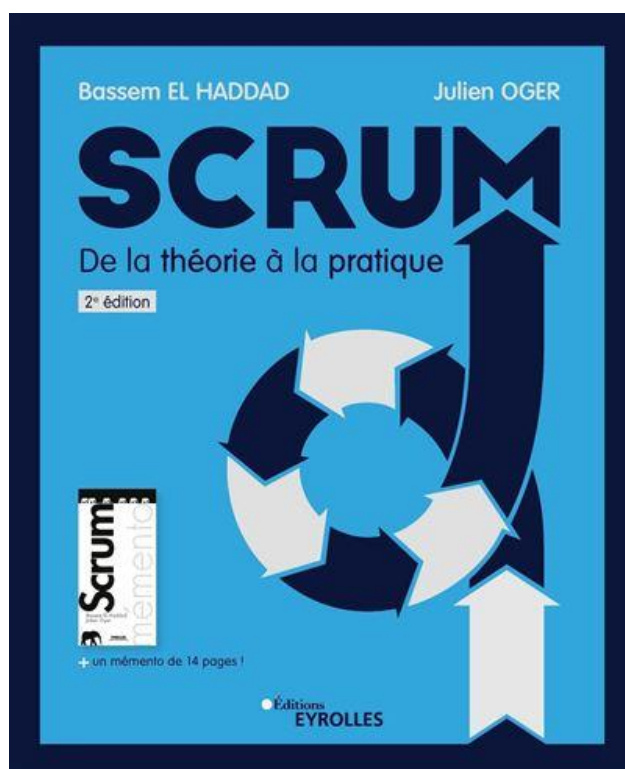


## Lecture individuelle semestre 2 Mars 2023

### Scrum de la théorie à la pratique

Figure 1 : Couverture du livre Scrum de la théorie à la pratique



Source : (El Haddad & Oger, 2019)

Etudiant : Jonathan Rapin

## Table des matières

1. Choix du livre .....	3
2. Informations générale sur l'ouvrage choisi .....	3
3. Présentation des concepts généraux de l'agilité .....	4
4. Différences entre méthode traditionnelle de gestion de projet et les méthodes agiles .....	6
5. Principes de bases de Scrum .....	9
6. Les différents rôles dans un projet Scrum .....	11
6.1 Scrum Master .....	11
6.2 Product Owner .....	11
6.3 L'équipe de développement.....	12
6.4 Autres acteurs.....	12
7. Organisation générale d'un projet avec Scrum .....	12
7.1 Phase d'initialisation du projet .....	13
7.2 Phase de préparation du projet .....	15
7.3 Phase de réalisation du projet .....	17
7.4 Phase de finalisation du projet .....	20
8. Logiciels de gestion de projets pour Scrum .....	20
9. Conclusion .....	22
Références .....	23

## Table des illustrations

Figure 1 : Couverture du livre Scrum de la théorie à la pratique .....	1
Figure 2 : Illustration des méthodes agiles .....	4
Figure 3 : Roue de Deaming .....	5
Figure 4 : Méthode en cascade ou Waterfall .....	6
Figure 5 : Le Cycle en V .....	7
Figure 6 : Processus de développement Scrum .....	7
Figure 7 : éléments constitutifs de Scrum .....	9
Figure 8 : Trois piliers de Scrum .....	10
Figure 9 : Triangle d'or d'un projet .....	11
Figure 10 : Vue d'ensemble d'un projet réalisé avec la méthode Scrum .....	13
Figure 11 : Exemple de matrice RACI .....	14
Figure 12 : Création du carnet de release à partir du carnet de produit .....	15
Figure 13 : Exemple de Sprint .....	16
Figure 14 : Le sprint dans un projet Scrum .....	17
Figure 15 : Exemple de Board Chart .....	18
Figure 16 : Exemple de Burn-up chart.....	18
Figure 17 : Exemple de Burndown Chart .....	19
Figure 18 : Interface de iceScrum .....	20
Figure 19 : Interface de Trello.....	21

## Liste des Tableaux

Tableau 1 : Fiche de l'ouvrage présenté.....	4
Tableau 2 : Tableau comparatif entre méthodes agiles et méthodes traditionnelles .....	8
Tableau 3 : Exemple de carnet de sprint .....	16

## 1. Choix du livre

La lecture de ce livre permet de mieux connaître les grands principes de la méthode de gestion de projet Scrum afin de pouvoir les appliquer dans des projets actuels ou futurs de la formation en Informatique de Gestion en filière Digital Team Academy. Le contenu de ce livre permet de compléter les compétences DTA suivantes (Rey, 2022) :

- ✓ B5 : Concevoir, défendre et mettre en œuvre un projet informatique
- ✓ M7 : Mettre en œuvre les méthodologies et les outils nécessaires aux différents champs d'action de l'organisation en les appliquant

Scrum est un cadre de travail qui est largement utilisé dans le monde du développement logiciel. La complexité croissante des projets informatiques a nécessité la mise en place de nouvelles méthodes de gestion de projet plus proche des besoins des clients et plus adaptées à un environnement technologique en constante évolution. Au début des années 2000, les méthodes agiles de gestion de projet sont progressivement adoptées par les équipes de projets informatiques. La méthode Scrum est l'une des méthodes de gestion de projet agiles les plus populaires et les plus utilisées par les équipes de développement à travers le monde.

Pour pouvoir évoluer efficacement dans une équipe de projet informatique, il est essentiel de connaître les fondements de la gestion de projet agile via la méthode Scrum. Ce cadre de travail nous permet d'être plus proche des besoins des clients et de gagner en efficacité lors de la conception de nouvelles applications ou logiciels par exemple.

## 2. Informations générale sur l'ouvrage choisi

Cet ouvrage est découpé en 8 chapitres qui permet de mieux se familiariser avec la méthode Scrum. Il est destiné à toutes les personnes travaillant dans le secteur informatique au sens large mais aussi à toutes les personnes qui veulent mieux comprendre le fonctionnement de la méthode Scrum. Il ne nécessite pas de connaissances préalables en termes de gestion de projet. Les auteurs donnent tout au long des différents chapitres des conseils pour gérer efficacement des projets informatiques.

Les deux auteurs de cet ouvrage ont commencé leurs carrières respectives par du développement Web. Ils ont travaillé sur des projets allant de quelques jours à plusieurs mois. Pour Bassem Hel Haddad « réussite d'un projet passe inévitablement par l'implication des individus et la bonne gestion des profils, tout en s'appuyant sur un cadre organisationnel en guise de fil conducteur ». (El Haddad & Oger, 2019). Les deux auteurs ont dirigés des projets IT en France dont les clients étaient des PME ou des entreprises plus grandes et qui touchaient des secteurs aussi variés que la recherche, la santé, l'éducation, le domaine bancaire, l'assurance.

Les quatre premiers chapitres de l'ouvrage sont consacrés aux concepts théoriques sur l'agilité et Scrum. Les chapitres 5 à 6 expliquent comment mettre en œuvre la méthode Scrum. Le chapitre 7 présente un exemple de projet géré avec la méthode Scrum. Ce cas pratique illustre différents éléments à prendre en compte pour mener à bien un projet dans le domaine IT. Le chapitre 8 donne quelques logiciels utiles pour gérer certains aspects de la méthode Scrum.

