

# Scrum

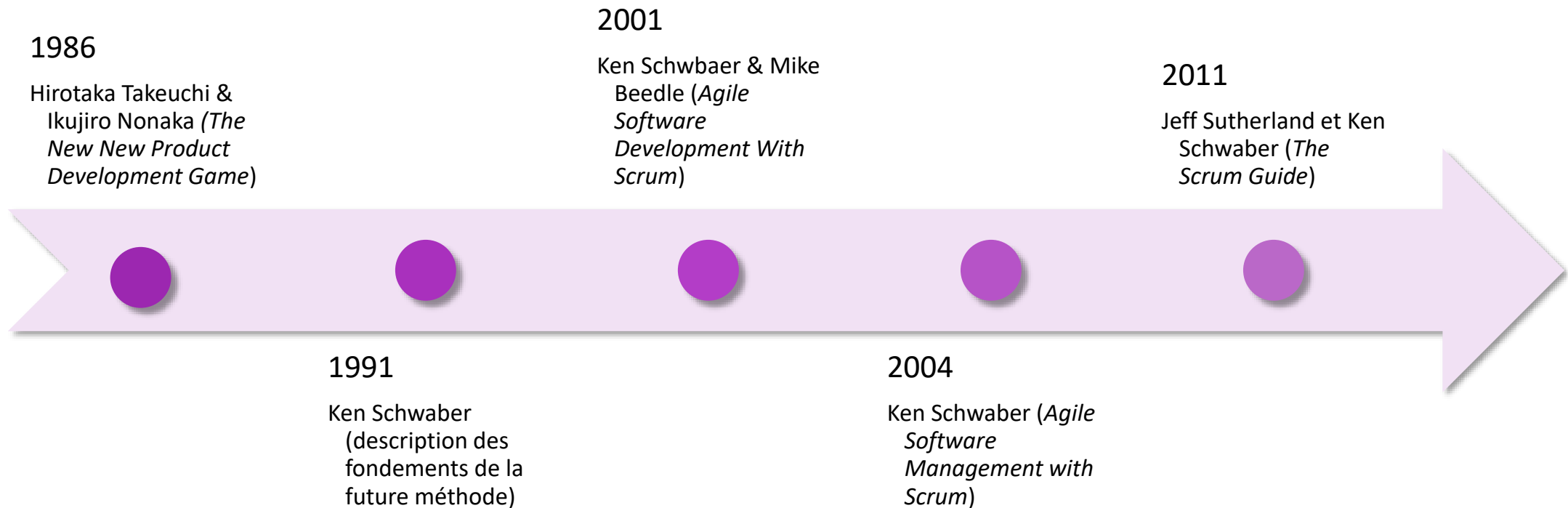
Emmanuel CHAUVET

# Ressources Lynda

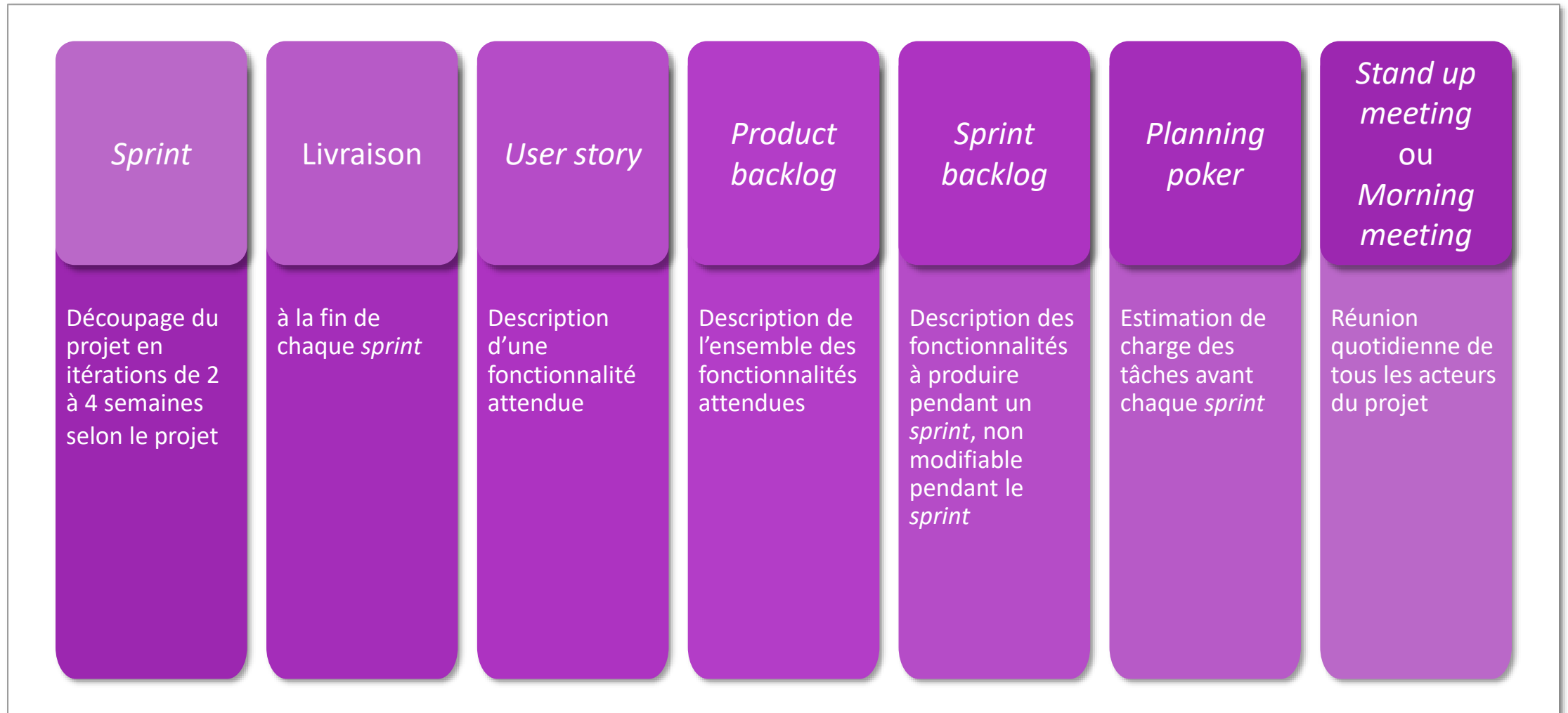
- Les fondements de la gestion de projet agile  
<https://www.lynda.com/fr/Project-Management-tutorials/fondements-gestion-projet-agile/568984-2.html?srchtrk=index%3a1%0alinktypeid%3a2%0aq%3ascrum%0apage%3a1%0as%3arelevance%0asa%3atrue%0aproducttypeid%3a2>
- Scrum : The Basics  
<https://www.lynda.com/Business-Skills-tutorials/Scrum-Basics/550619-2.html?srchtrk=index%3a2%0alinktypeid%3a2%0aq%3ascrum%0apage%3a1%0as%3arelevance%0asa%3atrue%0aproducttypeid%3a2>
- Scrum : Advanced  
<https://www.lynda.com/Business-Skills-tutorials/Scrum-Advanced/550574-2.html?srchtrk=index%3a1%0alinktypeid%3a2%0aq%3ascrum%0apage%3a1%0as%3arelevance%0asa%3atrue%0aproducttypeid%3a2>

