Le réseau de transport d'électricité

Comprendre l'Agile et sa déclinaison à RTE

Date : 17/01/2024

Accessibilité : RTE

Version : 3

SOMMAIRE

A. Qu’est-ce que l’agile?

- La méthode agile en quelques mots
- Quand et pourquoi choisir de développer en Agile?

B. Le fonctionnement de l’équipe agile chez RTE

- Les rôles
- Le déroulé d’un projet agile et les instances opérationnelles spécifiques à l’agile

C. Comment cadrer et spécifier mon projet en agile

- Les spécifications en agile
- Comment réaliser le Story mapping pour obtenir la story map ?
- Les différents types de user story, au cœur de la réussite du projet

D. Comment piloter mon projet agile

- Business Value et Story Points
- Les outils de pilotage de la performance

E. Et si plusieurs projets en agile se coordonnent?

- L’agilité à l’échelle à RTE
- Les rôles spécifiques à l’agilité à l’échelle adaptés à RTE
- Les rituels

A QU’EST CE QUE L’AGILE?

1 LA METHODE AGILE EN QUELQUES MOTS

L'AGILE EN QUELQUES MOTS

- L'agile est une méthode de gestion et de développement projet reposant sur des **cycles itératifs courts** appelés « **Sprints** » qui permet des **livraisons progressives** de fonctionnalités **aux utilisateurs**, les équipes travaillant de manière **itérative** et **incrémentale** et se basant sur les retours réguliers des utilisateurs pour adapter et améliorer le produit.

- La méthode valorise, d'après le manifeste agile :
  - **Les individus et leurs interactions**, de préférence aux processus et aux outils,
  - Des **solutions opérationnelles**, de préférence à une documentation exhaustive,
  - La **collaboration avec les clients**, de préférence aux négociations contractuelles,
  - La **réponse au changement**, de préférence au respect d'un plan

LES GRANDS PRINCIPES

Livrer souvent :

Pour moins se tromper

ou se tromper plus vite

Gérer les risques :

Validation régulière des livraisons

Réduire les échecs et augmenter la qualité :

Tester à chaque itération le technique et fonctionnel

Accroître la satisfaction utilisateur :

Priorisation de la valeur métier & validation fréquente

Limiter l'effet tunnel :

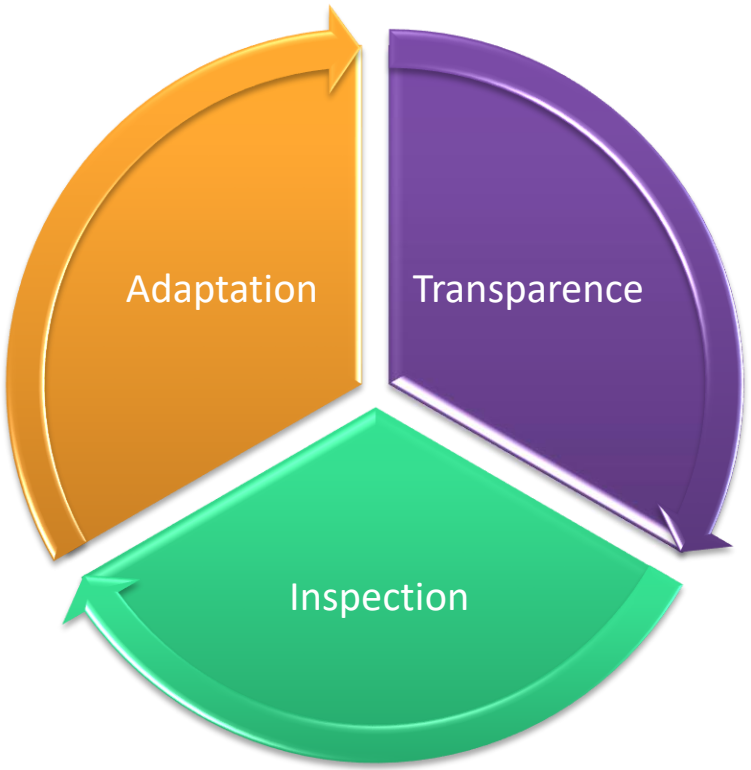
Livrer un produit fini en production à chaque itération (DEVOPS)

Maîtriser la complexité :

Découpage en cas d'utilisation

MISE EN ŒUVRE DE L'AGILE À RTE

- Différentes méthodes (frameworks) existent (ex : Scrum, Kanban, Lean, XP-Extreme)
- RTE a fait le choix d'adapter la méthode **Scrum** (mêlée en anglais), un framework favorisant l'amélioration continue et la livraison rapide de valeur ajoutée aux utilisateurs, qui se base sur 3 piliers :

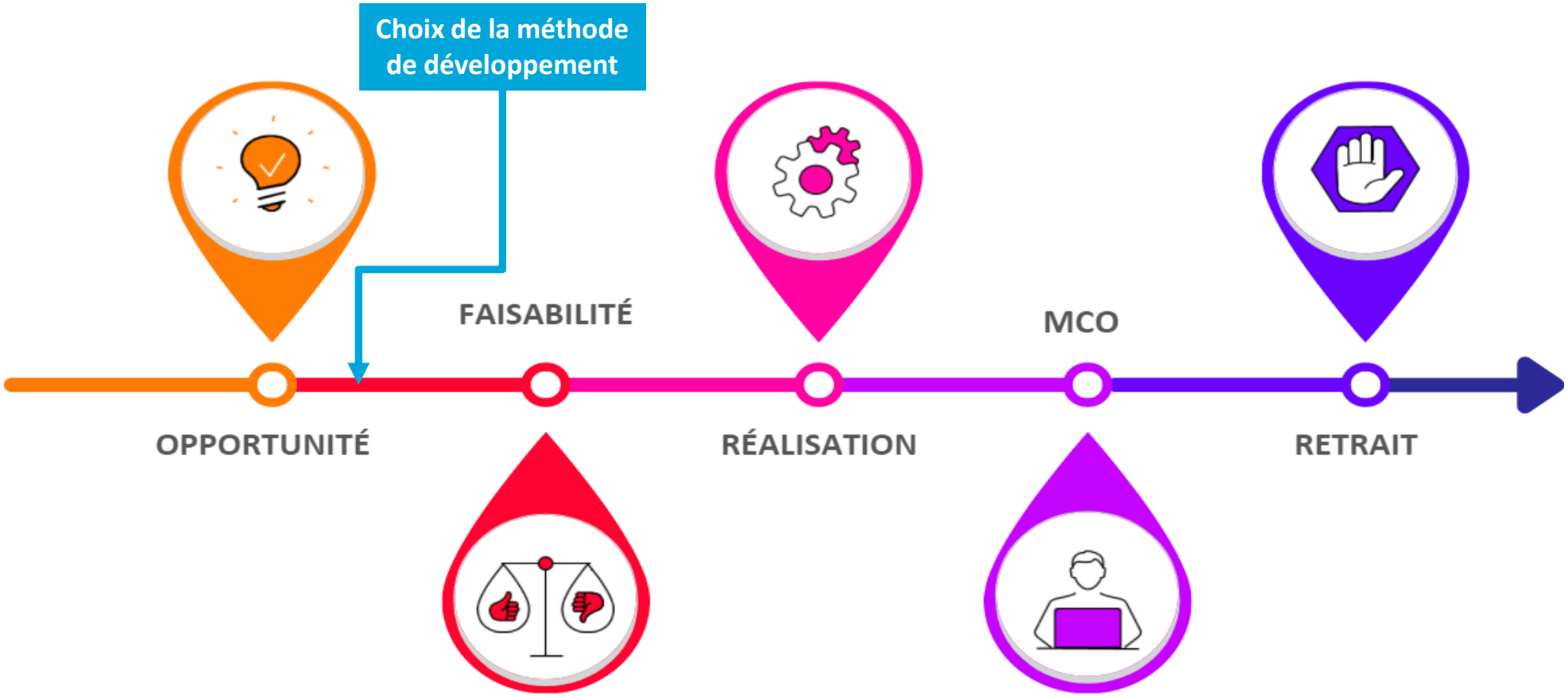


- La **transparence**:
  - Le processus, les standards et le travail doivent être visibles par les acteurs du projet (ceux qui exécutent le travail ou le reçoivent)
  - La transparence permet le partage de l'information et permet de s'accorder sur les standards, tel que la notion de « terminé »
- L'**inspection**
  - Le fonctionnement du projet et les progrès vers l'atteinte des objectifs doivent être inspectés fréquemment pour détecter les écarts ou les problèmes au plus tôt
  - Les évènements et rituels scrum permettent de provoquer le changement
- L'**adaptation**
  - Si certains aspects du fonctionnement du projet s'écartent des limites acceptables , ils doivent être adaptés, de même pour le produit
  - L'adaptation doit être la plus rapide possible pour minimiser les écarts supplémentaires

2 QUAND ET POURQUOI CHOISIR DE DÉVELOPPER EN AGILE?

A QUEL MOMENT S'INTÉRESSER AU CHOIX DE LA MÉTHODE DE DEVELOPPEMENT

Le choix de la méthode de développement doit s'effectuer au plus tôt, et doit être effectué de manière définitive **au cours de phase de faisabilité,** après éventuelle étude des différents scénarios possibles. Ce choix conditionnera le mode de réalisation du projet et donc planning et ressources.



JE CHOISIS LA MÉTHODE AGILE SI…

- Je suis en mesure de mobiliser une **équipe** (**métier**, **compétences techniques** ...) qui pourra participer de **manière active et continue** et **répondre rapidement** aux questions tout au long du projet
- J'ai une **vision claire** du **périmètre fonctionnel** attendu et de la cartographie des grandes fonctions sans connaître précisément toutes les règles de gestion en détail
- Le périmètre fonctionnel du projet est en partie **négociable**
- J'ai besoin de **montrer** et **faire valider** régulièrement au métier l'avancée des développements et **mettre à disposition** une application avec les **fonctionnalités clefs rapidement**
- Je suis **formé à l'agile** et mon équipe aussi
- Les équipes sont localisées sur place
- Je n'ai pas à faire (ou peu) de conduite du changement en préalable à chaque mise en production
- Mon projet comporte des risques liés à des incertitudes réglementaires et **demande de l'adaptabilité**

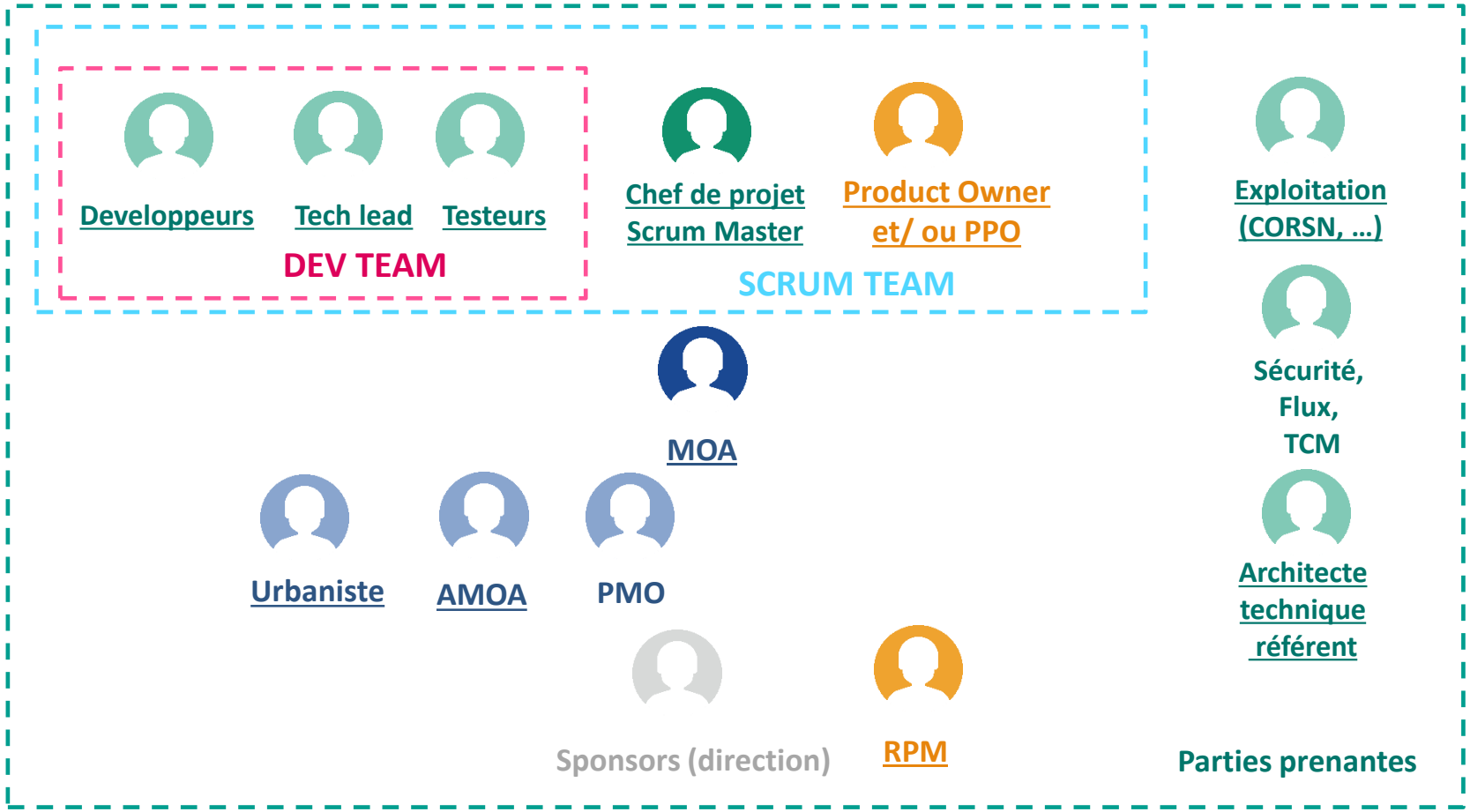
LES AVANTAGES ET INCONVÉNIENTS DE L'AGILE