**Tableau 1 :** Fiche de l'ouvrage présenté

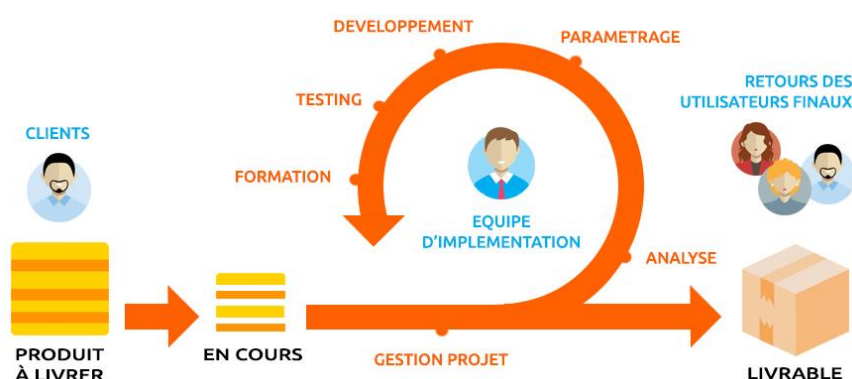
Fiche de l'ouvrage	
<b>Auteurs</b>	Bassem El Haddad , Julien Oger
<b>Maison d'édition</b>	Eyrolles
<b>Année de publication</b>	2019
<b>Nombre de pages</b>	254
<b>ISBN-13</b>	978-2212678628
<b>Mots clés</b>	Gestion de projet, agilité, Scrum

**Source :** (Amazon, 2023)

### 3. Présentation des concepts généraux de l'agilité

Les méthodes agiles ont longtemps été mis de côté par les universités et le monde professionnel qui les jugeaient trop expérimentales et inadaptées au début des années 2000. Les méthodes agiles reposent sur une meilleure communication entre les membres d'un projet et une plus grande souplesse dans l'organisation des différentes tâches à réaliser. Les méthodes agiles fonctionnent très bien dans le domaine informatique car le rythme des évolutions technologiques est rapide et des cycles de développement très rapprochés. Les méthodes agiles sont de plus en plus employées dans d'autres domaines comme l'aéronautique avec Airbus qui implémente des méthodes agiles pour développer des projets innovants en quelques mois. D'autres domaines industriels comme la construction automobile recourent aux méthodes agiles pour créer de nouveaux produits et services.

**Figure 2 :** Illustration des méthodes agiles

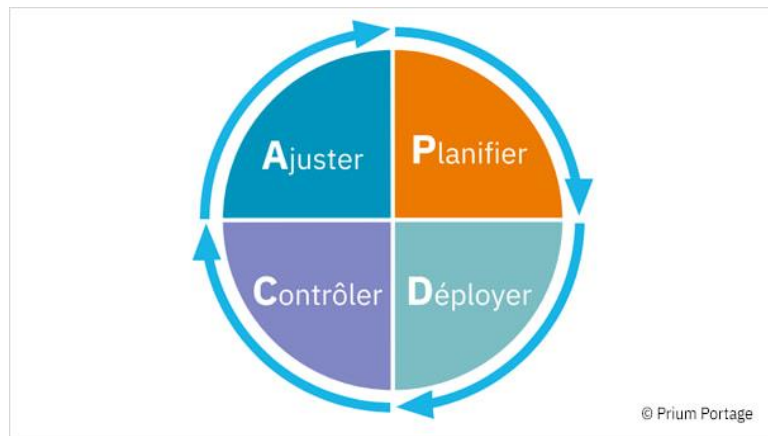


**Source :** (mercator.eu)

Dès les années 50, de nombreux travaux vont remettre en question l'organisation traditionnelle de l'industrie. Le taylorisme était largement appliquée par de nombreuses industries. Cette méthode de travail reposait sur une organisation rationnelle de la production afin de produire le meilleur rendement financier possible. L'individu n'est pas pris en compte dans l'organisation

du travail. Walter Shewhart et Edwards Deming proposent une nouvelle approche pour de l'organisation du travail qui est très éloignée du taylorisme. Toutes les actions des individus ont un impact sur le fonctionnement de l'entreprise. Parmi ces nouveaux concepts, nous avons la roue de Deaming ou le « Plan-Do-Check-Act » qui consiste à enchaîner 4 phases de manière itérative afin d'améliorer une tâche, un processus ou un produit. La roue de Deaming est un outil qui sert notamment lors de la rétrospective de Sprint dans le méthode Scrum pour identifier des éléments à améliorer et être plus efficace. Le domaine informatique commence à appliquer les théories de Deaming et Shewhart dès les années 50.

**Figure 3** : Roue de Deaming



**Source** : (Prium-MT, 2021)

En 1991, apparaît la méthode RAD (Rapid Application Développement). Cette nouvelle méthode de gestion de projet est itérative et incrémentale. Dans le domaine informatique, de nombreuses entreprises réfléchissent sur la meilleure manière de gérer les projets liés au développement de nouveaux logiciels. Les concepts des méthodes agiles ne sont pas encore standardisés dans les années 90. C'est en 2001 que 17 personnes issues du monde informatique et de la gestion de projet se réunissent pour définir les principes de bases des méthodes de gestion de projet agile.

Cette réunion va déboucher sur la création du Manifeste agile qui définit 4 grands préceptes déclinés en 12 principes.

Voici la liste des éléments importants à retenir des 12 principes de l'agilité (El Haddad & Oger, 2019) :

1. Satisfaire l'utilisateur
2. Accepter le changement
3. Livrer fréquemment
4. Travailler en synergie
5. Stimuler la motivation
6. Communiquer en direct avec les opérationnels
7. Un seul indicateur : les fonctionnalités implémentées
8. Bannir les rushs de production
9. Ne pas négliger la qualité de production
10. Rester concentré sur l'essentiel : mettre en place un produit
11. Favoriser une certaine autonomie des équipes
12. Intégrer la notion d'amélioration continue tout au long du projet

L'agilité propose 12 grands principes sur lesquels reposent les méthodes agiles telles que Scrum, eXtreme Programming ou DevOps.

La méthode Scrum reprend tous les principes de l'agilité. Nous pouvons appliquer partiellement ou intégralement la méthode Scrum dans un projet.

## 4. Différences entre méthode traditionnelle de gestion de projet et les méthodes agiles

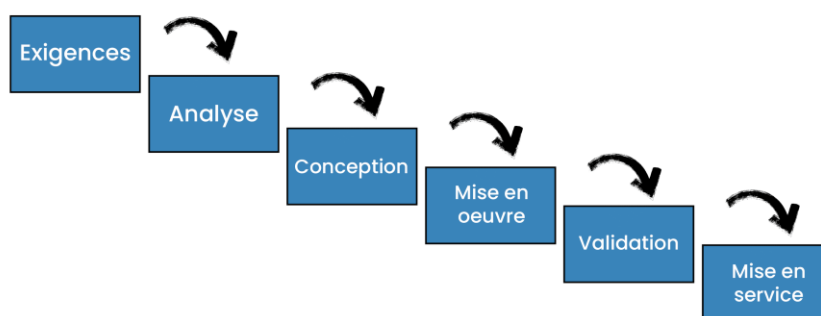
Dans l'approche traditionnelle de gestion de projet, le client donne tous les éléments nécessaires à la réalisation d'une application. Dès que l'on a terminé l'analyse préliminaire de ses besoins du clients, nous pouvons déterminer les différentes étapes nécessaires au développement de l'application. Tout est prévu dès le démarrage du projet. Toutes les étapes du projet s'enchaînent de manière linéaire et séquentiel. Les besoins du client sont bien définis. Tout est prévu à l'avance et donc les équipes de développement L'équipe de projet délivre l'application finale au client à la fin du projet. Le client n'intervient pas durant la phase d'élaboration de l'application. Nous ne pouvons pas effectuer de modifications en cours de route sur l'application. Nous disposons d'un cahier des charges bien défini et nous

Les méthodes traditionnelles de gestion de projet sont en général représentées selon deux manières :

La première est la méthode en cascade ou Waterfall. Cette méthode propose une série d'actions qui se suivent de manière linéaire pour transformer les besoins du client en une application fonctionnelle.

