - Où ? Question en rapport avec le lieu
  - Quand ? Question en rapport avec le temps, la durée
  - Combien ? Question en rapport avec les quantités
  - Comment ? Question en rapport avec les moyens, les méthodes, les outils
  - Pourquoi ? Question en rapport avec la raison d'utiliser telle procédure, tel moyen

### Les réunions « Brainstorming »

#### Présentation

- Outil de créativité orale de groupe, cette méthode, qui fait appel à l'imagination, favorise l'émergence d'idées neuves par l'échange productif d'un groupe de travail. Elle a été inventée par Alex F. Osborn en 1939. La traduction littérale de « Brainstorming » est « tempête des cerveaux », mais cette méthode est aussi connue sous le nom de « remue-méninges ». Son principe est de faire naître le plus d'idées possibles dans un temps défini. Cette méthode permet de collecter le maximum d'idées en un minimum de temps.

#### Principe

- Mettre en place un groupe hétérogène de 15 personnes maximum sous la conduite d'un animateur spécialisé. Les séances ne dureront pas plus d'une heure. Les idées exprimées ont toutes la même valeur, pas de jugement de valeur, ni vis-à-vis des participants, ni au sujet des idées exprimées. C'est tout l'art de l'animateur qui contribue par son expérience à la réussite de cet exercice.

#### Réalisation

- Toutes les idées exprimées sont inscrites sur un tableau par l'animateur et numérotées. Elles seront triées et classées par famille pour dégager une orientation.

## Méthodes de collecte des besoins

### La méthode de « Metaplan® »

#### Présentation

- Outil de créativité de groupe fondé sur la discussion par l'écrit. Cette méthode consiste à répondre aux questions de l'animateur, par écrit, sur des cartes autocollantes. On y recourt pour analyser une criticité, résoudre un problème donné.

#### Principe

- Mettre en place un groupe limité à 5 personnes, sous la conduite d'un animateur spécialisé.

#### Réalisation

- À partir d'une question ouverte, on exprimera une idée par carte. Affichées et lues à haute voix, les cartes seront regroupées par famille. Les participants hiérarchiseront les réponses en collant des gommettes sur les cartes. L'animateur, par l'observation des masses de gommettes, proposera des hypothèses.

### Le logigramme

#### Présentation

- Le logigramme ou flowchart est un outil simple et visuel permettant de construire un enchaînement séquentiel et logique de différentes étapes. Le logigramme est l'outil idéal d'aide à la rédaction d'une procédure.

#### Principe

- Dessiner le logigramme en utilisant des symboles (ovales ; rectangles ; cercles et losanges) permettant d'identifier à chaque étape de sa construction « qui » fait « quoi » et « comment ».

### Question 1

La finalité d'un projet est de :

1. Changer les comportements
2. Gagner de l'argent
3. Développer de nouvelles opérations
4. Gagner du temps

### Réponse Question 1

La finalité d'un projet est de :

1. Changer les comportements
2. Gagner de l'argent
- 3. Développer de nouvelles opérations**
4. Gagner du temps

### Question 2

Les livrables représentent :

1. Un document qui regroupe les outils réalisés dans la phase
2. Des documents qui valident la fin de la phase
3. Des points d'étapes
4. Des états d'avancement du projet

### Réponse Question 2

Les livrables représentent :

1. Un document qui regroupe les outils réalisés dans la phase
- 2. Des documents qui valident la fin de la phase**
3. Des points d'étapes
4. Des états d'avancement du projet

### Question 3

Chaque fin d'étape donne lieu à :

1. Une décision
2. L'ouverture d'une ligne budgétaire
3. Une réunion du comité de pilotage
4. Une prime pour le chef de projet

### Réponse Question 3

Chaque fin d'étape donne lieu à :

1. **Une décision**
2. L'ouverture d'une ligne budgétaire
3. Une réunion du comité de pilotage
4. Une prime pour le chef de projet



#### Question 4

Dans la réalisation d'un projet, le maître d'ouvrage est celui qui :

1. définit le cahier des charges d'ouvrage
2. est responsable de la coordination des intervenants
3. assure les études de conception
4. dirige le chantier de construction de l'ouvrage

### Réponse Question 4

Dans la réalisation d'un projet, le maître d'ouvrage est celui qui :

1. **définit le cahier des charges d'ouvrage**
2. est responsable de la coordination des intervenants
3. assure les études de conception
4. dirige le chantier de construction de l'ouvrage

### Question 5

Quelles sont les méthodes de définitions des besoins :

1. Questionnaire
2. Enquête
3. Groupes de discussion
4. QQQQCCP

### Réponse Question 5

Quelles sont les méthodes de définitions des besoins :

1. Questionnaire
2. Enquête
3. Groupes de discussion
4. QQQQCCP

# CHAPITRE 1

## FORMALISER DES EXIGENCES

1. Cahier des charges
2. **Diagramme de cas d'utilisation**

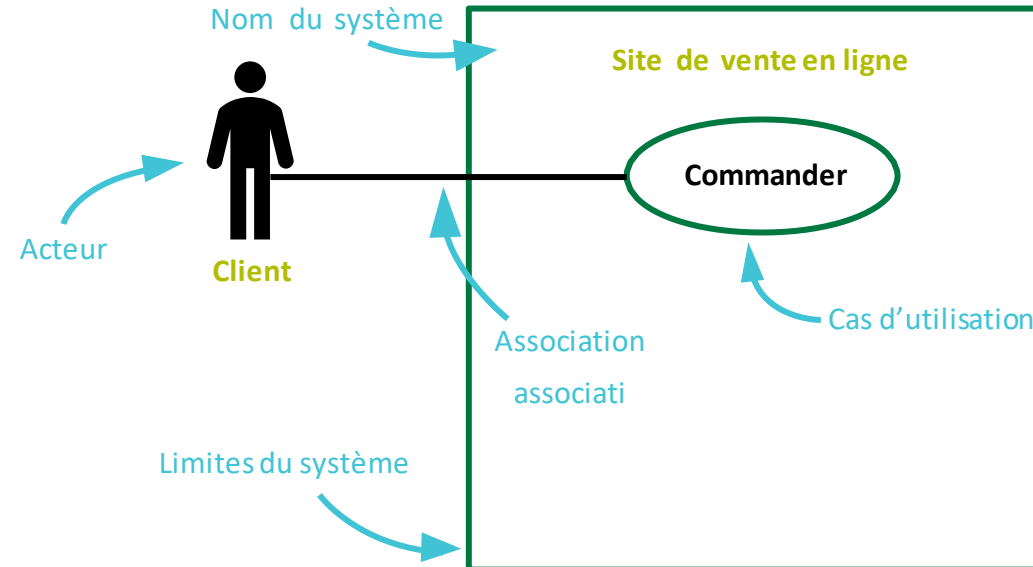


# 01 - Formaliser des exigences

## Diagramme de cas d'utilisation

### Les cas d'utilisation

- Représente une vue du système qui met l'accent sur le comportement du système tel qu'il apparaît aux utilisateurs externes. Il permet la représentation des fonctionnalités du système
- Les diagrammes de cas d'utilisation sont élaborés pour visualiser les relations entre les acteurs et les cas d'utilisation
- Les diagrammes de cas d'utilisation présentent une vue extérieure du système



# 01 - Formaliser des exigences

## Diagramme de cas d'utilisation

### Acteurs et cas d'utilisation

- Acteurs et cas d'utilisation permettent de décrire le système :
  - Les acteurs interagissent directement avec le système



Un acteur



- Les cas d'utilisation représentent l'utilisation du système par les acteurs



Un cas d'utilisation

# 01 - Formaliser des exigences

## Diagramme de cas d'utilisation

### Diagramme de Use case : Les acteurs

- Un acteur est un rôle d'un ou plusieurs objets situés à l'extérieur du système et qui interagissent avec lui pour remplir une fonctionnalité donnée de ce système
  - Un acteur caractérise le rôle joué par un objet à l'extérieur du système
  - Un acteur est :
    - Un humain (via une IHM)
    - Du soft
    - Du hard
- Un Cas d'utilisation ( use case ) est une fonctionnalité remplie par le système et qui se manifeste par un ensemble de messages échangés entre le système et un ou plusieurs acteurs.



Utilisateur



Utilisateur

Vérifier la Bonne  
Marche



Capteur de température

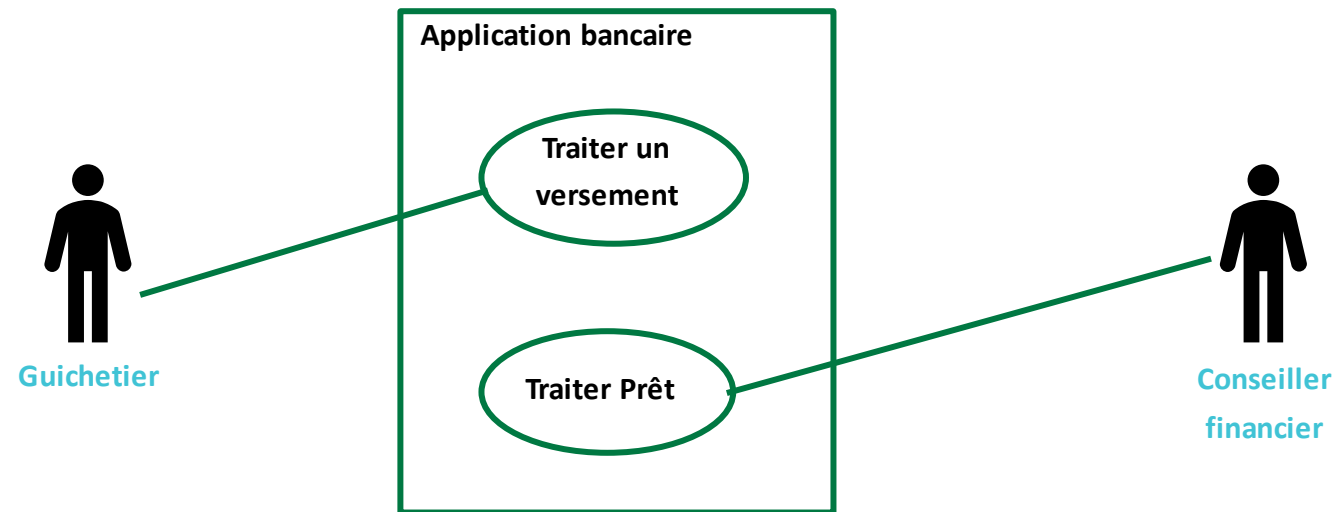


# 01 - Formaliser des exigences

## Diagramme de cas d'utilisation

### Diagramme de cas d'utilisation

- Un cas d'utilisation définit un comportement du système sans révéler sa structure interne ; il spécifie les services que le système fournit à ses utilisateurs et les interactions acteurs/système



# 01 - Formaliser des exigences

## Diagramme de cas d'utilisation



### Relations entre cas d'utilisation

#### Relation « include » :

- inclusion d'un cas d'utilisation dans un autre
- à utiliser quand on répète plusieurs fois la même séquence dans différents cas d'utilisation

#### Relation « extends » :

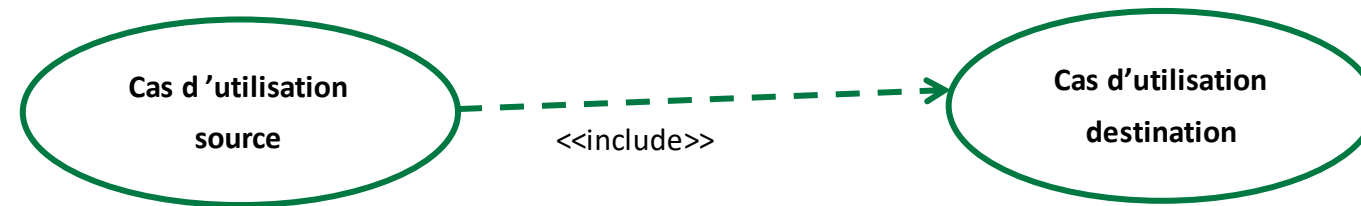
- ajout optionnel de comportement dans un cas d'utilisation
  - à définir :
    - condition d'extension
    - point d'extension dans le cas d'utilisation étendu
- à utiliser quand on décrit une variation sur un comportement normal.

# 01 - Formaliser des exigences

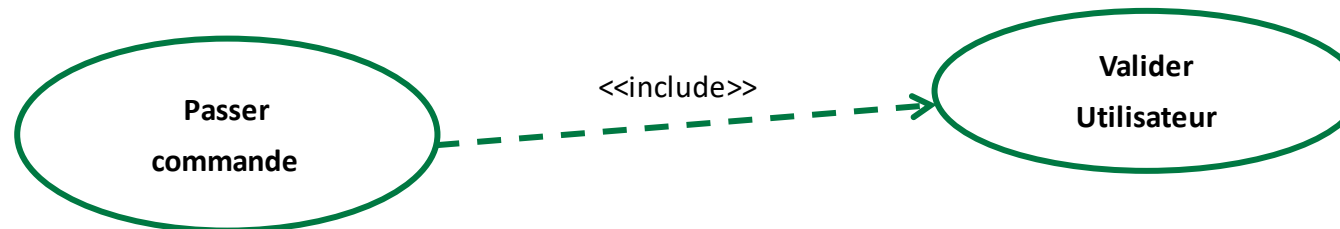
## Diagramme de cas d'utilisation

### Relation d'inclusion : définition

- La relation d'inclusion signifie que le cas d'utilisation source comprend le comportement décrit par le cas d'utilisation destination en un point d'insertion bien déterminé.



Exemple :



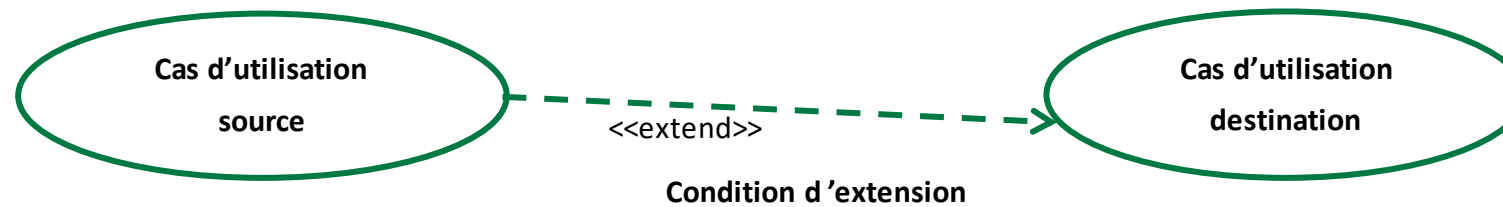
- La relation d'inclusion est un exemple de délégation. Un cas d'utilisation est partagé par plusieurs cas d'utilisation.

# 01 - Formaliser des exigences

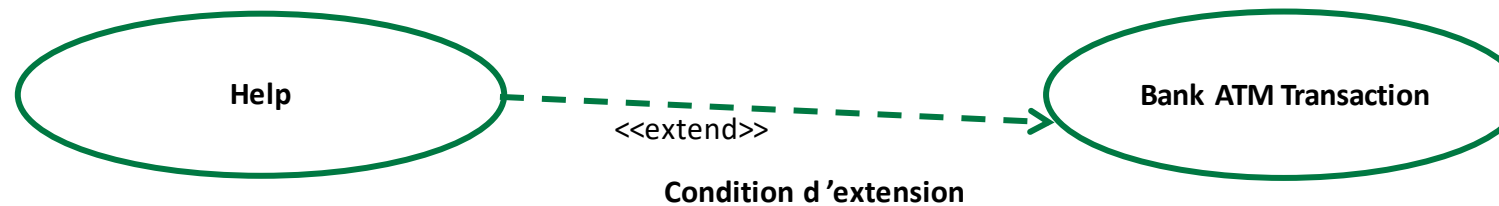
## Diagramme de cas d'utilisation

### Relation d'extension : définition

- Une relation d'extension entre cas d'utilisation signifie que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination.



Exemple :

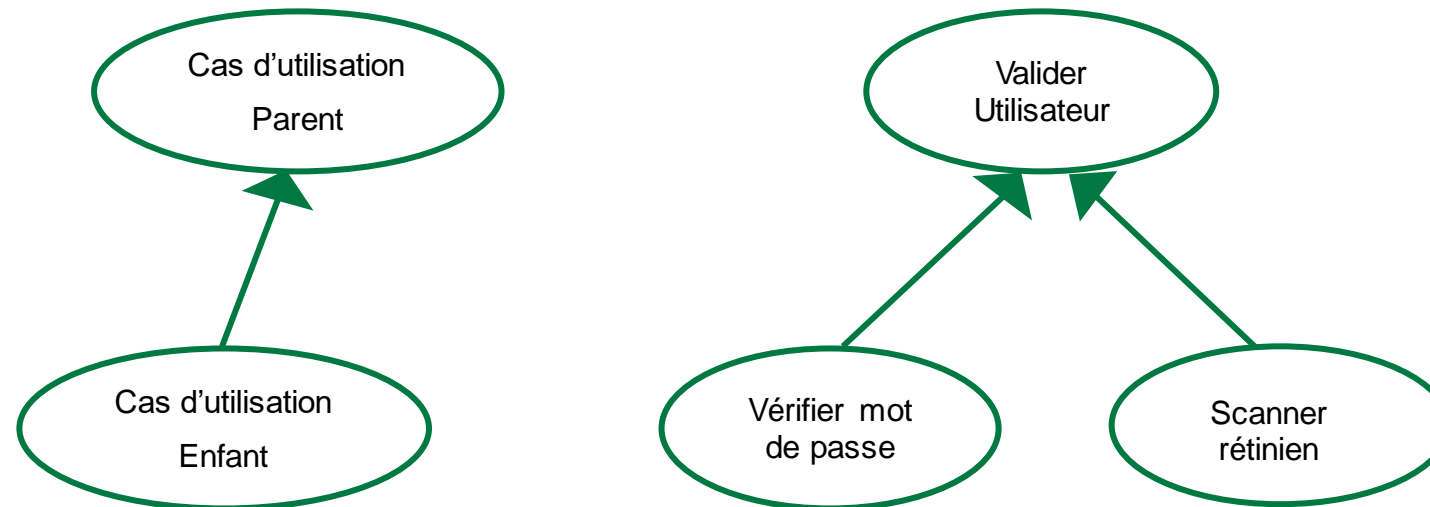


# 01 - Formaliser des exigences

## Diagramme de cas d'utilisation

### Relation de généralisation : définition

- Une relation de généralisation entre cas d'utilisation signifie que le cas d'utilisation enfant est une spécialisation du cas d'utilisation parent. Le cas d'utilisation parent peut être abstrait.



# 01 - Formaliser des exigences

## Diagramme de cas d'utilisation



### Construction des cas d'utilisation

- Un cas d'utilisation doit avant tout être simple, intelligible, décrit de manière claire et concise.
- Lors de la construction, il faut se demander :
  - Quelles sont les tâches de l'acteur ?
  - Quelles informations l'acteur doit-il créer, sauvegarder, modifier, détruire ou simplement lire ?
  - L'acteur devra-t-il informer le système des changements externes ?
  - Le système devra-t-il informer l'acteur des conditions internes ?

# 01 - Formaliser des exigences

## Diagramme de cas d'utilisation



### Relations standard entre cas d'utilisation

- Trois types de relations standard entre cas d'utilisation sont proposés par UML :
  - <<include>> : le cas d'utilisation incorpore explicitement et de manière obligatoire un autre cas d'utilisation à l'endroit spécifié ;
  - <<extend>> : le cas d'utilisation incorpore implicitement de manière facultative un autre cas d'utilisation à l'endroit spécifié ;
  - Généralisation : les cas d'utilisation descendants héritent des propriétés de leur parent.

## 01 - Formaliser des exigences

### Diagramme de cas d'utilisation



#### Exercice

Dans un établissement scolaire, on désire gérer la réservation des salles de cours ainsi que du matériel pédagogique (ordinateur portable ou/et Vidéo projecteur).

Seuls les enseignants sont habilités à effectuer des réservations (sous réserve de disponibilité de la salle ou du matériel).

Le planning des salles peut quant à lui être consulté par tout le monde (enseignants et étudiants).

Par contre, le récapitulatif horaire par enseignant (calculé à partir du planning des salles) ne peut être consulté que par les enseignants.

Enfin, il existe pour chaque formation un enseignant responsable qui seul peut éditer le récapitulatif horaire pour l'ensemble de la formation.

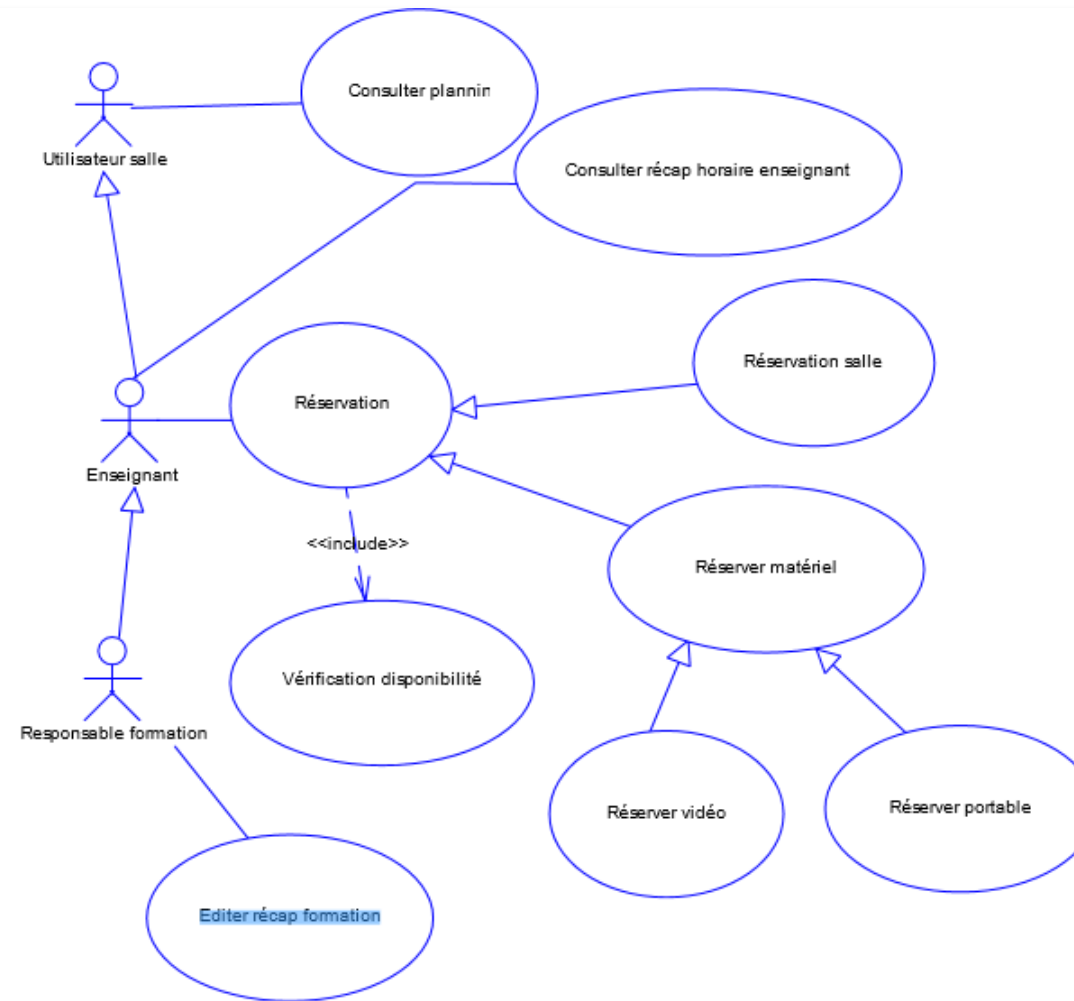
1. Modéliser cette situation par un diagramme de cas d'utilisation.



# 01 - Formaliser des exigences

## Diagramme de cas d'utilisation

### Solution



## CHAPITRE 2

### FORMALISER DES EXIGENCES

Ce que vous allez apprendre dans ce chapitre :

- Découvrir les itérations du projet
- Maîtriser le diagramme de Gantt
- Élaborer un diagramme de Pert



04 heures



## CHAPITRE 2

### PRIORISER DES "USER STORIES"

#### 1. **Itération d'un projet**

- 2. Diagramme de Gantt
- 3. Diagramme de Pert



## 02 - Prioriser des "User Stories"

### Itérations du projet



### Phases de la gestion d'un projet

Ces phases représentent le cycle de vie d'un projet dans divers secteurs professionnels.

#### 1. Initiation

- Déterminer le besoin et évaluer la viabilité

#### 2. Planification Ordonnancement

- Définir le budget, identifier les risques et établir des objectifs clairs

#### 3. Réalisation

- Créer les livrables, déléguer les tâches et maintenir une communication transparente

#### 4. Suivi et contrôle

- Suivre la progression de l'équipe et surveiller le projet avec un logiciel adéquat

#### 5. Clôture

- Faire le bilan des réussites et des échecs

## 02 - Prioriser des "User Stories"

### Itérations du projet



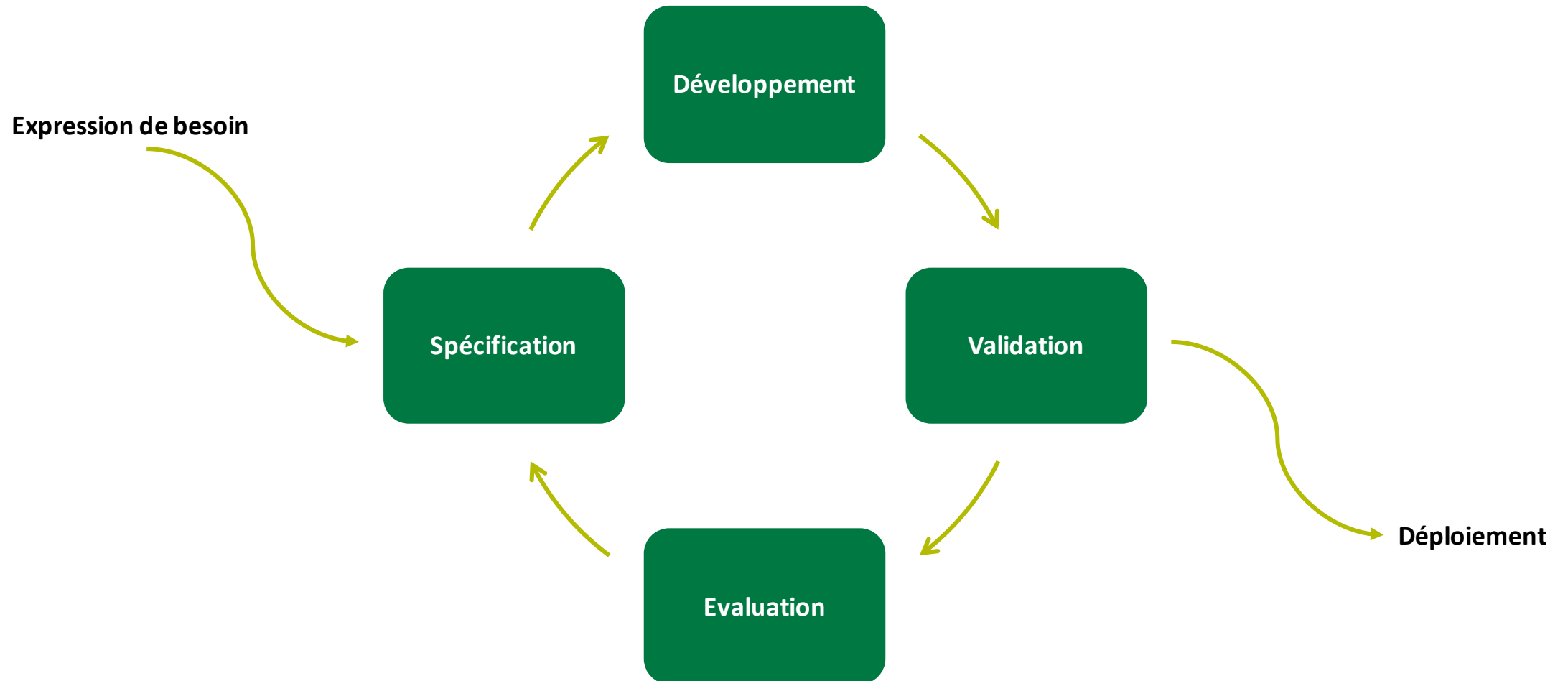
#### Itération

- Une itération c'est une courte période de temps, de durée fixe (1, 2, 3 ou 4 semaines) durant laquelle vont se dérouler une série d'activités (analyse, ergo, conception, codage, test...) et qui se termine par une livraison (interne ou externe).
- Le **développement logiciel itératif et incrémental** commence par la planification et se poursuit par des cycles de développement itératifs impliquant un retour continu des utilisateurs et l'ajout progressif de fonctionnalités se terminant par le déploiement du logiciel terminé à la fin de chaque cycle.
- Le **développement itératif et incrémental** peut être regroupé dans les phases suivantes :
  - **Phase de lancement** : traite de la portée du projet, des exigences et des risques aux niveaux supérieurs
  - **Phase d'élaboration** : offre une architecture de travail qui modère les risques identifiés lors de la phase de démarrage et satisfait les exigences fonctionnelles
  - **Phase de construction** : remplit progressivement les composants de l'architecture avec du code prêt pour la production, qui est produit par l'analyse, la mise en œuvre, la conception et le test des exigences fonctionnelles
  - **Phase de transition** : fournit le système à l'environnement d'exploitation de production.