	AGILE
AVANTAGES	• Développement commençant par les fonctions à **plus forte plus-value** • **Meilleur ROI** de la solution livrée car le besoin est régulièrement repriorisé • Plus **forte acceptation** de la solution par le métier, qui participe à sa construction tout au long du projet • Conduite du changement facilitée • Une **communication plus fluide** et moins de risques d'incompréhension au vu de la proximité des acteurs quand le métier joue le rôle de PO • **Pas d'effet tunnel**
INCONVÉNIENTS	• **Nécessite une forte implication du métier** tout au long du projet (Faisabilité et Réalisation) pour des prises de décision rapides • **Rythme soutenu** et régulier de l'équipe projet (spécifications, développements, tests durant chaque Sprint) • Une équipe projet qui doit être **organisée et dimensionnée pour être résiliente au rythme** et aux rituels de l'agile • Une **documentation qui demande plus d'efforts** pour être tenue à jour • Plus difficile à mettre œuvre si les acteurs de l'équipe sont sur des **sites géographiques différents**

B LE FONCTIONNEMENT DE L'ÉQUIPE AGILE CHEZ RTE

1 LES RÔLES SPÉCIFIQUES À L'AGILE À RTE

L'ÉQUIPE AGILE CHEZ RTE



Activités DSIT

Activités métier

Activités PiSSIN/MOA

LES RÔLES SPÉCIFIQUES À L'AGILE À RTE 1/3

Product Owner

- Le **Product Owner** ou **PO** (Propriétaire du produit) porte **la vision métier** du produit.
- Il est **le représentant des utilisateurs** et sa responsabilité est de **définir un produit** qui apporte le **maximum de valeur métier** aux utilisateurs dans **le temps** et **le budget** imparti au produit. Il est donc propriétaire du **Backlog du produit** (Product Backlog) qui est une liste priorisée de toutes les fonctionnalités attendues.
- Il porte la **responsabilité** de la **rédaction/validation des User Stories**
- Le périmètre du PO est le produit c'est à dire le développement qui va permettre de réaliser une application.
  - Si le projet nécessite la réalisation ou la modification de plusieurs produits/applications, la vision métier globale est portée par le **Product Manager.**
- Dans certains cas, quand le métier n'a ni la compétence ni le temps pour réaliser le rôle de product owner, le projet peut faire appel à des **Proxy-Product Owner (PPO)** auxquels le Product Owner délègue la tâche de rédaction des user stories et une partie des tests.
  - Ces PPO, pilotés par la DSIT, sont systématiquement rattachés à la Scrum Team, composée de la Dev team et du chef de projet DSIT.
  - Dans tous les cas, le **PO reste valideur des user stories** et **des tests utilisateurs** permettant de **valider la mise en production** de l'application.

LES RÔLES SPÉCIFIQUES À L'AGILE À RTE 2/3

Scrum Master

- Le **Scrum Master** appartient à l'équipe AGILE (**Scrum Team**).
  Celle-ci est constituée des Développeurs, des Testeurs, et du Tech Lead (**DEV Team**), du Scrum Master et du Product Owner (PO).

- Le Scrum Master sert d'interface entre le Product Owner (PO) et la Dev Team, protégeant ces derniers de tout élément susceptible de perturber la production. Il aide l'équipe à avancer de manière autonome et en cherchant en permanence à améliorer le fonctionnement de l'équipe.

- Au sein de l'équipe, il a notamment pour mission de **faire appliquer la méthode Agile**, d'animer les différents « rituels » de Scrum : daily meeting, poker planning, sprint planning, sprint rétrospective... Il assure le suivi de la réalisation et contribue à améliorer la vélocité de l'équipe.

- Il contribue :
  - à l'amélioration de la vélocité des Développeurs,
  - à la construction des Sprints Backlog, permettant de chiffrer et planifier ce qui sera développé aux prochains sprints.

- **A RTE il est préconisé que le Scrum Master soit le chef de projet SI si la taille de l'équipe le permet.**

LES RÔLES SPÉCIFIQUES À L'AGILE À RTE 3/3

Tech lead

- Le **Tech Lead** appartient comme les **Développeurs** et les **Testeurs**, à l'équipe de développement AGILE (**DEV Team**).
- Celle-ci est complétée du **Scrum Master** (SM) et du **Product Owner** pour constituer la **Scrum Team**.
- Le rôle de **Tech Lead** est de porter la vision technique dans les développements, de coordonner les champs techniques au sein de l'équipe de développement.

2 LE DÉROULÉ D'UN PROJET AGILE ET LES INSTANCES OPÉRATIONNELLES SPÉCIFIQUES À L'AGILE

LE SPRINT EN AGILE

- Le **sprint** d'un projet agile est **une période de travail** fixe, durant laquelle **l'équipe de développement** (Dev team) se **concentre** sur la **réalisation d'un ensemble spécifique d'objectifs** définis dans le Backlog (fonctionnalités projet prêtes à être développées) par le Product Owner

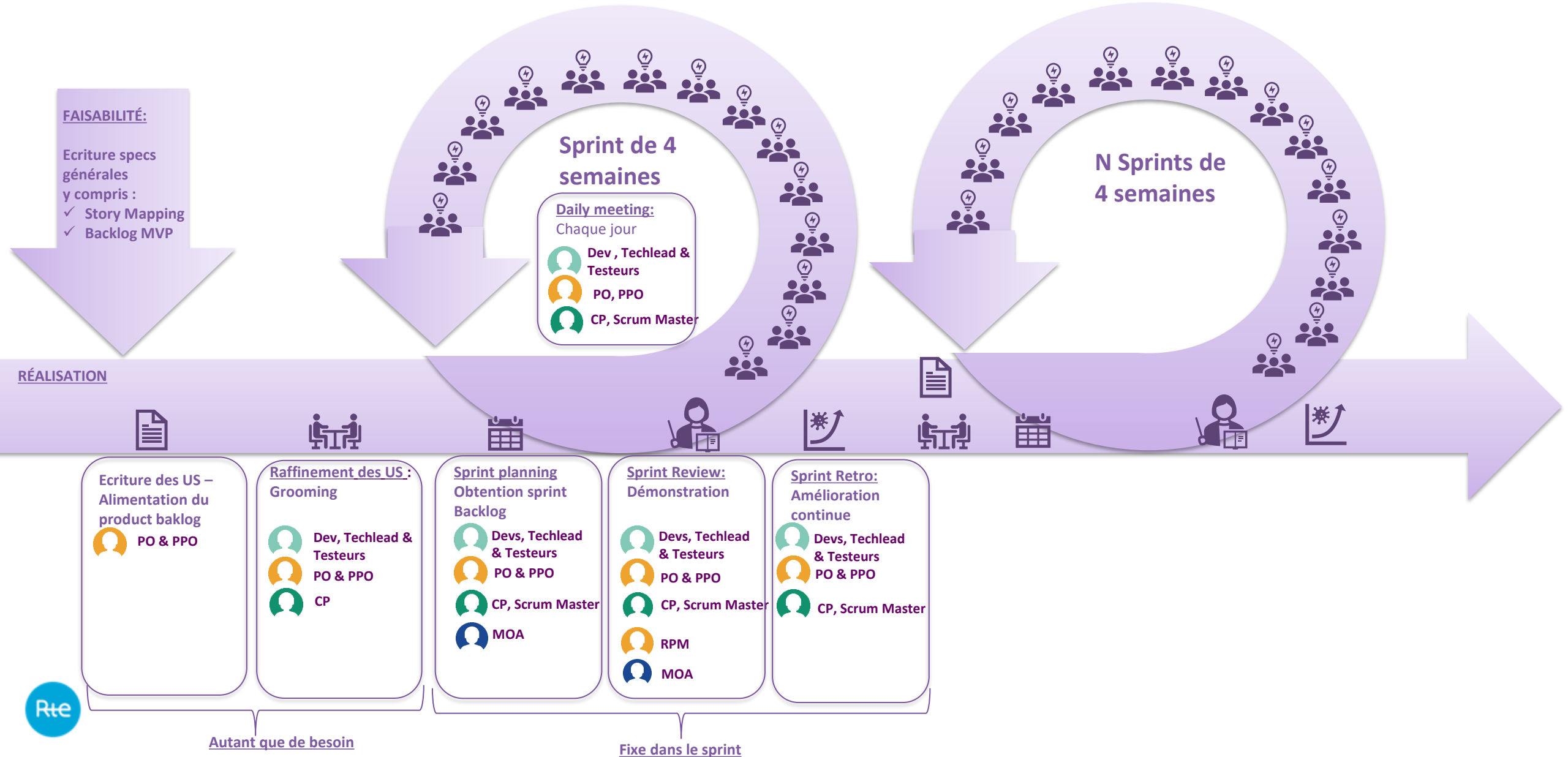
- L'**objectif** du sprint est de **livrer un produit** ou des fonctionnalités à la fin de la période.

- Le sprint, tout au long du projet, est d'une **durée fixe** (entre 2 et 4 semaines généralement) et courte pour limiter complexité et risques sur le Sprint.

- **Pendant le Sprint :**
    - **l'objectif à atteindre, défini au démarrage, ne peut être modifié** ;
    - la composition de l'équipe doit rester constante ;
    - **la qualité n'est pas négociable**

- Les principales activités incluent:
    - La planification du sprint : Sprint Planning,
    - Le développement,
    - Les réunions quotidiennes de suivi : Daily meetings
    - La revue du sprint pour démontrer les réalisations aux parties prenantes : sprint review ou sprint demo
    - L'amélioration des process futurs lors de la rétrospective

LE DÉROULEMENT DU PROJET EN AGILE : ZOOM SUR LE SPRINT



LES RITUELS DU SPRINT: LE DAILY MEETING

QUI

- Développeurs et Techlead
- Proxy Product Owner (PPO) ou PO si pas de PPO
- Scrum Master (en début de projet)

QUAND

Tous les jours

COMBIEN DE TEMPS

- Le **daily meeting** (ou stand-up) agile est une **réunion quotidienne de 15 minutes** environ de l'ensemble des **membres de l'équipe de développement.**
- Elle permet que chaque membre
  - Se synchronise et se **coordonne avec l'ensemble de l'équipe**,
  - Fasse part des avancées et des **difficultés rencontrées** en identifiant les points de blocages
  - **Ajuste** avec le reste de l'équipe **les plans** pour assurer l'atteinte en fin de sprint des objectifs
- Chaque membre de l'équipe répond à trois questions :
  1. Qu'ai-je accompli hier ?
  2. Que vais-je faire aujourd'hui pour permettre d'atteindre l'objectif du sprint ?
  3. Quels sont les obstacles rencontrés qui empêchent l'équipe d'atteindre l'objectif du sprint ?
- Cette réunion, auto-organisée, se déroule généralement debout, devant le sprint board, où sont affichés les objectifs et la progression du sprint.
- Si nécessaire, le scrum master peut intervenir en tant que facilitateur.