**Figure 4** : Méthode en cascade ou Waterfall

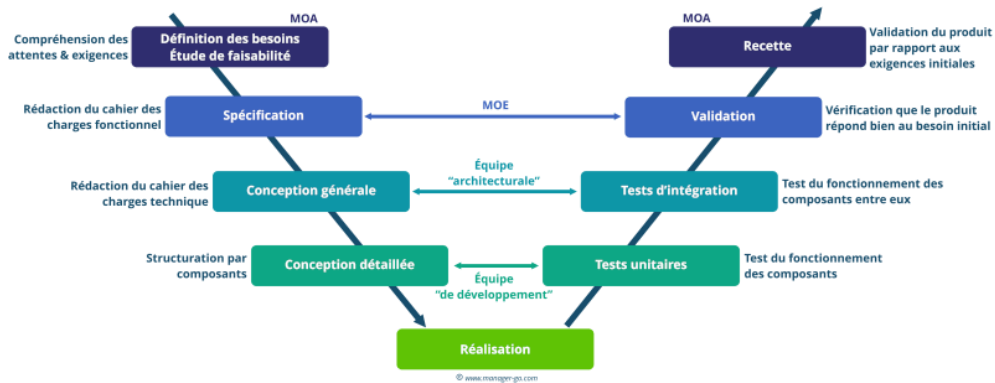
### Méthode en Cascade



**Source** : (Acca, 2021)

Le Cycle en V représente mieux les méthodes traditionnelles de gestion de projet. Toutes les étapes du projet sont représentées du début du projet jusqu'à la livraison finale de l'application. Les différentes étapes sont classés par paire. Il existe un lien de causalité entre certaines étapes du projet. La validation du produit final se base sur les spécifications décrites dans le cahier des charges. Les tests d'intégration sont effectués en fonction de ce qui a été défini lors de la conception générale de l'application.

**Figure 5 : Le Cycle en V**



**Source :** (Lardilleux, 2021)

Les méthodes agiles permettent plus de souplesse dans l'établissement de la conception de l'application. Au lieu de livrer une application totalement finie au client, nous livrons chaque fois une morceau de l'application. On vérifie que chaque élément de l'application correspond bien à ce que le client souhaite.

**Figure 6 : Processus de développement Scrum**



**Source :** (Crochet-Damais, 2022)

Voici le tableau comparatif entre les méthodes agiles et les méthodes classiques de gestion de projet. Ce tableau reprend les éléments présentés dans les pages 10 et 11 de l'ouvrage :

**Tableau 2** : Tableau comparatif entre méthodes agiles et méthodes traditionnelles

	<b>Méthodes agiles</b>	<b>Méthodes classiques</b>
<b>Objectif</b>	Satisfaire l'utilisateur	Respecter le besoin initial et les engagements
<b>Changement</b>	Accepter le changement	Opposé au changement
<b>Livraison</b>	Livrer fréquemment	Livrer en une seule fois une application « finalisée »
<b>Équipe</b>	Travailler en synergie	Travailler de façon segmentée
<b>Moteur</b>	Stimuler la motivation	Stimuler la productivité
<b>Communication</b>	direct avec les opérationnels	Communiquer de façon verticale en passant par des relais hiérarchiques
<b>Indicateurs</b>	fonctionnalités implémentées	Indicateurs clés de performance (KPI), écarts à la planification initiale, état d'avancement
<b>Rythme</b>	Éviter les périodes de rush	Adapter la production aux contraintes projets
<b>Qualité</b>	Que ce soit en méthode agile ou classique, l'excellence technique est essentielle	
<b>Autonomie</b>	Favoriser une certaine autonomie des équipes	Encadrer scrupuleusement le travail des équipes
<b>Amélioration</b>	Intégrer la notion d'amélioration continue tout au long du projet	Introspection possible mais uniquement en fin de projet

**Source** : (El Haddad & Oger, 2019)

Les méthodes de gestion de projet traditionnelles répondent aux besoins initiaux du clients. Les clients finaux découvrent l'application développée à la fin de projet. Il y a un risque de proposer un produit final qui correspond bien à ce que le client final avait demandé au début du projet mais qui ne correspond pas à ce qu'il veut. Les méthodes traditionnelles sont plus rigides et orientées sur le résultat final. Elles négligent la satisfaction client final. Il y a peu d'interactions entre l'équipe de développement, les clients finaux et les commanditaires du projet. L'approche classique a l'avantage d'être plus rigoureuse que les méthodes agiles. Ces méthodes de gestion de projets sont souvent appliquées dans des domaines critiques comme le médical, le nucléaire, l'aérospatial... Nous pouvons néanmoins appliquer une méthode Scrum pour des projets liés à des secteurs critiques. L'avantage principal des méthodes agiles et que nous pouvons effectuer des changements tout au long du projet afin que l'application finale puisse répondre aux besoins des clients finaux d'une application. Les méthodes agiles visent à mieux répartir l'effort à fournir par les membres de l'équipe de projet tout au long du processus de création d'une application. Plus le projet est complexe et plus le risque de retard ou de dépassement des coûts est grand. Les deux méthodes se rejoignent sur la qualité des livrables à fournir. L'application fournie aux clients finaux doit comporter le moins de problèmes techniques et être la plus ergonomique possible.



## 5. Principes de bases de Scrum

La terme « Scrum » signifie mêlée en français. C'est une référence direct au rugby. Dans un projet géré avec la méthode Scrum, il y a des réunions quotidiennes ou mêlées qui se déroulent chaque jour pour répartir les tâches à traiter, de faire le point sur les éléments en cours de réalisation et d'aborder les difficultés rencontrées dans le projet.

Comme toutes les méthodes agiles, Scrum ne dispose pas d'un cadre très rigide avec des méthodes de travail bien définies contrairement aux méthodes de gestion de projet traditionnelles.

Scrum propose une méthode de travail contenant les éléments suivants :

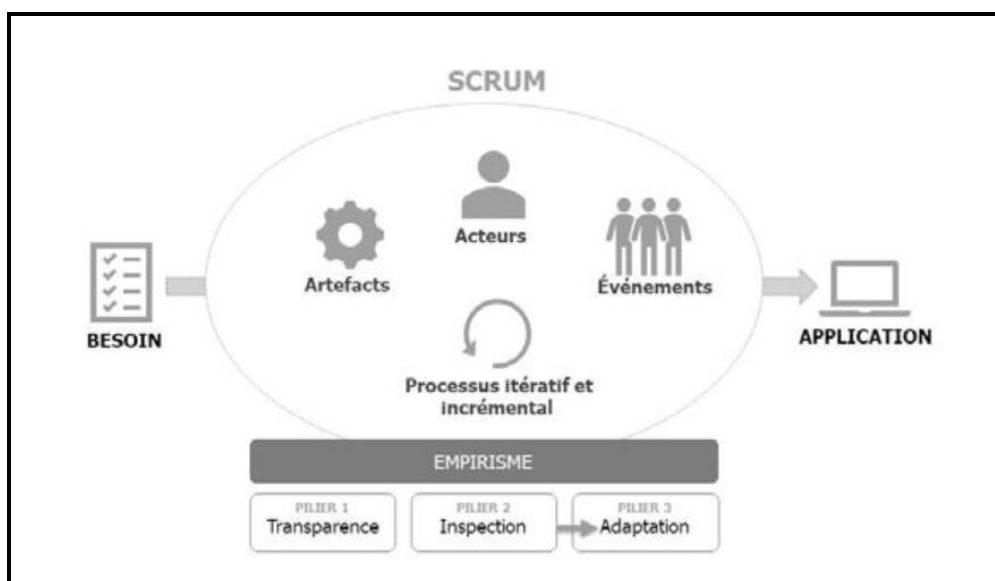
**Un objectif général** : construire des morceaux d'applications de manière progressive plutôt que de livrer une application totalement terminée.