# Un peu d'histoire

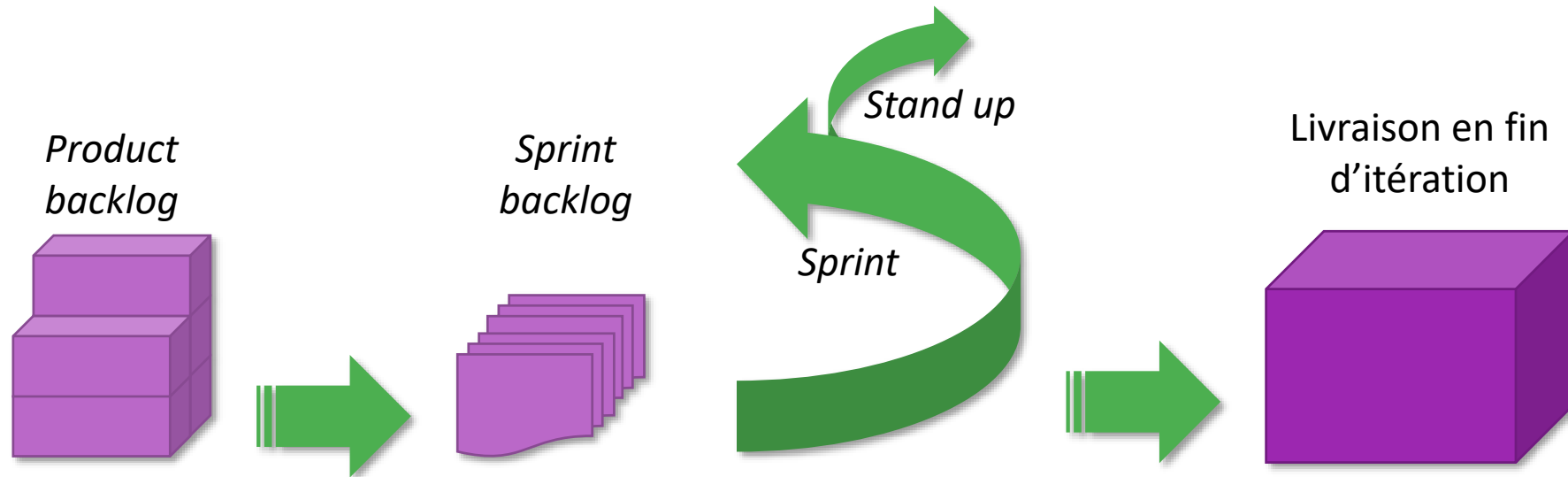
*Scrum* = mêlée de rugby



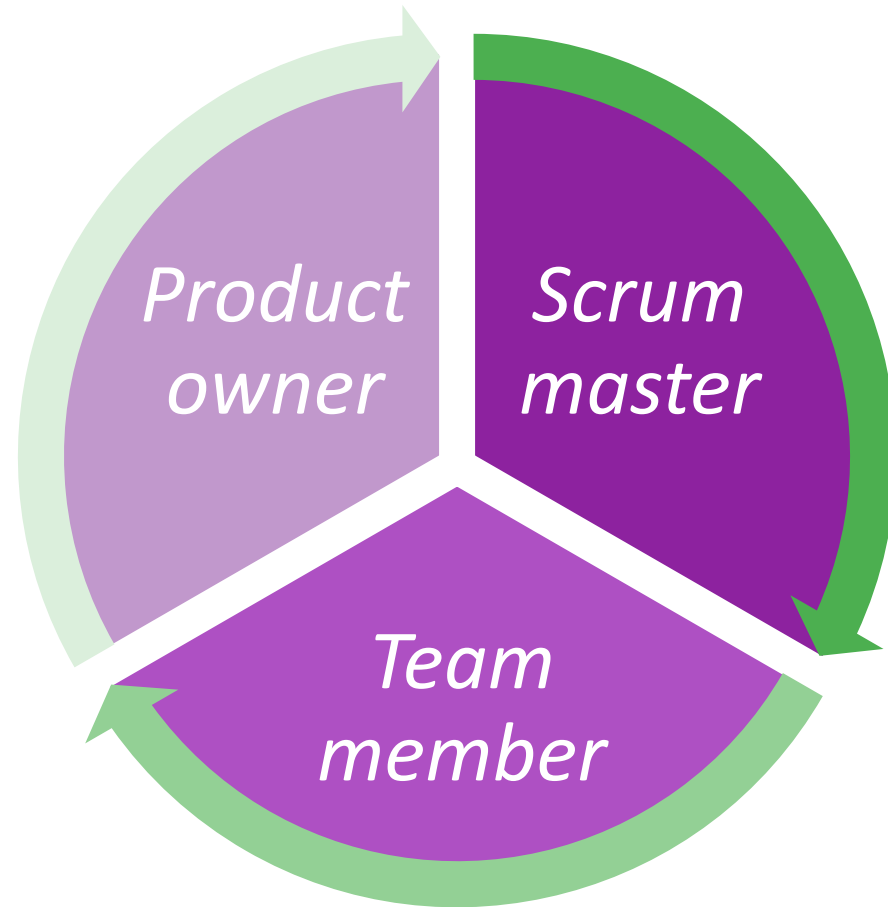
# Les fondamentaux



# Déroulement d'un *sprint*



# Les acteurs



# Product owner



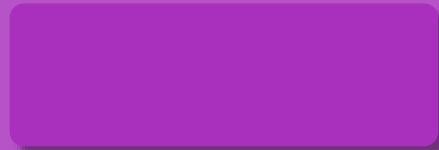
Représentant officiel du client, present avec l'équipe



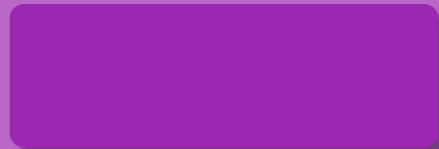
Interlocuteur principal du *scrum master* et des *team members*  
Répond aux questions qui surviennent (*stories*, maquettes...)