LES RITUELS DU SPRINT : LE SPRINT PLANNING

QUI

- Développeurs et Techlead
- Product Owner et PPO
- Chef de projet (CP)
- Scrum Master (en début de projet)
- MOA

QUAND

En fin de sprint

COMBIEN DE TEMPS

à

- Le **sprint planning** permet:
  - De définir le **but du prochain sprint** et d'identifier les éléments du **Product Backlog** qui y seront traités.
  - D'établir **le plan de travail** pour atteindre l'objectif

- Cette réunion se déroule en 2 temps:
  1. L'équipe de développement **prévoit** ce qui sera développé dans le prochain sprint :
     - En sélectionnant les **éléments du Product Backlog** , clarifiés et **priorisés par le Product Owner**;
     - En tenant compte de la **vélocité** ou **capacité de production** prévue pour ce sprint

     Celle-ci est basée pour le premier sprint projet sur une estimation théorique, puis sur la productivité réelle passée pour les suivants

  2. L'équipe de développement se focalise sur la manière dont ses membres atteindront le but du sprint, en
     - **Découpant le travail à effectuer** pendant le sprint en **activités** d'une durée limitée d'une journée au plus
     - **Planifiant** le travail

- La réunion permet d'obtenir le **Sprint Backlog** : l'ensemble des éléments du Product Backlog que l'équipe de développement s'engage à développer pendant ce sprint

LES RITUELS DU SPRINT: LA SPRINT REVIEW

QUI

- Développeurs et Techlead
- Product Owner et PPO
- Chef de projet (CP)
- Scrum Master (en début de projet)
- Responsable projet Métier (RPM) si différent du PO
- MOA

QUAND

Le dernier jour du sprint

COMBIEN DE TEMPS

à

- La **sprint review** est une réunion qui permet à la fin du sprint :
  - De **présenter au commanditaire** et aux parties prenantes **ce qui a été réalisé**
  - De **valider** ce qui a été réalisé
- Il se déroule donc en 4 temps:
  1. Présentation par l'équipe de développement de la liste des fonctionnalités terminées du sprint backlog
  2. Démonstration par l'équipe de développement des fonctionnalités terminées du produit
  3. Recueil du feedback des parties prenantes
  4. Echange sur les modifications possibles et ajustement si nécessaire du product backlog

LES RITUELS DU SPRINT : LA SPRINT RETROSPECTIVE

QUI

- Développeurs et Techlead
- Product Owner et PPO
- Chef de projet (CP)
- Scrum Master (si distinct du CP)

QUAND

Le dernier jour du sprint

COMBIEN DE TEMPS

à

- La **sprint retrospective** permet à l'équipe de développement de réfléchir sur le sprint écoulé, d'identifier ce qui a bien fonctionné, ce qui n'a pas bien fonctionné, et de **décider des actions à entreprendre pour s'améliorer lors des prochains sprints.**

- Elle se déroule en plusieurs temps:
  - Le **Scrum Master** accueille l'équipe et explique **l'objectif de la rétrospective** et la **technique d'animation** qui va être utilisée (ex « Start, Stop, Continue », «4 L - Liked, Learned, Lacked, Longed for»)
  - L'équipe discute des événements du sprint, en se concentrant sur les **points forts** et **les points d'amélioration**.
  - L'équipe **analyse les données collectées** pour identifier les tendances, les obstacles récurrents, et les succès.
  - L'équipe propose et discute des **actions concrètes à mettre en place** pour améliorer les processus et la collaboration.
  - Les actions sont **priorisées et assignées** à des membres de l'équipe si nécessaire.

- Cette réunion favorise une amélioration continue et aide l'équipe à s'adapter et à devenir plus efficace sprint après sprint.

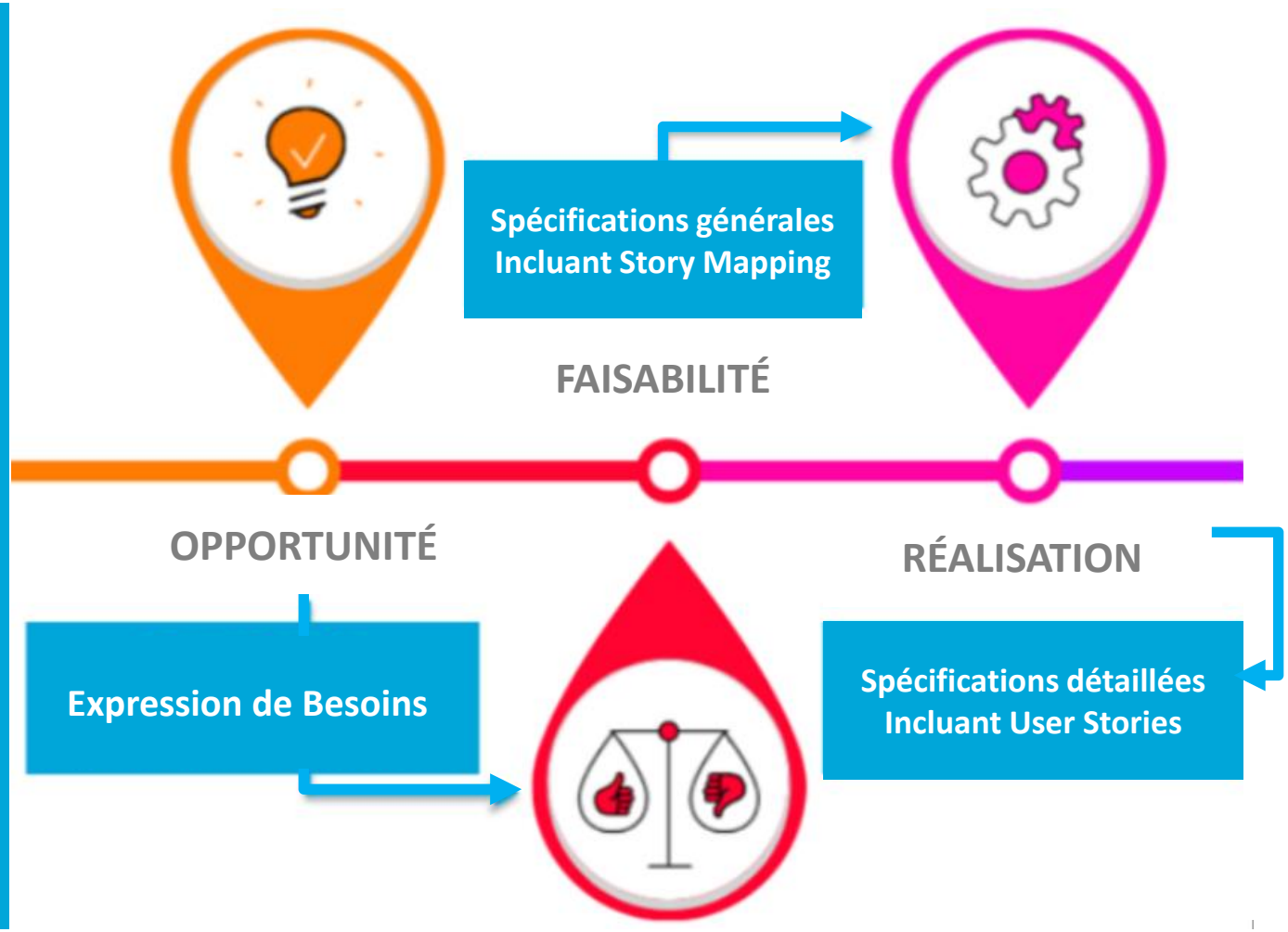
C COMMENT CADRER ET SPÉCIFIER MON PROJET EN AGILE?

1 LES « SPÉCIFICATIONS » EN AGILE

L'ÉTABLISSEMENT DE SPÉCIFICATIONS, UNE DÉMARCHE GLOBALE QUI CONCERNE TOUS LES ACTEURS DU PROJET

3

Phases
Acteurs
Livrables
Trames



RAPPEL

Les Spécifications Générales (SG) permettent d'affiner et analyser les besoins et attentes métiers exprimés dans **l'Expression des besoins** afin de les traduire en exigences qui fourniront le détail et la précision nécessaires pour faire une évaluation technico-économique et réaliser les développements.

- *Objectif : fournir un document unique au chef de projet pour analyse et chiffrage et lancement d'appel d'offres*

La rédaction des SG est obligatoire quelle que soit la méthode de réalisation choisie (**Cycle en V ou Agile**)

- **Une V0 des SG** doit être rédigée **en entrée de faisabilité** afin d'orienter les analyses.
- **La version finale**, livrable de la phase de faisabilité et indispensable pour la DI, sera remise aux CP afin qu'ils puissent démarrer leur travail.
- **En mode agile,** il s'agit en complément d'identifier l'ensemble des fonctions et exigences constituant le **story map** et le **backlog de chaque produit** et d'affecter les fonctions identifiées pour la solution à chacun des produits **pour la première version qui apporte de la valeur (MVP*).**

Dans le contexte d'un **projet d'évolution d'un système existant, qui repose sur un plateau agile déjà en place**, il est admis de remplacer les paragraphes des spécifications générales :

- 4.1 L'arborescence des fonctions
- 4.2 Nom de la macro fonction
- 4.2.1 Nom de la fonction
- 4.1.1.2 Exigences

→ Par un export de l'outil de spécifications (Confluence du projet, JIRA du projet) contenant le story map et les EPIC/Features qui seront couvertes par la future Décision d'investissement.

Plus les SG sont précises, meilleurs et plus rapides seront **le chiffrage** et **la réalisation**.

* Le MVP (Minimum Viable Product / Produit Minimum Viable) est une version fonctionnelle et élémentaire du produit.

2 COMMENT RÉALISER LE STORY MAPPING POUR OBTENIR LA STORY MAP?

QU'EST-CE QU'UNE STORY MAP?

- La **Story Map** est une représentation visuelle des grandes fonctionnalités métiers du produit attendu, déclinées en scénarios utilisateurs, qui sont priorisées et hiérarchisées.
- Elle est établie lors d'ateliers de **Story mapping** entre la MOA, le RPM et le chef de projet SI ou les membres de la scrum team et l'UX s'ils sont nommés.