**Un processus global** : reposant sur trois piliers

**Des évènements**

**Des outils** : pour la mise en place de la méthode

**Figure 7** : éléments constitutifs de Scrum



**Source** : (El Haddad & Oger, 2019)

Les grands de la méthode Scrum sont définis dans un document officiel d'une quinzaine de pages qui est disponible sur ce lien :

<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-French.pdf>

Les trois grands pilier dans la méthode Scrum sont les suivants :

### 1. La transparence

Tous les acteurs du projets ont accès à tous les documents du projet en cours. L'information doit être partagée entre les différents membres. Cela nécessite une grande confiance entre les différents acteurs de projet et de partager une vision commune.

### 2. L'inspection

L'état d'avancement du projet doit être régulièrement contrôlé. Nous devons détecter rapidement les écarts entre les prévisions et ce qui est effectivement réalisé.

### 3. L'adaptation

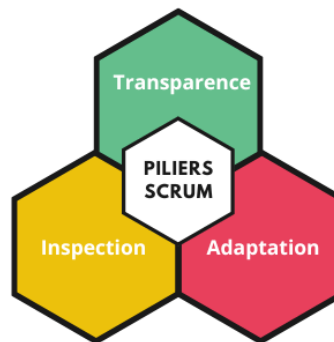
Si un écart est détecté lors de l'inspection, il est nécessaire de prendre des mesures pour éviter que nouveaux problèmes apparaissent. L'adaptation est essentielle pour maintenir l'avancement du projet.

Figure 8 : Trois piliers de Scrum



## PILIER SCUM

- 01 Nécessite une relation de confiance et une vision commune
- 02 Se fait durant tout le cycle Scrum pour détecter les écarts : Objectifs/réalité
- 03 Exige de s'adapter en fonction des résultats d'une inspection régulière



Source : (blog-gestion-de-projet.com)

## 6. Les différents rôles dans un projet Scrum

### 6.1 Scrum Master

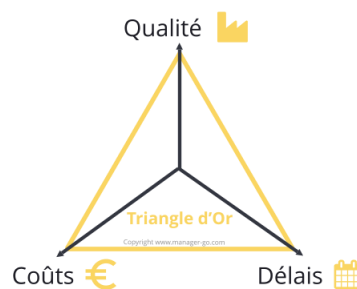
Le Scrum Master a un rôle central dans l'organisation d'un projet Scrum. Il est au même niveau hiérarchique que l'équipe de développement. Il veille à la bonne application de Scrum dans un projet. Il accompagne l'équipe de développement dans l'organisation des différentes tâches du projet. C'est lui qui contrôle la bonne marche du projet et s'assure que tous les parties de l'application à livrer soient correctement développées. Il gère les conflits qui pourraient survenir lors du projet. C'est lui qui s'occupe de l'échange d'informations entre l'équipe de projet et les autres acteurs du projet.

Le Scrum Master dispose de solides connaissances des méthodes agiles et de Scrum. Il est en général lié à l'équipe de développement du projet. Il aide l'équipe de développement à avancer dans le projet. Il motive les différentes personnes du projet afin qu'elles donnent le meilleur d'elles-mêmes et qu'elles se surpassent. Il veille à ce que l'équipe de développement s'améliore et progresse régulièrement. Le Scrum Master ne doit pas être assimilé à un chef de projet classique. Le rôle de Scrum Master peut être joué par différentes membres de l'équipe de développement à condition qu'ils disposent de grandes qualités relationnelles et de très bonnes connaissances de Scrum. Le rôle du Scrum Master est essentiellement opérationnel. Les membres de l'équipe de développement se tourne vers lui en cas de problèmes. Nous pouvons considérer le rôle de Scrum Master comme un grand frère qui va guider l'équipe durant toutes les étapes du projet.

### 6.2 Product Owner

Le Product Owner est responsable du produit à développer. Il doit veiller à ce que les contraintes de qualité, de coûts et délais soient respectées. C'est lui qui gère le carnet produit qui contient toutes les fonctionnalités à développer. Il valide progressivement toutes les fonctionnalités développées. C'est lui qui a le rôle de chef de projet dans la méthode Scrum. Le carnet produit est mis à jour par le Product Owner. Tous les membres du projet peuvent consulter à tout moment le carnet produit. C'est lui qui définit les besoins qui seront traduit en fonctionnalités de l'application par l'équipe de développement. Il se situe entre les utilisateurs finaux d'une application et les développeurs.

**Figure 9** : Triangle d'or d'un projet



**Source** : (Granger, 2021)

## 6.3 L'équipe de développement

L'équipe de développement se charge de traduire les besoins transmis par le Product Owner en fonctionnalités pour l'application. L'équipe de développement s'organise par elle-même pour élaborer les différentes fonctionnalités de l'application. Elle s'appuie sur le carnet produit qui est géré par le Product Owner. Elle estime par elle-même le temps qu'il sera nécessaire pour produire les différentes fonctionnalités. Elle doit garantir que le niveau de qualité de l'application soit le plus élevé possible et qu'elle puisse évoluer dans le futur. La méthode Scrum recommande la mise en place d'une équipe de développement pluridisciplinaire pour résoudre tous les aspects techniques liés à la conception de l'application. Le nombre de personnes qui composent l'équipe de développement doit être compris entre 3 et 9 personnes au maximum. Plus l'équipe grandit et plus les échanges deviennent compliqué entre les membres. L'équipe de développement doit être flexible et réactive. Elle doit respecter les délais imposés et livrer régulièrement des morceaux de l'application finale. Le Scrum Master et le Product Owner sont compris dans l'équipe de développement mais ils ont des tâches très différentes de celles de l'équipe de développement. Ils ont la possibilité d'aider l'équipe de développement à réaliser les fonctionnalités de l'application.

L'équipe de développement participe activement aux prises de décisions sur le projet. Ils fixent l'ordre dans lequel les différentes fonctionnalités vont être développées. C'est l'équipe de développement qui opèrent les choix techniques pour l'application à élaborer.

## 6.4 Autres acteurs

Les commanditaires ou les clients sont l'origine d'un projet. C'est eux qui s'assurent que le projet débouche sur la mise en place d'une application qui répond aux besoins d'utilisateurs et qui est potentiellement rentable. Ce sont eux qui évaluent l'utilisation finale de l'application qui est développée. Ils s'occupent de la mise sur le marché de l'application. Ils évaluent les besoins des utilisateurs finaux et les transmettent au Product Owner pour réaliser le carnet produit.

Les utilisateurs finaux doivent être intégrés dès le démarrage du projet afin de pouvoir effectuer les ajustements nécessaires tout au long du projet. Plus le public-cible d'une application est large et plus il sera difficile d'évaluer les retours des utilisateurs et de comprendre les besoins de chaque catégorie d'utilisateurs. L'équipe de développement se sert des avis des utilisateurs tests pour procéder à des ajustements dans les fonctionnalités afin d'améliorer l'expérience utilisateur.

## 7. Organisation générale d'un projet avec Scrum

Un projet réalisé avec la méthode Scrum est découpé en 4 grandes étapes. La phase d'initialisation consiste à mettre en place les bases de l'organisation du projet. La phase de préparation sert à préciser les besoins fonctionnels afin de préparer la réalisation des différentes fonctionnalités. Dans la phase de production, les différentes fonctionnalités de l'application sont réalisées et testées afin de vérifier qu'elles répondent bien aux besoins des clients. La dernière étape du projet consiste à livrer la version finale de l'application avec toutes les fonctionnalités.

Sur l'illustration ci-dessous, nous retrouvons l'ensemble des étapes pour aboutir à la réalisation de la version finale de l'application avec toutes les fonctionnalités :

**Figure 10** : Vue d'ensemble d'un projet réalisé avec la méthode Scrum



**Source** : (PomeloAdmin, 2021)

## 7.1 Phase d'initialisation du projet

Avant de commencer le projet, l'équipe de développement doit déjà être formée et le Product Owner et le Scrum Master ont déjà été désignés. Dans la phase initialisation du projet, tous les acteurs du projet doivent connaître leurs rôles et leurs responsabilités dans le projet.

