



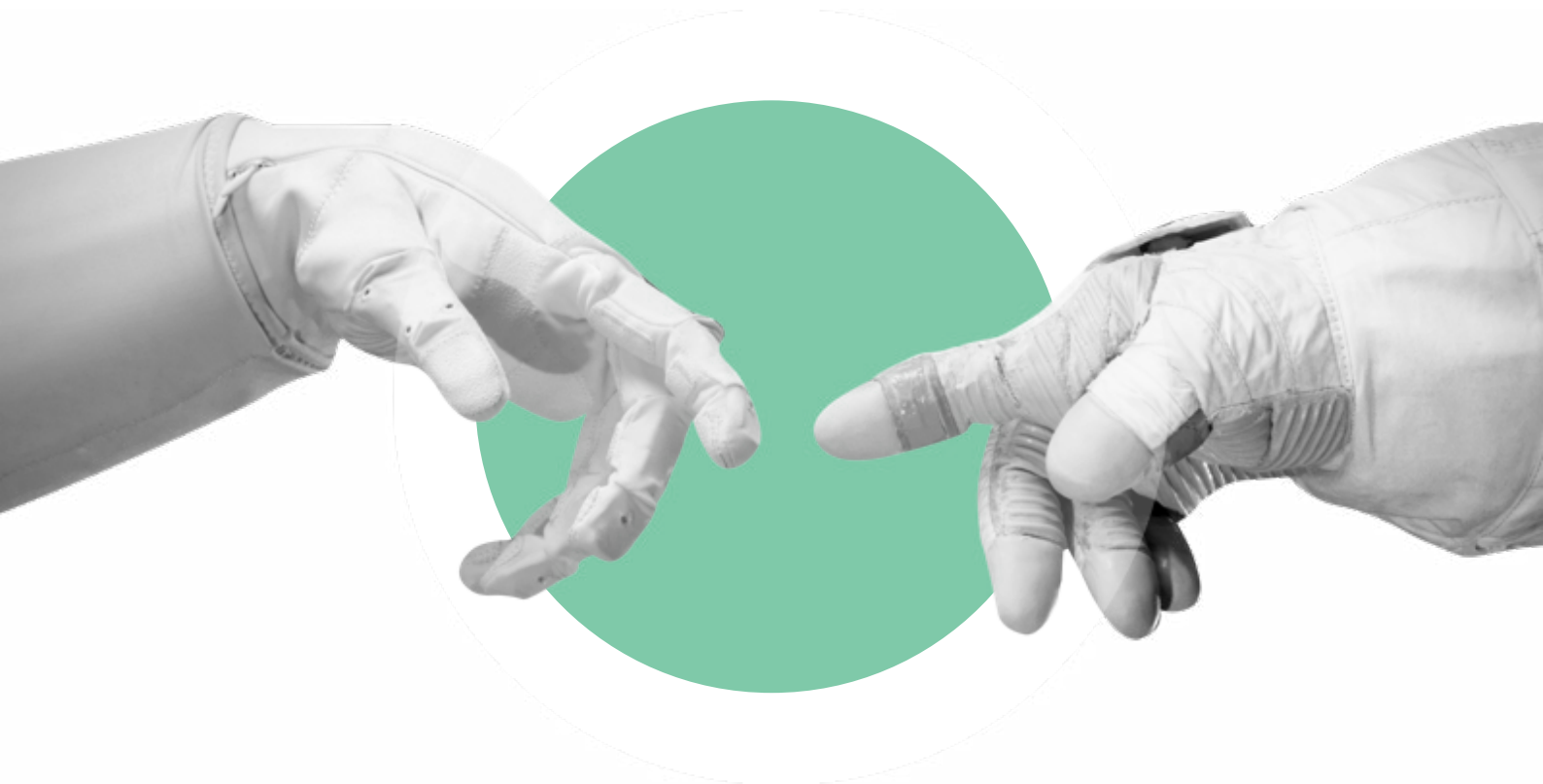
Eleven Labs



Glossaire Agile

Table des matières

A		O	
■ Agilité	3	■ OKR	13
B		P	
■ Backlog Refinement (anciennement Grooming)	4	■ Pain Points	14
■ BurnDown Chart	4	■ Persona	14
■ BurnUp Chart	4	■ Pivot	14
D		■ Poker Planning	15
■ Daily meeting		■ Product Backlog	15
■ Stand-up meeting	5	■ Prototype	15
■ Design thinking	5	R	
■ Design sprint	6	■ Refactorisation	16
■ Dette technique	6	■ Release/Version	16
E		■ Roadmap	16
■ EPIC	7	S	
F		■ SCRUM	17
■ Feature	8	■ SCRUMBAN	18
■ Feature Team	8	■ Sprint	18
I		■ Sprint Backlog	18
■ Impact Mapping	9	■ Sprint Planning	19
K		■ Sprint Review	19
■ Kanban	10	■ Story Map / Story Mapping	19
L		■ Story Points	19
■ Lean Start-Up	11	U	
M		■ User Journey Map (ou Customer Journey Map)	20
■ MEP	12	■ User Story	20
■ MoSCoW	12	■ User Test	21
■ MVP	12	V	
■ (Minimum Viable Product)		■ Vitesse	22
		W	
		■ Wireframe	23



A

Agilité

L'agilité caractérise une posture dans les projets de développement privilégiant le dialogue entre toutes les parties prenantes : clients, utilisateurs, développeurs et tout autre interlocuteur.

Dans les cadres et processus agiles, on privilégie :

- la souplesse en cours de projet, la capacité à modifier et s'adapter rapidement par exemple aux changements de priorités, aux retours des clients et des parties prenantes
- la rapidité de livraison et le développement évolutif : on privilégie le développement d'un MVP (Minimum Viable Product) pour avoir de premiers retours utilisateurs et faire évoluer le produit en fonction

Les cadres agiles permettent de rompre avec les pratiques plus traditionnelles de gestion de projet qui se basent sur un cahier des charges précis et non évolutif, pour passer vers de la création de produit en l'améliorant continuellement par de petites itérations.

Backlog Refinement (anciennement Grooming)

Cette pratique agile prend la forme d'une réunion pendant laquelle le Product Owner et la Dev Team procèdent ensemble à l'affinage des user stories présentes dans le backlog :

- Détectez les user stories qui n'ont plus de sens pour le projet ;
- Réévaluez la priorité des user stories ;
- Précisez éventuellement les critères d'acceptance techniques d'une story ;
- Corrigez les estimations en fonction de nouvelles informations ;
- Formulez et estimez de nouvelles user stories ;