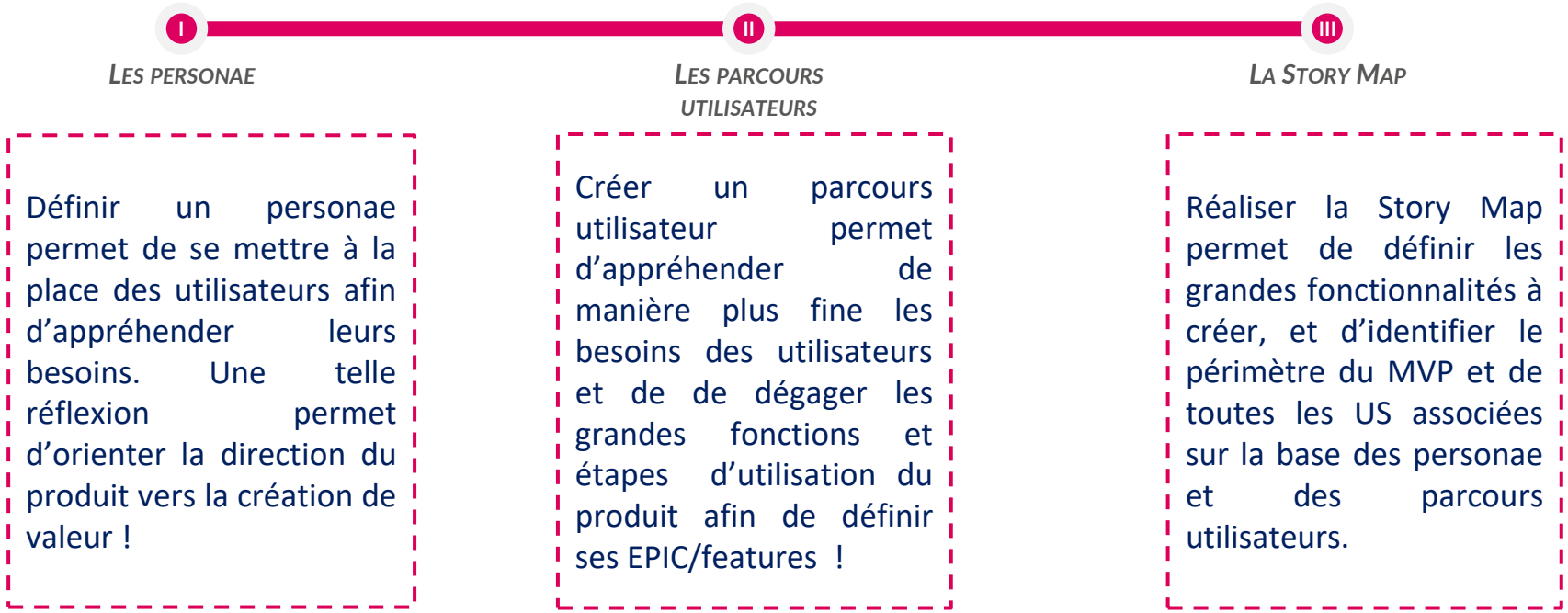
- La Story Map permet de
  - Visualiser **le parcours utilisateur** et les **fonctionnalités essentielles**
  - Définir le **Produit Minimum Viable ou MVP** via la priorisation et la hiérarchisation des fonctionnalités, le MVP étant le périmètre fonctionnel minimum du produit proposé aux utilisateurs pour répondre au besoin métier
  - Initier une **première version du backlog produit**, en décrivant les grandes fonctionnalités de l'application

- La Story Map facilite ainsi la **planification** et les **livraisons incrémentales** du projet en aidant à organiser les tâches, les fonctionnalités ou les étapes.

- **Lean Canvas** et **Impact Mapping** (cf annexe) peuvent aider à établir la story map

COMMENT ÉTABLIR LE STORY MAPPING?

- La démarche de Story Mapping vue dans son ensemble peut être découpée en 3 grandes phases :
  - La **création des personae** qui permet d'identifier qui sont les utilisateurs du produits
  - La **définition des parcours utilisateurs** de chacun des utilisateurs afin d'identifier les besoins
  - La **réalisation de la Story Map** qui va définir les fonctionnalités à implémenter dans les mois à venir



LA CRÉATION DES PERSONNAE

- Le **Personae** un utilisateur-type, utilisé pour décrire les différents scénarios d'utilisations
- Pour le construire, il faut, au travers d'ateliers ou d'interview d'utilisateurs ou futurs utilisateurs:
  - **Identifier les profils représentatifs** des utilisateurs (y compris l'administrateur) et leur donner à chacun un nom
  - Décrire le profil, la **fonction, le rôle, le contexte** de travail du personae
  - Décrire ses **problèmes**, ses **attentes**
  - Indiquer le **bénéfice** qu'aura le personae si on répond à ses attentes
  - Indiquer les **craintes** du personae

- Exemple : Le produit attendu est la mise en place d'une cuisine. Les interviews ont permis de définir 2 personae

Persona 1 : Nathalie, étudiante, 19 ans

Motivations & frustrations

- Faire des soirées
- Réussir ses études

- Pas le temps de cuisiner
- Petit budget
- Ne sait pas cuisiner

Besoins

- Économiser du temps dans la cuisine
- Avoir de quoi cuisiner rapidement notamment un micro-ondes
- Pouvoir manger debout dans la cuisine

Persona 2: Bruno, père de famille, 40 ans

Motivations & frustrations

- Faire de la cuisine savoureuse
- Faire à manger pour toute la famille

- Aimerait passer moins de temps à cuisiner avec des meilleurs équipements

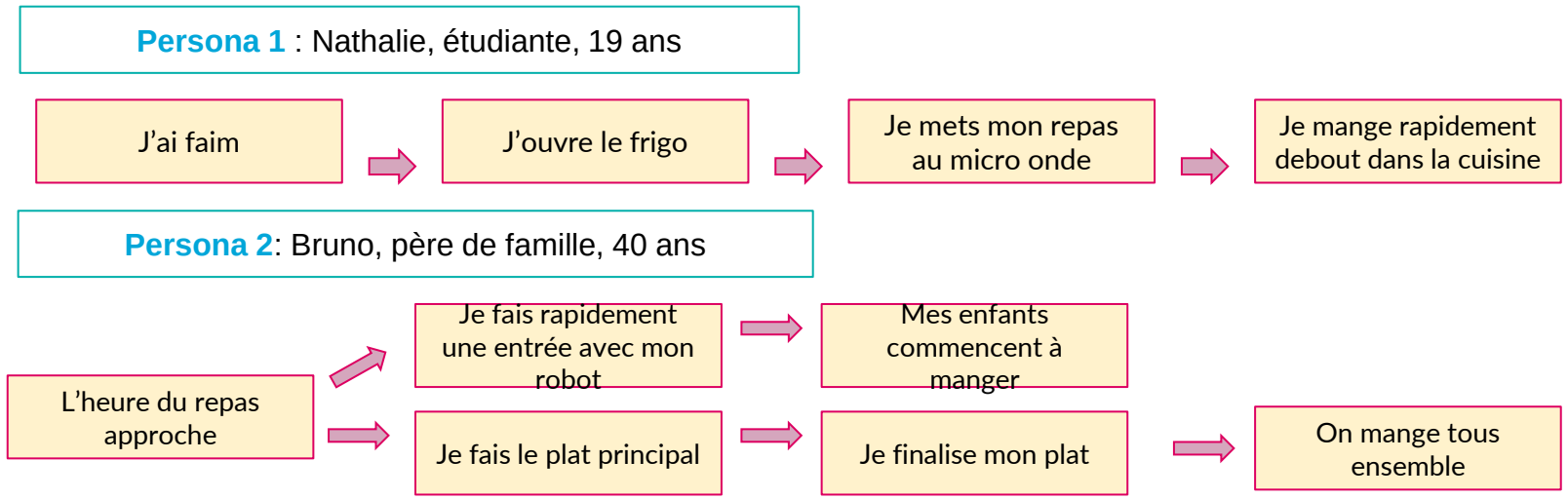
Besoins

- Avoir un grand plan de travail
- Gagner du temps même sur les recettes longues
- Avoir une cuisine pratique

LE PARCOURS UTILISATEUR

- Le **parcours utilisateur** est un scénario qui se compose d'une succession d'étapes correspondant à chaque **interaction entre l'utilisateur et le produit**.
  - Pour chaque Personae, en fonction des objectifs du produit ou de la fonctionnalité à développer, on va identifier les **activités clés** pour définir le parcours type d'utilisation, pour mieux comprendre l'utilisateur et l'impact des actions qu'il réalise.
  - Ces activités clés sont séquencées par ordre **logique ou chronologique et décomposées en tâches** avec les étapes spécifiques ou tâches que l'utilisateur doit accomplir, en détaillant les conditions, les points de décision des utilisateurs à chaque étape, ainsi que les obstacles qui peuvent être rencontrés.

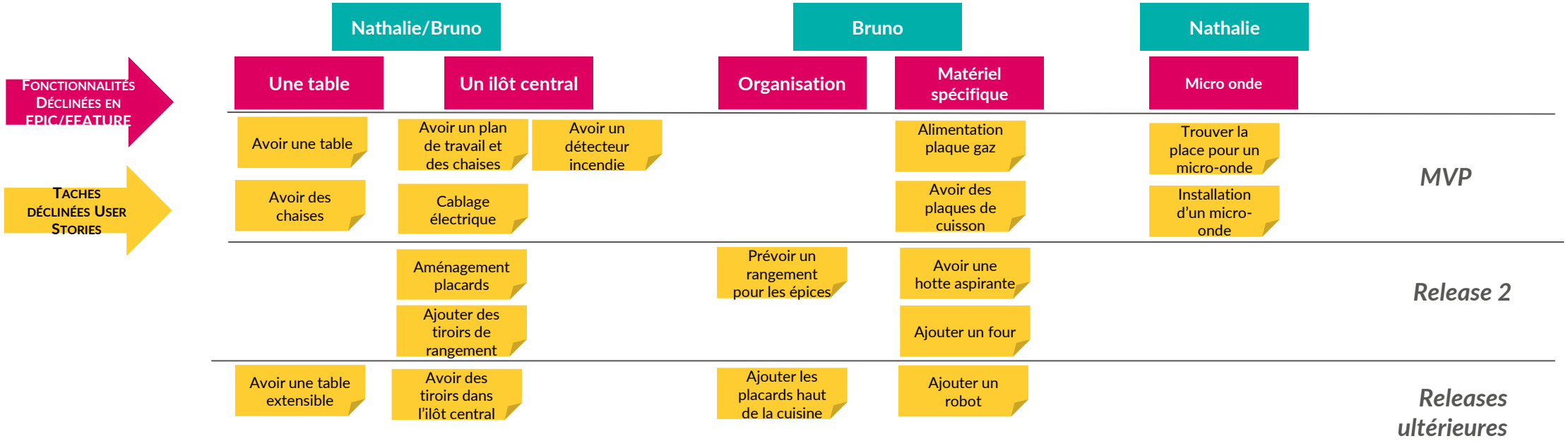
- Exemple : Le produit attendu est la mise en place d'une cuisine.



LA STORY MAP

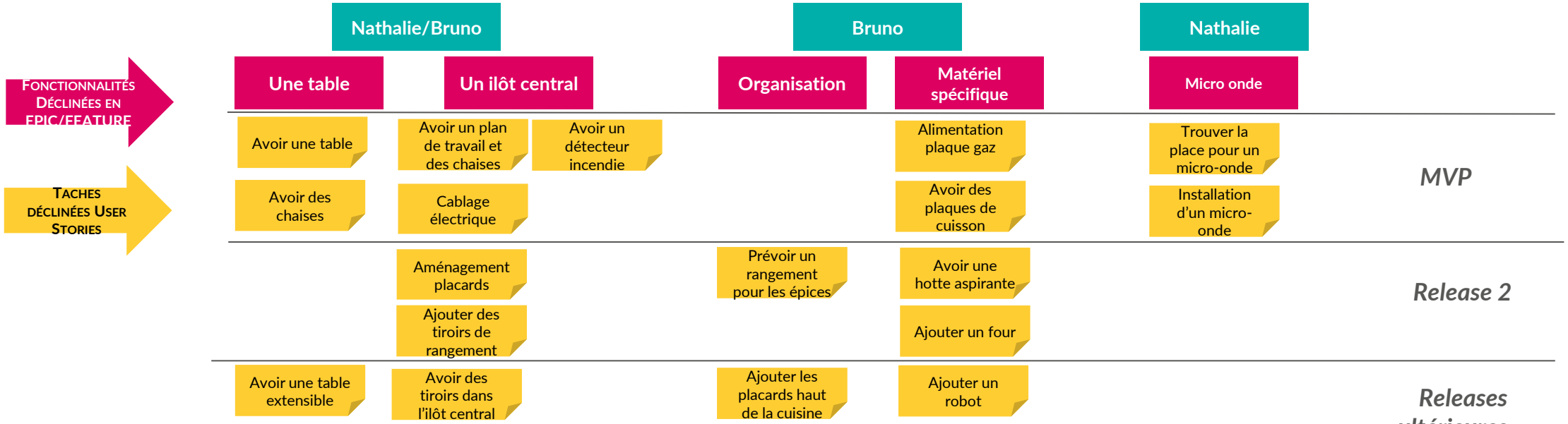
- La **Story Map** est une représentation de **l'ensemble des fonctionnalités ou EPIC/Features** qu'on souhaite développer à un instant t**, priorisées par la valeur et répartie par thématique**.
  - La création d'une Story Map a lieu en faisabilité au moment où on se lance dans la création du produit pour mettre à plat la vision du produit

- Elle permet de décomplexifier le besoin en:
  - Etablissant un premier découpage en tâches et en les répartissant dans 3 niveaux : le **MVP - Produit Minimum Viable, la deuxième version de l'application ou release 2, et la release ou les versions ultérieures**
  - Créant une première version du **backlog**.