## 02 - Prioriser des "User Stories"

### Itérations du projet

#### Cycle itératif



## CHAPITRE 2

### PRIORISER DES "USER STORIES"

1. Phase de la gestion de projet
2. **Diagramme de Gantt**
3. Diagramme de Pert



## 02 - Prioriser des "User Stories"

### Diagramme de Gantt

#### Définitions

- Un **diagramme de Gantt** est un outil de gestion de projet qui illustre un **plan de projet**. Il comprend généralement deux sections : la partie gauche **présente une liste de tâches**, tandis que la partie droite comporte **une ligne de temps** avec **des barres d'horaires** qui visualisent le travail ;
- Le **diagramme de Gantt** peut également inclure les **dates de début** et de **fin des tâches**, les **jalons**, les **dépendances entre les tâches** et les destinataires ;
- Quelques outils pour la mise en place d'un diagramme de Gantt :



*Gantt project*



*MS project*



## 02 - Prioriser des "User Stories"

### Diagramme de Gantt



### A quoi sert un diagramme de Gantt ?

Les chefs de projet utilisent les diagrammes de Gantt pour trois raisons principales :

#### 1. Construire et gérer un projet complet

- Les diagrammes de Gantt visualisent les éléments constitutifs d'un projet et les organisent en tâches plus petites et plus faciles à gérer. Les petites tâches qui en résultent sont planifiées sur la ligne de temps du diagramme de Gantt, avec les dépendances entre les tâches, les attributaires et les jalons.

#### 2. Déterminer la logistique et les dépendances des tâches

- Les diagrammes de Gantt peuvent être utilisés pour garder un œil sur la logistique d'un projet. Les dépendances entre les tâches garantissent qu'une nouvelle tâche ne peut commencer que lorsqu'une autre tâche est terminée. Si une tâche est retardée (cela arrive aux meilleurs d'entre nous), les tâches dépendantes sont automatiquement reprogrammées. Cela peut être particulièrement utile lors de la planification dans un environnement multi-équipes.

#### 3. Suivre l'avancement d'un projet

- Comme les équipes enregistrent le temps passé sur les points du plan, le gestionnaire peut surveiller la santé de ses projets et faire des ajustements si nécessaire. Le diagramme de Gantt peut inclure des dates de publication, des jalons ou d'autres mesures importantes pour suivre l'avancement de projet.

## 02 - Prioriser des "User Stories"

### Diagramme de Gantt



#### Origines du diagramme de Gantt

- Au début du XXe siècle, Henry Gantt a révolutionné la gestion de projet avec les diagrammes de Gantt. À l'époque, ils étaient écrits sur des feuilles de papier. Avec l'essor des ordinateurs dans les années 1980, les diagrammes de Gantt sont devenus de plus en plus complexes et élaborés. Aujourd'hui, les diagrammes de Gantt restent l'un des outils de gestion de projet les plus utilisés
- Aujourd'hui, les outils de diagramme de Gantt sont souvent appelés outils de feuille de route.

## 02 - Prioriser des "User Stories"

### Diagramme de Gantt



### Diagrammes de Gantt dans la planification en cascade ou agile

Les diagrammes de Gantt peuvent être un outil puissant pour les méthodologies en cascade et Agile.

#### Le modèle en cascade

- Le modèle de planification de projet en cascade suit une approche linéaire dans laquelle les exigences des parties prenantes et des clients sont recueillies au début du projet. À partir de là, les gestionnaires de projet créent un plan de projet séquentiel, assorti de jalons et de délais. Chaque élément du projet repose sur l'achèvement des tâches précédentes. Cette méthode est privilégiée par les équipes qui se concentrent sur le processus (comme la construction ou la fabrication) et moins sur l'idéation ou la résolution de problèmes, car les étapes doivent être planifiées à l'avance.
- Les diagrammes de Gantt sont généralement préférés par les chefs de projet utilisant la méthode de la cascade. Ils déterminent le calendrier d'un projet en le divisant en blocs de travail gérables et en attribuant des dates de début et de fin. Il est également utile pour identifier les étapes importantes de projet. Les jalons sont des réalisations que les équipes doivent atteindre dans les délais ou en avance sur le calendrier. Ils sont facultatifs mais recommandés.

#### Agile

- D'autre part, le modèle agile de planification de projet valorise la flexibilité et l'adaptabilité. Au lieu de créer un calendrier complet avec des dates fixes, les équipes agiles divisent les projets en itérations plus petites (également appelées sprints). Au début d'un sprint, l'équipe planifie son travail en fonction des objectifs du projet pour les deux semaines à venir. Une fois le sprint terminé, les réalisations et les développements qui en ont découlé permettent d'élaborer le plan du sprint suivant.
- Un diagramme de Gantt peut montrer comment la modification d'une tâche peut avoir un impact sur le plan ou la feuille de route du produit. Pour les équipes agiles, cela est essentiel, car le retour d'information des parties prenantes constitue une part importante de la méthodologie.

## 02 - Prioriser des "User Stories"

### Diagramme de Gantt



#### Utilisation des diagrammes de Gantt

- Les diagrammes de Gantt restent un outil important de gestion de projet dans divers secteurs. À la fin de la deuxième décennie du 21e siècle, le Project Management Institute a conclu que seulement 11 % des organisations étaient entièrement agiles. La plupart des organisations utilisent des méthodologies de gestion de projet en cascade (généralement au niveau de la direction) en plus de la méthodologie agile. C'est ce qu'on appelle une approche hybride.
- Si on pense en termes de "dates et d'échéances", ainsi on a besoin de diagrammes de Gantt basés sur des échéances.

## 02 - Prioriser des "User Stories"

### Diagramme de Gantt

### Étapes pour créer un diagramme de Gantt efficace

Voici les 6 étapes pour créer un diagramme de Gantt efficace.



## 02 - Prioriser des "User Stories"

### Diagramme de Gantt



### Méthodologie et démarche

#### 1. Définir le périmètre de projet

- L'objectif, la date de début, la date de fin du projet, le budget doivent être identifiés
- Les tâches doivent être identifiées
- Les tâches doivent être quantifiées en terme de délais, de charges ou de ressources
- La logique de l'ensemble des tâches doit être analysée

#### 2. Déterminer et structurer la liste des tâches en respectant au mieux une chronologie. Cette identification peut se faire par des techniques comme le Brainstorming ou les groupes de travail.

#### 3. Estimer les durées et les ressources

- Il faut ensuite remplir un tableau présentant, pour chaque tâche, la durée de celle-ci et les ressources affectées : utiliser la même unité de temps pour toutes les tâches dans un souci d'harmonisation du diagramme de GANTT. Quant aux ressources, elles peuvent être humaines ou matérielles.

#### 4. Réaliser le réseau logique (dépendances entre chaque tâche)

- Le réseau doit reprendre les hypothèses de priorité des tâches. Il se présente souvent sous la forme de tâches reliées entre elles par des liens logiques. Définir les tâches directement antérieures à chaque tâche. Une fois le réseau tracé, on retrouvera la chronologie du projet.

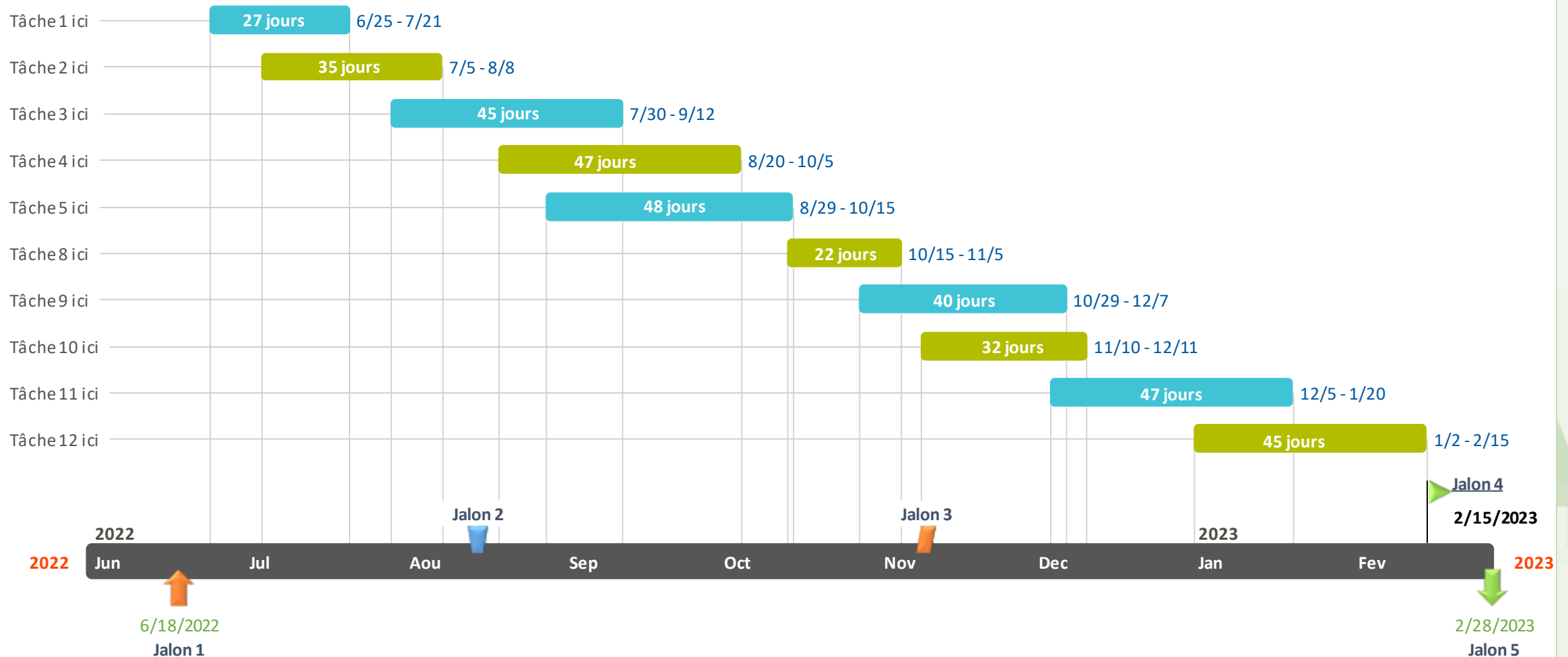
#### 5. Tracer le diagramme de GANTT (planning détaillé du projet)

- Les coordonnées du graphique font apparaître les tâches en ordonnée et la durée (heures, jours, semaines,...) en abscisse :
  - Chaque ligne représente donc une tâche, et chaque colonne, une unité de temps
  - Chaque tâche est représentée par une droite horizontale, dont la longueur est proportionnelle à la durée
  - Dessiner chaque tâche, en représentant au fur et à mesure, la contrainte en amont
  - Lorsque la dernière tâche est représentée, il convient de suivre le graphique "en marche arrière", pour déterminer le chemin critique

## 02 - Prioriser des "User Stories"

### Diagramme de Gantt

#### Exemple



## 02 - Prioriser des "User Stories"

### Diagramme de Gantt

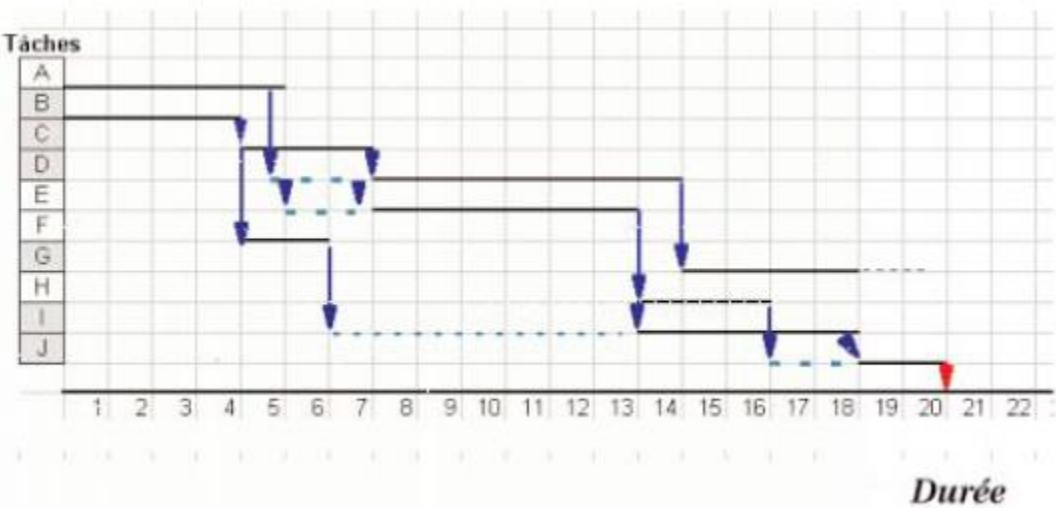


#### Exemple

- Lister les tâches, estimer les durées et identifier l'ordre dans lequel les tâches doivent être faites.

Tâches	Durée	Tâches précédentes
A	5	
B	4	
C	3	B
D	7	A C
E	6	A C
F	2	B
G	4	D
H	3	E
I	5	E F
J	2	H I

- Dessiner chaque tâche en faisant apparaître aussitôt et au fur à mesure, la ou les contraintes antérieures et/ou les marges de manœuvres.





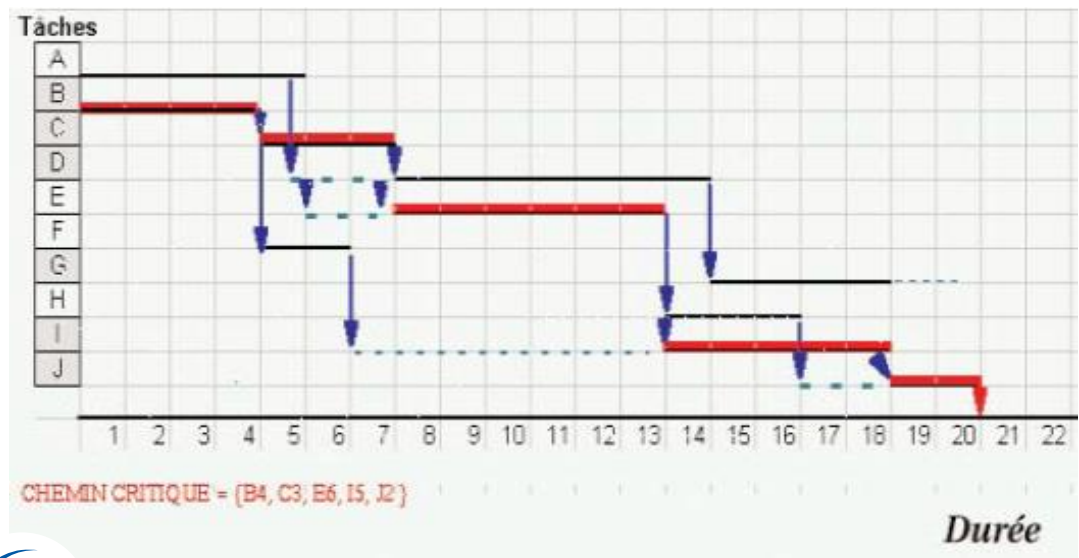
## 02 - Prioriser des "User Stories"

### Diagramme de Gantt



### Exemple

- Lire le graphique en sens inverse pour représenter le chemin critique (tâches immédiates derrières).



- Calculer l'effectif total par unité de temps.

Tâches	Durée	Tâches précédentes	Effectif
A	5		4
B	4		4
C	3	B	5
D	7	A C	1
E	6	A C	2
F	2	B	6
G	4	D	4
H	3	E	4
I	5	E F	3
J	2	H I	4

### À noter

- **Chemin critique** : "Le chemin critique se définit comme étant le temps minimum nécessaire pour réaliser le projet, il s'agit de la séquence la plus longue d'activités pour traverser le réseau du nœud de début jusqu'au nœud de fin. Les activités faisant partie du chemin critique sont des activités critiques, leurs durées ne peuvent être modifiées sans modifier la durée totale du projet"

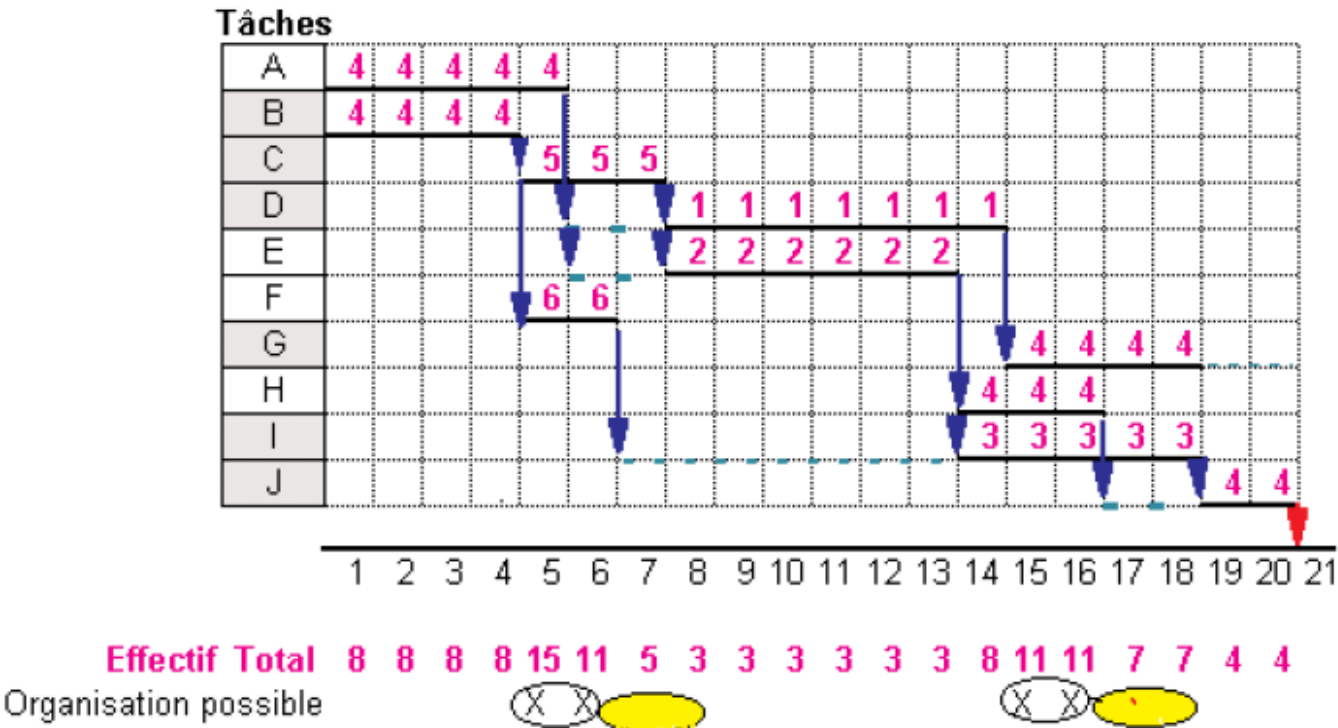
# 02 - Prioriser des "User Stories"

## Diagramme de Gantt



### Exemple

Faire apparaître la marge possible sur chaque tâche, suivre l'état d'avancement du projet, et corriger éventuellement.



## CHAPITRE 2

### PRIORISER DES "USER STORIES"

1. Itérations du projet
2. Diagramme de Gantt
3. **Diagramme de Pert**



## 02 - Prioriser des "User Stories"

### Diagramme de Pert



#### Définition

- Un diagramme PERT (Program Evaluation Review Technique) est une représentation graphique de l'échéancier d'un projet qui affiche toutes les tâches individuelles nécessaires à la réalisation du projet ;



#### Objectif

- trouver la meilleure organisation possible pour mener à bien un projet dans les meilleurs délais
- 
- En tant qu'outil de gestion de projet, le diagramme PERT est souvent préféré au diagramme de Gantt car il identifie les dépendances entre les tâches. Cependant, un diagramme PERT peut être plus difficile à interpréter.

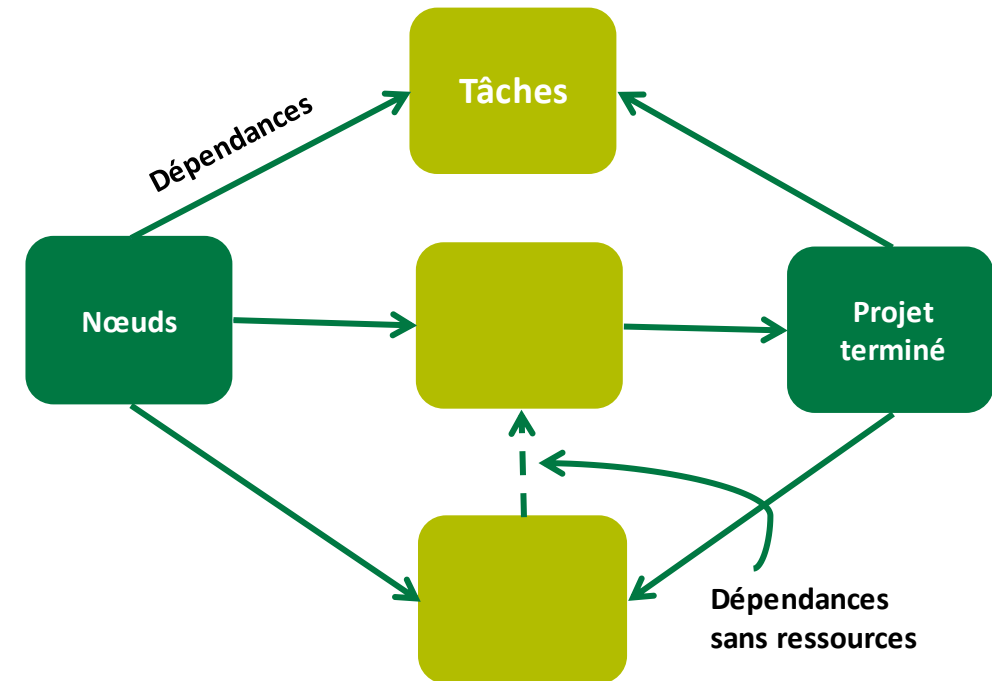
## 02 - Prioriser des "User Stories"

### Diagramme de Pert

#### Points clés

- Les diagrammes PERT ont été créés pour la première fois par le bureau des projets spéciaux de la marine américaine dans les années 1950 pour guider le projet de sous-marin nucléaire Polaris.
  - Un diagramme PERT utilise des cercles ou des rectangles appelés nœuds pour représenter les événements ou les jalons du projet. Ces nœuds sont reliés par des vecteurs, ou lignes, qui représentent les différentes tâches et leurs dépendances;
  - Un diagramme PERT permet aux managers d'évaluer le temps et les ressources nécessaires à la gestion d'un projet.
- Pour créer un diagramme de PERT, suivez les cinq étapes du cycle de vie du processus, depuis l'identification des tâches jusqu'à la gestion de l'achèvement du projet :
  - Étape 1** - identifier les tâches de projet
  - Étape 2** - définir les dépendances des tâches
  - Étape 3** - faire le lien entre les tâches de projet
  - Étape 4** - estimer la durée du projet
  - Étape 5** - gérer la progression des tâches

#### Éléments d'un diagramme de PERT :



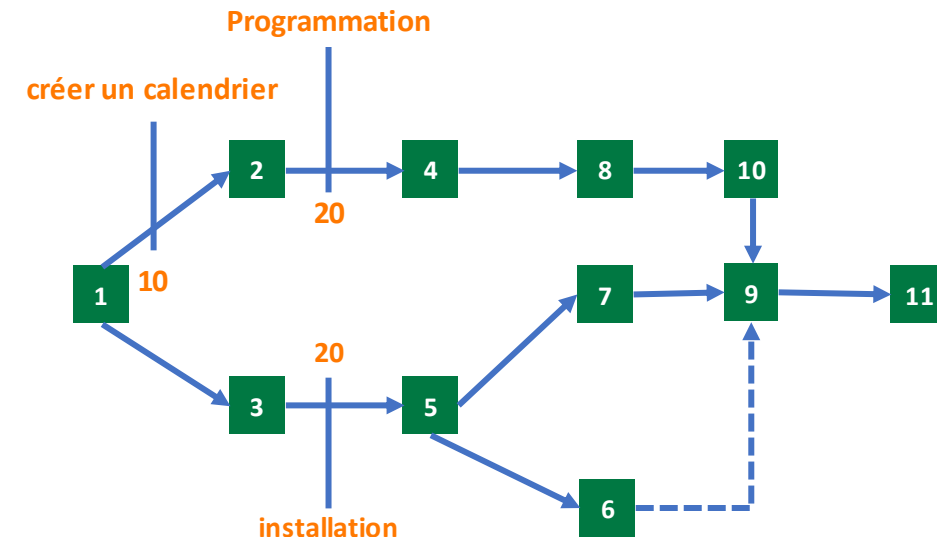
## 02 - Prioriser des "User Stories"

### Diagramme de Pert

### Fonctionnement d'un diagramme de PERT

- Un chef de projet crée un diagramme PERT afin d'analyser toutes les tâches d'un projet et d'estimer le temps nécessaire à la réalisation de chacune d'entre elles. Grâce à ces informations, le chef de projet peut estimer le temps minimum nécessaire à la réalisation de l'ensemble du projet
- Ces informations aident également le gestionnaire à élaborer le budget du projet et à déterminer les ressources nécessaires à sa réalisation
- Un diagramme PERT utilise des cercles ou des rectangles, appelés nœuds, pour représenter les événements ou les jalons du projet. Les nœuds sont reliés par des vecteurs ou des lignes qui représentent diverses tâches
- Les tâches dépendantes sont des éléments qui doivent être exécutés d'une manière spécifique. Par exemple, si une flèche est tracée de la tâche n° 1 à la tâche n° 2 sur un graphique PERT, la tâche n° 1 doit être achevée avant que le travail sur la tâche n° 2 ne commence
- Les éléments qui se trouvent au même stade de production mais sur des lignes de tâches différentes au sein d'un projet sont appelés tâches parallèles. Ils sont indépendants les uns des autres et se produisent en même temps

Un graphique PERT bien construit ressemble à ceci :



- Les rectangles numérotés sont des nœuds et représentent des événements ou des jalons
- Les flèches directionnelles représentent des tâches dépendantes qui doivent être accomplies de manière séquentielle
- Les flèches de direction divergentes (par exemple, 1-2 & 1-3) indiquent des tâches éventuellement simultanées
- Les lignes en pointillés indiquent des tâches dépendantes qui ne nécessitent pas de ressources

## 02 - Prioriser des "User Stories"

### Diagramme de Pert



#### Interprétation d'un diagramme de PERT

- Un graphique PERT est une représentation visuelle d'une série d'événements qui doivent se produire au cours de la durée de vie d'un projet. La direction des flèches indique le flux et la séquence des événements requis.
- Les lignes d'activité en pointillés représentent des activités fictives, c'est-à-dire des éléments qui se trouvent sur un autre chemin PERT. Des nombres et des temps alloués sont attribués et indiqués à l'intérieur de chaque vecteur.
- Ces graphiques ont leurs définitions et termes distincts, dont les plus importants anticipent le temps nécessaire à la finalisation d'un projet. "Le temps optimiste" fait référence à la durée la plus courte. "Le temps pessimiste" correspond à la durée la plus longue qui pourrait être nécessaire. Le "temps le plus probable" indique une estimation raisonnable du meilleur scénario, tandis que le "temps prévu" tient compte des problèmes et des obstacles.

## 02 - Prioriser des "User Stories"

### Diagramme de Pert



#### Avantages et inconvénients des diagrammes PERT

- Le diagramme PERT est similaire à l'analyse du chemin critique (CPA), une autre méthode utilisée par les chefs de projet pour planifier les tâches d'un projet. La principale différence entre les deux est qu'un diagramme PERT utilise différents délais et termes de probabilité pour estimer chaque étape du projet.
- En tant qu'outil de gestion de projet, les diagrammes PERT présentent des avantages et des inconvénients distincts.