Définit les besoins du produit et rédige les spécifications



Se fait aider de responsables fonctionnels pour la rédaction des spécifications



Définit et priorise les *users stories* pour chaque *sprint*

# Scrum master



Veille à la mise en application de la méthode et au respect de ses objectifs



Lève les obstacles éventuels qui empêcherait l'avancement de l'équipe et du projet pendant les *sprints*



! Ce n'est pas un chef de projet !



# Team members



Chargé de la réalisation du *sprint* et d'un produit utilisable en fin de *sprint*

Développeur, Architecte, Testeur, Infographiste, Intégrateur...

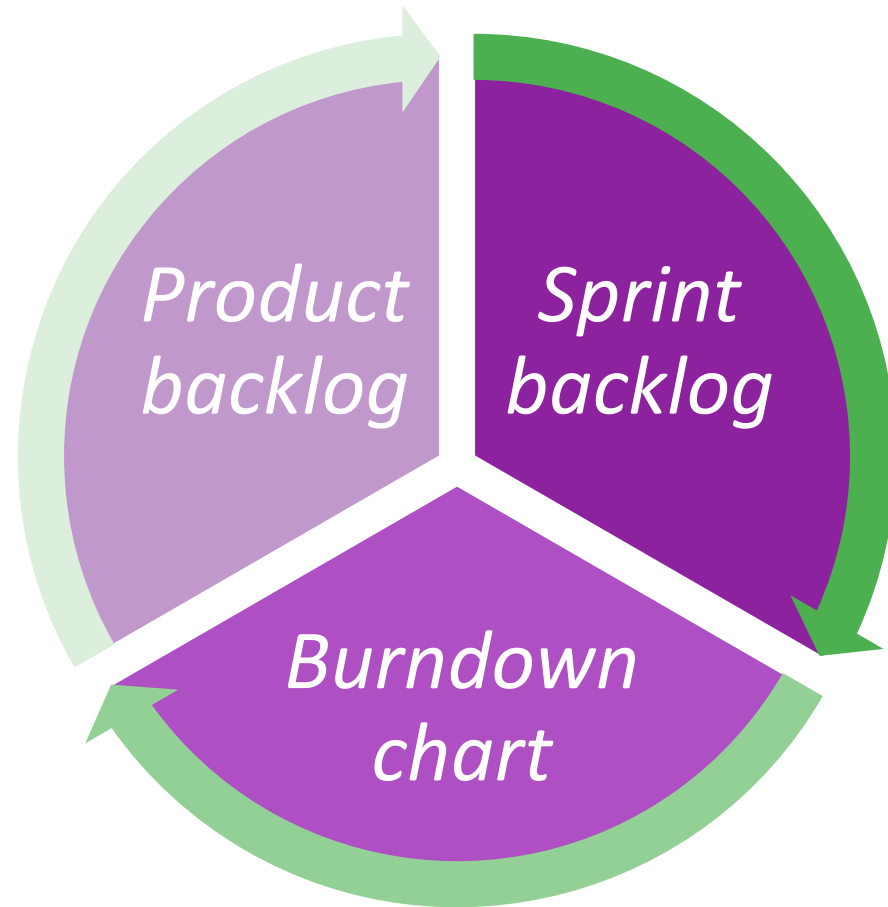


Impliqué très en amont dans le projet



Participe à l'évaluation de charge, à la conception, aux arbitrages techniques...

# Les artefacts



# Product backlog

## « Simple » inventaire des *user stories*

User Stories	Valeur métier	Story points
Story A	1	5
Story B	2	8
Story C	3	1
Story D	4	8
Story E	5	2
Story F	6	2
Story G	7	2

**Valeur métier** : priorisation selon les critères métier (la valeur accordée à la *story*)

**Story points** : estimation de l'effort nécessaire à la production de la *story*



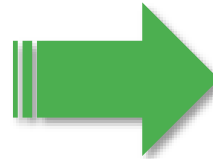
[Template de \*product backlog\* plus complet](#)

# Sprint backlog

## Sélection de *stories*

- à produire pendant un sprint
- par le *product owner*
- en accord avec le *scrum master* et les *team members*
- construit au fur et à mesure de l'avancement (1 à 2 *sprints* à l'avance)

User Stories	Priorité métier	Story points
Story A	1	5
Story B	2	8
Story C	3	1
Story D	4	8
Story E	5	2
Story F	6	2
Story G	7	2