L'EPIC/FEATURE

- Une **Epic/Feature** (Épopée en français) décrit une **macro fonction** qui englobe plusieurs fonctionnalités décrites dans les User Stories
- Les **Epics/Features** sont un moyen utile d'organiser le besoin sous une forme arborescente.
- L'Epic/Feature est rédigée par:
  - le Product Owner (PO) pour les macro-fonctions liées à un besoin métier,
  - le Tech Lead pour les macro-fonctions liées à un besoin technique.



3 LES DIFFÉRENTS TYPES DE STORYS, AU CŒUR DE LA RÉUSSITE DU PROJET

LA USER STORY ET SES CARACTÉRISTIQUES

- L'équipe agile va développer au sein de chaque Sprint les **User Stories** qui ont été choisies lors du Sprint planning

- Une User Story est une description simple et concise d'une fonctionnalité du point de vue de l'utilisateur final, formulée pour capturer ses besoins et attentes spécifiques.

- A chaque user story doivent être associés **Business Value** et **Story Point**.

- La **Business Value** :
  - Est une **mesure de l'importance** ou du bénéfice qu'une **fonctionnalité**, un produit ou un projet apporte à une entreprise, souvent en termes de revenus, d'efficacité, de satisfaction client ou d'avantage compétitif.
  - Elle **aide à prioriser les efforts** en fonction de leur impact sur les objectifs stratégiques de l'entreprise.
  - Elle est **définie par le product owner**

- Le **Story Point**:
  - Est une unité de mesure utilisée dans les méthodes agiles pour **estimer l'effort nécessaire** à la réalisation d'une user story, en tenant compte de sa complexité, de sa taille et des risques associés.
  - Cette estimation permet aux équipes de **planifier et de prioriser** leur travail de manière plus efficace.
  - Il est **défini par l'ensemble de l'équipe au cours du sprint planning**

QU'EST-CE QU'UNE USER STORY ET COMMENT L'ÉCRIRE ?

- Une **user story** ou récit utilisateur en français est une phrase simple permettant de définir avec suffisamment de précision le contenu d'une fonctionnalité à développer.
- Elle est relue et challengée par la Scrum team lors de sessions de **raffinement** du backlog (ou **Grooming**), qui ont lieu tout au long de la réalisation
- Elle contient trois éléments descriptifs de la fonctionnalité, sous la forme :

  En tant que <qui>, je veux <quoi> afin de <pourquoi>

  - Le **qui** indique l'utilisateur du point de vue duquel on se place.

    Il s'agit souvent d'un rôle dans l'usage du produit, basé sur un personna auquel on peut s'identifier pour mieux comprendre ses attentes.
  - Le **quoi** décrit la fonctionnalité ou le comportement attendu.

    Le détail exhaustif est rédigé par le PO en itérant avec le Responsable Projet Métier (RPM).

    Sur certains projets, le détail peut être rédigé par un PPO et doit être validé par le PO.
  - Le **pourquoi** permet d'identifier l'intérêt de la fonctionnalité et d'en justifier le développement. Il permet également de mieux évaluer la priorité du récit.

***Exemple** : User story "accéder pour un administrateur à l'application "*

- ***<En tant que>** administrateur de l'application*
- ***<Je Veux>** accéder à l'application. Pour cela je me connecte à l'URL de l'application et après avoir saisi mon NNI et mot de passe Windows.*
- ***<Afin de>** d'accéder à l'écran d'accueil de l'application, qui doit avoir l'aspect suivant (joindre la maquette de l'ergonome de la page d'accueil pour un administrateur.*

COMMENT ÉCRIRE MA USER STORY?

- Une **User Story** doit respecter les caractéristiques réunies sous le sigle **INVEST**.

- Elle est:
  - **Indépendante**, vis à vis des autres User Stories du Product Backlog.
    Cela permet d'éviter de devoir largement reprendre la User Story lors du traitement d'éventuelles User Stories liées, et diminue ainsi la dette technique;
  - **Négociable,** c'est à dire permettre un support de discussion en vue d'une amélioration du besoin initial;
  - **Valorisable**, c'est à dire apporter une valeur métier;
  - **Estimable**, c'est à dire définie et rédigée de façon à être facilement mesurable en points d'efforts;
  - **Suffisamment petite**, de façon à être réalisable sur un sprint; Il ne faut pas hésiter à découper une User Story si besoin. La User story doit être vue comme un atome de spécification.
  - **Testable**, c'est à dire accompagnée de ses critères d'acceptance pour faciliter sa validation.

LES AUTRES TYPES DE STORY

- Il existe d'autres types de story, apparentées à des **user stories,** qui sont identifiées dans JIRA :
  - **Les technical stories :**
    - Ce sont des user stories consacrées à des besoins purement techniques (exemple : amélioration des temps de requêtage de la base de données, prérequis techniques pour mise en place d'échanges de flux entre applications, …).
    - N'obéissant pas à un formalisme spécifique pour son écriture, elles seront validées par le Techlead avant d'être intégrées au backlog.
  - **Les évolutions:**
    - Ce sont des modifications de besoins fonctionnels qui conduisent à des modifications des User Stories ou une nouvelle user story
  - **Les anomalies:**
    - Ce sont des corrections tracées dans les User Stories ou les Technical stories en fonction du type d'anomalie
  - **Les tâches:**
    - Lorsque les User Stories, Technical Stories ou Evolutions sont trop grandes, elles peuvent être découpées en tâches ou sous-tâches. Cela permet d'éviter l'effet tunnel, de produire de la valeur rapidement et de contribuer à une meilleure organisation du travail.

COMMENT MESURER LA QUALITÉ DE MA USER STORY – LES CRITÈRES D'ACCEPTANCE?

- Chaque User Story doit être complétée avec des **critères d'acceptance définis par le PO.**
  - Cette liste d'éléments permet à l'utilisateur de confirmer que la fonctionnalité livrée correspond aux attentes.
  - Définir ces critères à l'avance assure que la description de la fonctionnalité est suffisamment précise pour être développée.

- Les critères d'acceptance incluent des exigences fonctionnelles et non fonctionnelles précises comme :
  - Les cas d'utilisation spécifiques et les résultats attendus, les conditions d'interface utilisateur.
  - Les contraintes de performance spécifiques à la fonctionnalité.

- Les **critères d'acceptance** sont fondamentaux car ils permettent de garantir la conformité du produit livré.
  - On utilise le langage **Gherkhin** « **Given –When –Then** » pour définir avec précision les scénarios
    - **Given** liste les conditions initiales nécessaires au test (les données en entrée souvent)
    - **When** décrit les actions à effectuer (ce qui doit être testé)
    - **Then** décrit le résultat attendu en cas de bon fonctionnement du produit (les données que l'on veut récupérer en sortie souvent)
  - Une fois le scenario de comportement décrit au travers de la syntaxe Gherkhin, plusieurs outils (ex Cucumber) permettent d'automatiser les tests. Chaque étape d'un scénario Gherkhin est implémentée comme une méthode java (à développer), appelée step.

***Exemple** :*

- *<Etant donné> les utilisateurs suivants : Nom/Prénom, …ou une situation de départ donnée = UN CONTEXTE*
- *<Quand ou lorsque> j'ajoute/je supprime/je modifie cette donnée : l'utilisateur effectue une ou plusieurs ACTIONS*
- *<et que> ….<mais>*
- *<Alors> je dois obtenir ce résultat : on doit pouvoir constater telles conséquences ou situation d'arrivée*

COMMENT MESURER LA QUALITÉ DE MA USER STORY: DEFINITION OF READY (DOR) ?

- Au démarrage du projet, la **scrum team** définit un certain nombre de critères qui devront être respectés pour tous types de user storys du projet, avant qu'elles ne soient considérées comme complètes pour partir en développement.
- Cette liste de critère est la « **Definition Of Ready** » ou DOR

- Tout au long du projet, **une user story pourra être intégrée à un sprint** et partir en développement **uniquement si elle respecte cette liste** de critères.

- La DOR peut évoluer durant la vie du projet et être complétée.

- A RTE, il est demandé de rendre obligatoire la DOR avec a minima (cf annexe):
  - L'ensemble des critères correspondant aux **éléments méthodologiques**
  - L'ensemble des critères correspondant aux **éléments descriptifs**
  - L'ensemble des critères correspondant aux **critères d'acceptance**

Exemple *: Cf annexe*

- *Maquette de l'écran disponible,*
- *Dépendances avec d'autres user stories ou équipes identifiées,*
- *Règle de gestion accompagnée d'un exemple,*
- *Description des exigences non fonctionnelles comme performance et sécurité documentées.*

COMMENT MESURER LA QUALITÉ DE MA USER STORY: DEFINITION OF DONE (DOD)?

- A l'issue du développement, la user story doit respecter la « **Definition of Done** » ou **DOD** que la **scrum team** également définit en début de projet
  - Cette **DOD est identique pour toutes les user stories** et est propre à chaque équipe.
  - Elle peut **évoluer dans le temps** mais **ne doit pas être modifiée** pendant un sprint.
  - Elle est en général challengée en fin de sprint lors de la rétrospective et peut se voir ajouter des critères en cours de projet.

- Les critères de cette DOD sont choisis parmi des critères :
  - Relatifs au code, aux tests, à la documentation, à la validation des user stories, à la qualité, au respect des normes UX/UI (Design System), à l'ergonomie, à l'accessibilité numérique, à la conformité RGPD, au déploiement et à la communication.
  - Il est d'usage de choisir entre 5 à 10 critères de DOD, pour que celle-ci puisse être exploitable

Exemple *:*

- *Toutes les tâches techniques sont terminées*
- *Le code est commenté, vérifié (par un autre développeur) et compilé sans erreur // la documentation projet est à jour*
- *Les tests unitaires (Développeurs) ont été réalisés*
- *Les tests fonctionnels (PO) ont été réalisés*
- *Les critères d'acceptation sont validés*
- *La procédure de déploiement a été mise à jour et elle a été validée*

COMMENT MESURER LA QUALITÉ DE MA USER STORY: DEFINITION OF DONE (DOD)?

- A RTE, il est demandé de rendre **obligatoire la DOD**.

- Elle doit comporter :
  - Les **items suivants** qui correspondent à la **qualité du code**:
    - Les **tests unitaires sont écrits** et couvrent **au moins 60% du code nouvellement ajouté** ou modifié
    - Le code ajouté ou modifié doit obtenir **une note A** sur les critères suivants : **maintenabilité, fiabilité, sécurité**
    - Le **code en doublon** doit être **inférieur à 5% du code total**

  - A minima un item relatif à **la documentation**, commentaire du code etc ;

  - A minima un item relatif à la **validation des critères d'acceptation de l'US**

  - Et **les autres critères** jugés pertinents par l'équipe

- Des exemples de critères sont disponibles en annexe.

D COMMENT PILOTER AU QUOTIDIEN MON PROJET AGILE?

1 BUSINESS VALUE ET STORY POINTS

LA BUSINESS VALUE ET SA MESURE POUR LA USER STORY

- La « **business value** » ou « valeur métier » en agilité est un concept clé qui guide les équipes dans la priorisation et la réalisation des tâches et des projets.
  - Elle représente l'**importance** ou l'impact économique qu'un produit, une fonctionnalité ou une tâche apporte à une entreprise et mesure de la valeur obtenue en réalisant un projet ou une tâche spécifique.
  - Sa mesure est propre à chaque projet, le product owner définissant en début de projet une unité de mesure en « points valeur métier »

- La « business value » peut être mesurée de différentes manières:
  - **Valeur financière** (mesure de l'impact financier direct, comme l'augmentation des revenus), **Valeur utilisateur** (Acquisition de nouveaux clients,…), **Valeur stratégique/juridique** (alignement avec les objectifs stratégiques l'entreprise ou les contraintes juridiques)

- La business value sert à :
  - **Évaluer et prioriser** les éléments du backlog, le product owner et la scrum team se concentrant sur ceux qui apportent le plus de valeur à l'entreprise.
  - **Planifier les sprints** en aidant lors de la planification des sprints à décider quelles fonctionnalités seront développées en priorité.
  - **Mesurer la performance** : En analysant les résultats par rapport aux attentes en termes de valeur apportée.
  - **Communiquer avec les parties prenantes** : La business value fournit un langage commun pour discuter des priorités et des résultats avec les parties prenantes, facilitant ainsi la prise de décisions et l'alignement des objectifs.