AVANTAGES	INCONVÉNIENTS
<ul style="list-style-type: none"><li>• Un diagramme PERT permet à un gestionnaire d'évaluer le temps et les ressources nécessaires à la réalisation d'un projet. Il permet également au gestionnaire de suivre les actifs requis à n'importe quelle étape de la production au cours du projet</li></ul>	<ul style="list-style-type: none"><li>• Les informations qui entrent dans un graphique PERT peuvent être très subjectives. Elles peuvent inclure des données non fiables ou des estimations déraisonnables en matière de coûts ou de délais</li></ul>
<ul style="list-style-type: none"><li>• L'analyse PERT intègre des données et des informations fournies par un certain nombre de départements. Cette combinaison d'informations encourage la responsabilité des départements et identifie toutes les parties responsables dans l'ensemble de l'organisation</li></ul>	<ul style="list-style-type: none"><li>• Les diagrammes PERT sont axés sur les délais et peuvent ne pas communiquer pleinement la situation financière d'un projet</li></ul>
<ul style="list-style-type: none"><li>• Il améliore également la communication au cours du projet et permet à l'organisation de s'engager dans des projets qui correspondent à son positionnement stratégique</li></ul>	
<ul style="list-style-type: none"><li>• Enfin, les diagrammes PERT sont des données utiles pour les analyses de simulation. Comprendre les possibilités concernant le flux des ressources et des étapes du projet permet à la direction de réaliser le parcours le plus efficace et le plus utile du projet</li></ul>	<ul style="list-style-type: none"><li>• La création d'un diagramme PERT demande beaucoup de travail, et le maintien et la mise à jour des informations nécessitent du temps et des ressources supplémentaires. Un examen continu des informations fournies, ainsi que du positionnement prospectif du projet, est nécessaire pour qu'un diagramme PERT soit utile</li></ul>



## 02 - Prioriser des "User Stories"

### Diagramme de Pert



### Elaboration d'un diagramme de PERT : Matrice des antériorités

- Pour établir la matrice des antériorités, nous allons créer un tableau à deux entrées identiques : la liste des tâches, suivie d'un tableau comportant des colonnes de niveaux.
- L'entrée horizontale correspond aux tâches antérieures. Le remplissage de la matrice se fait de la façon suivante :
  - A l'aide de la liste des tâches et de leurs antériorités compléter ligne par ligne en plaçant des "1" dans les colonnes où les tâches sont antérieures ;
  - Supposons que la tâche E est antérieure à la tâche B nous allons donc placer un "1" à l'intersection de la ligne B et de la colonne E ;
  - Supposons que la tâche F est directement postérieure à D et à E nous allons donc placer un "1" à l'intersection de la ligne F et de la colonne D et un "1" à l'intersection de la ligne F et de la colonne E. Idem pour les autres lignes.

		Il faut avoir terminé										niveaux				
		A	B	C	D	E	F	G	H	I	J	1	2	3	4	5
POUR FAIRE	A															
	B					1										
	C															
	D															
	E															
	F				1	1										
	G															
	H															
	I															
	J															

## 02 - Prioriser des "User Stories"

### Diagramme de Pert



### Elaboration d'un diagramme de PERT

- Avec une liste des tâches et de leur antériorités par exemple :

Pour faire	A	B	C	D	E	F	G	H	I	J
Il faut avoir fait		E	E	A	A	D, E	B	G	J, C, H F	A

- cela donne :

		Il faut avoir terminé										niveaux				
		A	B	C	D	E	F	G	H	I	J	1	2	3	4	5
POUR FAIRE	A															
	B					1										
	C					1										
	D	1														
	E	1														
	F				1	1										
	G		1													
	H							1								
	I			1			1		1		1					
	J	1														



### Remarque

- la tâche A n'a pas de tâches antérieures elle ne comporte pas de "1" sur sa ligne.

- Nous allons réaliser un ensemble de deux étapes que nous allons répéter n fois, n étant le nombre de niveaux.

Elaboration d'un diagramme de PERT

Première étape de l'ensemble

- Reporter la somme par ligne des "1" dans la colonne de niveau i (i variant de 1 à n).

Deuxième étape de l'ensemble

- Déterminer quelles sont les tâches de niveau i : tout simplement ce sont les tâches pour lesquelles la somme trouvée précédemment est nulle.
- Éliminer les "1" de chaque colonne de niveau i.

		Il faut avoir terminé										niveaux				
		A	B	C	D	E	F	G	H	I	J	1	2	3	4	5
POUR FAIRE	A											0				
	B					1						1				
	C					1						1				
	D	1										1				
	E	1										1				
	F				1	1						2				
	G		1									1				
	H							1				2				
	I			1			1		1		1	4				
	J	1										1				

- Nous avons réalisé la première étape ci-dessus en reportant la somme par ligne des "1" dans la colonne de niveau 1(i étant =1). pour la deuxième.

Elaboration d'un diagramme de PERT

- la tâche de niveau 1 est donc la tâche A car la somme des "1" de la ligne A est nulle.
- Nous allons donc éliminer tous les 1 de la colonne A, ce qui donne le tableau ci-dessous :

		Il faut avoir terminé										niveaux				
		A	B	C	D	E	F	G	H	I	J	1	2	3	4	5
POUR FAIRE	A											0				
	B					1						1				
	C					1						1				
	D											1				
	E											1				
	F				1	1						2				
	G		1									1				
	H							1				2				
	I			1			1		1		1	4				
	J											1				

Elaboration d'un diagramme de PERT

- Nous avons recommencé la première étape ci-dessous en prenant i = 2 et en reportant la somme par ligne des "1" dans la colonne de niveau 2.

		Il faut avoir terminé										niveaux				
		A	B	C	D	E	F	G	H	I	J	1	2	3	4	5
POUR FAIRE	A											0				
	B					1						1	1			
	C					1						1	1			
	D											1	0			
	E											1	0			
	F				1	1						2	2			
	G		1									1	1			
	H							1				2	1			
	I			1			1		1		1	4	4			
	J											1	0			

Elaboration d'un diagramme de PERT

- Pour la deuxième étape les tâches de niveau 2 sont donc les tâches D,E et J car la somme des "1" de leur ligne est nulle.

		Il faut avoir terminé										niveaux				
		A	B	C	D	E	F	G	H	I	J	1	2	3	4	5
POUR FAIRE	A											0				
	B					1						1	1			
	C					1						1	1			
	D											1	0			
	E											1	0			
	F				1	1						2	2			
	G		1									1	1			
	H							1				2	1			
	I			1			1	1			1	4	3			
	J											1	0			

- Nous allons donc éliminer tous les "1" des colonnes D,E et J.

## 02 - Prioriser des "User Stories"

### Diagramme de Pert



### Elaboration d'un diagramme de PERT

- ce qui donne :

		Il faut avoir terminé										niveaux				
		A	B	C	D	E	F	G	H	I	J	1	2	3	4	5
POUR FAIRE	A											0				
	B											1	1	0		
	C											1	1	0		
	D											1	0			
	E											1	0			
	F											2	2	0		
	G		1									1	1	1		
	H							1				2	1	1		
	I			1			1	1				4	3	3		
	J											1	0			

- En répétant ces deux étapes :
  - pour le niveau 3 on trouve B,C et F
  - pour le niveau 4 on trouve G
  - pour le niveau 5 on trouve H
  - pour le niveau 6 on trouve I

- ce qui nous donne une table des niveaux :

niveaux	1	2	3	4	5	6
tâches	A	D, E, J	B, C, F	G	H	I

## 02 - Prioriser des "User Stories"

### Diagramme de Pert



### Établir le graphe sagittal

- Prenons l'exemple précédent pour lequel la répartition des tâches en niveaux est :

niveaux	1	2	3	4	5	6
tâches	A	D, E, J	B, C, F	G	H	I

- et la liste des tâches et de leurs antériorités :

Pour faire	A	B	C	D	E	F	G	H	I	J
Il faut avoir fait		E	E	A	A	D, E	B	G	J, C, H F	A



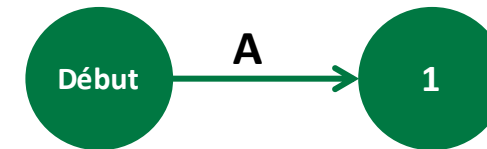
## 02 - Prioriser des "User Stories"

### Diagramme de Pert

### Établir le graphe sagittal

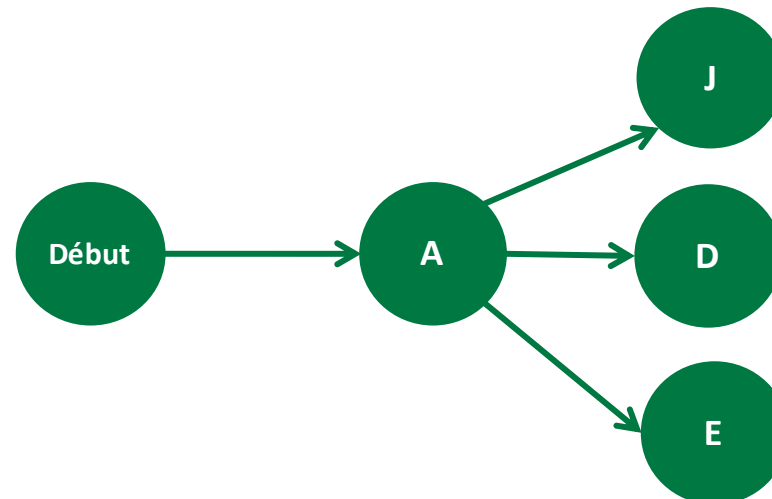
#### Niveau 1

- Nous allons débiter le graphe avec les tâches de premier niveau dans notre cas il s'agit de la tâche A :



#### Niveau 2

- Les trois tâches D,E et J sont de niveau deux et ont pour antécédent la tâche A, elles vont donc débiter en parallèle juste après A :



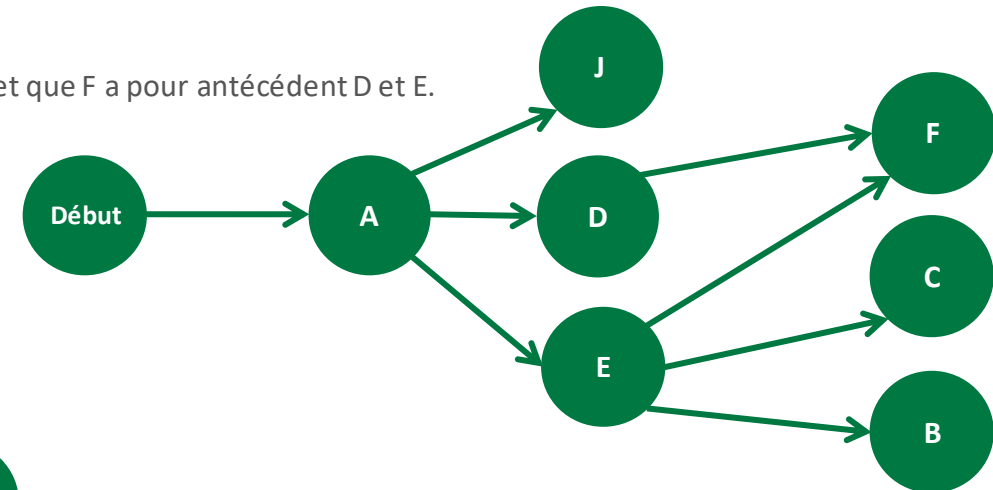
## 02 - Prioriser des "User Stories"

### Diagramme de Pert

#### Établir le graphe sagittal

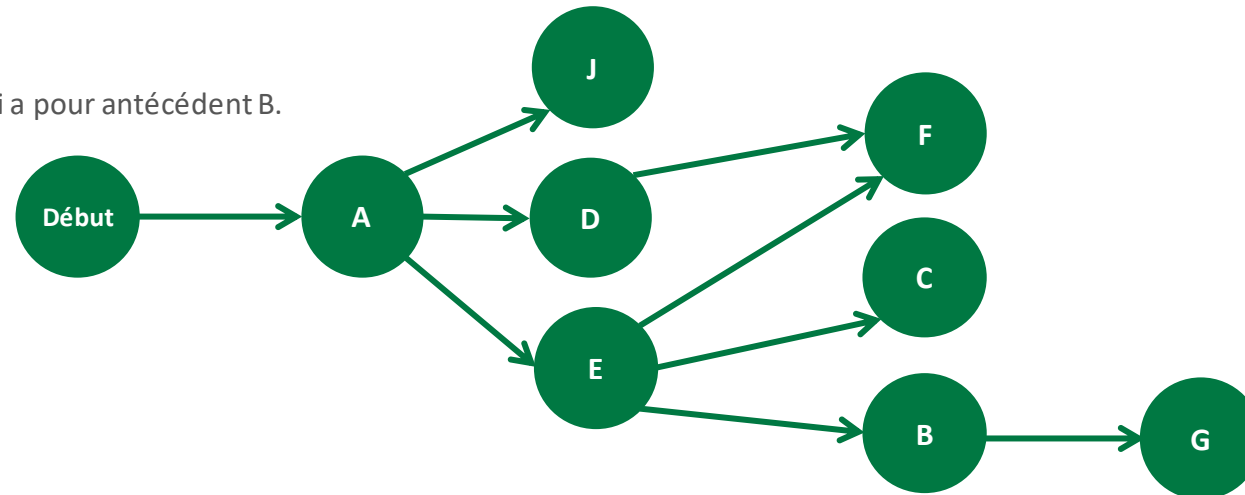
##### Niveau 3

- Les trois tâches B,C et F sont de niveau trois sachant que B et C ont pour antécédent E et que F a pour antécédent D et E.



##### Niveau 4

- Nous avons simplement la tâche G qui a pour antécédent B.



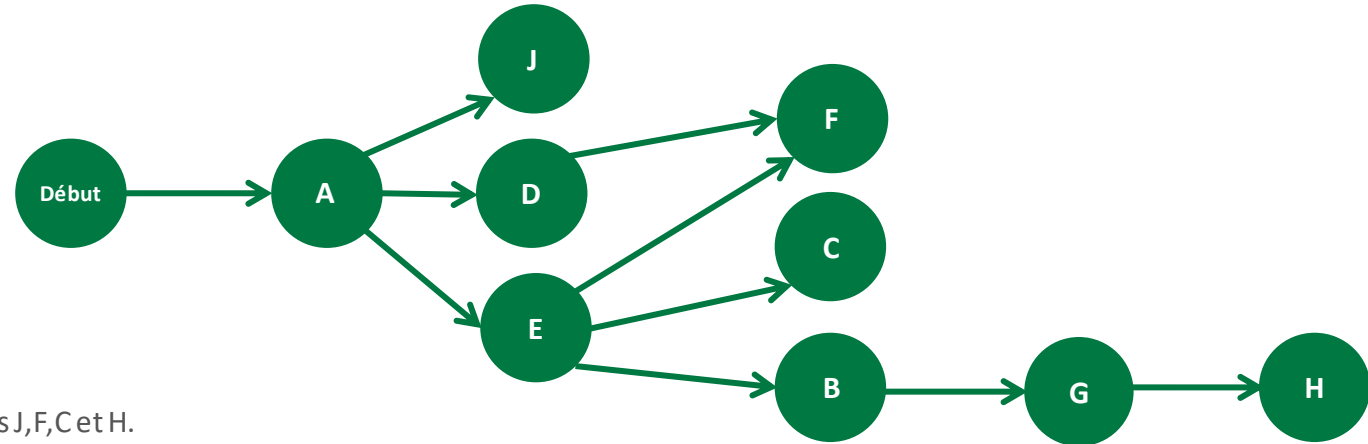
## 02 - Prioriser des "User Stories"

### Diagramme de Pert

#### Établir le graphe sagittal

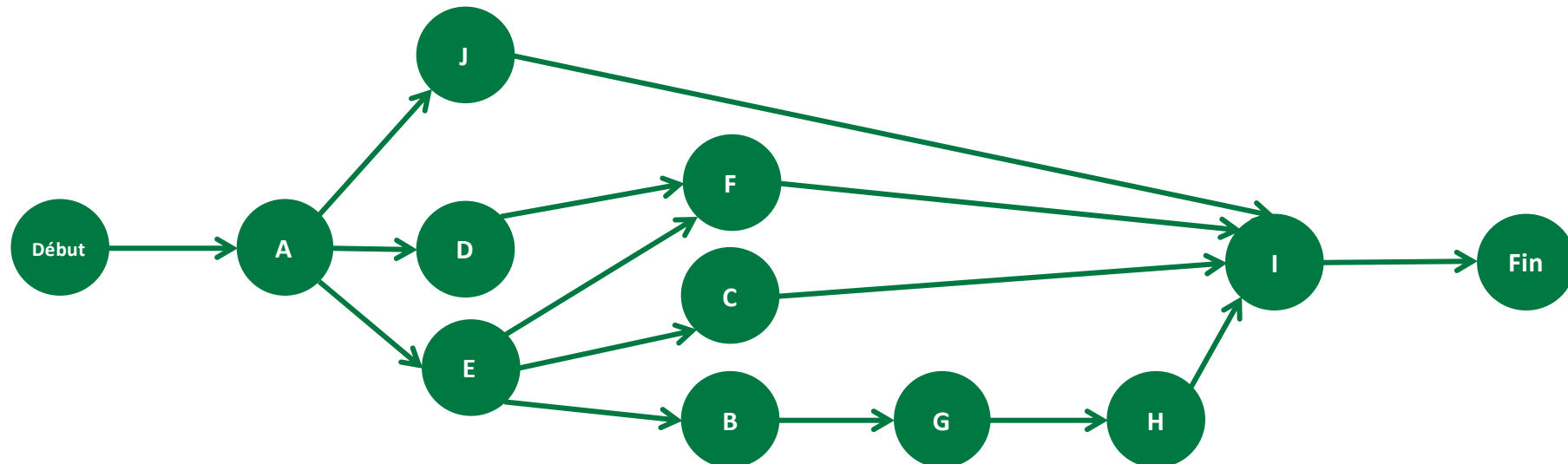
##### Niveau 5

- Nous avons simplement la tâche H qui a pour antécédent G.



##### Niveau 6

- Nous avons une seule tâche : I mais elle a quatre antécédents J, F, C et H.

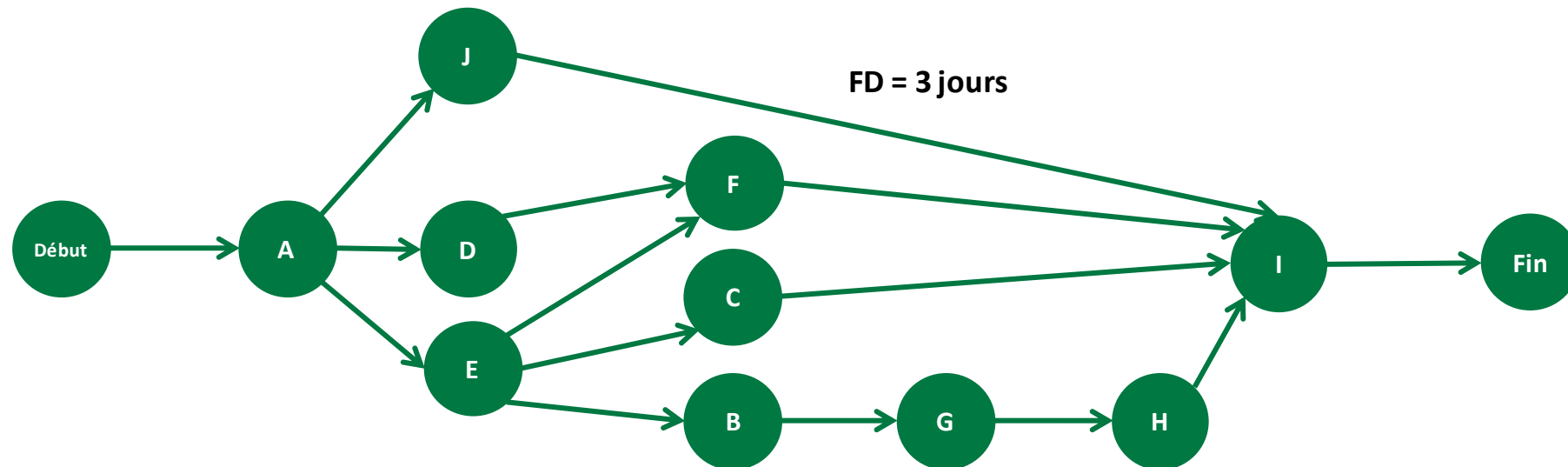


## 02 - Prioriser des "User Stories"

### Diagramme de Pert

#### Établir le graphe sagittal

- On peut noter sur ce graphe les contraintes qui peuvent exister entre deux tâches. Supposons que pour réaliser I il faut attendre deux jours après la fin de J, il s'agit d'une contrainte de type fin-début=3 entre la fin de J et le début de I.
- On peut la représenter sur le graphe :





## PARTIE 2

# APPLIQUER LES MÉTHODES AGILE

Dans ce module, vous allez :

- Appliquer la méthode SCRUM
- Appliquer la méthode SAFE



13 heures

# CHAPITRE 1

## APPLIQUER LA MÉTHODE SCRUM

### Ce que vous allez apprendre dans ce chapitre :

- Distinguer les rôles en agile : le Product Owner, le Scrum Master, l'équipe de développement, le testeur, le tracker, le coach
- Vous appuyer sur des périodes sprints ou les itérations du projet
- Organiser la planification journalière : daily scrum ou standing meeting (objectif, l'organisation)
- Vous initier au planning poker



08 heures



# CHAPITRE 1

## APPLIQUER LA MÉTHODE SCRUM

### 1. Gestion de projet agile

- 2. Processus Scrum
- 3. Cadre Scrum
- 4. Planification de la release
- 5. Planning de poker



# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



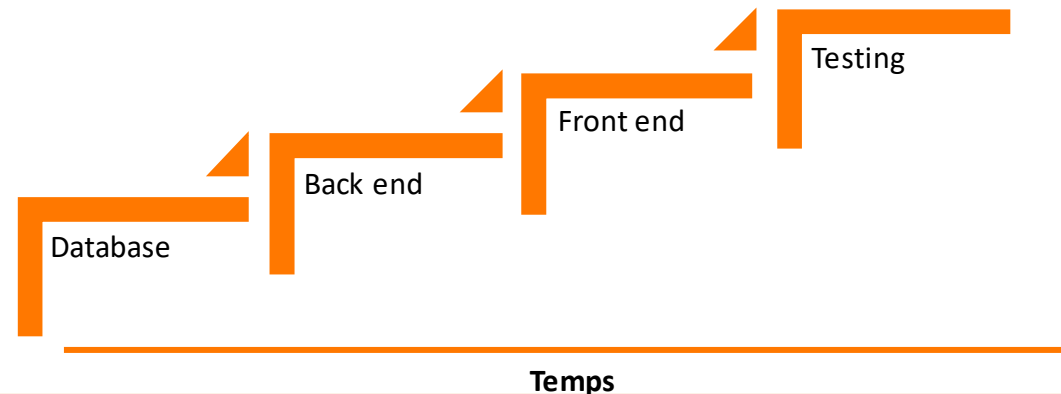
### Contexte

- La **gestion de projet agile** est une **approche itérative** de la réalisation d'un projet, qui met l'accent sur des versions continues intégrant les commentaires des clients à chaque itération. La possibilité de s'adapter au cours de chaque itération favorise la rapidité et l'adaptabilité. Cette approche diffère de l'approche linéaire de la gestion de projet en cascade, qui suit un chemin déterminé avec des écarts limités
- Les **clients et les entreprises d'aujourd'hui exigeant des réponses et des changements rapides**, la méthode agile offre la **souplesse** nécessaire pour s'adapter et itérer au cours du processus de développement. La gestion de projet agile est également une pierre angulaire des pratiques DevOps, où les équipes de développeurs et d'exploitation travaillent en collaboration
- Les équipes logicielles qui adoptent les méthodologies de gestion de projet agile augmentent leur vitesse de développement, étendent la collaboration et favorisent la capacité à mieux répondre aux tendances du marché
- Voici tout ce que vous devez savoir pour vous lancer ou affiner vos pratiques de gestion de projet agile



### Gestion de projet en cascade

- **L'approche de gestion de projet en cascade** implique une séquence d'exécution clairement définie avec des phases de projet qui n'avancent pas tant qu'une phase n'a pas reçu l'approbation finale. Une fois une phase achevée, il peut être difficile et coûteux de revenir sur une étape précédente. Les équipes agiles peuvent suivre une séquence similaire, mais en procédant par petits incréments avec des boucles de rétroaction régulières
- **L'approche de gestion de projet en cascade** suit une formule linéaire et séquentielle. Elle fonctionne bien pour les travaux dont les processus sont prévisibles et récurrents, mais elle peut laisser les équipes de développeurs sur la touche et les empêcher de s'adapter plus rapidement que leurs concurrents
- Un seul dépassement de délai ou une seule modification de la portée d'un projet en cascade peut avoir un impact considérable sur les versions suivantes. De plus, lorsqu'une équipe est entièrement concentrée sur la phase de travail suivante, la résolution de la dette technique ou la correction des bogues peut s'avérer pénible si l'équipe est entièrement affectée au travail sur les nouvelles fonctionnalités et qu'elle est toujours en train de passer à l'étape suivante
- Vous trouverez ci-dessous une illustration d'un projet standard en cascade avec des blocs de temps rigoureusement segmentés. Cette situation crée une mentalité de "l'utiliser ou le perdre" qui encourage les développeurs, les propriétaires de produits et les parties prenantes à demander autant de temps que possible dans chaque fenêtre de temps, car il n'y aura peut-être pas d'occasion d'itérer à l'avenir. En général, les équipes qui utilisent la méthode de la cascade essaient de contrôler la dérive de l'étendue du projet par le biais du "contrôle des changements", où tout le monde est d'accord pour ne pas modifier le contrat initial



# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Gestion de projet en cascade

Le modèle en cascade peut exacerber certains des défis connus de la création de produits :

- **Les blocages et la gestion des dépendances** : Les styles traditionnels de gestion de projet créent souvent des "chemins critiques", où le projet ne peut pas avancer tant qu'un problème bloquant n'est pas résolu ;
- **Difficulté à obtenir le retour des utilisateurs et la validation du produit** : Pour couronner le tout, le client final ne peut pas interagir avec le produit avant qu'il ne soit entièrement terminé. Ainsi, des problèmes importants dans la conception et le code du produit ne sont pas découverts avant la mise en production.

Les avantages et les inconvénients de l'approche en cascade :

AVANTAGES	INCONVÉNIENTS
<ul style="list-style-type: none"><li>• Nécessite moins de coordination en raison de phases clairement définies et de processus séquentiels</li></ul>	<ul style="list-style-type: none"><li>• Plus difficile de diviser et de partager le travail en raison de séquences de phases plus strictes ; les équipes sont plus spécialisées</li></ul>
<ul style="list-style-type: none"><li>• Une phase de projet claire permet de définir clairement les dépendances des travaux</li></ul>	<ul style="list-style-type: none"><li>• Risque de perte de temps en raison de retards et de contretemps lors des transitions de phase</li></ul>
<ul style="list-style-type: none"><li>• Le coût du projet peut être estimé après la définition des exigences</li></ul>	<ul style="list-style-type: none"><li>• Besoins d'embauche supplémentaires pour constituer des équipes de phase spécialisées alors que la méthode agile encourage la composition d'équipes plus transversales</li></ul>
<ul style="list-style-type: none"><li>• Meilleure attention portée à la documentation des conceptions et des exigences</li></ul>	<ul style="list-style-type: none"><li>• Frais de communication supplémentaires pendant le transfert entre les transitions de phase</li></ul>
<ul style="list-style-type: none"><li>• La phase de conception est plus méthodique et structurée avant l'écriture de tout logiciel</li></ul>	<ul style="list-style-type: none"><li>• L'appropriation du produit et l'engagement peuvent ne pas être aussi forts par rapport à la méthode agile puisque l'accent est mis sur la phase en cours</li></ul>

# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Agile et cascade

- La méthode agile a d'abord été adoptée par les équipes de développement de logiciels, qui ont abandonné l'approche traditionnelle et séquentielle de la cascade au profit d'une méthode permettant un retour d'information et des ajustements constants tout au long du cycle de développement.
- La gestion de projet agile adopte une approche itérative du développement en créant plusieurs étapes incrémentielles avec des intervalles de retour réguliers. Cette approche favorise l'adaptabilité puisque l'équipe peut s'adapter tout au long du processus de développement du produit, au lieu d'être confinée dans un parcours linéaire. Elle permet également de publier régulièrement des versions à fort impact, ce qui permet aux équipes de remporter une série de victoires au fil du temps.
- Les versions itératives offrent de multiples possibilités à une équipe pour :
  - s'adapter à des circonstances changeantes, qu'il s'agisse d'exigences nouvellement découvertes ou d'un travail bloqué
  - recueillir les commentaires des parties prenantes pendant le processus et itérer de manière réactive sans le stress d'une date limite de livraison finale
  - établir des relations et des connexions entre les rôles, ce qui facilite les contacts et la communication efficace
- Agile permet aux équipes d'être plus résistantes aux changements qui surviennent inévitablement au cours d'un projet.