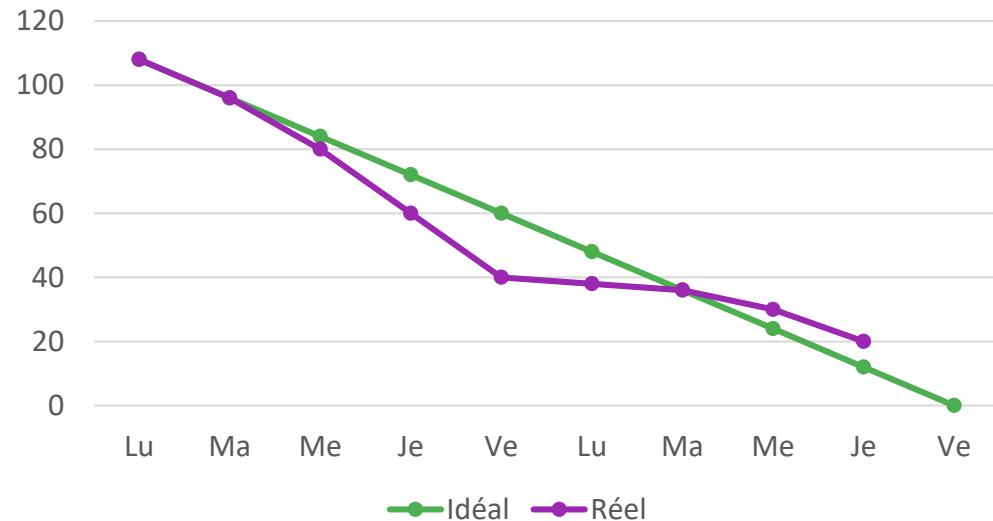


User Stories	Priorité métier	Story points
Story A	1	5
Story C	3	1

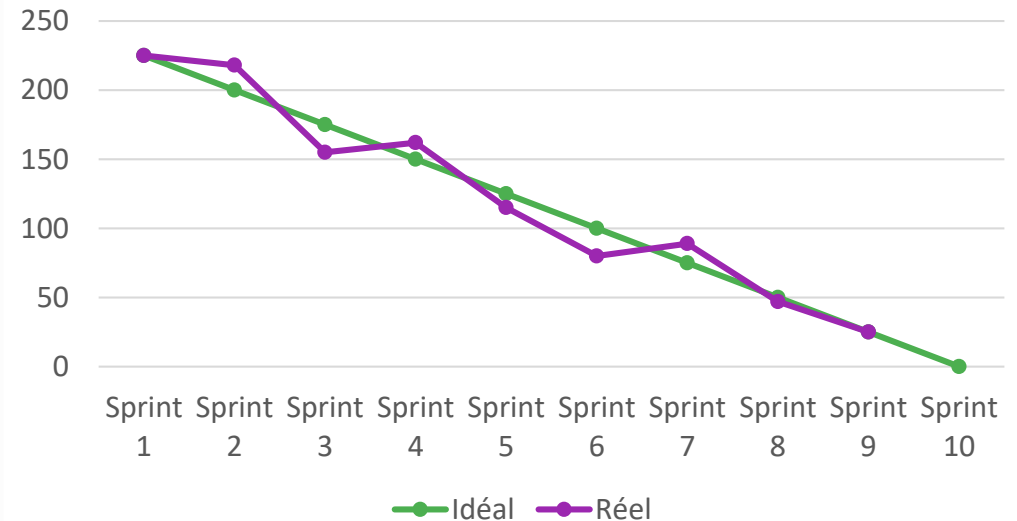
# Burndown chart

- Graphique d'avancement basé sur le « reste à faire »
- Indique l'effort en fonction du temps restant (jours, *sprints*...)
- Souvent utilisé pour piloter les *sprints* ou les *releases*
- Mise à jour fréquentes : quotidienne pour les *sprints*, fin de *sprints* pour les *releases*