- Découpez les user stories qui pourraient candidater aux prochains sprints.

Quand elles ne sont pas séparées en 2 cérémonies distinctes afin de ne pas dépasser 10% du temps de l'équipe, il arrive aussi que le Poker Planning se fasse à l'occasion du Backlog Refinement.

BurnDown Chart

Graphique utilisé pour suivre l'avancement du Sprint (Horizon court terme). Il permet de suivre l'évolution du travail restant. Le but consiste donc à atteindre le niveau zéro le plus tôt possible, d'où le terme « Down ». Il fournit une visualisation rapide et simple qui permet de monitorer l'avancée du sprint, de pouvoir anticiper les retards dans le sprint et re-prioriser éventuellement les US restantes, et le cas échéant d'informer les parties prenantes. Le BurnDown Chart permet également d'adapter les sprints suivants.

BurnUp Chart

Graphique d'avancement utilisé pour suivre l'avancement d'une Release (horizon moyen ou long terme). Il permet de suivre l'évolution de la quantité de travail terminée en fonction du temps. Le BurnUp Chart présente deux courbes, l'une représentant le périmètre du produit ou d'une release du produit. Cette ligne croît lorsque l'on ajoute des éléments dans le backlog, l'autre représentant le travail déjà réalisé. Lorsque les deux lignes se rejoindront, le projet ou le sprint seront terminés.

Daily meeting / Stand-up meeting

Réunion quotidienne permettant à la Dev Team (dev front, dev back, QA, Data, UI/UX, etc.) de se synchroniser, d'identifier les obstacles éventuels et de mesurer son avancement sur le Sprint en cours. Le PO et le Scrum Master peuvent y assister pour répondre aux questions de la Dev Team mais ne participent pas directement.

Design thinking

Méthode de gestion de l'innovation élaborée dans les années 1980 par Rolf Faste et popularisée par Stanford et IDEO, le Design Thinking est une approche découpée en plusieurs étapes claires avec pour objectif commun de définir, imaginer, synthétiser, prototyper et tester. Certaines méthodologies intègrent également la recherche et l'implémentation.



D

Design sprint

Le Design Sprint est une méthode inspirée du Design Thinking. C'est un processus qui dure généralement 5 jours (mais peut être fait en moins de temps en fonction du but recherché), et dont l'objectif est de répondre à une problématique de façon concrète en produisant des prototypes et en les testant auprès d'utilisateurs.