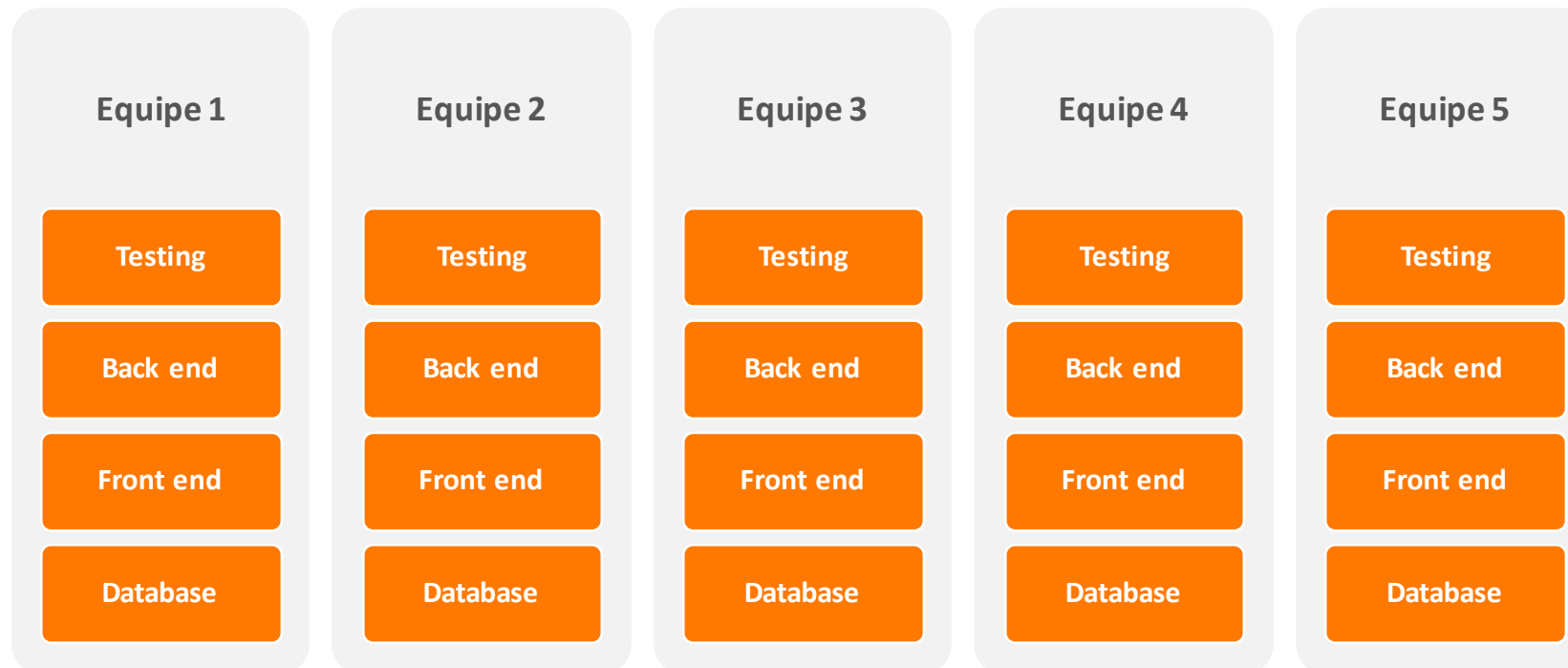
# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Agile et cascade

- Le partage des compétences au sein de l'équipe logicielle constitue un avantage encore plus important. Le chevauchement des compétences de l'équipe ajoute de la flexibilité au travail dans toutes les parties de la base de code de l'équipe. Ainsi, le travail et le temps ne sont pas perdus si la direction du projet change.



# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Principes agile

- Un projet agile est segmenté en plusieurs étapes incrémentielles qui comprennent des intervalles de rétroaction réguliers
- Une exigence de projet est segmentée en plus petits éléments, qui sont ensuite classés par ordre d'importance
- Favorise la collaboration, notamment avec le client
- S'ajuste à intervalles réguliers pour s'assurer que les besoins du client sont satisfaits
- Intègre la planification à l'exécution, ce qui permet à une équipe de répondre efficacement à l'évolution des exigences

Les avantages et les inconvénients d'une gestion de projet agile :

AVANTAGES	INCONVÉNIENTS
<ul style="list-style-type: none"><li>• Des cycles de retour d'information plus rapides</li><li>• Identification précoce des problèmes</li><li>• Meilleur potentiel de satisfaction du client</li><li>• Le délai de mise sur le marché est considérablement amélioré</li><li>• Meilleure visibilité / responsabilité</li><li>• Des équipes dédiées améliorent la productivité au fil du temps</li><li>• Priorités flexibles axées sur la création de valeur</li></ul>	<ul style="list-style-type: none"><li>• Le chemin critique et les dépendances inter-projets peuvent ne pas être clairement définis comme dans le cas de la cascade</li><li>• Il y a un coût de courbe d'apprentissage organisationnel</li><li>• Une véritable exécution agile avec un pipeline de déploiement continu a beaucoup de dépendances techniques et de coûts d'ingénierie à établir</li></ul>

# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Éléments à prendre en compte lors du passage à la méthode agile

- Passer à la méthode agile peut être un défi, en particulier lorsqu'une équipe ou une organisation est ancrée dans une approche de gestion de projet plus traditionnelle. Le passage aux pratiques agiles peut nécessiter un certain nombre de changements de processus, en particulier lors de l'adoption d'une approche DevOps, où les équipes de développement et d'exploitation travaillent en étroite collaboration pour développer et maintenir les logiciels. Lors de l'adoption des principes agiles, une équipe et les parties prenantes doivent adhérer à deux concepts importants :
  - Le propriétaire du produit (product owner's) se concentre sur l'optimisation de la valeur de la production de l'équipe. L'équipe compte sur le propriétaire du produit pour donner la priorité au travail le plus important
  - L'équipe de développeurs (développement équipe) ne peut accepter que le travail pour lequel elle a la capacité. Le propriétaire du produit n'impose pas de travail à l'équipe et ne lui impose pas de délais arbitraires. L'équipe de développement retire du travail du backlog du programme au fur et à mesure qu'elle peut en accepter de nouveaux
  - Explorons les mécanismes que les programmes agiles utilisent pour organiser, exécuter et structurer le travail de manière itérative.
- Explorons les mécanismes que les programmes agiles utilisent pour organiser, exécuter et structurer le travail de manière itérative.

# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Feuille de route (Roadmaps)

- Une feuille de route de produit décrit comment un produit ou une solution se développe au fil du temps. Dans le cadre du développement agile, une feuille de route fournit un contexte important qui permet aux équipes d'atteindre à la fois des objectifs progressifs et des objectifs à l'échelle du projet. Les feuilles de route sont composées d'initiatives, qui sont de grands domaines de fonctionnalité, et comprennent des échéances qui indiquent quand une fonctionnalité sera disponible. Au fur et à mesure que le travail avance et que les équipes en apprennent davantage, il est accepté que la feuille de route change pour refléter ces nouvelles informations, de manière subtile ou générale. L'objectif est de maintenir la feuille de route axée sur les conditions actuelles qui ont un impact sur le projet et les objectifs à long terme afin de travailler efficacement avec les parties prenantes et de répondre au paysage concurrentiel.
- Dans les diapositives suivants des exemples de deux feuilles de route :
  - feuille de route avec les plans de navigation et les détails du projet pour une gestion efficace du processus de développement d'une plateforme;
  - feuille de route de haut niveau avec un aperçu du projet, des équipes et des activités pour un programme de commerce électronique.

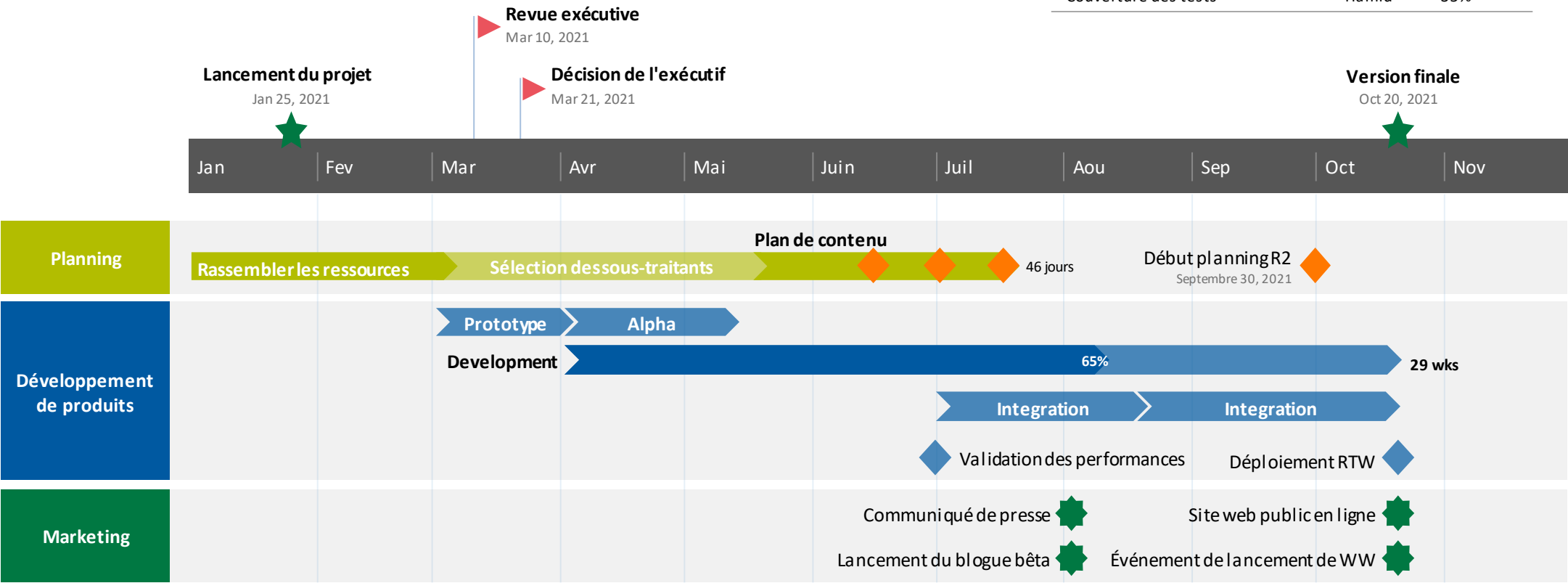
# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Feuille de route de la plate-forme

Flux de développement	Owner	Objectif
Couverture de l'API Foundation	Mido	90%
Architecture de l'infrastructure	Ali	100%
Couverture des tests	Hamid	35%



2021

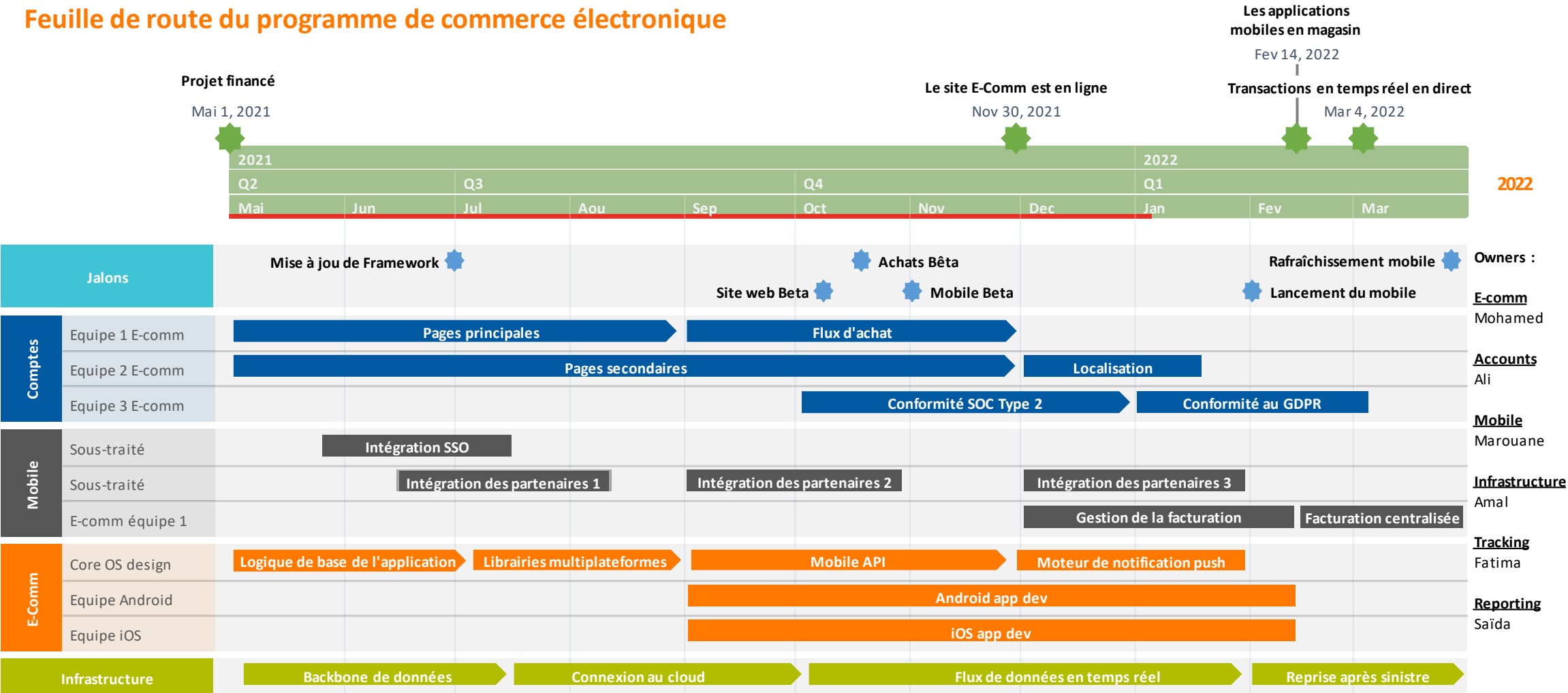


# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Feuille de route du programme de commerce électronique



# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Exigences

- Chaque initiative de la feuille de route se décompose en un ensemble d'exigences. Les exigences agiles sont des descriptions légères de la fonctionnalité requise, plutôt que les documents de 100 pages associés aux projets traditionnels. Elles évoluent au fil du temps et s'appuient sur la compréhension partagée par l'équipe du client et du produit souhaité. Les exigences agiles restent légères pendant que tous les membres de l'équipe développent une compréhension commune par le biais d'une conversation et d'une collaboration continues. Ce n'est que lorsque la mise en œuvre est sur le point de commencer qu'elles sont étoffées avec tous les détails.

### Backlog

- Le backlog fixe les priorités du programme agile. L'équipe inclut tous les éléments de travail dans le backlog : nouvelles fonctionnalités, bogues, améliorations, tâches techniques ou architecturales, etc. Le propriétaire du produit priorise le travail sur le backlog pour l'équipe d'ingénierie. L'équipe de développement utilise ensuite le backlog priorisé comme source unique de vérité sur le travail à faire.

# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Métriques agiles

- Les équipes agiles s'épanouissent grâce aux mesures. Les limites des travaux en cours (WIP-Work in progress) permettent à l'équipe, et à l'entreprise, de se concentrer sur la livraison des travaux les plus prioritaires. Des graphiques tels que les burndown charts et les cartes de contrôle aident l'équipe à prévoir sa cadence de livraison, et les diagrammes de flux continu permettent d'identifier les goulots d'étranglement. Ces mesures et artefacts permettent à chacun de se concentrer sur les grands objectifs et renforcent la confiance dans la capacité de l'équipe à fournir le travail futur.



# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### L'agilité repose sur la confiance

- Les processus agiles ne peuvent pas fonctionner sans un haut niveau de confiance entre les membres de l'équipe et donc créer la confiance. Il faut de la franchise pour avoir des conversations difficiles sur ce qui est bon pour le programme et le produit. Comme les conversations ont lieu à intervalles réguliers, les idées et les préoccupations sont régulièrement exprimées. Cela signifie que les membres de l'équipe doivent avoir confiance dans la capacité (et la volonté) de chacun à mettre en œuvre les décisions prises au cours de ces conversations.

### Synthèse

- La gestion de projet agile est une approche innovante, non seulement pour les projets logiciels, mais aussi pour les projets de toutes sortes. En offrant la souplesse nécessaire pour réagir aux changements pendant le cycle de développement, la méthode agile permet aux équipes de livrer des produits de meilleure qualité qui répondent aux besoins des clients. La méthode agile responsabilise les équipes, renforce l'obligation de rendre des comptes et encourage l'innovation tout en favorisant l'amélioration continue. La méthode agile donne la possibilité de réagir au changement sans dérailler.

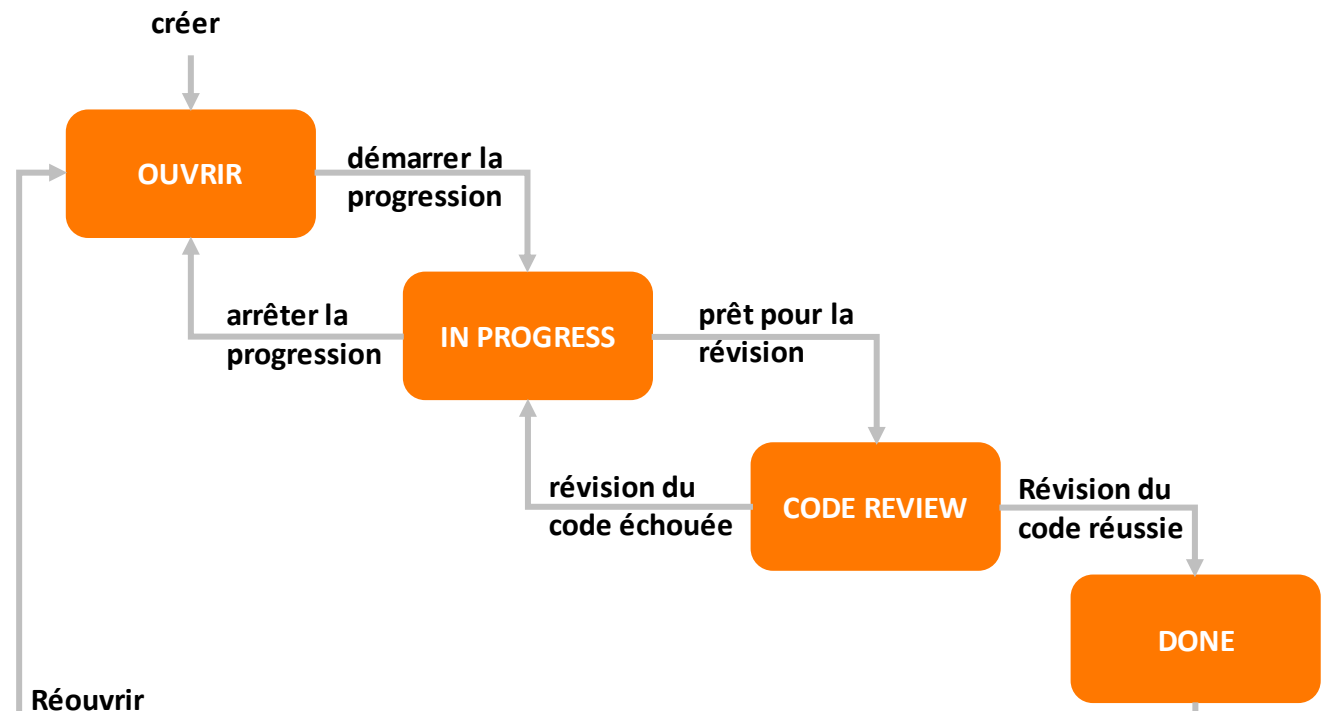
# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Flux de travail

- Chaque équipe logicielle a un processus qu'elle utilise pour accomplir son travail. Normaliser ce processus, c'est-à-dire l'établir en tant que workflow, le rend clairement structuré et reproductible, ce qui le rend évolutif.
- Adopter une approche itérative de la gestion des flux de travail aidera à atteindre les objectifs plus rapidement.
- Les états de base recommandée du flux de travail :



- Dans un outil de suivi des bogues, ces statuts passent de l'un à l'autre en utilisant des transitions qui structurent le flux de travail.

# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Stories, epics, et initiatives

- Imaginons qu'une entreprise souhaite lancer une satellite dans l'espace. Pour y parvenir, elle doit structurer le travail : en des objectifs plus grands aux plus petits détails. Elle doit être capable de réagir au changement, de rendre compte des progrès et de respecter un plan. Les **epics**, les **stories** et les **initiatives** sont précisément les outils dont elle aura besoin pour y parvenir
- **Stories, epics et initiatives :**
  - Les **Stories**, également appelées "**user stories**", sont de courtes exigences ou demandes rédigées du point de vue d'un utilisateur final
  - Les **epics** sont de grands ensembles de travail qui peuvent être décomposés en un certain nombre de tâches plus petites (appelées stories)
  - Les **initiatives** sont des collections d'epics qui tendent vers un objectif commun.

Initiative

Epic

Epic

Story

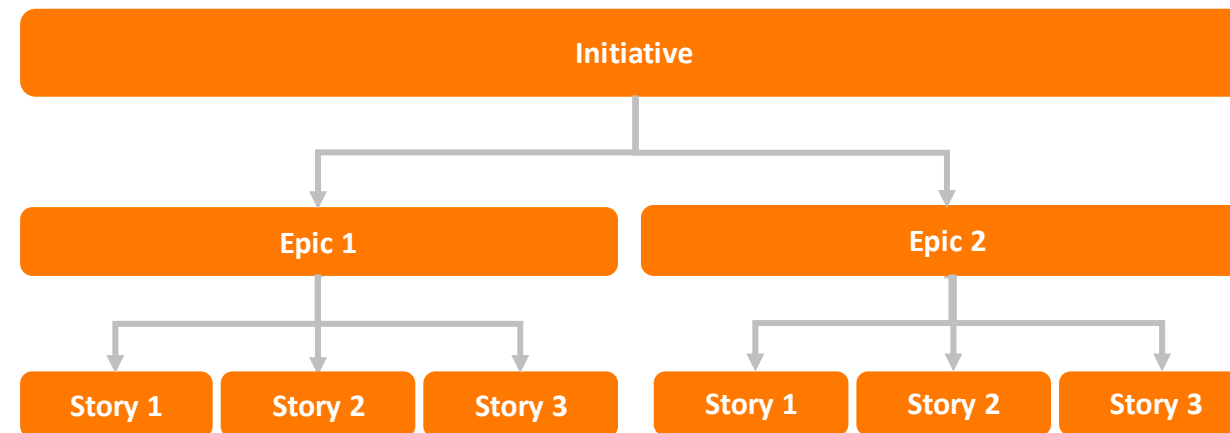
Story

### Stories vs Epics

- Les **stories** et les **epics** en agile sont similaires aux **stories** et aux **epics** dans le cinéma ou la littérature. Une **story** est un récit simple ; une série d'histoires liées et interdépendantes constitue une **epic**. Il en va de même pour la gestion de projet, où l'achèvement des stories liées conduit à l'achèvement d'une **epic**. Les stories racontent l'arc du travail accompli tandis que l'**epic** partage une vue de haut niveau de l'objectif unificateur.
- Dans une équipe agile, les **stories** sont quelque chose que l'équipe peut s'engager à terminer dans un sprint d'une ou deux semaines. Souvent, les développeurs travaillent sur des dizaines de **stories** par mois. Les **epics**, en revanche, sont peu nombreux et prennent plus de temps à réaliser. Les équipes travaillent souvent sur deux ou trois **epics** par trimestre.
- Si l'entreprise lançait des satellites dans l'espace et souhaitait améliorer le service de streaming pour ses lancements, elle pourrait structurer ses stories comme celles ci-dessous :
  - Les utilisateurs d'iPhone doivent avoir accès à une vue verticale du flux en direct lorsqu'ils utilisent l'application mobile ;
  - Les utilisateurs d'ordinateurs de bureau ont besoin d'un bouton "Afficher en plein écran" dans le coin inférieur droit du lecteur vidéo ;
  - Les utilisateurs d'Android doivent être reliés à Play store.
- Les **stories** ci-dessus sont toutes liées, et pourraient toutes être considérées comme des tâches individuelles qui mènent à l'achèvement d'un travail plus important (une **epic**). Dans ce cas, l'**epic** pourrait être "Améliorer le service de streaming pour le lancement du premier trimestre".

### Epic vs Initiative

- De la même manière que les **epics** sont constitués de **stories**, les **initiatives** sont constituées d'**epics**. Les initiatives offrent un autre niveau d'organisation au-dessus des **epics**. Dans de nombreux cas, une **initiative** compile les **epics** de plusieurs équipes pour atteindre un objectif beaucoup plus large et plus important que les **epics** elles-mêmes. Alors qu'une **epic** peut être réalisée en un mois ou un trimestre, les **initiatives** sont souvent réalisées en plusieurs trimestres, voire une année.



#### Exemple :

- Imaginons que votre l'entreprise de satellites souhaite réduire le coût par lancement de 5 % cette année. Il s'agit là d'une initiative idéale, car aucun epics ne pourrait à lui seul atteindre un tel objectif. Dans le cadre de cette initiative, il y aurait des epics tels que "Réduire la consommation de carburant en phase de lancement de 1 %", "Augmenter le nombre de lancements par trimestre de 3 à 4" et "Baisser tous les thermostats de 71 à 69 degrés #Dadmode".



# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### User stories

- Une **user story** est la plus petite unité de travail dans un cadre agile. C'est un objectif final, pas une fonctionnalité, exprimé du point de vue de l'utilisateur du logiciel.
- Une **user story** est une explication informelle et générale d'une fonctionnalité logicielle écrite du point de vue de l'utilisateur final ou du client.
- L'objectif d'une user story est d'expliquer comment un travail va apporter une valeur particulière au client. Notez que les "dients" ne doivent pas nécessairement être des utilisateurs finaux externes au sens traditionnel du terme, ils peuvent également être des clients internes ou des collègues au sein de votre organisation qui dépendent de votre équipe.
- Les **user stories** sont des phrases en langage simple qui décrivent le résultat souhaité. Elles n'entrent pas dans les détails. Les exigences sont ajoutées plus tard, après accord de l'équipe.
- Les stories s'intègrent parfaitement dans les cadres agiles tels que **scrum** et **kanban**. Dans le cadre de **scrum**, les user stories sont ajoutées aux sprints et « achevées » pendant la durée du sprint. Les équipes **Kanban** ajoutent les user stories à leur backlog et les font passer par leur workflow. C'est ce travail sur les user stories qui permet aux équipes **scrum** de s'améliorer dans l'estimation et la planification des sprints, ce qui conduit à des prévisions plus précises et à une plus grande agilité. Grâce aux stories, les équipes **Kanban** apprennent à gérer les travaux en cours (WIP) et peuvent affiner leurs flux de travail.
- Les **user stories** sont également les blocs de construction de cadres agiles plus grands comme les **epics** et les **initiatives**. Les **epics** sont de grands éléments de travail décomposés en un ensemble de **stories**, et plusieurs **epics** constituent une **initiative**. Ces structures plus larges garantissent que le travail quotidien de l'équipe de développeurs contribue aux objectifs organisationnels intégrés dans les **epics** et les **initiatives**.

# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Utilité des users stories

- Pour les équipes de développeurs qui découvrent la méthode agile, les **user stories** semblent parfois être une étape supplémentaire. Pourquoi ne pas simplement décomposer le grand projet (l'epic) en une série d'étapes et s'y mettre ? Mais les stories donnent à l'équipe un contexte important et associent les tâches à la valeur qu'elles apportent.
- Les user stories présentent un certain nombre d'avantages essentiels :
  - Les stories permettent de garder le focus sur l'utilisateur. Une liste de choses à faire permet à l'équipe de se concentrer sur les tâches à cocher, mais une collection de stories permet à l'équipe de se concentrer sur la résolution des problèmes des utilisateurs réels
  - Les stories permettent la collaboration. Une fois l'objectif final défini, l'équipe peut travailler ensemble pour décider de la meilleure façon de servir l'utilisateur et d'atteindre cet objectif
  - Les stories favorisent les solutions créatives. Les stories encouragent l'équipe à réfléchir de manière critique et créative à la meilleure façon de résoudre un objectif final
  - Les stories créent une dynamique. À chaque story, l'équipe de développeurs relève un petit défi et remporte une petite victoire, ce qui crée une dynamique.

# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Travailler avec les users stories

- Une fois qu'une story a été rédigée, il est temps de l'intégrer dans le workflow. En général, une story est rédigée par le Product Owner, le Product Manager ou le Program Manager et soumise pour révision.
- Au cours d'une réunion de planification de sprint ou d'itération, l'équipe décide des stories qu'elle va traiter pendant le sprint. Les équipes discutent maintenant des exigences et des fonctionnalités requises pour chaque user story. C'est l'occasion de faire preuve de technicité et de créativité dans la mise en œuvre de la story par l'équipe. Une fois l'accord obtenu, ces exigences sont ajoutées à la story.
- Une autre étape courante de cette réunion consiste à noter les stories en fonction de leur complexité ou du temps nécessaire à leur réalisation. Les équipes utilisent des t-shirts, la séquence de Fibonacci ou le poker de planification pour faire des estimations correctes. Une story doit être dimensionnée pour être achevée en un seul sprint, donc lorsque l'équipe spécifie chaque story, elle s'assure de séparer les stories qui dépasseront cet horizon d'achèvement.

# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### Ecriture des users stories

- Les points à prendre en compte lors de la rédaction des user stories :
  - **Définition de " terminé "** - L'user story est généralement " terminée " lorsque l'utilisateur peut accomplir la tâche décrite, mais assurer de définir ce que cela signifie
  - **Décrire les sous-tâches ou les tâches** - Décider quelles étapes spécifiques doivent être réalisées et qui est responsable de chacune d'entre elles
  - **Persona - Utilisateurs** - Pour qui ? S'il y a plusieurs utilisateurs finaux, envisager de créer plusieurs stories
  - **Étapes ordonnées** - Rédiger une story pour chaque étape d'un processus plus large
  - **Écouter les commentaires** - Parler aux utilisateurs et saisir le problème ou le besoin dans leurs mots. Inutile de deviner les stories quand on peut les obtenir des clients
  - **Le temps** - Le temps est un sujet délicat. De nombreuses équipes de développeurs évitent complètement les discussions sur le temps, se reposant plutôt sur leurs cadres d'estimation. Puisque les stories doivent pouvoir être complétées en un seul sprint, les stories qui peuvent prendre des semaines ou des mois doivent être divisées en plusieurs stories plus petites ou être considérées comme une epic à part entière.
- Une fois les user stories clairement définies, assurer qu'elles sont visibles pour l'ensemble de l'équipe.

### Modèle et exemples de user stories

- Les user stories sont souvent exprimées par une phrase simple, structurée comme suit :
- **"En tant que [persona], je [veux], [pour que]"**.

En décomposant cette phrase :

- **"En tant que [persona]"** : Pour qui construisons-nous cela ? Nous ne cherchons pas seulement un titre de poste, nous cherchons la personnalité de la personne. Max. Notre équipe doit avoir une compréhension commune de qui est Karim. Nous avons, je l'espère, interviewé beaucoup de Karim. Nous comprenons comment cette personne travaille, comment elle pense et ce qu'elle ressent. Nous avons de l'empathie pour Karim.
- **"Veut"** : Nous décrivons ici son intention - et non les fonctionnalités qu'il utilise. Qu'est-ce qu'ils essaient réellement d'atteindre ? Cette déclaration doit être exempte d'implémentation - si vous décrivez une partie de l'interface utilisateur et non l'objectif de l'utilisateur, vous passez à côté de l'essentiel.
- **"Pour que"** : comment leur désir immédiat de faire quelque chose de ce genre s'inscrit-il dans leur vision globale ? Quel est l'avantage global qu'ils essaient d'obtenir ? Quel est le grand problème à résoudre ?

Par exemple, les user stories peuvent ressembler à ceci :

- En tant que Karim, je veux inviter mes amis, pour que nous puissions profiter de ce service ensemble.
- En tant que Ali, je veux organiser mon travail pour me sentir plus en contrôle.
- En tant que manager, je veux être capable de comprendre les progrès de mes collègues, afin de pouvoir mieux rendre compte de nos succès et de nos échecs;
- Cette structure n'est pas obligatoire, mais elle est utile pour définir ce qui est fait. Lorsque ce personnage peut saisir la valeur qu'il souhaite, la story est terminée. Nous encourageons les équipes à définir leur propre structure, puis à s'y tenir.

Exemple d'un Backlog

Exemple de backlog détaillé.

User story	Tests d'acceptation	Niveau détaillé	Priorité	Effort, estimation, pointage
En tant que candidat potentiel pour un poste chez OFPPT, je veux pouvoir créer mon profil pour pouvoir postuler à des postes disponibles	<ul style="list-style-type: none"><li>Tous les champs du formulaire ont été validés</li><li>Les informations sont enregistrées</li><li>Un courriel de confirmation est envoyé au candidat</li></ul>	Créer le formulaire d'inscription chez OFPPT avec tous les champs : en d'autres mots créer le profil du candidat	M	4
		Indiquer les champs obligatoires(validation)	M	1
		Faire les validations sur le formulaire puis l'enregistrer	M	5
		Envoyer une confirmation par courriel au candidat	M	2

# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



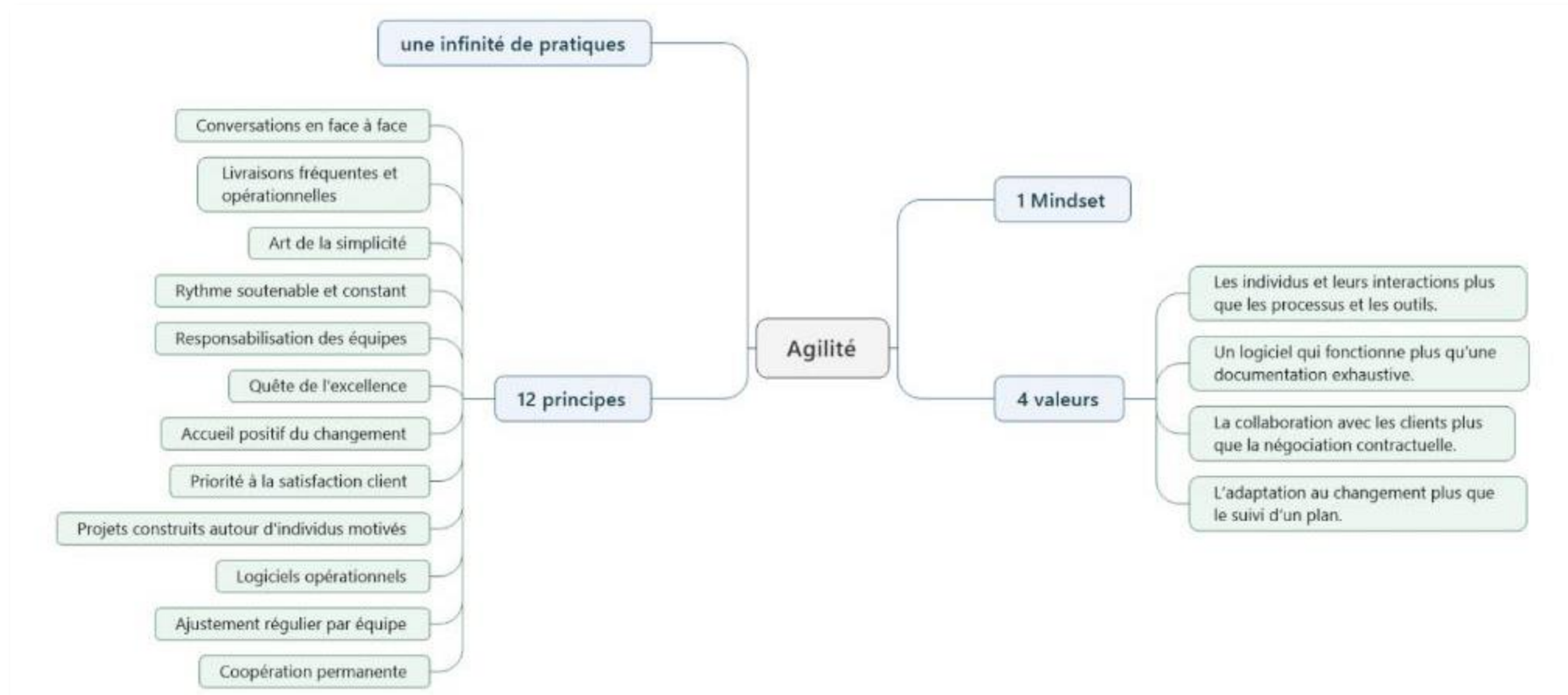
### Synthèse

- Les user stories décrivent le pourquoi et le comment du travail quotidien des membres de l'équipe de développeurs, souvent exprimé sous la forme persona + besoin + objectif. Comprendre leur rôle en tant que source de vérité pour ce que l'équipe livre, mais aussi pourquoi, est la clé d'un processus fluide.
- Commencer par évaluer le prochain grand projet, ou le plus urgent (par exemple, un epic). Décomposer-le en user stories plus petites, et travailler avec l'équipe de développeurs pour les affiner. Une fois que les stories sont dans la nature et que toute l'équipe peut les voir, vous êtes prêt à vous mettre au travail.

# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile

L'approche Agile est constituée par 1 Mindset, 4 Valeurs, 12 Principes, une infinité de Pratiques





# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### QCM : Question 1

Étant donnée la User Story suivante : 'En tant que président, toutes les données que je télécharge ne devraient pas être visibles par un autre utilisateur du système'  
Lors de la première séance de planning de poker, les points de story suivants ont été donnés en fonction du risque, de l'effort, de la complexité et de l'étendue des tests.

- Clients: 5
- Développeurs : 5
- Testeurs : 20

**Quel est le meilleur résultat suivant cette séance de planning Poker ?**

1. Parce que les estimations du développeur et du client correspondent, les membres de l'équipe peuvent être certains que cette estimation est bonne et doivent passer à la User Story suivante
2. L'équipe devrait dialoguer pour comprendre pourquoi les testeurs ont estimé que cette user story représentait nettement plus de travail. Un autre tour planning de poker devrait se produire après cette discussion
3. Parce que le client est le propriétaire final du système, les estimations des clients devraient être considérées comme justes quand il y a un conflit
4. Les séances de planning poker devraient continuer jusqu'à ce que tous les points de la story soient discutés entre les clients, les développeurs et les testeurs

### QCM : Réponse Question 1

Étant donnée la User Story suivante : 'En tant que président, toutes les données que je télécharge ne devraient pas être visibles par un autre utilisateur du système'  
Lors de la première séance de planning de poker, les points de story suivants ont été donnés en fonction du risque, de l'effort, de la complexité et de l'étendue des tests.

- Clients: 5
- Développeurs : 5
- Testeurs : 20

Quel est le meilleur résultat suivant cette séance de planning Poker ?

1. Parce que les estimations du développeur et du client correspondent, les membres de l'équipe peuvent être certains que cette estimation est bonne et doivent passer à la User Story suivante
2. **L'équipe devrait dialoguer pour comprendre pourquoi les testeurs ont estimé que cette user story représentait nettement plus de travail. Un autre tour planning de poker devrait se produire après cette discussion**
3. Parce que le client est le propriétaire final du système, les estimations des clients devraient être considérées comme justes quand il y a un conflit
4. Les séances de planning poker devraient continuer jusqu'à ce que tous les points de la story soient discutés entre les clients, les développeurs et les testeurs

### QCM : Question 2

Les résultats suivants proviennent d'une analyse des risques produit qui a eu lieu au début d'une itération :

- User Story 1 (Performance) : probabilité : élevée, impact : fort
- User Story 2 (Sécurité) : probabilité : élevée, impact : fort
- User Story 3 (Fonctionnalité) : probabilité : moyenne, impact : fort
- User Story 4 (Fonctionnalité) : probabilité : élevée, impact : moyen
- User Story 5 (Compatibilité) : probabilité : faible, impact : faible
- User Story 6 (Récouvrabilité) : probabilité : faible, impact : faible

**Lesquels des DEUX énoncés suivants décrivent le mieux ce que l'équipe doit faire avec cette information ? Sélectionnez DEUX options.**

1. Utiliser la session de planning poker pour estimer l'effort pour les User Story et déterminer ce qui peut être fait dans l'itération courante et ce qui doit être ajouté au Backlog
2. Retirer les User Story 5 et 6 de l'itération actuelle et les placer une itération plus tard.
3. En raison du nombre élevé de probabilités, de risques d'impacts, pour cette itération, l'équipe n'a pas d'autre choix que de prolonger le délai de l'itération de deux semaines
4. L'équipe doit collaborer par des moyens efficaces pour atténuer la forte probabilité et les risques d'impact élevés
5. L'équipe devrait planifier afin de compléter tous les éléments dans le Sprint actuel, mais garder les éléments à risque faible pour la fin du sprint et alors seulement tester ces articles s'il y a le temps

### QCM : Réponse Question 2

Les résultats suivants proviennent d'une analyse des risques produit qui a eu lieu au début d'une itération :

- User Story 1 (Performance) : probabilité : élevée, impact : fort
- User Story 2 (Sécurité) : probabilité : élevée, impact : fort
- User Story 3 (Fonctionnalité) : probabilité : moyenne, impact : fort
- User Story 4 (Fonctionnalité) : probabilité : élevée, impact : moyen
- User Story 5 (Compatibilité) : probabilité : faible, impact : faible
- User Story 6 (Récouvrabilité) : probabilité : faible, impact : faible

Lesquels des DEUX énoncés suivants décrivent le mieux ce que l'équipe doit faire avec cette information ? Sélectionnez DEUX options.

1. **Utiliser la session de planning poker pour estimer l'effort pour les User Story et déterminer ce qui peut être fait dans itération courante et ce qui doit être ajouté au Backlog**
2. Retirer les User Story 5 et 6 de l'itération actuelle et les placer une itération plus tard.
3. En raison du nombre élevé de probabilités, de risques d'impacts, pour cette itération, l'équipe n'a pas d'autre choix que de prolonger le délai de l'itération de deux semaines
4. **L'équipe doit collaborer par des moyens efficaces pour atténuer la forte probabilité et les risques d'impact élevés**
5. L'équipe devrait planifier afin de compléter tous les éléments dans le Sprint actuel, mais garder les éléments à risque faible pour la fin du sprint et alors seulement tester ces articles s'il y a le temps

# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### QCM : Question 3

Étant données les User Stories suivantes :

- 'En tant que caissier de banque, je peux facilement naviguer dans le menu et les liens système, et trouver l'information que je recherche'
- 'Pour tous les utilisateurs, le système doit afficher toutes les requêtes en moins de 2 secondes, dans 90% du temps'

Et les cas de tests associés suivants :

- **TC1** : Se connecter comme caissier de banque : Entrez l'ID du client. Vérifiez que l'historique des transactions de la clientèle est facile à trouver, et que la navigation dans les menus est intuitive.
- **TC2** : Se connecter comme caissier de banque : Entrez le nom du client. Vérifiez que les comptes clients sont faciles à trouver et que la navigation dans les menus est intuitive.
- **TC3** : Simuler le trafic attendu sur le système et valider que le temps d'affichage de l'historique de la transaction client est inférieur à secondes.

De quels DEUX quadrants de tests, les cas de test décrits ci-dessus font-ils partie ?

1. Niveau Q1, Orienté technologie et niveau Q2, Orienté Métier
2. Niveau Q2, Orienté Métier et niveau Q3 ou niveau d'acceptation utilisateurs, Orienté Métier
3. Niveau Q3 ou niveau d'acceptation utilisateurs, Orienté Métier et niveau Q4 ou niveau d'acceptation opérationnel, Orienté technologie
4. Niveau Q2, Orienté Métier et niveau Q4 ou niveau d'acceptation opérationnel, Orienté technologie

# 01 - Appliquer la méthode SCRUM

## Gestion de projet agile



### QCM : Réponse Question 3

Étant données les User Stories suivantes :

- 'En tant que caissier de banque, je peux facilement naviguer dans le menu et les liens système, et trouver l'information que je recherche'
- 'Pour tous les utilisateurs, le système doit afficher toutes les requêtes en moins de 2 secondes, dans 90% du temps'

Et les cas de tests associés suivants :

- **TC1** : Se connecter comme caissier de banque : Entrez l'ID du client. Vérifiez que l'historique des transactions de la clientèle est facile à trouver, et que la navigation dans les menus est intuitive.
- **TC2** : Se connecter comme caissier de banque : Entrez le nom du client. Vérifiez que les comptes clients sont faciles à trouver et que la navigation dans les menus est intuitive.
- **TC3** : Simuler le trafic attendu sur le système et valider que le temps d'affichage de l'historique de la transaction client est inférieur à secondes.

De quels DEUX quadrants de tests, les cas de test décrits ci-dessus font-ils partie ?

1. Niveau Q1, Orienté technologie et niveau Q2, Orienté Métier
2. Niveau Q2, Orienté Métier et niveau Q3 ou niveau d'acceptation utilisateurs, Orienté Métier
3. **Niveau Q3 ou niveau d'acceptation utilisateurs, Orienté Métier et niveau Q4 ou niveau d'acceptation opérationnel, Orienté technologie**
4. Niveau Q2, Orienté Métier et niveau Q4 ou niveau d'acceptation opérationnel, Orienté technologie

## 01 - Appliquer la méthode SCRUM

### Gestion de projet agile



#### QCM : Question 4

**Laquelle des activités de test suivante est réalisée typiquement pendant les projets Agile, mais n'est pas habituelle sur les projets traditionnels?**

1. Les testeurs écrivent des plans de test détaillés pour que les membres de l'équipe puissent comprendre ce qui sera testé à chaque itération.
2. Les testeurs sont fortement impliqués dans la création de cas de test automatisés qui sont alors utilisés pour vérifier l'implémentation des exigences
3. Les testeurs réalisent des tests exploratoires pour trouver rapidement des défauts importants.
4. Les testeurs collaborent avec les développeurs pour mieux comprendre ce qu'il est nécessaire de tester

## 01 - Appliquer la méthode SCRUM

### Gestion de projet agile



#### QCM : Réponse Question 4

Laquelle des activités de test suivante est réalisée typiquement pendant les projets Agile, mais n'est pas habituelle sur les projets traditionnels?

1. Les testeurs écrivent des plans de test détaillés pour que les membres de l'équipe puissent comprendre ce qui sera testé à chaque itération.
2. **Les testeurs sont fortement impliqués dans la création de cas de test automatisés qui sont alors utilisés pour vérifier l'implémentation des exigences**
3. Les testeurs réalisent des tests exploratoires pour trouver rapidement des défauts importants.
4. Les testeurs collaborent avec les développeurs pour mieux comprendre ce qu'il est nécessaire de tester



# CHAPITRE 1

## APPLIQUER LA MÉTHODE SCRUM

1. Gestion de projet agile
- 2. Processus Scrum**
3. Cadre Scrum
4. Planification de la release
5. Planning de poker



# 01 - Appliquer la méthode SCRUM

## Processus Scrum



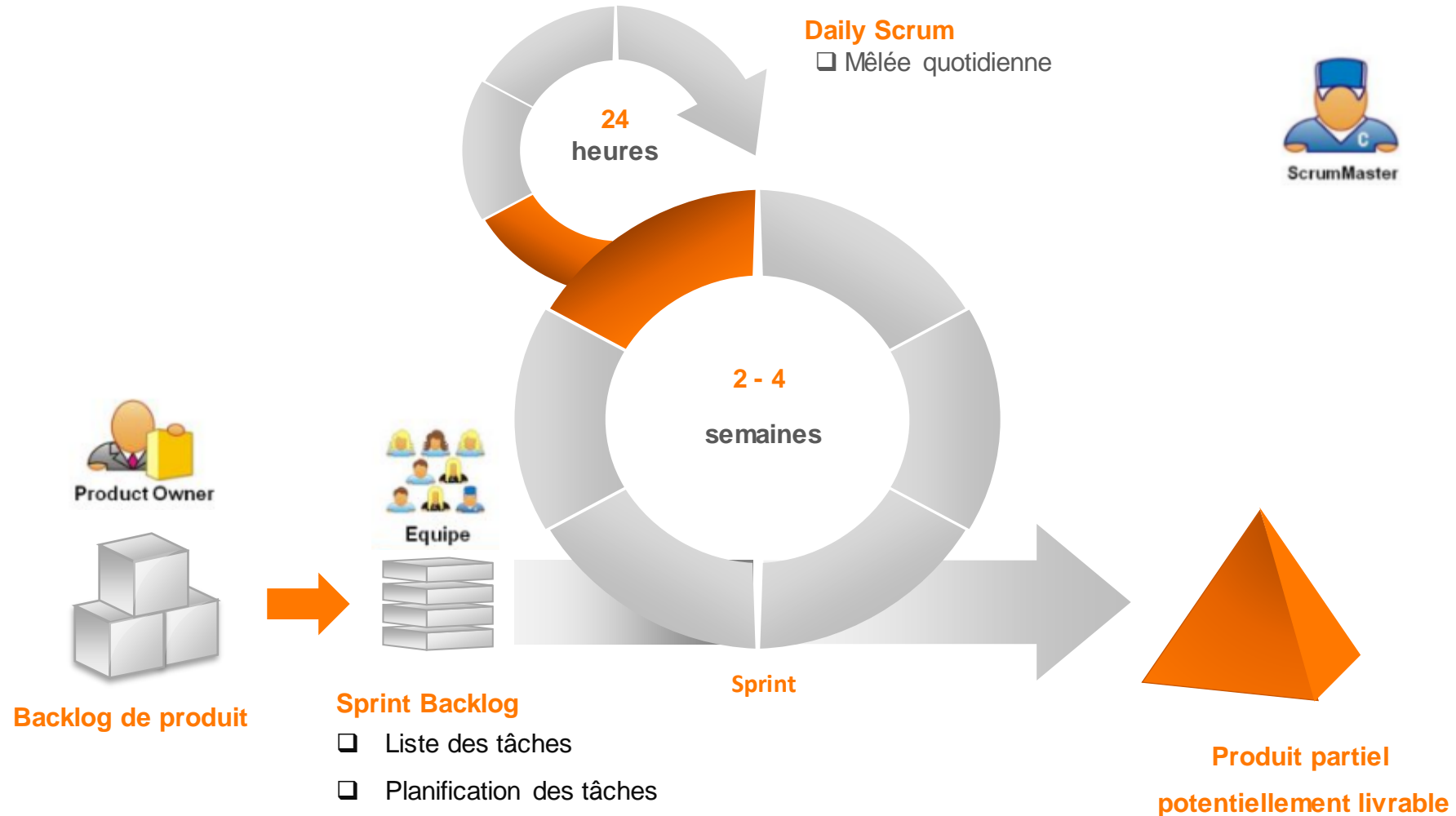
### Définitions

- Scrum est un cadre qui aide les équipes à travailler ensemble. À l'instar d'une équipe de rugby (d'où son nom) qui s'entraîne pour le grand match, scrum encourage les équipes à apprendre par l'expérience, à s'auto-organiser lorsqu'elles travaillent sur un problème et à réfléchir à leurs victoires et leurs défaites pour s'améliorer en permanence.
- Scrum est le plus souvent utilisée par les équipes de développement de logiciels, ses principes et ses leçons peuvent être appliqués à tous les types de travail d'équipe. C'est l'une des raisons pour lesquelles Scrum est si populaire. Souvent considéré comme un cadre de gestion de projet agile, Scrum décrit un ensemble de réunions, d'outils et de rôles qui fonctionnent de concert pour aider les équipes à structurer et à gérer leur travail.

# 01 - Appliquer la méthode SCRUM

## Processus Scrum

### Processus



Modèle de tableau des tâches de Scrum

Stories	A faire		En cours	A vérifier	Terminé
<div>Ceci est un exemple de texte. Remplacez-le par votre propre texte.</div>	<div>Ceci est un exemple de texte. Remplacez-le par votre propre texte.</div>	<div>Ceci est un exemple de texte. Remplacez-le par votre propre texte.</div>	<div>Ceci est un exemple de texte.</div> <div>Ceci est un exemple de texte.</div> <div>Ceci est un exemple de texte.</div>	<div>Ceci est un exemple de texte.</div> <div>Ceci est un exemple de texte.</div> <div>Ceci est un exemple de texte.</div>	<div>Ceci est un exemple de texte. Remplacez-le par votre propre texte.</div> <div>Ceci est un exemple de texte. Remplacez-le par votre propre texte.</div>
<div>Ceci est un exemple de texte. Remplacez-le par votre propre texte.</div>	<div>Ceci est un exemple de texte. Remplacez-le par votre propre texte.</div> <div>Ceci est un exemple de texte. Remplacez-le par votre propre texte.</div>	<div>Ceci est un exemple de texte. Remplacez-le par votre propre texte.</div>	<div>Ceci est un exemple de texte.</div> <div>Ceci est un exemple de texte.</div>	<div>Ceci est un exemple de texte.</div>	<div>Ceci est un exemple de texte. Remplacez-le par votre propre texte.</div>

# 01 - Appliquer la méthode SCRUM

## Processus Scrum



### Sprint

- Un sprint est une période courte et délimitée dans le temps pendant laquelle une équipe Scrum travaille à la réalisation d'une quantité de travail déterminée. Les sprints sont au cœur même de la méthode Scrum et des méthodologies agiles, et une bonne gestion des sprints aidera l'équipe agile à livrer de meilleurs logiciels avec moins de maux de tête.
- "Avec la méthode Scrum, un produit est construit en une série d'itérations appelées sprints qui décomposent les grands projets complexes en petits morceaux", explique **Megan Cook**, chef de produit pour le logiciel Jira chez Atlassian.
- "Les sprints rendent les projets plus gérables, permettent aux équipes de livrer un travail de haute qualité plus rapidement et plus fréquemment, et leur donnent plus de flexibilité pour s'adapter au changement« .
- Beaucoup associent les sprints Scrum au développement agile de logiciels, à tel point que l'on pense souvent que Scrum et Agile sont la même chose. Ce n'est pas le cas. Agile est un ensemble de principes et scrum est un cadre pour faire avancer les choses.
- Les nombreuses similitudes entre les valeurs agiles et les processus de Scrum conduisent à une association juste. Les sprints aident les équipes à suivre le principe agile de "livrer fréquemment des logiciels fonctionnels", ainsi qu'à vivre la valeur agile de "répondre au changement plutôt que de suivre un plan". Les valeurs Scrum de transparence, d'inspection et d'adaptation sont complémentaires à l'agilité et centrales au concept de sprints.

# 01 - Appliquer la méthode SCRUM

## Processus Scrum



### A faire

- S'assurer que l'équipe définit et comprend l'objectif du sprint et la façon dont le succès sera mesuré. C'est la clé pour que tout le monde reste aligné et avance vers une destination commune.
- À faire : assurez-vous d'avoir un backlog bien soigné avec vos priorités et dépendances en ordre. Il s'agit d'un défi important qui peut faire dérailler le processus s'il n'est pas correctement géré.
- Assurez-vous d'avoir une bonne compréhension de la vélocité, et qu'elle reflète des éléments tels que les congés et les réunions d'équipe.
- Profitez de la réunion de planification du sprint pour préciser les détails du travail à accomplir. Encouragez les membres de l'équipe à esquisser des tâches pour toutes les stories, bugs et tâches qui entrent dans le sprint.
- Laissez de côté le travail dont vous ne pourrez pas assurer les dépendances, comme le travail d'une autre équipe, les conceptions et les signatures légales.
- Enfin, une fois qu'une décision ou un plan est pris, assurez-vous que quelqu'un capture cette information dans votre outil de gestion de projet ou de collaboration, comme vos tickets Jira. De cette façon, la décision et sa justification seront facilement accessibles à tous par la suite.

### A ne pas faire

- N'introduisez pas trop de stories, ne surestimez pas la vélocité et n'introduisez pas de tâches qui ne peuvent pas être réalisées pendant le sprint. Vous ne voulez pas que votre équipe ou vous-même soyez voués à l'échec.
- N'oubliez pas la qualité ou la dette technique. Veillez à prévoir du temps pour l'assurance qualité et les tâches non liées aux fonctionnalités, comme les bogues et la santé technique.
- Ne laissez pas l'équipe avoir une vision floue de ce qui est prévu dans le sprint. Précisez-le et ne vous concentrez pas sur la rapidité d'exécution au point d'oublier de vous assurer que tout le monde va dans la même direction.
- De même, ne prenez pas en charge une grande quantité de travail inconnu ou à haut risque. Décomposez les stories qui sont importantes ou qui présentent une grande incertitude, et n'ayez pas peur de laisser une partie de ce travail pour le prochain sprint.
- Si l'équipe vous fait part de ses préoccupations, qu'il s'agisse de la vélocité, d'un travail peu sûr ou d'un travail qu'elle pense être plus important que ce qu'elle a estimé, ne l'ignorez pas. Traitez le problème et recalibrez si nécessaire.

## 01 - Appliquer la méthode SCRUM

### Processus Scrum

### Développement séquentiel vs superposé



Plutôt que de faire toute une discipline d'un coup...

...Les équipes Scrum font un peu de tout tout le temps



Source: "The New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review*, January 1986.