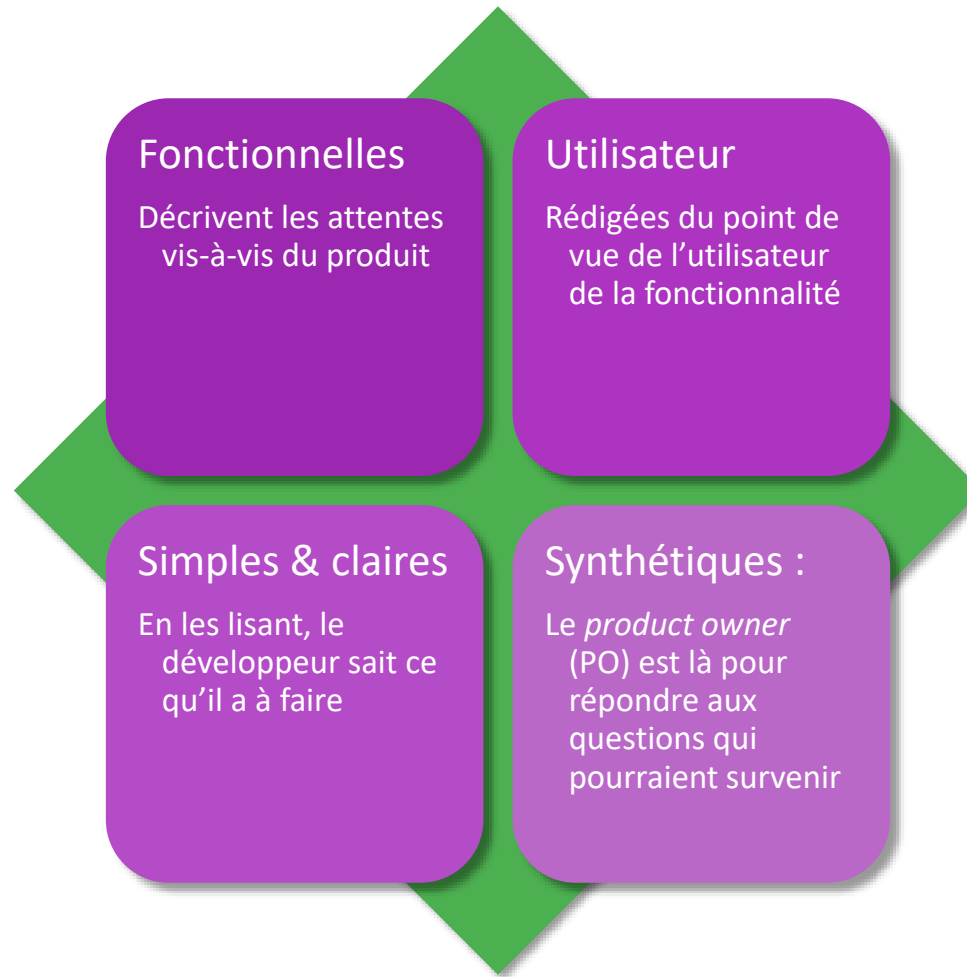
Burndown chart de sprint (sprint de 2 semaines)



Burndown chart de release



# User story – les règles



# User story... au format A5

Titre	Rattachement	Titre	Rédigé par le PO
Valeur métier	Propriétaire	Rattachement	Défini par le PO et l'équipe
Effort	Affectation	Valeur métier	Définie par le PO
Type de carte	Statut	Propriétaire	Défini par le PO, sauf pour les US <i>technical</i> ou <i>defect</i>
Objectif	Critères d'acceptation	Effort	Estimé par l'équipe
<i>En tant que...</i>		Affectation	Définie par l'équipe
<i>Je souhaite...</i>		Type de carte	Défini par le PO (US fonctionnelle) ou par l'équipe (US <i>technical</i> ou <i>defect</i> )
<i>Afin de...</i>		Statut	Défini par l'équipe
	BDD	Objectif	Rédigé par le PO
	<i>Given that (Etant donné que)</i>	Critères d'acceptation	Définis par le PO, enrichis par l'équipe
	<i>When (Quand)</i>	BDD ( <i>Behaviour Driven Development</i> )	Définis par le PO, enrichis par l'équipe
	<i>Then (Alors)</i>		