La réunion de lancement ou le kick-off meeting sert à présenter le projet à toute l'équipe de développement ainsi qu'au Scrum Master le Product Owner. La présentation du projet est réalisée par les commanditaires du projet ou l'entreprise qui a mandaté l'équipe de projet. Le Scrum Master explique le rôle de chaque participant du projet et décrit l'organisation générale du projet.

Après la réunion de lancement, un atelier découverte peut être organisé pour mieux comprendre les attentes sur le logiciel à développer et détecter la plupart des risques liés au projet. L'atelier d'immersion permet à l'équipe de développement de mieux se familiariser avec le métier en rapport avec le projet.

Le Scrum Master commence par définir les règles d'organisation du projet. Pour effectuer cette tâche, il s'appuie sur la matrice des responsabilités ou la matrice RACI. Cette matrice se présente sous la forme d'un tableau dans lequel nous retrouvons le nom de tous les participants aux projets, les tâches à effectuer et les rôles de chaque personne.

Il existe 4 rôles possibles dans la matrice RACI :

- R : pour responsable
- A : pour approbateur
- C : pour consulté
- I : pour informé

Chaque personne peut avoir un ou plusieurs rôles dans la matrice RACI.

**Figure 11** : Exemple de matrice RACI



## MATRICE RACI

Résultat de l'exemple

Tâche	Nom de l'activité	Laure	Jean	Amadou	Sophie	Rachid
1	Définir	R	A	I	C	C
2	Concevoir	R	A	C	C	C
3	Développer	I	A	R	I	I
4	Tester	I	A	R	I	I

**Source** : (blog-gestion-de-projet.com)

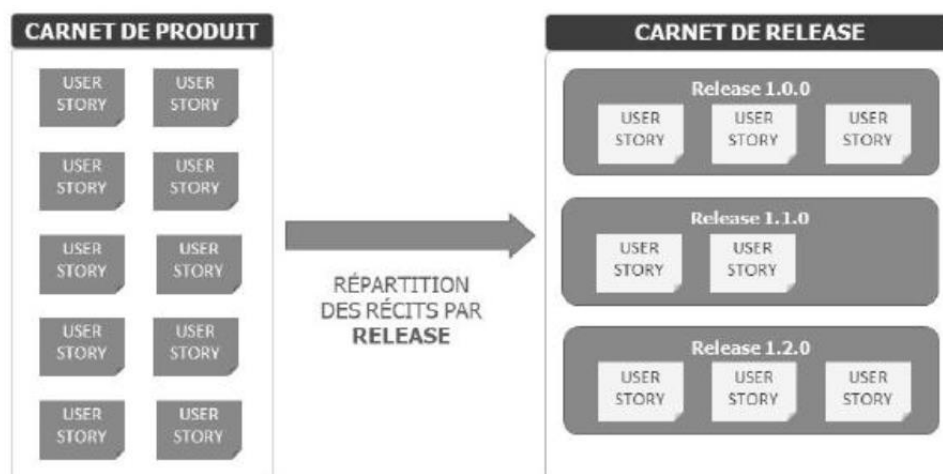
Des ateliers utilisateurs ou focus group regroupent des utilisateur finaux de l'application à développer afin de mieux comprendre leurs besoins. Dès que cette étape est achevée, il est nécessaire d'analyser les besoins des utilisateurs afin de créer le carnet de produit. Pour créer le carnet de produit nous devons passer par 5 grandes étapes :

1. Définir les personas ou les types d'utilisateurs
2. Définir les épics ou les grandes tâches du projet
3. Créer les user stories
4. Estimer les efforts nécessaires à fournir pour réaliser l'application
5. Prioriser les tâches à effectuer en fonction des contraintes techniques et des besoins

Le Product Owner doit maintenir à jour le carnet produit tout au long du projet.

Dès que le carnet produit est réalisé, l'équipe de développement avec l'aide du Product Owner et du Scrum Master établissent le carnet de release. Les User Stories du carnet de produit sont rassemblées sous forme de morceaux d'applications. L'application finale est décompensée en différents morceaux.

**Figure 12** : Création du carnet de release à partir du carnet de produit



**Source** : (El Haddad & Oger, 2019)

Dès que le carnet de release est opérationnel, l'équipe de développement se charge d'estimer le temps nécessaire pour réaliser chaque release de l'application. L'estimation du temps va dépendre de la complexité technique du release, des compétences techniques de chaque membres de l'équipe et de l'étendu des tâches à réaliser.

A ce stade du projet, l'équipe de développement dispose de toutes les informations nécessaires pour mettre en place le socle technique et sa configuration.

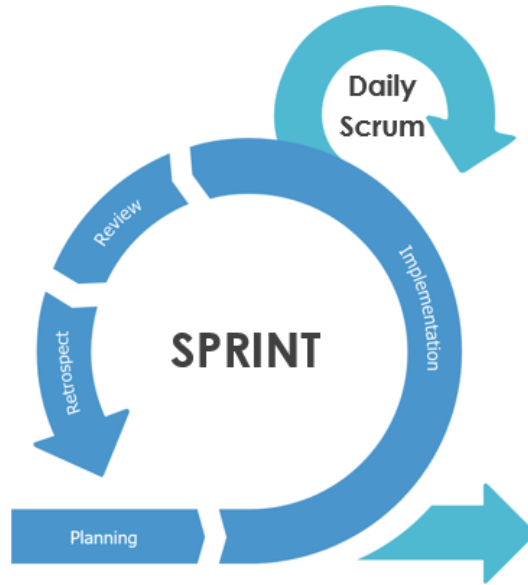
## 7.2 Phase de préparation du projet

Dans la méthode Scrum, le Sprint est une période limitée dans le temps durant laquelle l'équipe de développement crée un morceau d'application ou release et la présente au Product Owner et au commanditaire du projet. Si le morceau d'application est validé, l'équipe effectue une rétrospective pour analyse tous les éléments qui se sont bien passés et ce qu'il faut améliorer. Dès que cette analyse est effectuée, l'équipe de développement passe à la suite.

Les différents Sprint du projet vont permettre de réaliser progressivement toutes les morceaux de l'application. A la fin du dernier Sprint, l'application contient toutes les fonctionnalités et est prête à être mise sur le marché.

Pour chaque release, nous devons encore préparer un plan de release dans lequel nous précisons ce qui va être développé pour chaque sprint. Dans le plan de release, nous retrouvons les objectifs de production, le contenu de chaque sprint et un aperçu de ce que contient chaque release.

**Figure 13 : Exemple de Sprint**



**Source :** (Visual Paradigm)

Pour chaque Sprint, l'équipe de développement prépare un carnet dans lequel elle liste l'ensemble des User Stories à réaliser avant de livrer le morceau de l'application au commanditaire du projet et au Product Owner. Une User Stories va être décomposées en tâches à réaliser.