Afin de réussir un Design Sprint et parvenir à répondre rapidement aux besoins des utilisateurs au moindre coût, une petite équipe (jusqu'à 7 membres) pluri-disciplinaire du business, du développement et du design avec un décideur passe d'une problématique de base à un test utilisateur d'un système prototypé.

Dettes techniques

Comme une La dette technique décrit un retard technique accumulé à cause d'un code ou d'une documentation de mauvaise qualité, parfois pour gagner du temps dans le développement de nouvelles fonctionnalités à un instant T. La dette technique implique l'intervention future des équipes de développement pour "nettoyer" le code, et va donc nécessiter du temps là où l'équipe a pu en faire gagner pour livrer une fonctionnalité plus rapidement. La montée de version des bibliothèques utilisées ou la montée de version du langage utilisé dans le développement du produit peut également entraîner la nécessité d'une "refactorisation" du code (nom donné à l'opération de retravailler le code source d'un programme sans en changer le fonctionnement).

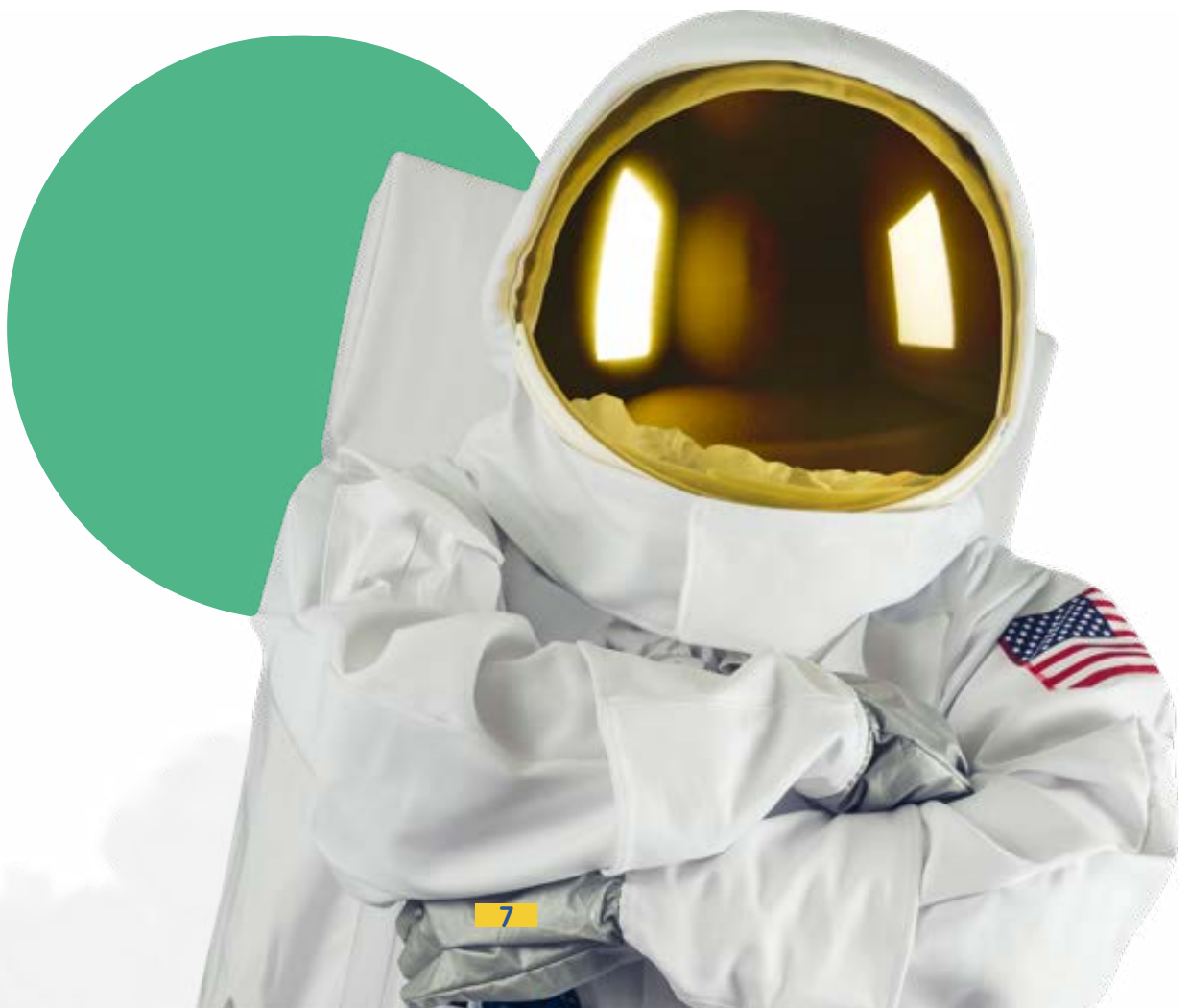


E

Epic

Selon Mike Cohn, les Epics sont vulgairement des « Big user-stories ». Autrement dit, il s'agit d'une fonctionnalité macro qui n'est pas détaillée, qui est souvent posée dès le début du projet ou par la suite à l'apparition de nouveaux besoins. Une EPIC est de ce fait détaillée par la suite en plusieurs User Stories afin de pouvoir alimenter le Product Backlog.

Plusieurs sprints pourront être nécessaires pour compléter un même EPIC.



F

Feature

Les features, à l'image des "Feature Teams" d'un produit, sont des caractéristiques ou des attributs d'un produit qui créent de la valeur d'un point de vue consommateur. Selon la définition, une feature peut représenter soit une grosse partie du produit, et se découper en plusieurs EPICS, qui se découpent eux-mêmes en un ensemble de multiples users stories.

Feature Team

Une Feature Team est une équipe autonome, pluridisciplinaire et à long terme.

Autonome car elle n'a pas de dépendance fonctionnelle avec les autres équipes, mais peut avoir des dépendances techniques.

Pluridisciplinaire, car elle est composée de toutes les compétences nécessaires pour créer une partie d'un produit (de manière non exhaustive : dev front, dev back, dev API, designer UX/UI, data scientist, product manager, coach, business analyst...), à contrario des équipes "composants" qui regroupent des compétences spécifiques par équipe.

Long terme, car l'équipe reste le plus possible stable, et peut durer pendant des années.

Les feature teams travaillent ensemble sur une macro-fonctionnalité d'un produit, et non plus sur le produit dans sa globalité, et donc doivent se mettre ensemble d'accord sur les différentes responsabilités fonctionnelles.