# CHAPITRE 1

## APPLIQUER LA MÉTHODE SCRUM

1. Gestion de projet agile
2. Processus Scrum
- 3. Cadre Scrum**
4. Planification de la release
5. Planning de poker





# 01 - Appliquer la méthode SCRUM

## Cadre Scrum



### Le cadre Scrum

#### Rôles

- Product Owner
- ScrumMaster
- Equipe

#### Cérémonial

- Planification du sprint
- Revue du sprint
- Rétrospective
- Scrum quotidien

#### Artefacts

- Backlog de produit
- Liste des tâches
- Burndowns

# 01 - Appliquer la méthode SCRUM

## Cadre Scrum



### Le cadre Scrum

#### Rôles

- Product Owner
- ScrumMaster
- Equipe

#### Cérémonial

- Planification du sprint
- Revue du sprint
- Rétrospective
- Scrum quotidien

#### Artefacts

- Backlog de produit
- Liste des tâches
- Burndowns

# 01 - Appliquer la méthode SCRUM

## Cadre Scrum

### Product Owner

- L'un des principaux rôles d'un Product Owner est de gérer le backlog du produit. Cela peut inclure les activités suivantes :
  - Le backlog de produit doit être clairement défini, et tous les éléments doivent être mentionnés de manière détaillée
  - Hiérarchiser et ordonner le backlog de produit de la bonne manière afin de donner la priorité aux tâches importantes
  - Hiérarchisez les éléments de travail et le backlog du produit, cela doit être en accord avec la vision et les objectifs du client
  - Évaluer le travail effectué par l'équipe de développement et fournir un feedback constant
  - Le Product Owner doit s'assurer que le backlog de produit est communiqué clairement à tous les membres de l'équipe
  - L'équipe Scrum doit avoir une vision claire des exigences du produit et des attentes des utilisateurs
  - Accepte ou rejette les résultats

GESTION DU BACKLOG  
DE SCRUM



GESTION DES VERSIONS

GESTION DES PARTIES  
PRENANTES

# 01 - Appliquer la méthode SCRUM

## Cadre Scrum



### Product Owner

- Les équipes agiles sont, par conception, flexibles et réactives, et il est de la responsabilité du Product Owner de s'assurer qu'elles fournissent la meilleure valeur possible. L'entreprise est représentée par le Product Owner qui indique au développeur ce qu'il est important de livrer. La confiance entre ces deux rôles est cruciale.
- Le Product Owner doit non seulement comprendre le client, mais aussi avoir une vision de la valeur que l'équipe Scrum apporte au client. Le Product Owner doit également tenir compte des besoins des autres parties prenantes de l'organisation.
- Le Product Owner doit donc prendre en compte toutes ces données et hiérarchiser le travail. Il s'agit probablement de sa responsabilité la plus importante, car des priorités conflictuelles et des orientations peu claires réduiront non seulement l'efficacité de l'équipe, mais pourraient également briser l'importante relation de confiance que l'entreprise entretient avec l'équipe de développeurs.
- Les équipes agiles sont conçues pour inspecter et s'adapter. Cela signifie qu'un changement de priorité peut entraîner un changement massif de la structure de l'équipe, des produits de travail, ainsi que du résultat final. Il est donc crucial, pour le succès des équipes Scrum, qu'une seule personne fixe les priorités. Cette personne est le Product Owner.
- Le Guide Scrum définit les responsabilités du Product Owner comme suit :
  - **Gérer le backlog de Scrum** - Cela ne signifie pas qu'ils sont les seuls à introduire de nouveaux éléments dans le backlog de produit. Mais en fin de compte, il est responsable du backlog que l'équipe de développeurs utilise pour livrer. Cela signifie que le Product Owner doit être au courant de tout ce qui se trouve dans le backlog et que les autres personnes qui ajoutent des éléments au backlog de produit doivent s'assurer de communiquer avec le Product Owner.
  - **Gestion des versions** - Le sprint n'est pas un cycle de publication, mais plutôt un cycle de planification. Cela signifie que les équipes Scrum peuvent livrer à tout moment. Idéalement, elles devraient livrer fréquemment tout au long du sprint, ce qui permet à la revue du sprint d'examiner l'utilisation réelle et le feedback des clients. Cependant, la livraison continue n'est pas toujours possible et d'autres modèles de publication sont nécessaires. Il est important pour le Product Owner de savoir quand les choses peuvent et doivent être publiées.
  - **Gestion des parties prenantes** - Tout produit implique de nombreuses parties prenantes, qu'il s'agisse des utilisateurs, des clients, de la gouvernance ou de la direction de l'organisation. Le Product Owner devra travailler avec toutes ces personnes pour s'assurer que l'équipe de développeurs apporte de la valeur ajoutée. Cela peut signifier une grande quantité de gestion et de communication avec les parties prenantes.

# 01 - Appliquer la méthode SCRUM

## Cadre Scrum



### Scrum Master

- Un Scrum Master est chargé de veiller à ce qu'une équipe Scrum fonctionne aussi efficacement que possible dans le respect des valeurs de Scrum. Cela signifie qu'il maintient l'équipe sur la bonne voie, qu'il planifie et dirige les réunions et qu'il résout les obstacles auxquels l'équipe peut être confrontée. Les Scrum Masters peuvent également jouer un rôle plus important au sein d'une organisation pour l'aider à intégrer les concepts Scrum dans son travail. Parce qu'ils sont à la fois un leader et un soutien dans les coulisses, ils sont souvent décrits comme le "leader serviteur" de l'équipe Scrum.
- Scrum peut être différent d'une organisation à l'autre et d'une équipe à l'autre, ce qui rend les tâches spécifiques d'un Scrum Master variées. De manière générale, cependant, un Scrum Master peut avoir les responsabilités suivantes :
  - animer les réunions quotidiennes de Scrum (également appelées "standups quotidiens")
  - diriger les réunions de planification du sprint
  - mener des revues "rétrospectives" pour voir ce qui a bien fonctionné et ce qui peut être amélioré pour le sprint suivant
  - prendre le pouls des membres de l'équipe, par le biais de réunions individuelles ou d'autres moyens de communication
  - gérer les obstacles qui se présentent à l'équipe en communiquant avec les parties prenantes extérieures à l'équipe

# 01 - Appliquer la méthode SCRUM

## Cadre Scrum



### Equipe Scrum

- Une équipe de développeurs est composée de professionnels qui effectuent le travail pratique de réalisation des tâches d'un sprint Scrum. Cela signifie que les membres de l'équipe de développement peuvent être des ingénieurs informatiques, des concepteurs, des rédacteurs, des analystes de données, ou tout autre rôle nécessaire pour atteindre les objectifs du sprint. L'équipe de développeurs ne se contente pas d'attendre les ordres ; elle collabore généralement à la définition des objectifs et des plans pour les atteindre.
- Tous les membres de l'équipe de développeurs n'auront pas toujours les mêmes responsabilités. Par exemple, si vous mettez à jour un site Web, il se peut que l'ingénieur frontal, le concepteur UX, le rédacteur et le spécialiste du marketing travaillent tous au sein de la même équipe Scrum. Les responsabilités d'une équipe de développeurs dépendront également des objectifs finaux de l'équipe Scrum. Toutefois, de manière générale, les tâches suivantes peuvent être confiées à une équipe de développeurs :
  - aider à la planification du sprint et à la définition des objectifs ;
  - apporter son expertise pour programmer, concevoir ou améliorer des produits ;
  - utiliser les données pour trouver les meilleures pratiques en matière de développement ;
  - tester les produits et les prototypes, ainsi que d'autres formes d'assurance qualité.

# 01 - Appliquer la méthode SCRUM

## Cadre Scrum



### Le cadre Scrum

#### Rôles

- Product Owner
- ScrumMaster
- Equipe

#### Cérémonial

- Planification du sprint
- Revue du sprint
- Rétrospective
- Scrum quotidien

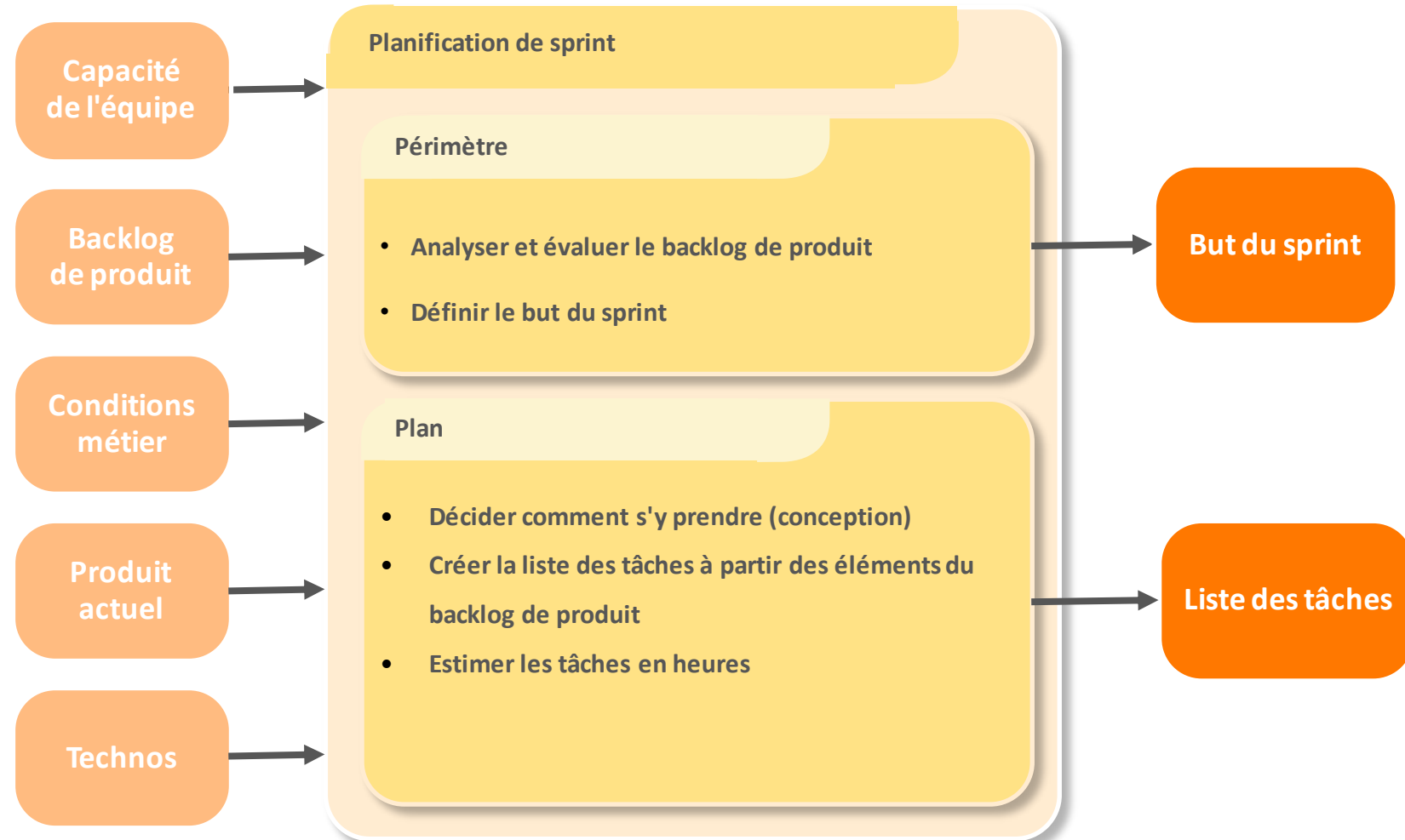
#### Artefacts

- Backlog de produit
- Liste des tâches
- Burndowns

# 01 - Appliquer la méthode SCRUM

## Cadre Scrum

### Planification de sprint





### Planification de sprint

- C'est une réunion d'équipe qui se tient avant le prochain sprint agile. L'équipe examine son backlog pendant la planification du sprint et décide des éléments à prioriser pour le prochain sprint.
- L'équipe aura deux éléments à la fin de la réunion de planification du sprint. Le premier est un objectif de sprint (un résumé du plan pour le prochain sprint). Le deuxième élément est le backlog du sprint (la liste des tâches sur lesquels l'équipe va travailler pendant le sprint) :
  - Permet à l'équipe de s'aligner sur les priorités du sprint avant le début de celui-ci
  - Il donne à tous les membres de l'équipe une compréhension de leurs responsabilités dans le prochain sprint
- Les spécialistes de Scrum ont vraiment pensé à tout. Afin de planifier le prochain sprint, vous utilisez la réunion de planification du sprint ! La planification du sprint est un événement collaboratif au cours duquel l'équipe répond à deux questions fondamentales : Quel travail peut être fait dans ce sprint et comment le travail choisi sera-t-il fait ?
- Le choix des bons éléments de travail pour un sprint est un effort de collaboration entre le Product Owner, le Scrum Master et l'équipe de développement. Le Product Owner discute de l'objectif que le sprint doit atteindre et des éléments du backlog de produit qui, une fois terminés, permettront d'atteindre l'objectif du sprint.
- L'équipe élabore ensuite un plan sur la manière dont elle va construire les éléments du backlog et les faire aboutir avant la fin du sprint. Les éléments de travail choisis et le plan pour les réaliser s'appellent le backlog du sprint. À la fin de la planification du sprint, l'équipe est prête à commencer à travailler sur le backlog du sprint, en faisant passer des éléments du backlog à "En cours" et "Terminé".
- Pendant le sprint, l'équipe fait le point sur l'avancement du travail lors de la mêlée quotidienne, ou standup. L'objectif de cette réunion est de faire apparaître les obstacles et les défis qui pourraient avoir un impact sur la capacité de l'équipe à atteindre l'objectif du sprint.
- Après un sprint, l'équipe démontre ce qu'elle a réalisé lors de la revue du sprint. C'est l'occasion pour l'équipe de montrer son travail aux parties prenantes et aux coéquipiers avant qu'il ne soit mis en production.
- Terminer le cycle de sprint par la rétrospective du sprint. C'est l'occasion pour l'équipe d'identifier les points à améliorer pour le prochain sprint. Avec cela, vous êtes prêt à commencer votre prochain cycle de sprint. En avant !

# 01 - Appliquer la méthode SCRUM

## Cadre Scrum



### Planification de sprint

- **Objectif(s)** : définir le périmètre et les objectifs du sprint puis découpage en tâches de développement
- **Responsable** : Product Owner
- **Participants** : Product Owner, Scrum Master, équipe de développement
- **Fréquence** : Avant le sprint
- **Durée maximale** : 4 heures
- **Document(s) en entrée** : Product Backlog priorisé
- **Document(s) en sortie** : Sprint Backlog

## 01 - Appliquer la méthode SCRUM

### Cadre Scrum



### Revue de sprint

A la fin du Sprint, toute l'équipe Scrum ainsi que les parties prenantes se réunissent pour passer en revue l'incrément développé durant le Sprint.

Cette démonstration se focalise uniquement autour des éléments finis, les autres ne sont pas présentés. Ceci donne lieu à un bilan de livraison et de validation d'éléments présentés :

- Les éléments présentés ne correspondant pas aux exigences ne sont pas validés. Ils seront reprogrammés dans un autre Sprint. A l'issue de cette réunion, le carnet du produit (Product Backlog) est tenu à jour.
- La durée de cette réunion pour un Sprint de 4 semaines ou d'un mois calendaire ne doit pas dépasser 4 heures. Pour un Sprint dont la durée est inférieure, la durée de la réunion peut s'ajuster de façon proportionnelle.

# 01 - Appliquer la méthode SCRUM

## Cadre Scrum



### Revue de sprint

- **Objectif(s)** : présentation des fonctionnalités produites au cours du Sprint au Product Owner et utilisateurs finaux, réception des feedbacks
- **Responsable** : Equipe de développement
- **Participants** : équipe de développement, Product Owner, Scrum Master, utilisateurs finaux, invités
- **Fréquence** : A la fin du Sprint
- **Durée maximale** : 2 heures
- **Document(s) en sortie** : Liste des feedbacks

## 01 - Appliquer la méthode SCRUM

### Cadre Scrum



### Mêlée quotidienne

- Mêlée quotidienne, également appelée standup, est une courte réunion quotidienne destinée à permettre à l'équipe de planifier son travail de la journée et d'identifier les obstacles qui pourraient avoir un impact sur ce travail.
- La plupart des équipes tiennent ces réunions le matin et les limitent à 10 ou 15 minutes. On les appelle parfois "standups" parce que de nombreuses équipes les tiennent debout pour qu'elles soient courtes et ciblées.
  - permet à l'équipe d'identifier les obstacles afin qu'elle puisse les résoudre rapidement
  - permet à l'équipe d'effectuer fréquemment des corrections mineures de la trajectoire, afin qu'elle ne passe pas trop de temps à faire fausse route
- Chacun répond à 3 questions :

1

Qu'as-tu fait hier ?

2

Que vas-tu faire aujourd'hui ?

3

Y a t-il un obstacle qui te freine ?

# 01 - Appliquer la méthode SCRUM

## Cadre Scrum



### Mêlée quotidienne

- **Objectif(s)** : coordination de l'équipe de développement, identification des obstacles qu'elle rencontre, mesure de l'avancement du Sprint
- **Responsable** : Scrum Master
- **Participants** : équipe de développement, Scrum Master, Product Owner/AMOA (optionnel)
- **Fréquence** : quotidienne
- **Durée maximale** : 15 minutes
- **Document(s) en entrée** : Sprint Backlog
- **Document(s) en sortie** : Sprint Backlog et graphique d'avancement mis à jour

### Rétrospective de sprint

- Il s'agit d'une discussion post-sprint. L'équipe Scrum examine ce qui s'est passé pendant le sprint pour déterminer ce qui a fonctionné, ce qui n'a pas fonctionné et comment ils peuvent améliorer le processus pendant le prochain sprint.
- Une rétrospective de sprint est une version plus restreinte et plus fréquente de la rétrospective de produit : une réunion tenue après le lancement d'un produit pour évaluer ce qui a fonctionné et ce qui doit être amélioré.
  - permet à l'équipe de réfléchir à son travail après chaque sprint et de célébrer ses succès
  - Permet à l'équipe de continuer à améliorer ses processus

Commencer à faire

Juste une façon parmi  
d'autres de faire une  
rétrospective.

Arrêter de faire

Continuer à faire

# 01 - Appliquer la méthode SCRUM

## Cadre Scrum



### Rétrospective de sprint

- **Objectif(s)** : améliorer la productivité (vélocité) de l'équipe de développement
- **Responsable** : Scrum Master
- **Participants** : équipe de développement, Scrum Master, Product Owner / AMOA (optionnel)
- **Fréquence** : Après la revue de Sprint
- **Durée maximale** : 1h30
- **Document(s) en sortie** : compte rendu de réunion (bilan de Sprint + plan d'actions)



# 01 - Appliquer la méthode SCRUM

## Cadre Scrum



### Le cadre Scrum

#### Rôles

- Product Owner
- ScrumMaster
- Equipe

#### Cérémonial

- Planification du sprint
- Revue du sprint
- Rétrospective
- Scrum quotidien

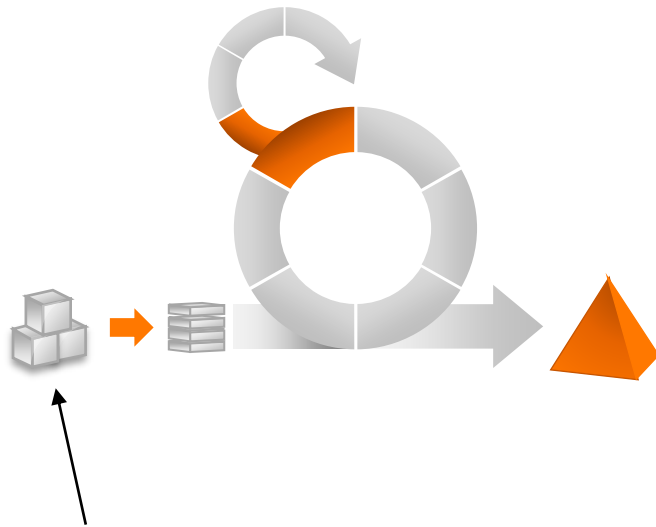
#### Artefacts

- Backlog de produit
- Liste des tâches
- Burndowns

## 01 - Appliquer la méthode SCRUM

### Cadre Scrum

### Backlog de produit (Product backlog)



Ceci est le backlog de produit

- Un backlog de produit est une liste de travaux prioritaires pour l'équipe de développement, établie à partir de la feuille de route et de ses exigences.
- Les éléments les plus importants figurent en haut du backlog de produit afin que l'équipe sache ce qu'elle doit livrer en premier.
- L'équipe de développement ne travaille pas sur le backlog au rythme du Product Owner et le Product Owner n'impose pas de travail à l'équipe de développement. Au lieu de cela, l'équipe de développement extrait le travail du backlog du produit au fur et à mesure qu'elle en a la capacité, soit de manière continue (Kanban), soit par itération (Scrum).
- Les priorités sont définies par le Product Owner.
- Les priorités sont revues à chaque sprint.

### Construire le backlog du produit

Pour produire un backlog initial, il est important de prioriser chacune des stories qui sera contenue dans le backlog.

#### Prioriser les stories : La méthode 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 faite 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 faite, on peut la contourner et la livrer plus tard.
  - **C** pour **Could Have** : Il s'agit d'une exigence souhaitable. Elle **POURRAIT** être faite 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** faite cette fois mais plus tard, mais intéressante et à garder pour la prochaine version.

#### La méthode priority Poker (pour les fonctionnalités)

- 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 la fonctionnalité. Chaque participant vote avec une carte. On fait le total des points.
- Le deuxième vote, porte sur la pénalité de ne pas avoir la fonctionnalité 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.

Un backlog de produit

Exemple de backlog initial, pour une application de réservation de chambre d’hôtel.

Elément de backlog	Priorité	Estimation
Un invité peut faire une réservation	M	3
En tant qu'invité, j'annule une réservation	M	5
En tant qu'invité, je change les dates d'une réservation	M	3
En tant qu'employé de l'hôtel, je produis les rapports de revenu par chambre	M	8
Améliorer la gestion des exceptions	S	8

Un backlog de produit

Exemple de product backlog — © The Scrum Primer

				Nouvel effort estimé					
				Reste à faire sur le Sprint					
Priorité	Elements (Stories)	Détails (URL Wiki)	Effort initial estimé	1	2	3	4	5	6
1	En tant qu'acheteur, je veux placer un livre dans un panier de courses (cf croquis de l'IHM sur le wiki)		5						
2	En tant qu'acheteur, je veux supprimer un live de mon panier		2						
3	Améliorer les performances des transactions (cf performances cibles sur le wiki)		13						
4	Investiguer une solution pour accélérer la validation des cartes de crédit (cf métriques de performance sur le wiki)		20						
5	Mise à jour des serveurs avec Apache 2.2.3		13						
6	Diagnostiquer et réparer l'erreur de calcul de commande (ID bugzilla 14923)		3						
7	En tant qu'acheteur, je veux créer et sauvegarder une liste d'envies		40						
8	En tant qu'acheteur, je veux ajouter ou supprimer des articles dans ma liste d'envies		20						

# 01 - Appliquer la méthode SCRUM

## Cadre Scrum



### Vie de la liste des tâches

- Chacun s'engage sur du travail qu'il choisit :
  - Le travail n'est jamais attribué par un autre
- L'estimation du reste à faire est ajustée tous les jours
- N'importe qui peut ajouter, supprimer ou changer la liste des tâches
- Le travail du sprint émerge progressivement
- Si un travail n'est pas clair, définir une tâche avec plus de temps et la décomposer après
- Mise à jour du travail restant quand il est mieux connu

# 01 - Appliquer la méthode SCRUM

## Cadre Scrum



### Backlog de Sprint

- Le Sprint Backlog comporte la liste des tâches du Sprint (son périmètre donc) ainsi que la charge de travail associée à ces dernières. Chaque jour, le Reste A Faire de chaque tâche est actualisé par l'équipe de développement afin de tracer le graphique d'avancement de Sprint.

#### Backlog de produit

- Le Product Owner a le contrôle du backlog produit
- Se focalise sur la vision produit globale
- Pas de limite de temps pour épuiser le backlog
- Entité indépendante qui évolue jusqu'à la finalisation du produit

#### Backlog de sprint

- L'équipe de développement est responsable de chaque backlog de sprint
- Se focalise sur l'atteinte de l'objectif du sprint
- À terminer sur une période de temps bien définie
- Sous-ensemble du backlog produit, avec des tâches précises

# CHAPITRE 1

## APPLIQUER LA MÉTHODE SCRUM

- 
1. Gestion de projet agile
  2. Processus Scrum
  3. Cadre Scrum
  - 4. Planification de la release**
  5. Planning de poker



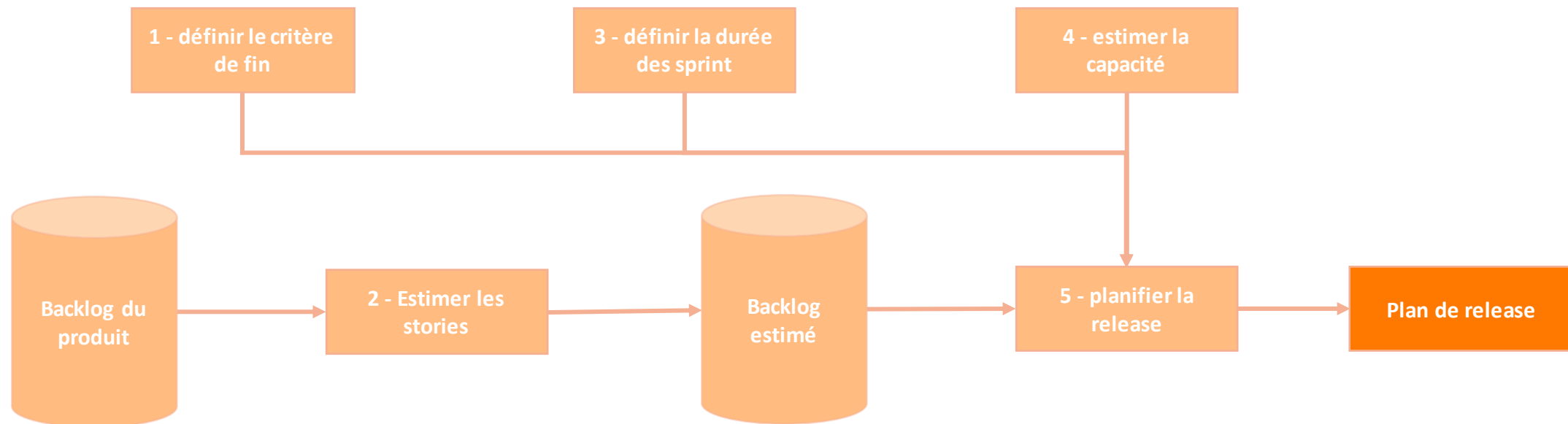
# 01 - Appliquer la méthode SCRUM

## Planification de la release



### Planifier la release

- Une release est une séquence de sprints, mais quand finit-elle? Comment la planifier pour livrer un produit fini et à temps? Combien de sprints la release contiendra-t-elle? Quelle est la durée d'un sprint? Pour planifier une release, nous allons procéder par étapes.



### Planifier la release

#### Étape 1 : définir le critère de fin de la release :

1. Finir quand le backlog est vide. Mais le backlog est vivant. Il évolue avec le temps. Il est difficile de figer le backlog quand une amélioration est possible.

Une variante serait de fixer un sous-ensemble du backlog pour la release. Comme les éléments sont rangés par priorité, il suffit de fixer une limite. « Pour la release courante on livre le sous-ensemble sélectionné, le reste des éléments du backlog seraient dans la prochaine version de la release ». C'est-ce qu'on appelle une release à paramètres fixés. Mais, nous avons toujours le même souci : le backlog est vivant. Il serait stupide de figer le backlog en refusant un changement qui apporte une valeur.

2. Fixer la date de fin à l'avance : la meilleure façon de procéder est de fixer une date de fin et de s'y tenir. L'objectif d'une release à date fixée est d'estimer le contenu qui sera livré à cette date. La release à date fixe présente les avantages suivants :

- a) Elle donne un objectif précis pas lointain ---> motivation de l'équipe ;
  - b) Elle impose au Product Owner d'avoir une réflexion poussée sur les priorités des éléments du backlog ;
  - c) Des éléments du backlog ayant peu d'intérêt ne seront pas développés ;
  - d) On passe moins de temps à planifier puisque la date de livraison est connue. Une release à date fixée et à durée uniforme (ex emple une durée de 3 mois) est la formule la plus facile à mettre en œuvre.
- Une release à date fixée et à durée uniforme (exemple une durée de 3 mois) est la formule la plus facile à mettre en œuvre.

# 01 - Appliquer la méthode SCRUM

## Planification de la release



### Planifier la release

Étape 2 : estimer les stories du backlog : La planning Pocker

Étape 3 : définir la durée des sprints :

Lorsqu'on lance utilise une approche agile pour le développement logiciel, la question fondamentale est de déterminer la durée d'un sprint. Chaque projet est différent. Il n'y a donc pas de réponse universelle.

Pour SCRUM, la pratique est de faire des sprints de maximum UN MOIS. Dans la plupart des cas, des sprints de 2 à 3 semaines sont recommandés.