LES STORY POINTS ET LE POKER PLANNING

- Le **Story Point** est une unité de mesure en "**points de complexité**" qui permet de faciliter l'estimation de charge pour la réalisation des User Story.
- Cette unité de mesure, définie par l'équipe de développement, est **propre à chaque équipe**.
- Cette unité de mesure se base fréquemment sur la suite de Fibonacci et est un **poids relatif**.
    - Les valeurs pouvant être attribuées sont:
        - 1 : US extrêmement simple (ex: correction d'un libellé par exemple),
        - 2, 3, 5: US un peu plus complexe (ex: création d'un formulaire simple),
        - 8, 13, 21, 34, 55, 89, 144 : tâches plus complexes & longues

- L'estimation de chaque user story du Product Backlog est faite de façon empirique par toute l'équipe souvent en utilisant la technique du **Poker planning** :
    - Chaque développeur estime l'US en tenant compte de la difficulté de développement, des interfaces, du risque, … ce qui permet de lever les éventuelles incompréhensions ou points non explicités encore dans la user story
    - Les développeurs dévoilent ensemble les cartes avec leurs estimations
    - Tour à tour, chacun explique son estimation,
    - La dev team doit tomber d'accord sur l'estimation de la user story

- L'équipe de développement définit ensuite le nombre de Story Point qu'elle estime pouvoir réaliser dans le temps imparti du sprint.
    - Ce nombre est initié lors du sprint 0 sur la base de l'expérience de l'équipe de développement , il est revu à chaque rétrospective de sprint. Ce nombre de points est **la vélocité**.

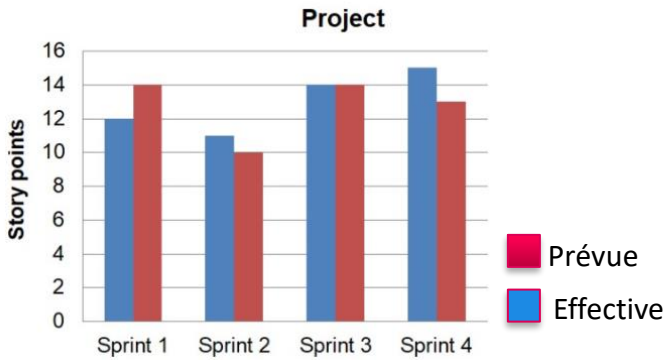
2 LES OUTILS DE PILOTAGE DE LA PERFORMANCE

SUIVI DE LA VÉLOCITÉ

- **La vélocité** agile représente **la capacité de production** de l'équipe.
  - Elle mesure l'**effort** qu'est capable de fournir une équipe de développement pour la réalisation des tâches programmées dans un sprint.
  - Elle est représentée par **la somme des story points** des user stories terminées du sprint, répondant aux critères de DOD du projet.
  - Elle est estimée au sprint 0 et réajustée à l'issue de chaque sprint.

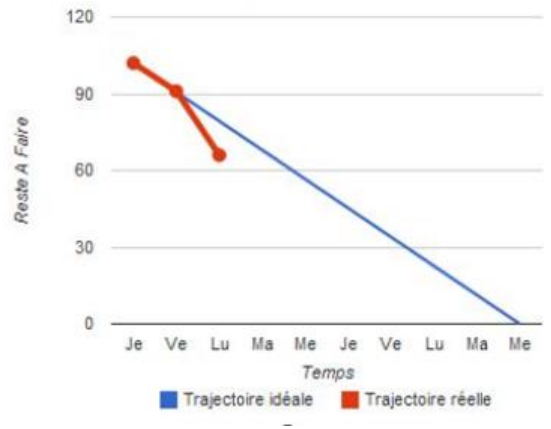
- Au démarrage du projet, il faut entre trois et cinq sprints pour que la vélocité de l'équipe de développement se stabilise:
  - Au premier sprint, il s'agit de la mise en route de l'équipe dont les membres ne se connaissent pas nécessairement.
  - La vélocité augmente progressivement durant les trois ou quatre premiers sprints, jusqu'à ce qu'elle se stabilise.
  - La vélocité agile est alors optimale et est relativement représentative, même si elle reste une estimation, de l'effort qu'est capable de produire l'équipe durant un sprint.

- Suivre l'évolution de la vélocité permet , de manière objective de:
  - **Planifier** avec précision, sur la base de la capacité de production de l'équipe
  - Identifier les tendances de production de l'équipe
  - **Réajuster les priorités**
  - **Reconnaître les progrès**

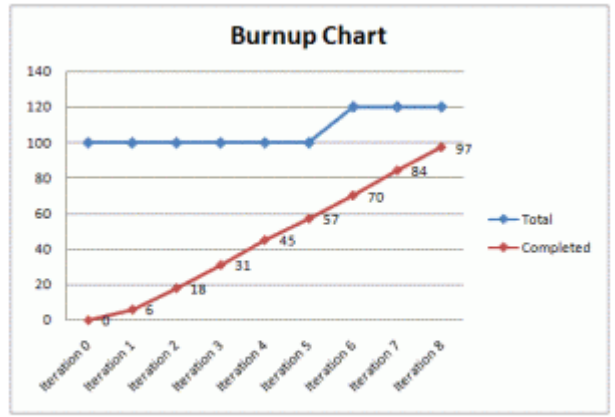


LE BURNDOWN ET BURNUP CHART

- Le **Burn Down Chart** est un outil visuel utilisé en méthodologie Agile pour suivre la progression du travail restant à accomplir dans un sprint ou un projet.
- Il s'agit d'un graphique qui montre la **quantité de travail restant à faire** (sur l'axe vertical) par rapport au temps (sur l'axe horizontal), mis à jour régulièrement
- En fin de sprint, idéalement, la courbe réelle doit atteindre 0 (pas de reste à faire)



- Le **BurnUp chart** est un graphique d'avancement utilisé pour suivre **l'avancement** du sprint.
- Il peut se réaliser en story point pour mesurer la progression des réalisations du sprint, ou en business value acquise, pour suivre l'évolution de la valeur acquise en fonction du temps. Le but consiste donc à atteindre la cible (haut du graphique) le plus tôt possible, d'où le terme « Up».
- Ce graphique n'est pas implémenté en standard dans JIRA, il faut le construire sous Excel en faisant un extrait des tickets.



Burnup Chart

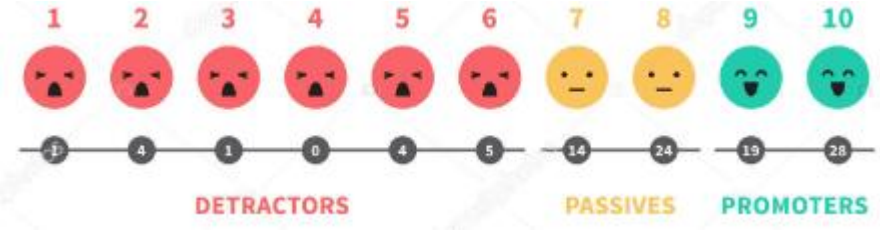
- Ces 2 graphiques peuvent être partagés lors de la démonstration avec le reste de l'équipe pour montrer l'avancement acquis lors du sprint et la valeur acquise du produit

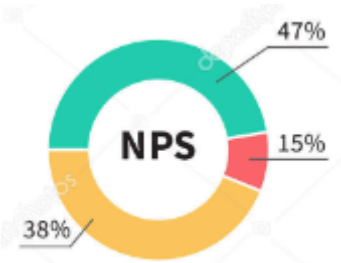
LE NET PROMOTER SCORE

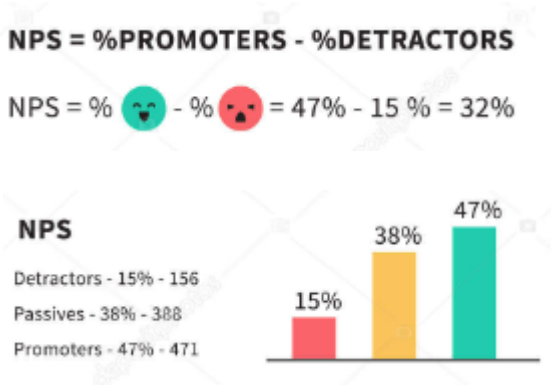
- Le **Net Promoter Score (NPS)** est une métrique utilisée pour mesurer la **satisfaction et la fidélité des utilisateurs**, qui peut être employée en agile

- Il permet un/une :
  - **Feedback rapide** et itératif en mettant l'accent sur l'amélioration continue et la réactivité aux besoins des utilisateurs.
  - **Alignement avec la satisfaction client** en aidant à mesurer si les itérations successives des produits répondent aux attentes des utilisateurs et génèrent de la satisfaction.
  - **Priorisation des améliorations** via l'identification des domaines spécifiques nécessitant des améliorations et prioriser les changements qui auront le plus grand impact sur la satisfaction client.
  - Culture l'amélioration continue en incitant les équipes à innover et à s'améliorer constamment pour augmenter la satisfaction

- Il est calculé à partir d'une seule question posée aux utilisateurs : "Sur une échelle de 0 à 10, quelle est la probabilité que vous recommandiez notre produit/service à un ami ou un collègue ?"







E ET SI PLUSIEURS PROJETS EN AGILE DOIVENT SE COORDONNER?

1 L'AGILITÉ À L'ÉCHELLE À RTE

AGILITÉ À L'ÉCHELLE

- **L'agilité à l'échelle** est une approche qui applique les principes et les pratiques Agile au-delà d'une seule équipe pour **coordonner et harmoniser le travail de multiples équipes dans une grande organisation, afin de construire plusieurs produits répondant à un besoin**

- Cela permet **d'aligner les objectifs**, de favoriser la collaboration et **d'améliorer la livraison continue de valeur à grande échelle**, notamment sur des chaînes applicatives comme l'équilibre offre demande, ou la mise en place de projets importants comme Plasma

- Différents Framework d'agilité à l'échelle existent (LeSS : Large Scale Scrum, Nexus qui se concentre sur les interactions entre équipes).

- RTE a fait le choix de mettre en place et **d'adapter Safe** (Scale Agile Framework) à son contexte
    - Ce framework, qui comprend plusieurs niveaux (équipe, programme par exemple) fournit des orientations structurées sur les rôles, les responsabilités et les activités nécessaires à une mise à l'échelle réussie.
    - Il met l'accent sur l'alignement, la collaboration et la livraison entre plusieurs équipes Agile.