Voici un exemple de carnet de sprint avec l'ensembles des tâches liées aux User Sories :

**Tableau 3 : Exemple de carnet de sprint**

Élément de Backlog	Story Points	Responsable	Statut	Estimation Originale	Jour 1	Jour 2	Jour 3	Jour 4	Jour 5	Revue de la Sprint
<b>Scénario utilisateur #1</b>	<b>8</b>									
Tâche				7	5	3	0	0	0	0
Tâche				3	1	1	5	0	1	0
Tâche				1	0.5	0	3	0	0	0
Tâche				0.5	1	2	3	1	0	0
<b>Scénario utilisateur #2</b>	<b>1</b>									
Tâche				3	3	0.5	0.5	0	0	2
Tâche				3	5	5	1	1	1	0
Tâche				2	2	5	0	1	0	1
Tâche				5	5	9	5	1	0	1
<b>Scénario utilisateur #3</b>	<b>5</b>									
Tâche				8	6	0	0	0	0	0
Tâche				3	1	3	3	3	0	0
Tâche				1.5	1	0.5	0.5	1	1	0
Tâche				2	0.5	0	0	0	0	3
<b>Scénario utilisateur #4</b>	<b>8</b>									
Tâche				9	4	2	2	1	1	0
Tâche				6	6	3	3	3	1	1
Tâche				6	2	8	8	1	0	1
Tâche				0.5	0.5	0.5	0.5	0	0	0
<b>Scénario utilisateur #5</b>	<b>3</b>									
Tâche				2	1	1	1	0.5	1	1
Tâche				6	6	6	0.5	3	9	0
Tâche				9	9	9	4	3	3	3
Tâche				0.5	0.5	0.5	1	0.5	0	1
Total				78	60	59	41	20	18	14

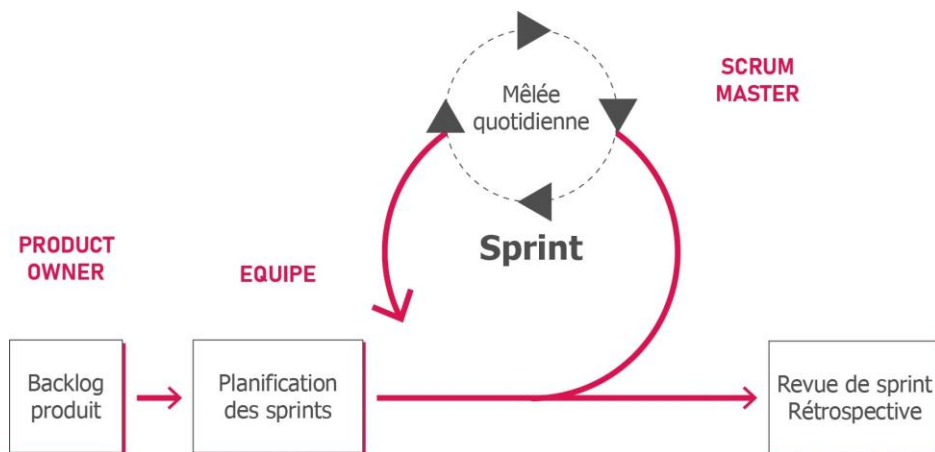
**Source :** (smartsheet.com)



### 7.3 Phase de réalisation du projet

Dès que le carnet de Sprint est prêt, l'équipe de développement passe à la phase concrète de réalisation du projet. Le premier sprint du projet commence et l'équipe de développement se sert du carnet de sprint et du plan de release pour développer le premier release de l'application. Chaque release correspond à un sprint. L'équipe de développement s'auto-gère. C'est elle qui gère la production des différents morceaux de l'application. Elle met à jour avec le Scrum Master le carnet de Sprint afin de connaître l'état d'avancement du projet et des tâches en cours.

Figure 14 : Le sprint dans un projet Scrum



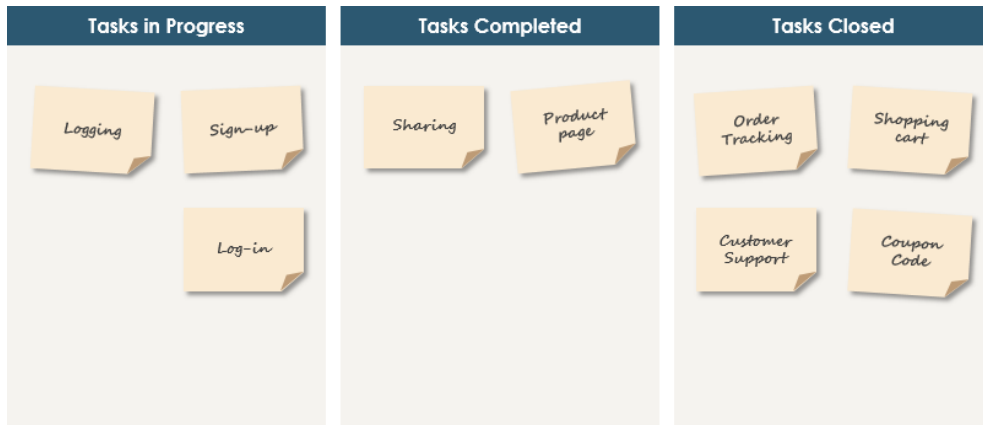
Source : (Acca, 2021)

Chaque jour, le Scrum Master organise une réunion (dénommée mêlée dans la méthode Scrum) d'une quinzaine de minutes avec l'équipe de développement afin de faire le point sur ce qui a été réalisé et sur ce qui sera réalisé dans la journée.

Durant la réunion, les participants tentent de trouver une solution aux problèmes rencontrés. A la fin de la réunion, le carnet de sprint est mis à jour. Le Product Owner se charge de mettre à jour le carnet de produit et le plan de release.

Afin d'améliorer l'efficacité des réunions quotidiennes ou mêlées, nous pouvons utiliser le Scrum Board pour représenter les tâches déjà effectuées, montrer ce qu'il reste à faire, ce qu'il faut tester et les tâches terminées sous forme d'un tableau sur lequel on colle des post-it. Divers logiciels de gestion de projets avec Scrum permettent de créer un Scrum Board. Nous pouvons aussi prendre un tableau blanc et des post-it pour représenter les tâches en cours, celles qu'il reste à finir et celles à réaliser.

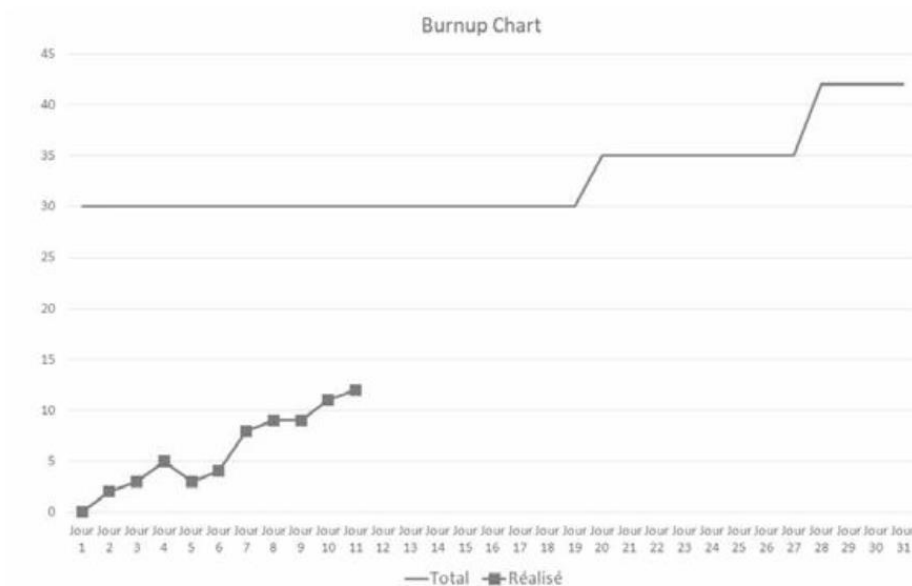
**Figure 15 :** Exemple de Board Chart



**Source :** (Visual Paradigm)

Pour suivre l'avancée des tâches lors d'un sprint, il existe deux graphiques que les équipes de développement peuvent utiliser. Le premier outil est le Burn-up chart. Il montre sous forme de deux graphiques l'avancement des tâches et le périmètre du morceau d'application qui est en cours de réalisation.