Pour définir la durée d'un sprint, il faudra tenir compte des facteurs suivants :

1. **L'implication des clients et des utilisateurs** : Il faut tenir compte de leur disponibilité à utiliser les versions partielles produites à la fin de chaque sprint
2. **Le coût supplémentaire engendré par la préparation du sprint** : Un sprint ajoute du travail supplémentaire pour préparer le produit partiel. Faire les tests de non régression, préparer la démonstration pour la revue de sprint
3. **La taille de l'équipe** : plus il y a du monde dans l'équipe, plus il faudra du temps pour se synchroniser
4. La date de fin de la release : idéalement une release comporte au moins quatre sprints pour profiter des bénéfices des avantages de l'itératif
5. Un sprint trop long risque de démotiver l'équipe
6. La stabilité de l'architecture: il est plus facile d'obtenir un produit qui fonctionne si l'architecture est stable

# 01 - Appliquer la méthode SCRUM

## Planification de la release



### Planifier la release

- **La Vitesse** : la vitesse de l'équipe mesure la partie du backlog réalisée par l'équipe à l'intérieur d'un sprint. À la fin d'un sprint, on mesure ce que l'équipe a été capable de réaliser. La vitesse se calcule à la fin d'un sprint après la démonstration et lors de la revue de sprint
- La vitesse est volatile, elle peut varier sensiblement entre les sprints.
- Si un projet vient de commencer et que l'équipe n'a pas de sprint passé, une façon de faire pour déterminer la vitesse (une mesure) est de simuler une planification du premier sprint.
- La vitesse est une mesure de l'équipe et non d'une personne individuelle.
- La capacité est une prévision de ce que l'équipe est capable de faire en tenant compte de sa vitesse.
- Exemple de prévision: Trois stories sont étudiées qui avaient été estimées à 3, 2 et 5 points. Les tâches identifiées pour ses stories sont estimées à 60 heures.
- L'équipe dispose de 300 heures pour le sprint
- En 60 heures, l'équipe pourrait réaliser 10 points.  $(3 + 2 + 5)$
- En 300 heures (sprint) l'équipe pourrait réaliser  $(300 / 60) * 10$  points
- La capacité de l'équipe serait 50 points (estimation)

# 01 - Appliquer la méthode SCRUM

## Planification de la release



### Planifier la release

#### Étape 5 : produire un plan de release

1. Après avoir fait les étapes précédentes produire un plan de release devient facile, un jeu d'enfant. On procède comme suit :
  2. On prend le backlog du produit priorisé et estimé ;
  3. On commence par le premier sprint de la release. On y associe les stories en commençant par les prioritaires ;
  4. On continue dans ce sprint en additionnant la taille en points de stories jusqu'à atteindre la capacité de l'équipe ;
  5. Quand on y arrive, on passe au sprint suivant.
- Si on ne tombe pas exactement sur la capacité de l'équipe dans un sprint en ajoutant une story, il faut décider: Est-ce qu'on va au-dessus ou en dessous de la capacité de l'équipe ou prendre une story moins prioritaire mais qui se rapproche le plus de la capacité de l'équipe?
  - Garder du « LOUSSE » ou du mou pour les incertitudes.
  - Dans le plan de release, tout est basé sur l'estimation des stories du backlog. Même si l'estimation est faite par ceux qui réalisent, même si l'estimation est collective il y a une part d'incertitude surtout au début de la release et il faudra en tenir compte.
  - Pour les release à date fixe, le lousse porte sur les stories.

# 01 - Appliquer la méthode SCRUM

## Planification de la release



### Le sprint zéro

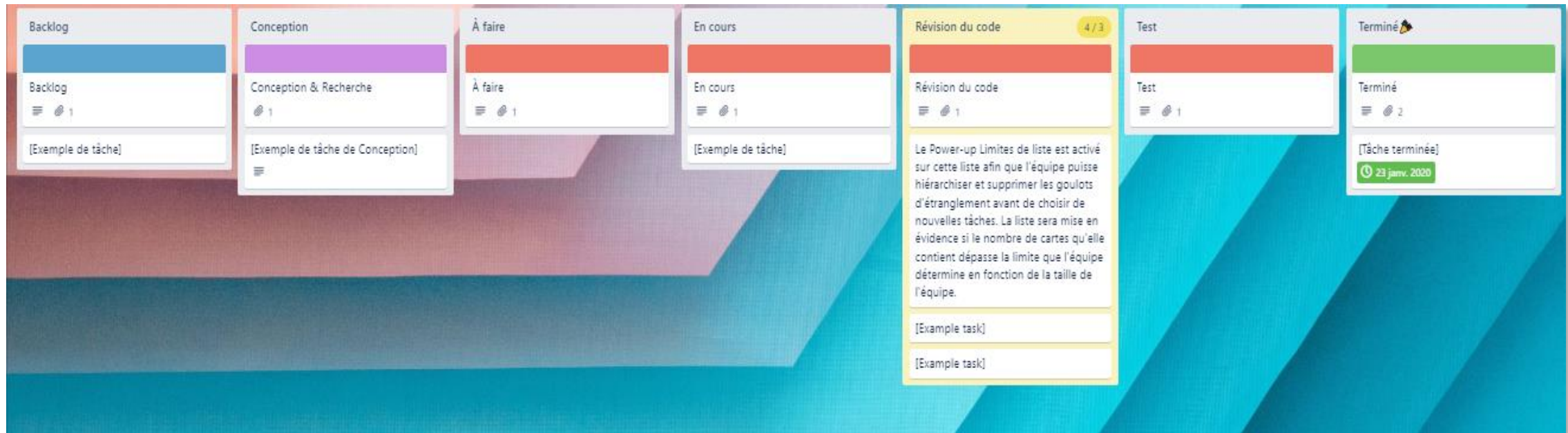
- Le développement agile a besoin d'un sprint de départ, qui ne se termine pas nécessairement par une livraison. D'une durée variable sert à mettre le projet sur de bons rails et d'apprendre à l'équipe de travailler ensemble. Concrètement, ce que l'on doit faire durant le sprint Zéro est :
  1. Partager une vision claire du projet
  2. Préparer l'environnement de développement
  3. Produire un backlog du produit estimé et priorisé
  4. Roder l'équipe sur le backlog initial
  5. Définir la posture ergonomique de l'interface
  6. Déterminer un plan de Release
  7. Selon le contexte, travailler sur l'architecture et la BD • S'offrir une belle rétrospective

# 01 - Appliquer la méthode SCRUM

## Planification de la release

### Tableau des tâches

- Ce tableau comporte la liste des tâches du Sprint Backlog, il permet à l'équipe de développement de se coordonner. Il comporte généralement 4 colonnes : « A faire », « En cours », « A vérifier », « Terminé ». Il est mural et simple à interpréter, permettant à n'importe quel acteur du projet de connaître en un coup d'œil l'avancement « temps réel » du Sprint.



# CHAPITRE 1

## APPLIQUER LA MÉTHODE SCRUM

- 
1. Gestion de projet agile
  2. Processus Scrum
  3. Cadre Scrum
  4. Planification de la release
  5. **Planning de poker**



## 01 - Appliquer la méthode SCRUM

### Planning Poker



#### Définition

- Le Planning Poker est une technique d'estimation de coût d'exigences. Cette technique se pratique en équipe et permet de procéder à des estimations rapides et aussi précises que possible selon le niveau de précision du besoin disponible.
- Au cours des séances d'estimation, le Product Owner (assisté de l'AMOA au besoin) soumet une à une à l'équipe de développement les exigences dont il souhaite connaître l'estimation de coût. Il répond aux questions de l'équipe de développement qui en cas d'absence de réponse (besoin encore flou) établit des hypothèses. Les estimations de coût (dont l'unité est le point) ainsi obtenues viennent alimenter le Product Backlog et aideront le Product Owner à prioriser ses exigences. La technique de Planning Poker peut également être utilisée pour estimer en équipe (Product Owner, AMOA, utilisateurs, marketing,...) la valeur métier en points des exigences du Product backlog.

## 01 - Appliquer la méthode SCRUM

### Planning Poker



#### Estimer les stories

- Chaque story du backlog doit être estimée si on veut en tenir compte dans la planification. Les stories ne sont pas toutes de même taille; ce qui veut dire : pour planifier son projet, donc des sprints, on ne se basera pas sur le nombre de stories à faire dans le sprint. Exemple, le sprint 2 pourrait contenir une seule story alors que le sprint 3 va contenir 3 stories. Dans SCRUM, l'estimation se fait de manière collective et la technique fréquemment utilisée est celle **du planning poker**.
- De nombreux jeux de cartes sont vendus sur internet ou fournis lors de conférences.
  - Une valeur de 0 représente une fonctionnalité déjà mise en place, ou ne demandant pas d'effort particulier.
  - Les valeurs 0.5 et 1 peuvent être utilisées pour des tâches particulièrement simples.
  - 3 et 5 pour des tâches un peu plus complexes.
  - Au-delà, la réalisation demande des travaux plus conséquents ou plus complexes.
  - La valeur 100 représente une grande complexité et un grand nombre d'heures de travail. Il faudra sans doute découper.
  - La carte portant le symbole « ∞ » (infini) représente toute tâche valant plus de 100. Il s'agit donc d'un travail particulièrement long ou complexe nécessitant une attention particulière. Généralement, une fonctionnalité recevant cette note a peu de chance de pouvoir être embarquée dans le sprint à venir.
- Ces chiffres complètent les premiers éléments d'une suite Fibonacci.



<https://www.linkedin.com/pulse/realistic-planning-poker-game-training-teams-agile-estimation-paz/>

## 01 - Appliquer la méthode SCRUM

### Planning Poker



#### Déroulement d'un planning Poker

- Chaque participant reçoit un jeu de cartes.
- Sur chaque carte, il y a une valeur positive pour estimer la story.
  1. Le PO présente la story.
  2. Les membres de l'équipe posent des questions pour clarifier la story.
  3. Tous les participants présentent en même temps la carte choisie pour l'estimation.
  4. L'équipe discute des différences éventuelles entre les estimations.
  5. On recommence jusqu'à une convergence des estimations.
  6. On passe à la prochaine story.
- Pour commencer la séance de planification, il suffit de choisir une story connue de TOUS, (baseline) pour laquelle l'équipe décide en commun de lui fixer une valeur. Et, il est préférable de choisir une story de taille moyenne (3 ou 5) pour laisser une marge vers le bas et vers le haut.

#### Estimation en effort ou en temps

- Vous pouvez estimer les user-stories, en effort. Dans ce cas il faudra déterminer c'est quoi l'effort.

##### Exemple :

- 1 Effort = 5 jours/homme :
- A une durée de 5 jours si le nombre de ressources qui travaillent en parallèle et à temps plein est égal à 1.
- A une durée de 10 jours si le nombre de ressources qui travaillent en parallèle et à mi-temps est égal à 1.
- A une durée de 1 jour si le nombre de ressources qui travaillent en parallèle et à temps plein est égal à 5.

### Question 1

Quels types de projets de développement logiciels peuvent être exécutés par le cadre de gestion de projet Scrum ?

1. Des logiciels complets
2. Projets de clients
3. Sous-systèmes, composants ou parties de systèmes plus importants
4. Tous les types de projets de développement de logiciels
5. Aucune des réponses données

### Réponse Question 1

Quels types de projets de développement logiciels peuvent être exécutés par le cadre de gestion de projet Scrum ?

1. Des progiciels complets
2. Projets de clients
3. Sous-systèmes, composants ou parties de systèmes plus importants
- 4. Tous les types de projets de développement de logiciels**
5. Aucune des réponses données

### Question 2

Quels sont les avantages du cadre Scrum ?

1. Les exigences fines ne sont définies que lorsqu'elles sont vraiment nécessaires
2. Toutes les activités de conception, de construction et de test d'une certaine fonctionnalité sont regroupées en une seule phase
3. Les changements sont attendus et bien accueillis par l'équipe Scrum
4. Toutes les réponses données
5. Aucune des réponses données

### Réponse Question 2

Quels sont les avantages du cadre Scrum ?

1. Les exigences fines ne sont définies que lorsqu'elles sont vraiment nécessaires
2. Toutes les activités de conception, de construction et de test d'une certaine fonctionnalité sont regroupées en une seule phase
3. Les changements sont attendus et bien accueillis par l'équipe Scrum
4. **Toutes les réponses données**
5. Aucune des réponses données

### Question 3

Qu'est-ce qui ne fait PAS partie des fondements du manifeste agile ?

1. Les individus et les interactions plutôt que les processus et les outils
2. Un logiciel fonctionnel plutôt qu'une documentation complète
3. Les processus plutôt que les personnes
4. La collaboration avec les clients plutôt que la négociation de contrats
5. Réagir au changement plutôt que suivre un plan



### Réponse Question 3

Qu'est-ce qui ne fait PAS partie des fondements du manifeste agile ?

1. Les individus et les interactions plutôt que les processus et les outils
2. Un logiciel fonctionnel plutôt qu'une documentation complète
3. **Les processus plutôt que les personnes**
4. La collaboration avec les clients plutôt que la négociation de contrats
5. Réagir au changement plutôt que suivre un plan

#### Question 4

Qu'est-ce qui est défini dans le cadre de Scrum ?

- A. Règles et rôles
  - B. Directives relatives aux documents
  - C. Artéfacts et événements
1. A
  2. B
  3. C
  4. A, B ,C
  5. A, C

### Réponse Question 4

Qu'est-ce qui est défini dans le cadre de Scrum ?

- A. Règles et rôles
  - B. Directives relatives aux documents
  - C. Artéfacts et événements
1. A
  2. B
  3. C
  4. A, B ,C
  5. A, C

#### Question 5

Où sont stockées les exigences du client ?

1. Dans le Backlog de produit
2. Dans le backlog de sprint
3. Dans une base de données
4. Dans une spécification des exigences du produit Scrum.
5. Nulle part. Le Product Owner de Scrum les connaît.

### Réponse Question 5

Où sont stockées les exigences du client ?

1. Dans le Backlog de produit
2. Dans le backlog de sprint
3. Dans une base de données
4. Dans une spécification des exigences du produit Scrum.
5. Nulle part. Le Product Owner de Scrum les connaît

### Question 6

Parmi les rôles principaux suivants, lesquels sont définis par le Scrum Framework ?

- A. Scrum Tester
  - B. The Scrum Team
  - C. Scrum Manager
  - D. Scrum Master
  - E. Scrum Product Owner
1. A, B, C, D, E
  2. B, C, D, E
  3. B, D, E
  4. A, B, D, E
  5. A, B, C, D

### Réponse Question 6

Parmi les rôles principaux suivants, lesquels sont définis par le Scrum Framework ?

- A. Scrum Tester
  - B. The Scrum Team
  - C. Scrum Manager
  - D. Scrum Master
  - E. Scrum Product Owner
1. A, B, C, D, E
  2. B, C, D, E
  3. **B, D, E**
  4. A, B, D, E
  5. A, B, C, D

### Question 7

Que sont les exigences pour Scrum ?

1. Utilisation de JIRA
2. Être discipliné dans l'application du processus
3. Avoir un Scrum Master distinct
4. Avoir des sprints de 2 semaines



### Réponse Question 7

Que sont les exigences pour Scrum ?

1. Utilisation de JIRA
2. Être discipliné dans l'application du processus
3. Avoir un Scrum Master distinct
4. Avoir des sprints de 2 semaines

### Question 8

Où pouvez-vous voir la progression de votre équipe Scrum?

1. Scrum boards
2. Backlog
3. Backlog de sprint
4. Backlog boards

### Réponse Question 8

Où pouvez-vous voir la progression de votre équipe Scrum?

1. Scrum boards
2. Backlog
3. Backlog de sprint
4. Backlog boards

### Question 9

Combien de temps doit durer un Scrum/Stand-up quotidien ?

1. 1 heure
2. Une demi-journée
3. 5 minutes
4. 15 minutes

### Réponse Question 9

Combien de temps doit durer un Scrum/Stand-up quotidien ?

1. 1 heure
2. Une demi-journée
3. 5 minutes
4. **15 minutes**

### Question 10

Toutes les barrières, tous les obstacles et toutes les interférences sont éliminés par le \_\_\_\_\_

1. Product Owner
2. Scrum Master
3. Chef de projet
4. Équipe de développement

### Réponse Question 10

Toutes les barrières, tous les obstacles et toutes les interférences sont éliminés par le \_\_\_\_\_

1. Product Owner
2. **Scrum Master**
3. Chef de projet
4. Équipe de développement



## CHAPITRE 2

### APPLIQUER LA MÉTHODE SAFE

**Ce que vous allez apprendre dans ce chapitre :**

- Vous initier à la gestion de portefeuille SAFE ;
- Vous appuyer sur les trains de release agile (ART), la planification des itérations et l'organisation de travail avec les autres équipes dans le train.



**05 heures**



## CHAPITRE 2

### APPLIQUER LA MÉTHODE SAFE

1. **Framework SAFE**
2. Trains de release agile (ART)



## 02 - Appliquer la méthode SAFE

### Framework SAFE



#### Définitions

- Le Scaled Agile Framework® (SAFe®) est un ensemble de modèles d'organisation et de flux de travail permettant de mettre en œuvre des pratiques agiles à l'échelle d'une entreprise. Le cadre est un ensemble de connaissances qui comprend des conseils structurés sur les rôles et les responsabilités, la façon de planifier et de gérer le travail, et les valeurs à respecter.
- SAFe favorise l'alignement, la collaboration et la livraison à travers un grand nombre d'équipes agiles. Il s'articule autour de trois grands domaines de connaissances : le développement logiciel agile, le développement de produits allégés et la pensée systémique.
- Au fur et à mesure que la taille des entreprises augmente, SAFe fournit une approche structurée pour la mise à l'échelle agile. Il existe quatre configurations dans SAFe pour s'adapter à différents niveaux d'échelle : Essential SAFe, Large Solution SAFe, Portfolio SAFe et Full SAFe.
- Dean Leffingwell et Drew Jemilo ont publié SAFe en 2011 pour aider les organisations à concevoir de meilleurs systèmes et logiciels qui répondent mieux aux besoins changeants des clients. À l'époque, les équipes utilisaient des processus de gestion de projet traditionnels pour livrer des logiciels. Mais à mesure que le besoin de répondre rapidement à l'évolution des conditions du marché augmentait, de nouveaux cadres ont émergé pour aider les entreprises à améliorer la livraison de solutions à l'échelle de leur entreprise, et SAFe est né. Aujourd'hui, SAFe est l'un des cadres de livraison agiles les plus populaires, et la communauté mondiale des praticiens de SAFe continue de le faire évoluer.

## 02 - Appliquer la méthode SAFE

### Framework SAFE



### Principes et valeurs fondamentaux

#### Valeurs fondamentales

- Les valeurs fondamentales de SAFe décrivent la culture que le leadership doit favoriser et la façon dont les gens doivent se comporter au sein de cette culture afin d'utiliser efficacement le cadre de travail.

#### Alignement

- SAFe exige que les entreprises mettent en place des cadences de planification et de réflexion à tous les niveaux de l'organisation. Grâce à ces cadences, tout le monde comprend l'état actuel de l'entreprise, les objectifs et la manière dont chacun doit agir pour atteindre ces objectifs. En synchronisant régulièrement les personnes et les activités, tous les niveaux du portefeuille restent alignés. L'information circule vers le haut et vers le bas en temps opportun, contrairement aux structures traditionnelles de commandement et de contrôle.

#### Qualité intégrée

- Dans le cadre de SAFe, l'agilité ne doit jamais se faire au détriment de la qualité. SAFe exige des équipes à tous les niveaux qu'elles définissent ce que signifie "terminé" pour chaque tâche ou projet et qu'elles intègrent des pratiques de développement de qualité dans chaque accord de travail. Selon SAFe, il existe cinq dimensions clés de la qualité intégrée : le flux, la qualité de l'architecture et de la conception, la qualité du code, la qualité du système et la qualité des versions.

## 02 - Appliquer la méthode SAFE

### Framework SAFE



### Principes et valeurs fondamentaux

#### Transparence

- SAFe encourage les comportements visant à instaurer la confiance, notamment la planification du travail en lots plus petits afin que les problèmes puissent être détectés plus tôt, la visibilité en temps réel de l'avancement du backlog à tous les niveaux et les rituels d'inspection et d'adaptation.

#### Exécution de programme

- L'exécution du programme est le cœur de SAFe et alimente tout le reste du cadre. Les équipes et les programmes doivent être en mesure de fournir régulièrement des logiciels de qualité et fonctionnels, ainsi que de la valeur commerciale.

#### Leadership

- SAFe exige un comportement de leadership lean-agile car seuls les leaders peuvent changer le système et créer l'environnement nécessaire à l'adoption de toutes les valeurs fondamentales.

## 02 - Appliquer la méthode SAFE

### Framework SAFE



#### Itérations

- Le processus Portefeuille SAFe 4.0 définit une itération Feuille de route du flux de valeurs de niveau supérieur, qui contient une itération Cadence d'incrément de programme du flux de valeurs. L'itération Cadence d'incrément de programme du flux de valeurs est composée de plusieurs itérations Incrément de programme (PI).

#### Rôles

- Le processus SAFe Portefeuille 4.0 définit le niveau de portefeuille et les rôles utilisateur au niveau programme suivants :

- Tableau 1. Rôles de processus Portefeuille SAFe 4.0 : niveau Portefeuille*

Rôle	Description
Gestionnaire de portefeuille	Dirige l'équipe de gestion du portefeuille, pilote des programmes volumineux du développement au déploiement, en fournissant des rapports sur leur statut et budget.
Analyste métier	Définit les initiatives (epics métier) et détermine l'impact sur les flux de valeurs internes et externes de l'entreprise. SAFe dénomme les analystes métier Propriétaires d'Epic.
Directeur informatique	Cadre responsable du portefeuille.
Partie prenante	Membre de l'équipe de gestion du portefeuille qui contribue à l'émergence de nouvelles idées dans le système.
Architecte informatique	Définit une vision technologique et des scénarios d'implémentation qui épaulent la stratégie métier. Ce rôle articule généralement les epics architecturaux et collabore avec les architectes système au niveau des programmes.

Rôles

• *Tableau 2. Rôles de processus Portefeuille SAFe 4.0 : niveau Flux de valeurs*

Rôle	Description
Ingénieur du flux de valeurs	Facilite les processus et l'exécution des flux de valeurs ; fait remonter les entraves ; gère les risques ; et permet un apport de valeur et une amélioration continue.
Architecte de solution	Responsable de la conception globale de la solution en termes d'architecture et d'ingénierie. Aligne le flux de valeurs et les trains d'éditions Agile pour obtenir une vision technologique et architecturale commune.
Gestionnaire de solution	Responsable du développement et de la hiérarchisation du journal des éléments en attente du flux de valeurs. Travaille avec les parties prenantes pour comprendre leurs besoins ; définit une vision et une feuille de route ; définit des exigences et achemine le travail via le Kanban de flux de valeurs.

• *Tableau 3. Rôles de processus Portefeuille SAFe 4.0 : niveau Programme*

Rôle	Description
Responsable de programme	Rôle défini dans un portefeuille pour permettre aux rôles Responsable produit, Ingénieur de train d'édition et Propriétaire fonctionnel de participer à la planification du portefeuille.

## 02 - Appliquer la méthode SAFE

### Framework SAFE



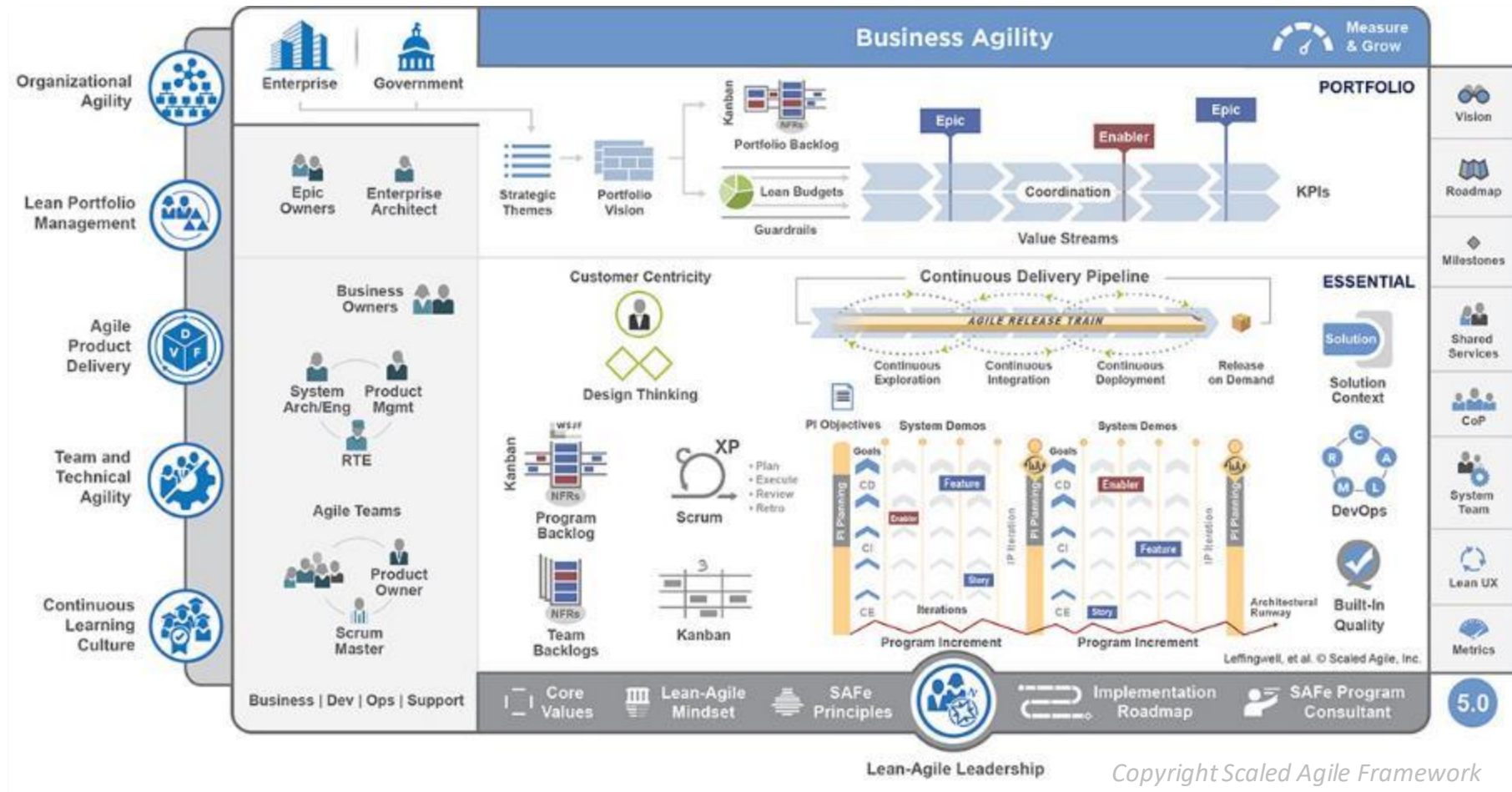
#### Scaled Agile Framework

- Le Scaled Agile Framework ou SAFe, est un ensemble de modèles d'organisation et de flux de valeur destiné à la mise en œuvre d'une méthodologie agile à l'échelle de l'entreprise. SAFe cherche à répondre au besoin des grandes organisations de faire travailler entre elles plusieurs équipes agiles. Il vise également à la mise en place d'une culture agile à tous les niveaux de l'entreprise.

## 02 - Appliquer la méthode SAFE

### Framework SAFE

### Scaled Agile Framework





## 02 - Appliquer la méthode SAFE

### Framework SAFE



### Scaled Agile Framework

Le cadre SAFe a été conçu autour de trois principaux corps de connaissances :

1. Le développement logiciel agile
2. Le développement de produits lean
3. La pensée systémique

Et repose sur les sept compétences de base de l'entreprise Lean.