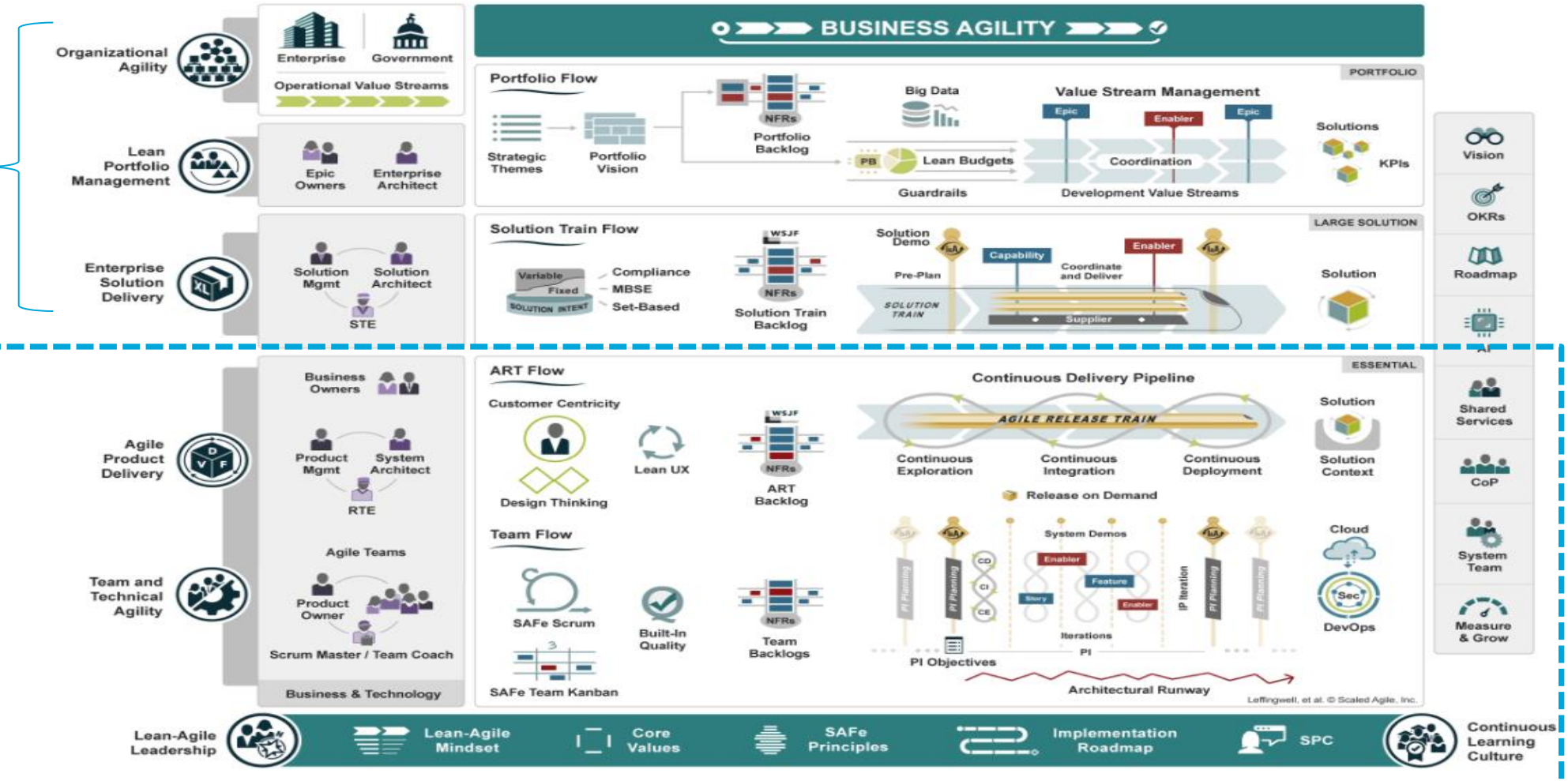
Pour en savoir plus [SAFe 6.0 (scaledagileframework.com)]

AGILITÉ À L'ÉCHELLE

- SAFe intègre des principes d'Agile, Lean et DevOps pour aider les grandes organisations à améliorer leur productivité, leur qualité et leur délai de mise sur le marché.

Le niveau Portefeuille et Solution delivery ne sont pas utilisés à RTE

NIVEAU IMPLÉMENTÉ À RTE



LE VOCABULAIRE DE L'AGILITÉ À L'ÉCHELLE

- Le **ART Backlog** est la liste de toutes les fonctionnalités qui sont destinées à répondre aux besoins des utilisateurs de la solution.
  Il est sous la responsabilité du **Product Manager (PM)**.
- Le **Product Increment** ou Incrément de produit est l'ensemble des éléments du l'**ART Backlog** fini pendant le dernier sprint, et aussi ceux finis pendant les sprints précédents. Cet incrément doit être utilisable, qu'il soit publié ou non.
- **Le PI** (**Program Increment** ou increment de programme) est l'intervalle de temps cadencé pour construire et valider un incrément complet de la solution, alimenté par le solution backlog.
  - A RTE, le PI dure généralement 12 semaines, et le temps consacré à l'élaboration de la solution est appelé wagon
  - La répétition de plusieurs itérations, permettant d'obtenir une solution complète, est appelée train
- Chaque incrément de programme **synchronise le travail de plusieurs équipes** agiles pour fournir une agrégation de valeur digne d'intérêt pour les utilisateurs.
- Outre les **instances de synchronisation régulières** des produits, tout au long du **Program Increment (PI),** qui est l'équivalent du sprint de l'équipe agile, les principales instances sont:
  - Le **PI (Program increment)** est l'équivalent du sprint de l'équipe agile
  - **La PI Demo** permet de **démontrer les fonctionnalités développées** durant un PI (Program Increment) à toutes les parties prenantes.
  - **Le PI Planning** est une réunion clé dans la méthode SAFe permettant aux équipes Agile de planifier et aligner leurs travaux aboutir à la construction du « Program Increment (PI) » ou produit. Il est animé par le **Release Train Engineer (RTE)** et le **Product Manager**
  - **La PI Retro** permet de prendre du recul sur les méthodes de travail et d'améliorer la méthodologie permettant d'aboutir au prochain product increment

2 LES ROLES SPÉCIFIQUES À L'AGILITÉ À L'ÉCHELLE ADAPTÉS À RTE

LES RÔLES SPÉCIFIQUES À L'AGILITÉ A L'ÉCHELLE 1/2

Product Manager

- Le **Product Manager** est un rôle en méthodologie AGILE. Dans le cadre des projets RTE ses activités peuvent être complétées par les activités des Responsables Projet Métier (RPM) dans le cas de projets multi-métiers.
- Il est responsable de **l'identification et de l'arbitrage des besoins de tous les métiers** pour un ou plusieurs produits donnés.
- Sa responsabilité est de définir un ensemble de produits qui **apporte le maximum de valeur métier aux différents utilisateurs dans le temps et le budget** imparti aux différents projets qui sollicitent les produits en question. A ce titre, il est **propriétaire du Backlog** des produits qu'il priorise, en lien avec les Product Owner (PO).
- Pour les gros projets ou les thématiques (Agile à l'échelle), la vision métier est à deux niveaux maximum pour RTE:
  - **Equipe** – Les Products Owners portent la vision métier et prennent des décisions rapides concernant le contenu local (un produit) au nom de l'Agile Team.
  - **Produits** – Le Product Manager porte la vision métier des produits et est responsable des décisions relatives au niveau de **l'Agile Release Train** (ARTs)
- Si le produit est multi-métiers ou multi-projets, le rôle de Product Manager est porté par la Maîtrise d'Ouvrage (MOA).

LES RÔLES SPÉCIFIQUES À L'AGILITÉ A L'ÉCHELLE 2/2

Release Train Engineer (RTE)

- En Agile à l'échelle, le Release Train Engineer (RTE) a un rôle essentiel de **coordination** des équipes Agile au travers de la notion d'Increment.

- A ce titre il :
  - Aide le Product Manager à piloter la réunion PI planning,
  - Suit la production du PI Planning (Backlog Increment)
  - Pilote la réunion de démonstration associé au PI Planning.

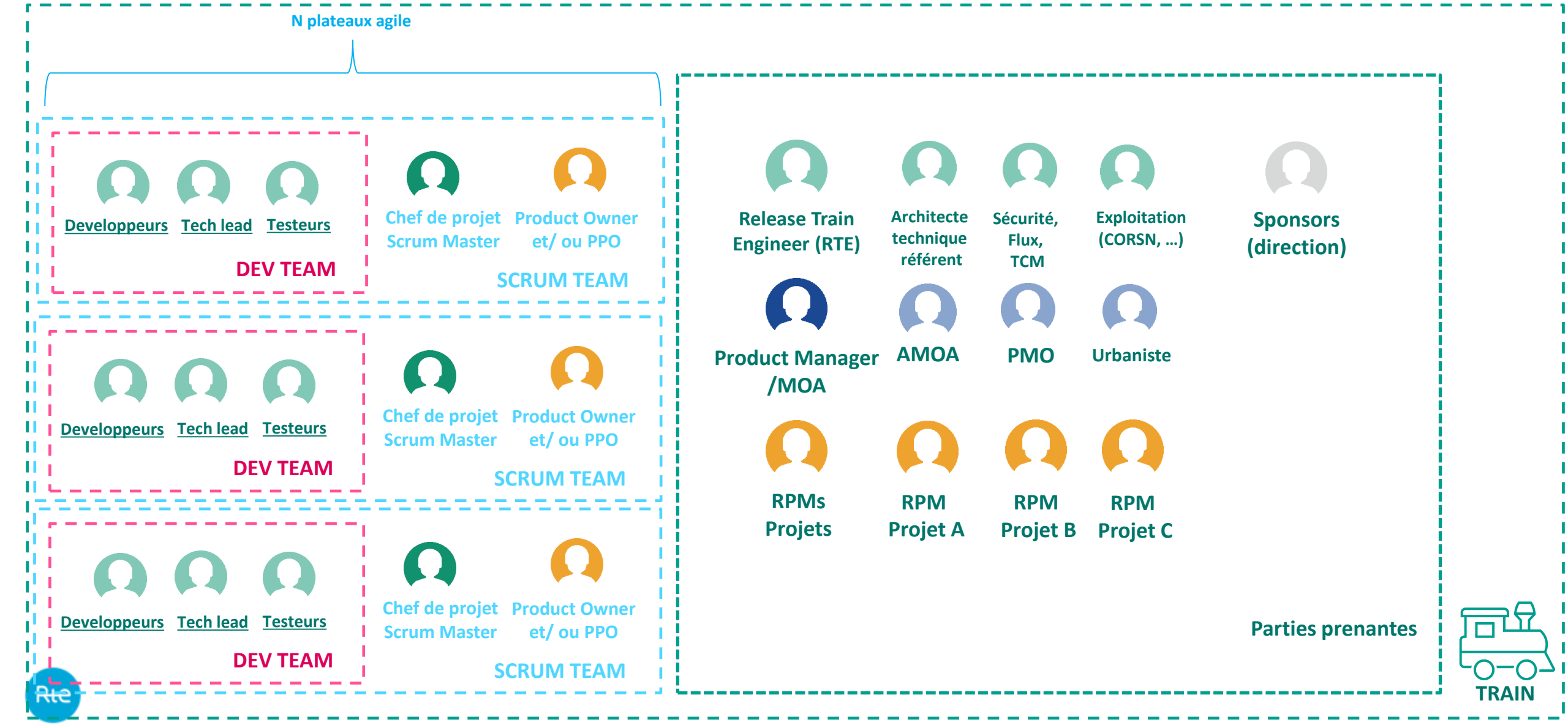
- Ses interlocuteurs les plus importants sont :
  - Le Product Manager (responsable de la partie fonctionnelle et du produit répondant à tous les besoins)
  - La Maîtrise d'Ouvrage (MOA), qui joue le rôle de product manager dans le cas de projets multi-métiers,
  - L'Architecte d'Entreprise et l'Architecte Technique Référent
  - Les Scrum Master (SM) des différentes équipes de développement.

- Il est **garant du cadre méthodologique en Agile à l'échelle**. Il participe à **la gestion des risques**, **contribue à l'amélioration continue** et aide à la **résolution des difficultés**.

- Ce rôle de coordination, en général tenu par un coordinateur SI à RTE est **essentiel dans le cas de projets faisant intervenir plusieurs équipes de développement et/ou plusieurs métiers** répondant à tous les besoins.

- **Le Product Manager et le Release Train Engineer** sont un binôme clé pour la réussite du projet

L'ÉQUIPE D'AGILITÉ À L'ÉCHELLE CHEZ RTE



3 LES RITUELS SPÉCIFIQUES À L'AGILITÉ À L'ÉCHELLE

LES RITUELS: LE PROGRAM INCREMENT PLANNING (PI PLANNING)

QUI

- RTE
- Product Manager
- RPM

N plateaux agiles coordonnés

- Développeurs et Techlead et Testeurs
- Product Owner et PPO
- Chef de projet (CP), Scrum Master

QUAND

A chaque incrément, généralement toutes les 12 semaines

COMBIEN DE TEMPS

- Le **PI Planning** vise à **synchroniser et fédérer les équipes** autour d'objectifs communs et à coordonner les efforts pour **maximiser la valeur livrée**.