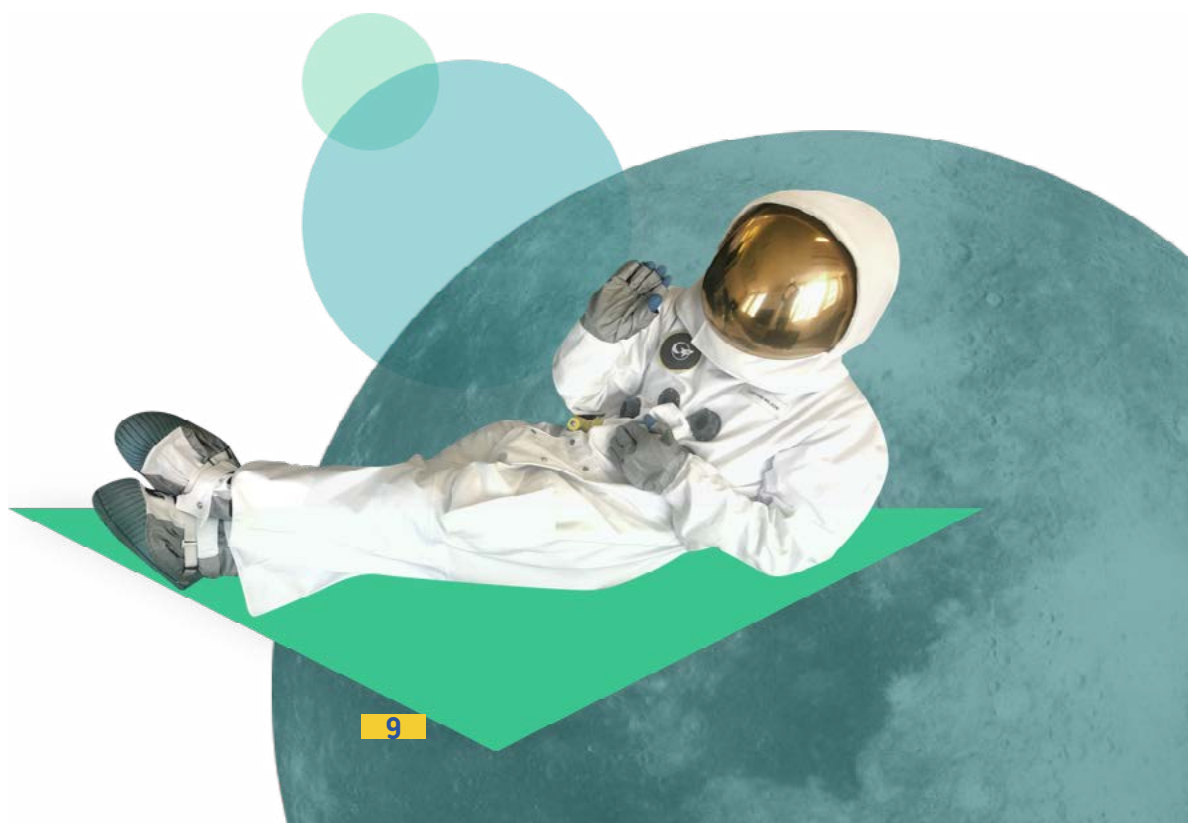
Impact Mapping

L'impact mapping est une technique permettant la visualisation des hypothèses les plus probables et des plus importantes vis-à-vis du produit et du périmètre à livrer. Elle permet de ce fait aux équipes d'adopter un plan d'action efficace et de pouvoir prévoir via cette feuille de route les problématiques et changements importants que le produit va peut-être rencontrer.

La création d'une impact map consiste à agencer graphiquement les éléments suivants :

- Les objectifs : Pourquoi souhaitez-vous créer ou faire évoluer votre produit ?
- Les acteurs : De qui avez-vous besoin pour atteindre vos objectifs ? Quelle est votre audience ?
- Les impacts : Comment souhaitez-vous modifier le comportement des acteurs afin d'atteindre votre objectif ?
- Les fonctionnalités : Quelles fonctionnalités ou actions devez-vous mettre en place pour obtenir les comportements identifiés ?

Pour en savoir plus, un site est dédié à l'approche de l'Impact Mapping, réunissant de nombreux articles et vidéos de la communauté :
<https://www.impactmapping.org/fr/index.html>



K

Kanban

Kanban est une stratégie permettant d'optimiser le flux de valeur en visualisant un processus de travail en flux-tiré.

Elle contient trois pratiques principales :

- Définir et visualiser un processus de travail,
- Limiter et activement gérer le travail en cours dans le processus,
- Améliorer le processus collectivement.

D'autres pratiques annexes existent, comme la création de cadences (réunions cycliques) ou de l'affichage de normes de travail en commun.

On appelle aussi « kanban » (avec un 'k' minuscule) ou encore parfois « Système kanban », le tableau visuel utilisé pour suivre visuellement l'avancement des tâches et la progression du Sprint.

Le point central de Kanban est la notion de Flux, qui correspond au mouvement de valeur au sein d'un système. Si les processus existent pour augmenter la valeur, Kanban existe pour optimiser ces processus en améliorant le flux.