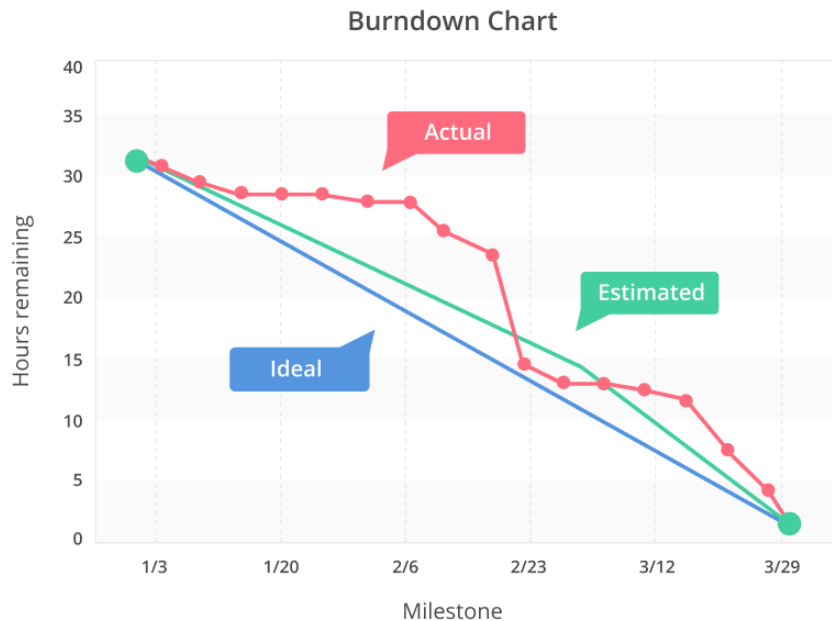
**Figure 16 :** Exemple de Burn-up chart



**Source :** (El Haddad & Oger, 2019)

Le deuxième outil est le Burndown chart. Il permet d'indiquer de montrer l'ensemble des tâches à faire pour finir le morceau d'application en cours. Nous pouvons comparer l'état d'avancement des tâches effectif avec celui qui avait été planifié initialement.

**Figure 17 :** Exemple de Burndown Chart



**Source :** (Guthrie, 2021)

Dès que l'équipe de développement a terminé les différentes tâches du release, elle organise une revue de sprint avec le Product Owner et le commanditaire du projet. Lors de la revue de sprint, l'équipe de développement présente le morceau d'application qu'elle a réalisée. Le Product Owner et le commanditaire du projet contrôlent que le morceau d'application répondent bien aux besoins des utilisateurs. Si des corrections sont à effectuer dans le morceau d'application présenté, elles seront planifiées dans le prochain sprint. Dès que la revue de sprint est terminée, le carnet de sprint est mis à jour ainsi que le plan de release.

Si le Product Owner et le commanditaire du projet valident le morceau d'application présenté, il sera mis à disposition d'utilisateurs pilotes.

Après la revue de sprint, une rétrospective de sprint est mise en place entre l'équipe de développement, le Product Owner et le Scrum Master. Durant cette réunion, tous les problèmes organisationnels, relationnels et techniques sont abordés. Le but de la revue de sprint est d'améliorer le fonctionnement globale de l'équipe afin qu'elle puisse être plus efficace lors du prochain sprint.

L'équipe de développement va passer par les différents sprints pour arriver à la fin du projet. Il est possible d'organiser des sprints supplémentaires si certains éléments sont encore à corriger après la dernière revue de sprint.

## 7.4 Phase de finalisation du projet

Dès que l'équipe de développement termine le dernier morceau de code de l'application, elle effectue la revue du dernier sprint et la rétrospective de sprint. La dernière rétrospective de sprint marque la fin du projet. Cette dernière séance permet de faire le bilan du projet. L'application contient maintenant toutes les fonctionnalités demandées pour répondre aux besoins des utilisateurs. Toutes les bugs et les différentes anomalies techniques ont été corrigés. A ce stade, l'application est prête à entrer sur le marché. Le Scrum Master, le Product Owner et l'équipe de développement peuvent organiser un évènement pour célébrer la fin de projet.

## 8. Logiciels de gestion de projets pour Scrum

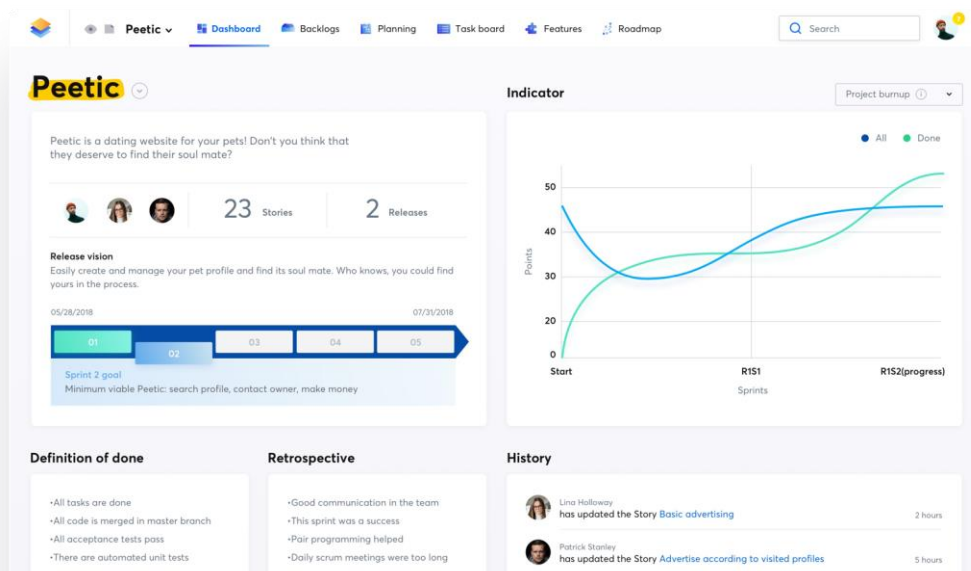
Il existe plusieurs logiciels pour gérer un projet avec la méthode Scrum qui sont proposés par les auteurs de l'ouvrage dans le chapitre 8.

Le premier logiciel s'appelle IceScrum. Il est disponible en allant sur cette page :

<https://www.icescrum.com/fr/>

Voici l'interface de iceScrum :

Figure 18 : Interface de iceScrum



Source : (iceScrum)

Il est disponible en version gratuite avec un nombre de fonctionnalités restreintes pour 3 utilisateurs. Selon les besoins divers abonnements sont proposés allant de 8,9 euros par mois à 349 euros par mois. Il suffit de s'inscrire en ligne pour se connecter au portail en ligne de l'application.

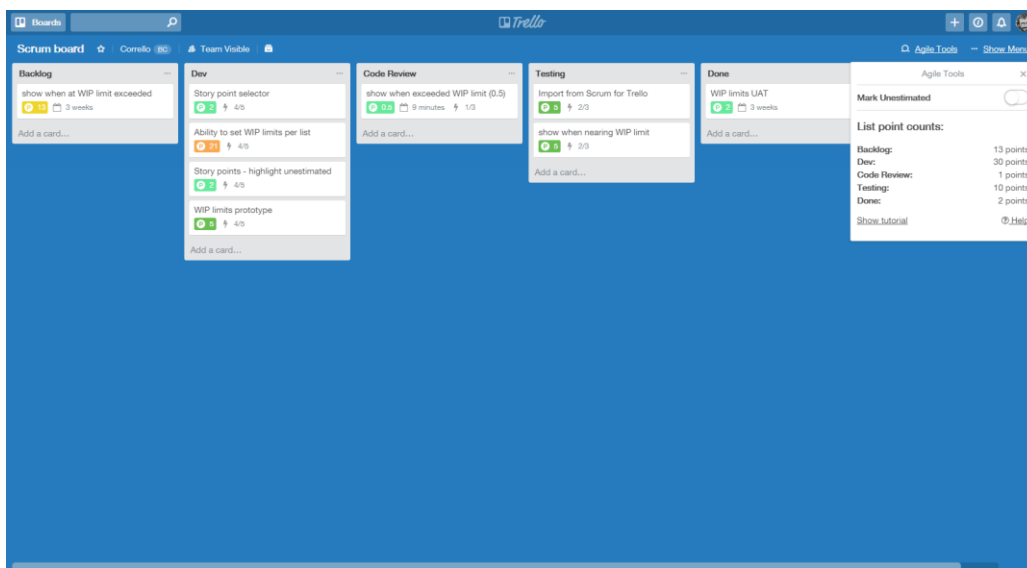
Nous retrouvons toutes les fonctionnalités nécessaires pour gérer un projet avec Scrum comme l'estimation de la durée des tâches, la création des User Stories, le Scrum board et des indicateurs comme le Burdown chart et le Burnup chart.