# User story – en détail

Titre	identifie la <i>US</i> au sein du <i>backlog</i> , donc clair et concis	Objectif	description de l'action utilisateur, purement fonctionnelle (pas de technique, pas d'IHM) respecte la formulation rituelle « <b>En tant que</b> » (caractéristiques utilisateurs) « <b>Je veux</b> » (exigence utilisateur) « <b>Afin de</b> » (bénéfice utilisateur)
Rattachement	généralement à une <i>feature</i> (ensemble de fonctionnalités liées)	Critères d'acceptation	description des critères que le propriétaire de la <i>US</i> va contrôler pour valider ou rejeter la livraison (ex : règles métiers à appliquer, textes à afficher...) peut contenir des informations/contraintes techniques (ex : regexp email, captcha, dédoublonnage...)
Valeur métier	indicateur d'importance et de priorité (de 0 à 100 par ex.)	BDD ( <i>Behaviour Driven Development</i> )	description des cas de test ; permettent au d'écrire les scénarios des test en même temps (voire avant) qu'il développe respecte la formulation rituelle « <b>Etant donné que</b> » (contexte) « <b>Quand</b> » (action utilisateur) « <b>Alors</b> » (comportement/résultat attendus)
Propriétaire	désigne le responsable de la validation de la <i>US</i> produite (généralement le <i>product owner</i> )		
Effort	estimation de l'effort nécessaire pour la réalisation de la <i>US</i> (cf. <i>planning poker</i> , <i>t-shirt sizing</i> )		
Affectation	qui va développer la <i>US</i>		
Type de carte	<i>user story</i> fonctionnelle (la base), <i>technical story</i> (ex : traitement auto), <i>defect story</i> (bug)		
Statut	état à l'instant <i>t</i> (ex : à faire, en cours, en test, validé, à livrer...)		



# User stories « INVEST »

I

## Indépendante

Aucune dépendance avec les autres *stories* afin d'éviter des problèmes de tests et de planification

N

## Négociable

Discussion entre PO et équipe... tant que la *story* n'est pas dans un *sprint*

V

## Valeur

Apporte une valeur à l'utilisateur, donc exprime l'objectif de l'utilisateur

E

## Estimable

Suffisamment claire pour que l'équipe puisse estimer l'effort nécessaire

S

## Small

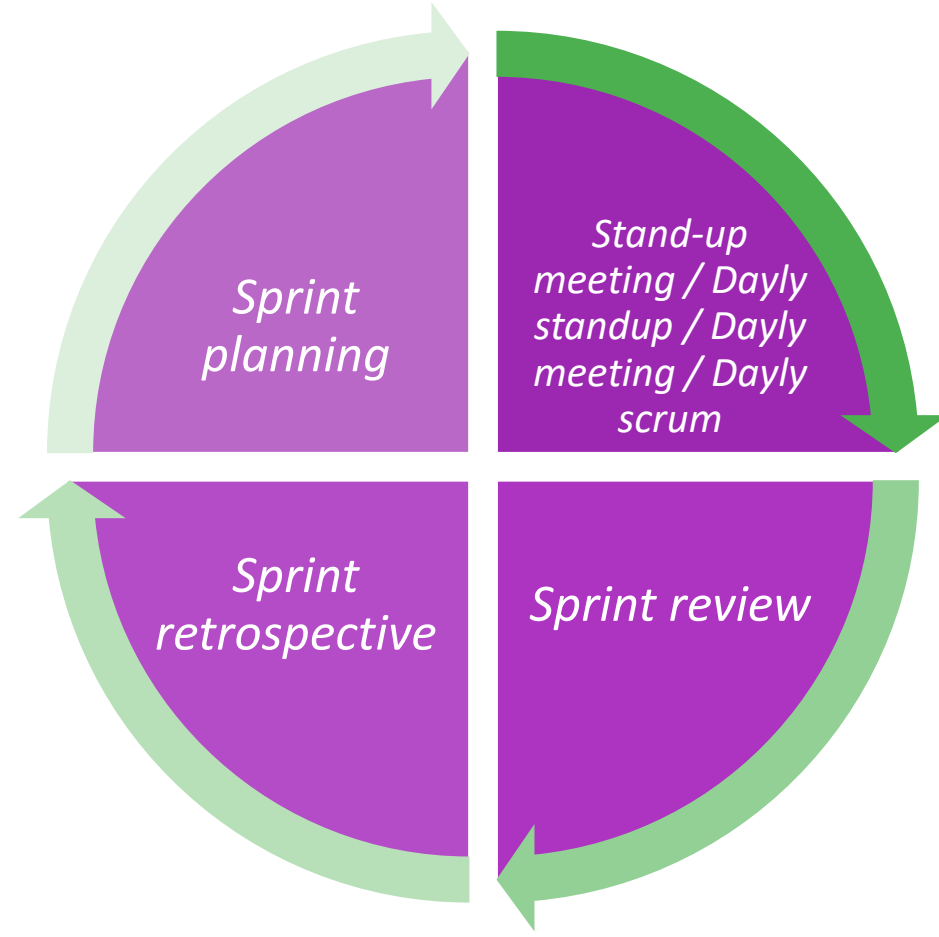
Suffisamment petite pour être planifiée simplement et sur un seul *sprint*

T

## Testable

Les tests doivent découler de la *story* de façon évidente

# Les artefacts



# Sprint planning

Qui ?