### Types d'élément de travail

Le processus Portefeuille SAFE 4.0 définit les types d'éléments de travail suivants :

- **Epic de portefeuille** : Les epics de portefeuille sont des initiatives d'entreprise qui affectent plusieurs organisations et incréments de programmes. Un epic de portefeuille affecte plusieurs trains d'édition.
- **Epic de flux de valeurs** : Les epics de flux de valeurs sont des initiatives qui sont suffisamment étendues pour justifier une analyse et une étude de rentabilité simple mais qui sont limitées à un flux de valeurs unique.
- **Fonctionnalité** : Les fonctionnalités sont comparables à des fonctions mais permettent la mise en œuvre de comportements de niveau supérieur de la solution. Elles incluent souvent plusieurs trains d'édition Agile (ART). Elles sont gérées dans le journal des éléments en attente du flux de valeurs et sont adaptées à un incrément de programme afin que chaque incrément de programme apporte de la valeur à la solution.
- **Jalon d'apprentissage** : Les jalons d'apprentissage permettent de représenter des points de progression spécifiques dans la chronologie et peuvent être très utiles pour mesurer et surveiller la progression et les risques d'un programme.
- **Objectif d'incrément de programme** : décrit un objectif spécifique avec une valeur métier planifiée et évaluée pour les capacités, fonctions et cas d'utilisation distribués dans un incrément de programme ou une itération spécifiée(e). L'objectif d'incrément de programme peut se trouver au niveau du programme, de l'équipe ou du flux de valeurs.
- **Tâche** : Définit une unité d'œuvre planifiée pour un incrément de programme et est estimée en heures.
- **Incident** : Définit un bogue, une erreur ou un défaut dans la solution.
- **Risque** : Un risque est un événement potentiel ou une situation à venir susceptible d'affecter, d'empêcher ou de limiter la réussite d'un projet. Les risques liés aux projets peuvent être considérés comme des menaces ou des opportunités.
- **Rétrospective** : Une rétrospective est un événement permettant de discuter des réussites, des échecs et des mesures à prendre dans la période à venir. Utilisez le type d'élément de travail Rétrospective pour garantir que l'événement se produit et assurer le suivi des commentaires et des plans de l'équipe

## 02 - Appliquer la méthode SAFE

### Framework SAFE



### Principes SAFe

Les principes du Scaled Agile Framework ont pour but d'améliorer l'entreprise dans son ensemble en inspirant une prise de décision lean-agile à travers les frontières fonctionnelles et organisationnelles. Les principes sont destinés à influencer les décisions non seulement des dirigeants et des gestionnaires, mais aussi de tous les membres de l'organisation, et à conditionner leur état d'esprit pour qu'ils passent de la pensée traditionnelle en cascade à la pensée agile allégée, où des pratiques comme la gestion de portefeuille allégée sont appliquées.

#### Principe n° 1 : Adopter une vision économique

- Inspiré par les théories sur le flux de développement de produits tirées des livres les plus vendus de Donald Reinertsen, l'obtention du délai d'exécution durable le plus court exige que chaque personne de la chaîne de décision comprenne les implications économiques des retards. Livrer tôt et souvent n'est pas toujours suffisant. Selon SAFe, le fait de séquencer les tâches pour en tirer le maximum d'avantages, de comprendre les compromis économiques et d'opérer dans le cadre de budgets réduits sont autant de responsabilités qui doivent être partagées au sein de l'organisation. De nombreux concepts et outils sont tirés des théories de Reinertsen sur le flux de développement des produits.

#### Principe n°2 : Appliquer la pensée systémique

- SAFe encourage les personnes qui utilisent le cadre à appliquer la pensée systémique à trois domaines clés : la solution elle-même, l'entreprise qui construit le système et les flux de valeur. Les solutions peuvent faire référence à des produits, des services ou des systèmes fournis au client, qu'ils soient internes ou externes à l'entreprise.
- Les solutions de grande envergure comportent de nombreux composants interconnectés, de sorte que les membres de l'équipe doivent avoir une perspective de plus haut niveau sur la façon dont leur partie s'intègre dans le tableau d'ensemble. En pensant à l'entreprise qui construit le système, les personnes qui suivent SAFe doivent prendre en compte les personnes, la gestion et les processus de l'organisation. Ainsi, si une organisation cherche à optimiser la façon dont les gens travaillent, elle peut avoir besoin d'éliminer les silos, de devenir interfonctionnelle et de former de nouveaux accords de travail avec les fournisseurs et les clients. Enfin, l'entreprise doit définir clairement comment la valeur passe du concept à l'argent dans les flux de valeur du développement de solutions. Les dirigeants et la direction doivent maximiser le flux de valeur au-delà des frontières fonctionnelles et organisationnelles.

### Principes SAFe

#### Principe n° 3 : Assumer la variabilité ; préserver les options

- Par défaut, la conception de systèmes et de logiciels est un exercice incertain. Ce principe traite de l'incertitude en introduisant le concept de conception par ensembles, qui demande de conserver de multiples exigences et options de conception pendant une période plus longue du cycle de développement. La conception par ensembles s'appuie également sur des données empiriques pour se concentrer sur l'option de conception finale à un stade plus avancé du processus.
- La conception par ensembles permet d'éclairer la prise de décision en période d'incertitude en identifiant les options et les résultats escomptés, un peu comme un pari stratégique. Le concept d'intégration de "jalons d'apprentissage", qui fait référence à une date limite pour une décision, est essentiel à la conception par ensembles. Plus les équipes apprennent au fil du temps, plus elles peuvent éliminer de choix. Plus elles éliminent de choix, plus il est facile d'identifier la meilleure voie à suivre et de produire le meilleur résultat possible pour les clients.

#### Principe n°4 Construire de manière incrémentielle avec des cycles d'apprentissage rapides et intégrés

- Comme le principe n° 3, ce principe aborde le risque et l'incertitude par le biais de jalons d'apprentissage. Il ne suffit pas que chaque élément du système s'avère fonctionnel, il faut considérer l'ensemble du système pour évaluer la faisabilité des choix de conception actuels. Les points d'intégration doivent être planifiés à une cadence régulière pour accélérer les cycles d'apprentissage. Ces points d'intégration sont un exemple du cycle planifier-faire-vérifier-ajuster de Walter A Shewhart, un cadre pour l'amélioration continue de la qualité et un mécanisme pour contrôler la variabilité du développement. Les travaux de Shewart et ceux qu'il a inspirés se retrouvent souvent dans SAFe.

#### Principe n° 5 : Baser les jalons sur une évaluation objective des systèmes en fonctionnement

- La démonstration d'un système opérationnel réel constitue une meilleure base pour la prise de décision qu'un document sur les exigences ou une autre évaluation superficielle du succès. L'inclusion des parties prenantes dans ces décisions de faisabilité dès le début favorise l'instauration de la confiance et la pensée systémique.

## 02 - Appliquer la méthode SAFE

### Framework SAFE



### Principes SAFe

#### Principe n° 6 : Visualiser et limiter les en-cours, réduire la taille des lots et gérer la longueur des files d'attente.

- Limiter le travail en cours aide les parties prenantes à voir exactement comment le travail se déroule.
- Les trois éléments de ce principe représentent les principaux moyens de maximiser le débit et d'accélérer la livraison de la valeur - ou, en d'autres termes, de mettre en œuvre le "flux". Comme le dit le proverbe : "Comment mange-t-on un éléphant ? Une bouchée à la fois".
- Lorsque vous appliquez cela au développement de logiciels, cela signifie limiter la quantité de travail qui se chevauche, la complexité de chaque élément du travail et la quantité totale de travail abordée à un moment donné. Les lots de petite taille permettent de valider en permanence que le travail va dans la bonne direction. Et la gestion de la longueur des files d'attente ...
- Ce principe vise à offrir des conseils sur la façon de l'optimiser pour obtenir les meilleurs résultats.

#### Principe n° 7 Appliquer la cadence, synchroniser avec la planification inter-domaines

- Les équipes agiles appliquent naturellement la cadence par le biais de sprints ou d'itérations. La création d'une cadence pour toutes les questions possibles réduit la complexité, répond à l'incertitude, développe la mémoire musculaire, renforce la qualité et incite à la collaboration. La synchronisation de ces cadences permet aux personnes et aux activités de se déplacer comme les rouages d'une roue où les informations acquises informent les décisions et la planification incrémentale.

#### Principe n° 8 Libérez la motivation intrinsèque des travailleurs du savoir

- Inspiré par l'influent consultant en management Peter Drucker et l'auteur Daniel Pink, ce principe est l'un de nos préférés ! Il s'agit de libérer le potentiel des équipes et d'aider les dirigeants à adopter une perspective de coaching et de service à leurs équipes plutôt qu'une mentalité de commandement et de contrôle.

#### Principe n° 9 : Décentraliser la prise de décision

- Réduire la longueur des files d'attente et adopter une approche économique en décentralisant la prise de décision, donne aux équipes l'autonomie dont elles ont besoin pour accomplir leur travail. Les dirigeants doivent conserver leur pouvoir de décision pour les sujets d'importance stratégique et permettre aux équipes de faire des choix éclairés pour tout le reste.

Scrum vs SAFe : principales différences

Eléments	Scrum	SAFe
Contexte de mise en œuvre	<ul style="list-style-type: none"><li>Scrum est un cadre de travail agile pour les projets.</li><li>Il fonctionne pleinement lorsqu'il est mis en place dans des entreprises possédant une structure organisationnelle simple.</li></ul>	<ul style="list-style-type: none"><li>SAFe est un cadre de travail agile généralement utilisé lorsque de grandes entreprises souhaitent étendre la « méthode de travail agile » au-delà du périmètre des équipes opérationnelles.</li></ul>
Structure et rôles des équipes	<ul style="list-style-type: none"><li>L'équipe Scrum est constituée généralement de 3 à 9 personnes ayant les rôles suivants :</li><li>Product Owner (Propriétaire du produit)</li><li>Scrum Master</li><li>Équipe de développement</li></ul>	<ul style="list-style-type: none"><li>SAFe possède différents rôles clés sur 4 niveaux différents. Parmi les rôles clés, nous avons :<ul style="list-style-type: none"><li>Releaser Train Engineer</li><li>Program Management</li><li>Value Stream Engineer</li><li>Solution Architect</li><li>Epic Owners</li><li>Etc</li></ul></li><li>NDRL : SAFe est un modèle pouvant être mis en place à partir de 2 niveaux organisationnel. Ce modèle est compatible avec Scrum et ses rôles associés.</li></ul>
Dépendances d'équipe	<ul style="list-style-type: none"><li>Dans Scrum, le niveau de dépendance entre les équipes a tendance à être inférieur à celui de SAFe. Les équipes de développement ont la liberté de s'organiser et de gérer leur propre travail.</li></ul>	<ul style="list-style-type: none"><li>SAFe nécessite un niveau élevé de coordination et d'alignement entre les équipes et les niveaux de gestion, ce qui accroît la dépendance dans l'activité quotidienne.</li><li>A noter, ceci créé des limitations dans l'adaptation et l'innovation du système.</li></ul>
Cérémonies et délais	<ul style="list-style-type: none"><li>Dans Scrum, les projets sont menés durant une période de temps, appelé « sprint », où différentes cérémonies récurrentes ont lieu. Un cycle dure de 1 à 4 semaines. On parle de cycle court.</li></ul>	<ul style="list-style-type: none"><li>SAFe suit plus ou moins la même structure. Le temps est décomposé en cycle de 10 semaines (itération au niveau de l'équipe), soit 4 sprints de 2 semaine + 1 sprint de préparation à la nouvelle itération programme.</li><li>Chaque sprint donne l'occasion de suivre l'avancement du programme, de vérifier le travail accompli et ajuster les choses.</li></ul>

## CHAPITRE 2

### APPLIQUER LA MÉTHODE SAFE

1. Framework SAFE
2. **Trains de release agile (ART)**

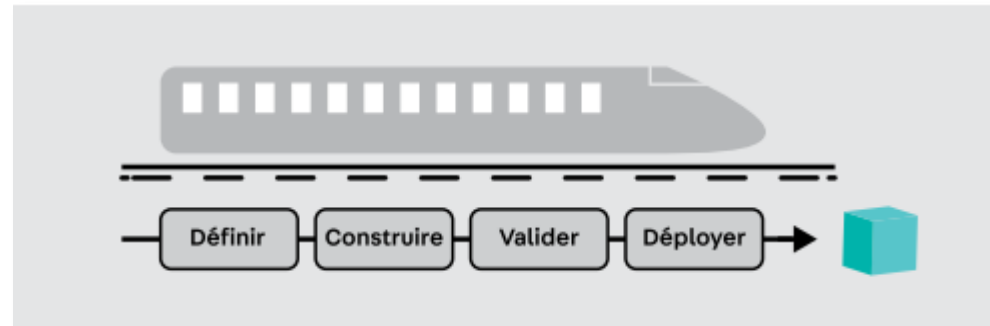


## 02 - Appliquer la méthode SAFE

### Trains de release agile (ART)

#### Les trains Agile (Agile Release Train – ART)

- Pour livrer de la valeur, SAFe introduit le concept de ART pour Agile release train. Un ensemble de gens ayant toutes les expertises nécessaires pour livrer de la valeur.



- Ces trains permettent d'aligner les équipes autour d'objectifs d'affaires et technologiques communs. Chaque train est une mini entreprise en soi, qui comprend le plus souvent entre 50 et 125 personnes, qui planifient, s'impliquent, développent et déploient ensemble.
- Ce genre d'organisation du travail vise à briser les silos (affaires, matériel, logiciel, QA, Opérations...) pour créer des synergies plus horizontales, et ainsi éviter les écueils habituels (difficulté de communication, enjeux de politique interne qui contreviennent à la collaboration, livraison de valeur réduite à cause des retards, etc.).

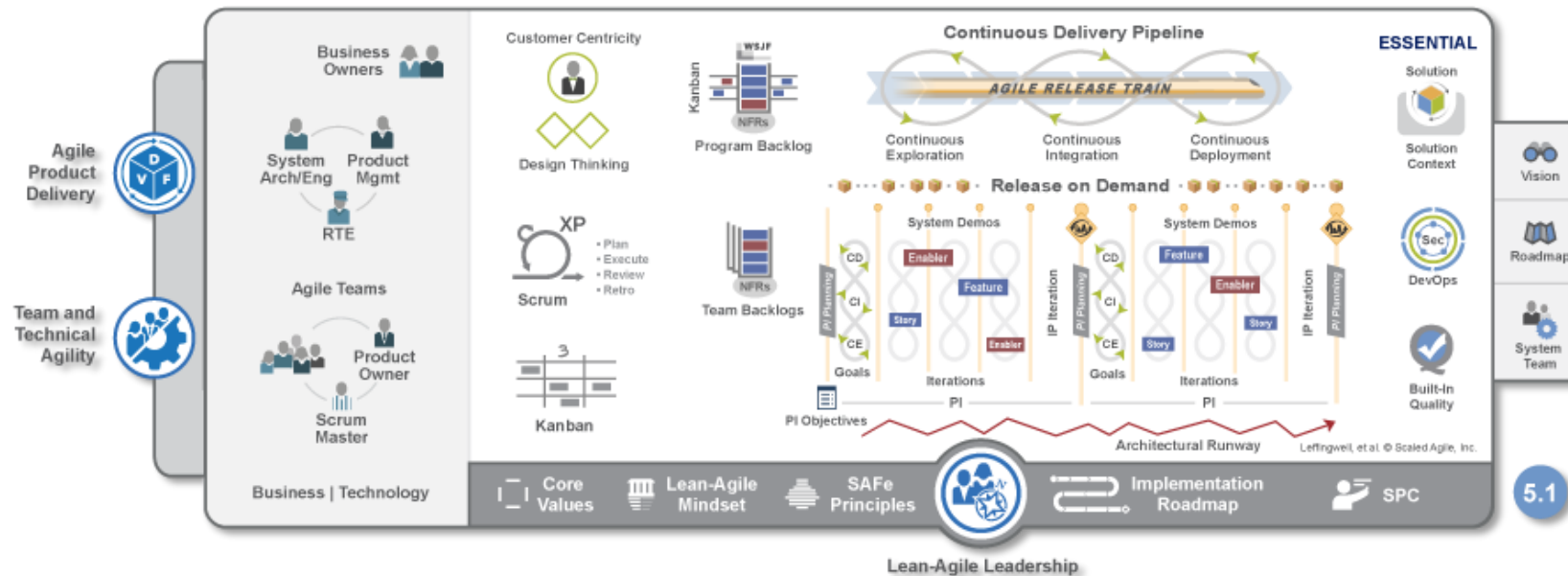


## 02 - Appliquer la méthode SAFE

### Trains de release agile (ART)

### Les trains Agile (Agile Release Train - ART)

- Pour appuyer le bon fonctionnement du train, SAgE propose un ensemble d'événements. Parmi ces événements, il y a le PI planning (PI = Program Increment). C'est une rencontre de planification de 2 jours qui revient à tous les trimestres (environ) et qui permet aux équipes agiles de se synchroniser et d'identifier les risques et les dépendances. Lors des rencontres de PI Planning il y a un backlog commun qui est partagé (et donc discutable). Par la suite, les équipes peuvent partager leur tableau d'équipe (qui énonce les objectifs à haut niveau jusqu'au prochain PI planning) et surtout il y a le tableau de Programme (et de dépendances) où les équipes indiquent sommairement sur quoi elles travaillent, ce qui leur permet de visualiser rapidement si d'autres équipes vont être impactées et d'en discuter sur le champ. C'est donc un outil qui centralise les liaisons (dependancies) pour permettre ensuite aux équipes de travailler de façon autonome. Enfin il est possible aux Business Owners d'avoir accès à ces divers outils et donc de favoriser la transparence et la rétroaction.



## 02 - Appliquer la méthode SAFE

### Trains de release agile (ART)



#### Les trains Agile (Agile Release Train - ART)

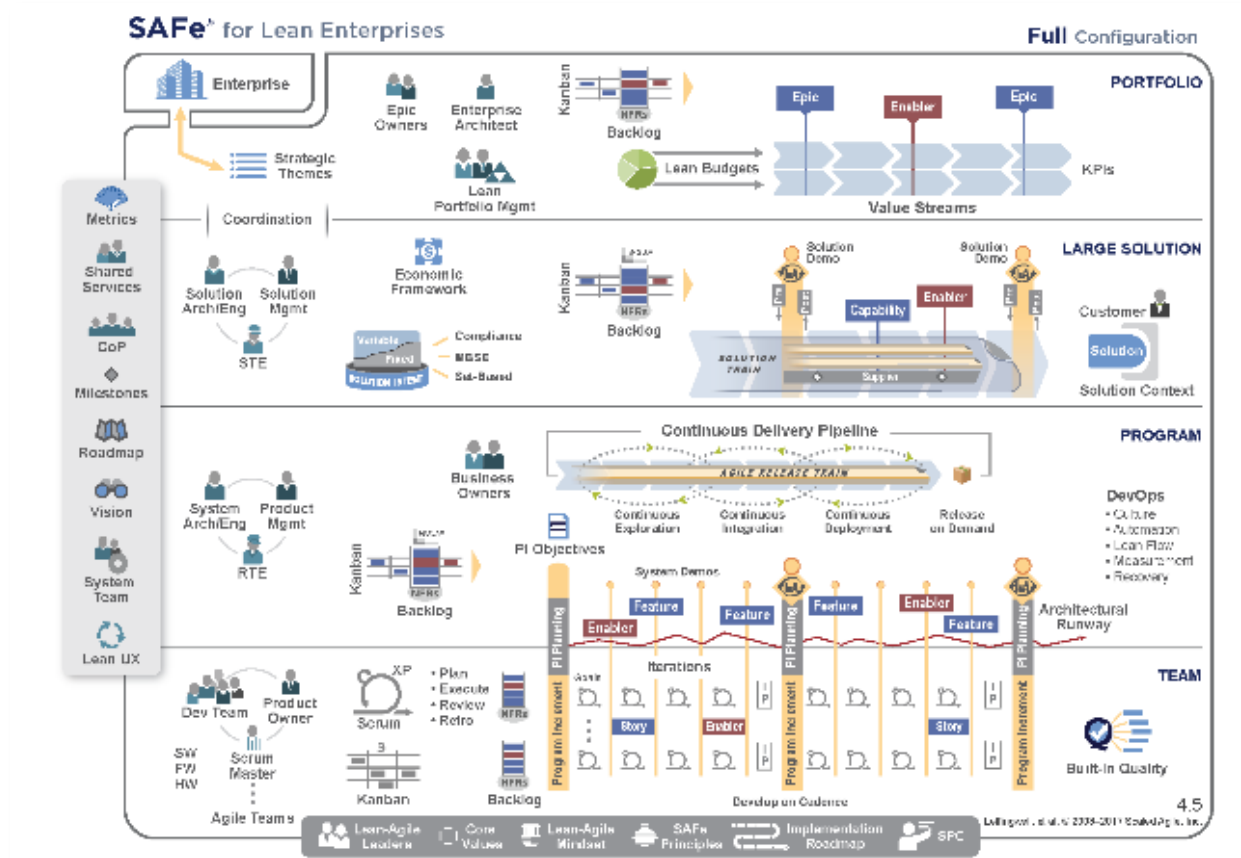
- Et il y a de nombreux rôles qui viennent appuyer le cadre de travail, en particulier :
  - **Le RTE (Release Train Engineer)**, le conducteur de train (sorte de Scrum Master en chef), dont le rôle consiste à veiller à la performance du train. Pour ce faire, il devra interagir avec les parties prenantes, faciliter la gestion des risques, faciliter les rencontres du train dont le PI planning, coacher pour garder les pratiques et l'état d'esprit Agile, et encourager sans cesse l'amélioration. En un mot c'est un facilitateur et pas un gestionnaire. Dans son rôle il s'appuie aussi sur le PM et le SAE.
  - **Le PM (Product Manager)**, coordonne la vision des divers Product Owners. Il est responsable de ce qui va être « fabriqué », en réponse aux attentes consommateurs et selon le cycle de vie du produit. Il agit avec de nombreuses personnes en amont pour clairement identifier les attentes et en conséquence bâtir une feuille de route comprenant les fonctionnalités attendues. Il aide le train à livrer de la valeur en continue.
  - **Le SAE (System Architect Engineer)**, est celui qui s'assure que le système dans lequel la solution va évoluer, fonctionne bien. Cela sous-entend son implication à l'élaboration de la solution, à la validation des orientations techniques, à l'évaluation des alternatives possibles et à l'établissement d'un pipeline de livraison continue.
  - **Le Business Owner**, qui est directement responsable de ce que le train va livrer, en termes de gouvernance, de conformité et de retour sur investissement. Le BO est souvent un petit groupe composé des parties prenantes, dont la responsabilité première est d'assurer la gouvernance technique et d'affaires, la conformité et le retour sur investissement, de la solution qui sera livrée par l'ART.

## 02 - Appliquer la méthode SAFE

### Trains de release agile (ART)

### Les trains Agile (Agile Release Train – ART)

Safe propose plusieurs configurations, comme on peut le voir ici :



Copyright Scaled Agile Framework

## 02 - Appliquer la méthode SAFE

### Trains de release agile (ART)



#### Les trains Agile (Agile Release Train – ART)

- **Essential** : il s'agit de la configuration de base du cadre de travail. Il comprend tous les éléments minimums nécessaires pour avoir du succès avec SAFe.
- **Large Solution** : Il s'agit de la configuration utilisée par les organisations qui construisent de très grosses et complexes solutions. À titre d'exemples, plusieurs trains (incluant notamment des fournisseurs externes) travailleront à la réalisation d'une solution. Ici, dans cette configuration la configuration Portfolio n'est pas requise.
- **Portfolio** : La configuration Portfolio comprend les éléments en lien avec la gouvernance Lean, la stratégie du portefeuille et d'investissement ainsi que l'Agile portfolio operations qui va coordonner et supporter l'exécution, décentraliser des programmes et l'excellence opérationnelle.
- **Full SAFe** : Il s'agit de la configuration la plus complète permettant de réaliser et d'intégrer de grosses et complexes solutions nécessitant des centaines de personnes pour le développement et la maintenance.

Ces rôles (et les autres) permettent des livraisons cadencées, aux moments prévus, même si l'on est dans un mode agile, propre aux incréments. Bien qu'il s'agisse de train de livraison, les backlogs restent la colonne vertébrale, du plus bas au plus haut niveau.

## 02 - Appliquer la méthode SAFE

### Trains de release agile (ART)



#### Et le SAFe Scrum Master dans tout ça ?

- Soyons clairs, dans Safe on fait du Scrum Le rôle initial du Scrum Master consiste à :
  - Faciliter les évènements d'équipes
  - Coacher l'équipe pour améliorer ses processus
  - Enlever les bloquants
  - Protéger l'équipe en lui faisant garder son attention sur l'objectif
  - Être un leader au service de l'organisation, de l'équipe, des individus
- Et en SAFe ? Tous les éléments cités plus haut plus les suivants :
  - Travailler avec le RTE (et le SAE) pour s'assurer que le train rencontre ses objectifs
  - Se coordonner avec les autres Scrum Masters impliqués dans l'ART
  - Faciliter les estimations des équipes auto-gérées pour qu'elles rencontrent leurs livrables
  - Éduquer les équipes en Scrum, Extrem Programming, Kanban, etc. pour s'assurer que la méthode convenue est bien suivie
  - Aider les équipes à opérer dans une grande organisation

### Question 1

Quel est le rôle de Gestionnaire de portefeuille dans le processus Portefeuille SAFe ?

1. Dirige l'équipe de gestion du portefeuille, pilote des programmes volumineux du développement au déploiement, en fournissant des rapports sur leur statut et budget.
2. Définit une vision technologique et des scénarios d'implémentation qui épaulent la stratégie métier. Ce rôle articule généralement les épics architecturaux et collabore avec les architectes système au niveau des programmes.
3. Définit les initiatives (épics métier) et détermine l'impact sur les flux de valeurs internes et externes de l'entreprise. SAFe dénomme les analystes métier Propriétaires d'Epic.
4. Cadre responsable du portefeuille.

### Réponse Question 1

Quel est le rôle de Gestionnaire de portefeuille dans le processus Portefeuille SAFe ?

1. **Dirige l'équipe de gestion du portefeuille, pilote des programmes volumineux du développement au déploiement, en fournissant des rapports sur leur statut et budget.**
2. Définit une vision technologique et des scénarios d'implémentation qui épaulent la stratégie métier. Ce rôle articule généralement les épics architecturaux et collabore avec les architectes système au niveau des programmes.
3. Définit les initiatives (épics métier) et détermine l'impact sur les flux de valeurs internes et externes de l'entreprise. SAFe dénomme les analystes métier Propriétaires d'Epic.
4. Cadre responsable du portefeuille.

### Question 2

Quelle définition de "Done" est requise pour la revue d'itération ?

1. Release
2. Incrément d'équipe
3. Incrément de système
4. Incrément de solution



### Réponse Question 2

Quelle définition de "Done" est requise pour la revue d'itération ?

1. Release
- 2. Incrément d'équipe**
3. Incrément de système
4. Incrément de solution

### Question 3

Quel est l'objectif principal des thèmes stratégiques ?

1. Déterminer l'ordre dans lequel les epics doivent être exécutés.
2. Favoriser une mise en œuvre progressive dans l'entreprise.
3. Relier le portefeuille à la stratégie commerciale de l'entreprise.
4. Définir la séquence des étapes utilisées pour fournir de la valeur au client.

### Réponse Question 3

Quel est l'objectif principal des thèmes stratégiques ?

1. Déterminer l'ordre dans lequel les epics doivent être exécutés.
2. Favoriser une mise en œuvre progressive dans l'entreprise.
3. **Relier le portefeuille à la stratégie commerciale de l'entreprise.**
4. Définir la séquence des étapes utilisées pour fournir de la valeur au client.

#### Question 4

Quel est l'objectif principal de la prise de décision décentralisée ?

1. Réduire le coût des retards.
2. Résoudre les dépendances.
3. Permettre un flux de valeur plus rapide.
4. Augmenter l'alignement.

### Réponse Question 4

Quel est l'objectif principal de la prise de décision décentralisée ?

1. Réduire le coût des retards.
2. Résoudre les dépendances.
- 3. Permettre un flux de valeur plus rapide.**
4. Augmenter l'alignement.

### Question 5

Quel est le résultat d'une longueur de file d'attente plus courte ?

1. Une qualité inférieure
2. Un risque accru
3. Moins de variabilité
4. Des temps de cycle plus longs

### Réponse Question 5

Quel est le résultat d'une longueur de file d'attente plus courte ?

1. Une qualité inférieure
2. Un risque accru
3. **Moins de variabilité**
4. Des temps de cycle plus longs

### Question 6

Que permet le principe SAFE® n°3, "Assumer la variabilité ; préserver les options" ?

1. De meilleurs résultats économiques
2. Traçabilité des spécifications
3. Conception anticipée des systèmes
4. Définition plus forte de ce qui est fait



### Réponse Question 6

Que permet le principe SAFe® n°3, "Assumer la variabilité ; préserver les options" ?

1. De meilleurs résultats économiques
2. Traçabilité des spécifications
3. Conception anticipée des systèmes
4. Définition plus forte de ce qui est fait

1. Atlassian, "Agile with Atlassian Jira"
2. Team Asana, "Les secrets d'une bonne analyse des besoins, pour davantage d'efficacité d'équipe", 2022, <https://asana.com/fr/resources/needs-assessment>
3. Saliha Yacoub, "Projet dirigé, Scrum, pour la gestion de projet", 2020. [http://www.salihayacoub.com/420Kbe/Theorie/Gestion\\_de\\_projetsv1.pdf](http://www.salihayacoub.com/420Kbe/Theorie/Gestion_de_projetsv1.pdf)
4. Aïm, Roger, "L'essentiel de la gestion de projet : Avec un index franco-anglais des principales expressions Ed. 14", 2022
5. Technologia, "SAFe pour livrer des solutions complexes"
6. Mountain Goat, "An Introduction to Scrum"
7. Gérard Casanova - Denis Abécassis, "Gestion de projet - réaliser le diagramme de PERT"
8. Kent Beck et al., "Manifeste pour le développement Agile de logiciels"