Trello est un autre outil de gestion de projet qui est très utile pour gérer un projet Scrum. Il est disponible à cette adresse :

<https://trello.com/home>

Il dispose de modules complémentaires pour la gestion de projet agiles. Nous retrouvons un calendrier qui permet d'organiser les différentes tâches du projet, des outils pour estimer le temps de chaque sprint et des indicateurs de suivis comme le Burdown chart.

**Figure 19** : Interface de Trello



Source : (trello.com, 2022)

Trello est accessible via un portail en ligne. Il est disponible en version gratuite mais avec des fonctions limitées. Les abonnements vont de 5 dollars à 17,50 dollars par mois et par utilisateur. Trello est aussi disponible sur iOS et Android.

## 9. Conclusion

Cet ouvrage m'a permis de mieux comprendre les différences qui existent entre les méthodes de gestion de projet traditionnelles et agiles. Je me suis familiariser avec les concepts généraux de l'agilité. Les premiers chapitres du livre m'ont donné les bases théoriques sur Scrum. J'ai découvert comment on passe des besoins des utilisateurs aux différentes fonctionnalités à créer dans une application. Je trouve que la méthode Scrum a l'avantage d'être plus flexible et plus adaptée à la création de nouvelles applications. Toutes les étapes d'un projet réalisé avec Scrum sont détaillées et des indicateurs de suivis sont présentés afin de mieux gérer la phase de production d'une application. Les rôles de chaque membre d'un projet géré avec Scrum est décrit en détail afin de bien comprendre ce que fait chaque personne dans un projet Scrum. Les auteurs donnent plein de conseils pour gérer efficacement un projet avec la méthode Scrum. Le chapitre 7 présente le cas d'un projet de création de plateforme de formations en ligne qui est géré avec la méthode Scrum. Les auteurs montrent tout ce qui pourrait ne pas fonctionner à chaque étape du projet et on arrive à mieux comprendre comment mettre en place la méthode Scrum dans un projet concret. Ce livre me donne toutes les clés nécessaires pour évoluer dans une équipe de projet qui recourt à la méthode Scrum.

## Références

- Acca, F. (2021, Mars 29). *De Natura Agilium*. Récupéré sur actin-co.com: <https://actin-co.com/methode-agile/>
- Amazon. (2023). *Scrum, de la théorie à la pratique*. Récupéré sur amazon.fr: <https://www.amazon.fr/Scrum-th%C3%A9orie-pratique-Julien-Oger/dp/2212678622>
- blog-gestion-de-projet.com. (s.d.). *Piliers Scrum : Comment ils contribuent à l'amélioration continue dans un projet agile*. Récupéré sur blog-gestion-de-projet.com: <https://blog-gestion-de-projet.com/piliers-scrum/>
- blog-gestion-de-projet.com. (s.d.). *RACI : comprendre les 4 rôles de la matrice, avec exemple et Modèle*. Récupéré sur blog-gestion-de-projet.com: <https://blog-gestion-de-projet.com/quest-ce-que-la-matrice-raci/>
- Crochet-Damais, A. (2022, Novembre 16). *Scrum : C'est quoi ? Définition, guide...* Récupéré sur journaldunet.fr: <https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1443834-scrum-maitriser-le-framework-star-des-methodes-agiles/>
- El Haddad, B., & Oger, J. (2019). *Scrum, de la théorie à la pratique*. Paris: Eyrolles.
- Ferlis, S. (s.d.). *Comprendre le but de la Sprint Rétrospective*. Récupéré sur savoiragile.com: <https://savoiragile.com/2021/07/22/comprendre-la-sprint-retrospective/>
- Granger, L. (2021, Novembre 9). *Gestion de projet : qu'est-ce que le Triangle d'or ?* Récupéré sur manager-go.com: <https://www.manager-go.com/gestion-de-projet/glossaire/triangle-d-or>
- Guthrie, G. (2021, Mai 28). *Get started using a burndown chart to track your project*. Récupéré sur nulab.com: <https://nulab.com/learn/project-management/get-started-using-burndown-chart-track-project/>
- henryford.fr. (s.d.). *Qu'est ce que le Taylorisme ? Définition et explications*. Récupéré sur henryford.fr: <https://www.henryford.fr/fordisme/taylorisme/>
- iceScrum. (s.d.). *Tout ce dont vous avez besoin pour gérer vos projets agiles*. Récupéré sur icescrum.com: <https://www.icescrum.com/fr/features/>
- Lardilleux, R. (2021, Février 12). *Cycle en V en gestion de projet : définition et méthode*. Récupéré sur manager-go.com: <https://www.manager-go.com/gestion-de-projet/cycle-en-v.htm>
- mercator.eu. (s.d.). *La méthode Agile comme méthode de travail chez Mercator : explications*. Récupéré sur mercator.eu: <https://www.mercator.eu/fr/la-methode-agile-comme-methode-de-travail-chez-mercator-explications.html>
- Morin, A. (2023, Janvier 9). *Les grandes étapes d'un projet Web* . Récupéré sur adimeo.com: <https://www.adimeo.com/blog/etapes-projet-web>
- PomeloAdmin. (2021, Mars 17). *9 points clés de la méthode agile Scrum*. Récupéré sur hello-pomelo.com: <https://www.hello-pomelo.com/articles/9-points-cles-de-la-methode-agile-scrum>
- Prium-MT. (2021, Octobre 5). *Qu'est-ce que la méthode PDCA ou Roue de Deming ?* Récupéré sur prium-transition.com: <https://prium-transition.com/methode-pdca-roue-de-deming/>
- Rey, J.-P. (2022). *Référentiel de compétences du programme « Digital Team Academy » de la Filière*

*Informatique de Gestion de Sierre. Sierre.*

smartsheet.com. (s.d.). *Les meilleurs modèles de gestion de projet Agile pour Excel*. Récupéré sur fr.smartsheet.com: <https://fr.smartsheet.com/agile-project-management-excel-templates>

Tech agilist. (s.d.). *Burnup Chart – Sprint & Release Tracking*. Récupéré sur techagilist.com: <https://www.techagilist.com/agile/scrum/burnup-chart-use-in-sprint-release-tracking/>

trello.com. (2022). *Agile Tools by Corrello*. Récupéré sur trello.com: <https://trello.com/power-ups/59d4ef8cfea15a55b0086614/agile-tools-by-corrello>

tuleap.org. (s.d.). *Comprendre la méthode Agile Scrum en 10 min*. Récupéré sur tuleap.org: <https://www.tuleap.org/fr/agile/comprendre-methode-agile-scrum-10-minutes>

Visual Paradigm. (s.d.). *How to Use Scrum Board for Agile Development?* Récupéré sur visual-paradigm.com: <https://www.visual-paradigm.com/scrum/how-to-use-scrum-board-for-agile-development/>

Visual Paradigm. (s.d.). *What is a Sprint in Scrum?* Récupéré sur visual-paradigm.com: <https://www.visual-paradigm.com/scrum/what-is-sprint-in-scrum/>