- Il se déroule donc plusieurs temps:
  - ✓ **Avant le PI**, **préparation** par **chaque product owner** et chaque équipe des **backlogs** des fonctionnalités à prioriser, selon la vision et les objectifs donnés par les sponsors et le product manager
  - ✓ Le premier jour du PI:
    1. La **vision stratégique** et les priorités sont présentées à l'ensemble des membres par le sponsor ou le product manager
    2. Les product owners présentent les fonctionnalités et les priorités et chaque équipe planifie ses sprints, identifie les risques et les dépendances avec les autres applications
  - ✓ Le second jour du PI:
    1. Les équipes **partagent leurs projections** et les ajustent en fonction des retours et **dépendances associées**
    2. Elles discutent entre elles pour s'assurer de la bonne compréhension des interactions
    3. Toutes les équipes se réunissent pour:
       - **Consolider les objectifs de l'increment** et valider les moyens de l'atteindre
       - **Identifier les risques globaux**, ainsi que réfléchir à des méthodes d'atténuation des risques
       - Réaliser un vote de confiance sur l'atteinte des objectifs du program increment

- Le PI Planning est essentiel pour **co-construire et aligner les efforts** des équipes Agile à l'échelle de l'entreprise, garantissant une **livraison de valeur cohérente et coordonnée en alignant les équipes autour d'une vision partagée du produit.**

LES RITUELS: LE PROGRAM INCREMENT DEMO (PI DEMO)

QUI

RTE

Product Manager

N plateaux agiles coordonnés

Développeurs et Techlead et Testeurs

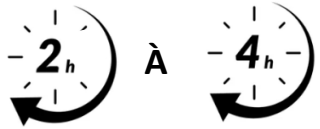
Product Owner et PPO

Chef de projet (CP), Scrum Master

QUAND

A chaque incrément, généralement toutes les 12 semaines

COMBIEN DE TEMPS



- La **PI Demo** vise à **montrer les progrès réalisés**, obtenir des retours, et **s'assurer que les produits livrés répondent aux attentes et aux besoins.**

- Comme le PI Planning, elle se déroule en plusieurs temps:
  - Avant le PI, les équipes préparent les scénarii de démonstration, ainsi que les environnements techniques
  - Au cours du PI Demo,
    - ✓ Le Release Train Engineer ou le Product Manager rappelle les objectifs du PI et le contexte des démonstrations
    - ✓ Chaque équipe Agile présente les fonctionnalités qu'elle a développées pour le product increment en se concentrant sur la valeur apportée à l'utilisateur final
    - ✓ Les parties prenantes, incluant les sponsors et les RPM donnent leur feedback.
    - ✓ Les équipes et les Business Owners évaluent si les objectifs de PI ont été atteints, tout en sachant que les points non atteints ou partiellement complétés sont discutés pour comprendre les obstacles rencontrés.
    - ✓ Le RTE résume les principaux résultats et les retours des démonstrations et explique la vision pour la suite

Le PI Demo permet:

- **D'offrir une vue d'ensemble des progrès** et des réalisations du PI.
- De recueillir du **Feedback Immédiat** des parties prenantes et des utilisateurs finaux et valider que les fonctionnalités développées répondent aux attentes et aux besoins des utilisateurs.
- **D'aligner toutes les équipes** et parties prenantes alignées sur les **objectifs et les résultats** attendus.

- La PI Demo est essentielle pour valider les progrès, recueillir des retours constructifs et garantir que le produit évolue dans la bonne direction, tout en renforçant la transparence et l'engagement des parties prenantes.

ANNEXES

LEAN CANVAS

- Le **Lean Canvas** est un outil qui permet en faisabilité de valider la compréhension du besoin en se posant 9 questions :
  - **Personae** : Quel est le personae qui a un ou des problèmes ? Qu'est ce qui le motive? Qu'est-ce qu'il essaie d'accomplir ? Il est nécessaire d'établir un Lean Canvas par personae afin d'adresser clairement les problèmes de chacun.
  - **Problème(s)** : Quels sont les 3 problèmes principaux à résoudre ?
  - **Valeur ajoutée :** En quoi votre projet répond-t-il efficacement au besoin du métier ? En quoi apporte-t-il plus de valeur que la solution existante ?
  - **Solutions** : Quelles sont les fonctionnalités qui pourraient résoudre le ou les problèmes du personae et répondre aux exigences des parties prenantes ?
  - **Équipe** : Quelles sont les compétences nécessaires pour que le projet mette en ouvre la solution ?
  - **Accompagnement** : Quelle conduite du changement est à mettre en place pour faire adopter la solution ?
  - **Gains et coûts évités :** Quels seront les couts évités grâce à votre projet ? Quel sont les gains par métier ?
  - **Coûts induits :** Quels sont les coûts (ponctuels et récurrents) liés au lancement et au fonctionnement de votre projet ?
  - **Indicateurs** : Quels sont les indicateurs mesurables qui permettront s'ils sont atteints de dire que le projet est réussi ? Se limiter à 3 indicateurs maximum. Déterminer quand, comment et par qui ils seront modifiés. Si plusieurs Lean Canvas sont produits il faut s'assurer de la cohérence des indicateurs.
- Les réponses à ces 9 questions sont apportées dans le cadre d'un atelier regroupant les acteurs métier et SI.
- Le **Lean Canvas** (vision partagée sur une page) est ensuite la **référence** permettant de **statuer si un nouveau besoin** ou une nouvelle idée **contribue** ou pas **à l'atteinte des objectifs** mesurés par les indicateurs.

LEAN CANVAS TYPE

IMPACT MAPPING

- **L'impact Mapping** permet d'identifier les fonctions qui ont le plus de valeur pour les utilisateurs.
  Cette méthode consiste à répondre à 4 questions :
  - **Pourquoi ?** - Déterminer l'objectif mesurable à atteindre (reprendre chaque indicateur cible du Lean Canvas). Un Impact Mapping est fait pour chaque objectif.
  - **Qui ?** - Lister les Personae (impactés d'une manière plus ou moins directe) qui devront interagir pour l'atteinte de l'objectif précédemment identifié.
  - **Comment ?** - Énumérer pour chaque acteur le processus ou le geste métier qui contribue à l'atteinte de l'objectif.
  - **Quoi ?** - Lister les fonctions à mettre à disposition de l'utilisateur pour lui permettre d'atteindre l'objectif.
- La liste des fonctions identifiée au niveau "Quoi?" va permettre d'initialiser la première version (**MVP**).
- Cette méthode permet d'améliorer la communication entre SI et métier en créant une image partagée des fonctions prioritaires. Elle permet de réduire le coût du projet en éliminant les fonctions superflues. De plus la construction en atelier permet d'imaginer les multiples moyens pour atteindre l'objectif ce qui est une première étape pour la conduite du changement.
- L'Impact Mapping est construit dans le cadre d'un atelier regroupant les acteurs métier et SI.

ANNEXE : ITEMS DE DEFINITION OF READY - DOR

Eléments méthodologiques

- La user story est ajoutée dans le backlog JIRA
- Business Value et Story points associés ont été estimés et sont rentrés dans JIRA

Eléments descriptifs

- La user story comprend un titre clair et succinct, une description
- Elle est rédigée conformément aux recommandations « En tant que… je veux… Afin de »
- Les documents spécifiques nécessaire lui sont associés (pour un écran , la maquette utilisateur, pour une interface, le contrat d'interface)
- Elle a été relue par les membres de l'équipe qui l'ont comprises et ont validé sa faisabilité

RGPD & Confidentialité

- Si des données personnelles sont utilisées, les mesures de sécurité associées et droits d'accès sont décrits

Dépendances et risques

- Les dépendances (au sein du projet ou avec l'externe) sont identifiées et décrites
- Les parties prenantes concernant les dépendances sont informées du planning prévisionnel
- Les spécifications ont été relues et validées par les parties prenantes si nécessaire

Critères d'acceptance

- Les étapes de test et validation de la user story sont décrites : données utilisées en entrée, comportement et résultats attendus, profils utilisateurs et rôles à utiliser, niveaux d'accès (droits et permission) décrits, ainsi que critères de confidentialité
- Les tests sont écrits en GHERKHIN

ANNEXE : ITEMS DE DEFINITION OF DONE - DOD (1/3)

Code

- Le code est écrit, revu par un pair et merge dans la branche principale.
- Les standards de codage de l'équipe sont respectés.
- Les commits sont bien documentés et suivent les conventions de nommage.

Tests

- **Les tests unitaires sont écrits et couvrent au moins 60% du code nouvellement ajouté ou modifié (critère Obligatoire)**
- **Le code ajouté ou modifié doit obtenir une note A sur les critères suivants : maintenabilité, fiabilité, sécurité (critère Obligatoire)**
- **Le code en doublon doit être inférieur à 5% du code total (critère Obligatoire)**
- Tous les tests unitaires passent sans erreur.
- Les tests d'intégration sont écrits et exécutés avec succès.
- Les tests de régression sont passés et aucun bug critique n'a été introduit.
- Les tests d'acceptation sont créés et réussis.
- Les tests automatisés sont concluants et ne remontent pas d’erreur.

Documentation (a minima un critère concernant la documentation doit figurer dans la DOD)

- La documentation technique et fonctionnelle est mise à jour pour refléter les changements effectués.
- Les commentaires pertinents sont ajoutés au code pour expliquer les parties complexes ou critiques.
- La documentation utilisateur (si nécessaire) est rédigée ou mise à jour.

ANNEXE : ITEM DE DEFINITION OF DONE – DOD (2/3)

Validation (a minima un critère concernant la validation doit figurer dans la DOD)

- Le code est déployé dans un environnement de test.
- Les Product Owners ou les parties prenantes ont validé les fonctionnalités développées.
- Les critères d'acceptation des user stories sont satisfaits et validés.

Qualité

- Le code a passé un scan technique (Sonar) qui permet de vérifier différents points de qualité, écoconception, sécurité (si applicable) et aucun problème majeur n'a été détecté.
- Les performances de l'application sont testées et respectent les critères définis (par exemple, le temps de réponse doit être inférieur à 200ms).
- Aucune dette technique critique n'est introduite.

Respect du Design System

- Les éléments visuels et les composants UI respectent les directives du design system.
- Les styles, les couleurs, les typographies et les espacements sont conformes aux spécifications du design system.
- Les nouveaux composants ou modifications sont revus par l'équipe de design pour assurer la cohérence.

Ergonomie

- Les interfaces sont conçues pour être intuitives et faciles à utiliser.
- Les principes de design centrés sur l'utilisateur sont appliqués pour améliorer l'expérience utilisateur.
- Les prototypes et les maquettes ont été testés avec des utilisateurs finaux pour recueillir des feedbacks et faire des ajustements.

ANNEXE : ITEM DE DEFINITION OF DONE – DOD (3/3)

Accessibilité Numérique

- Les pages et les composants respectent les normes d'accessibilité (par exemple, WCAG 2.1).
- Les éléments interactifs sont accessibles via le clavier et les lecteurs d'écran.
- Les contrastes de couleur, les tailles de police et autres critères d'accessibilité sont vérifiés et conformes.

Conformité RGPD

- Les données personnelles des utilisateurs sont traitées en conformité avec le RGPD.
- Les utilisateurs sont informés de la collecte et du traitement de leurs données personnelles.
- Les consentements sont obtenus et documentés avant la collecte des données personnelles.
- Les données personnelles sont sécurisées et protégées contre les accès non autorisés.

Déploiement

- La fonctionnalité est intégrée dans la branche principale sans conflits.
- La fonctionnalité est déployée dans l'environnement de production ou prête à être déployée selon les procédures de déploiement.
- Les scripts de déploiement (si nécessaire) sont créés et testés.

Communication

- Les parties prenantes sont informées des nouvelles fonctionnalités ou des modifications apportées.
- Les membres de l'équipe sont au courant des changements via une réunion de sprint ou un email récapitulatif.
- Les tickets Jira (ou tout autre outil de gestion des tâches) sont mis à jour pour refléter l'état actuel du travail.