*Team members, Scrum Master, Product Owner*

Quand ?

Début de *sprint*

Combien de temps ?

Environ 1 h par semaine de *sprint*

Quoi ?

- Construction du *sprint backlog* (*sprints* longs) ou basée sur le *sprint backlog* (*sprints* courts)
- Définition, estimation (en heures) et attribution des tâches en fonction des *user stories* du *sprint backlog*

Contraintes

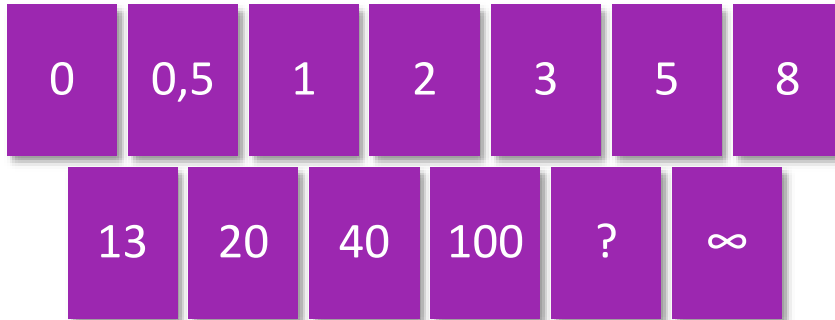
Respect du cycle agile :

- Spécifications détaillées des *user stories* (tests d'acceptation)
- Remaniement de l'architecture si nécessaire
- Conception, implémentation et tests des

composants (tests unitaires)

- Intégration et des composants (tests d'intégration)
- *Package* produit (livraison)
- Passage des tests d'acceptation

# Planning poker



Principe

- Evaluations de complexités relatives de *user story*
- Choix d'1 ou 2 étalons (fonctionnalités connues)
- Attributions de valeurs (non extrêmes) aux étalons
- Cycles d'évaluation "secrète"/débat jusqu'à obtention de l'unanimité
  - Peut être fait en début de projet, puis affiné ensuite

Planning  
poker

- Ludique (utilisation des cartes)
- Communication en amont
- Implication de l'équipe

<https://scrumpoker.online/>  
<https://www.pointingpoker.com/>

Avantages

Inconvénients

- **Evaluation de complexités relatives, pas de charges**
- Traduction en charge de travail à réaliser par la suite (souvent difficile)
- Ne remplace pas forcément les autres méthodes

# Stand-up meeting / Dayly standup / Dayly meeting / Dayly scrum

Qui ?

*Team members, Scrum Master, Product Owner*

Quand ?

Tous les jours, en début de journée

Combien  
de temps ?

15 mn **maximum**

Comment ?

Tout le monde debout

A quoi ça  
sert ?

Chaque *team member* expose en 1 mn :

- Ce qu'il a fait la veille pour l'atteinte des objectifs du projet
- Ce qu'il va faire aujourd'hui pour l'atteinte de ces objectifs
- Les éventuels freins pour lui ou pour l'équipe

# Sprint review

Qui ?

*Team members, Scrum Master, Product Owner*  
Eventuellement certaines parties prenantes du projet

Quand ?

Fin de *sprint*

Combien  
de temps ?

30 à 60 mn

Quoi ?

Présentation du produit du *sprint* (démonstration)  
Le tout en 5 étapes

- Rappel des objectifs du *sprint* définis pendant le *sprint planning*
- Démonstration du produit (*user stories* terminées uniquement, et par ceux qui les ont réalisées)
- Bilan du *sprint*, intégrant les retours en temps réel du *Product Owner* (y compris modifications du *product backlog*)
- Calcul de la vélocité réelle (capacité de production de l'équipe)
- Mise à jour du planning de release et du *burndown chart*

# Sprint retrospective

Qui ?

*Team members, Scrum Master, Product Owner*  
Eventuellement certaines parties prenantes du projet

Quand ?

Fin de *sprint*

Combien  
de temps ?

60 mn

Quoi ?

- Bilan du *sprint* en mode « amélioration continue » :
- Ce qui a bien fonctionné et comment capitaliser dessus => communiquez vos *best practices*
  - Ce qui n'a pas fonctionné et comment le corriger et s'améliorer => imaginez des solutions