Lean Start-Up

Le Lean Start-up est une approche spécifique au lancement d'un produit ou d'une activité. Elle est basée sur trois étapes : BUILD, pour construire un prototype par une hypothèse, MEASURE en mesurant la validité de l'hypothèse, LEARN en apprenant de nos mesures pour améliorer le produit. Puis l'on recommence le cycle.

Des pratiques viennent compléter les étapes, comme la vérification de la validité des concepts, l'expérimentation scientifique et le design itératif.

Sur des fondements similaires à l'agilité, le lean start-up cherche à mesurer régulièrement les impacts des réalisations et à obtenir des retours utilisateurs afin de répondre au mieux à la demande des consommateurs avec un investissement optimal.



MEP

Une mise en production concerne le déploiement d'un ensemble de features développées pendant un ou plusieurs sprints (souvent appelé «release»), finalisées et validées, d'un environnement de recette vers l'environnement de production (celui accessible par tous les utilisateurs).

MoSCoW

Le méthode de priorisation MoSCoW permet de prioriser les user stories à moyen terme selon les critères suivants :

- M – Must have : doit être réalisée.
- S – Should have : devrait être réalisée si possible.
- C – Could have : pourrait être réalisée s'il n'y a pas d'impact sur les autres tâches en cours.
- W – Won't have : ne sera pas réalisée tout de suite mais serait souhaitable pour une version ultérieure.

MVP (Minimum Viable Product)

Le MVP est la version la plus épurée d'un produit. Il est constitué uniquement des fonctionnalités les plus prioritaires et il peut être lancé sous 3 conditions :

- Il a suffisamment de valeur d'un point de vue utilisateur pour que ceux-ci l'utilisent ou l'achètent ;
- Il procure suffisamment de valeur aux early adopters pour les fidéliser ;
- Il inclut une feedback loop, permettant de valider des idées ou de pivoter.

Le concept de MVP a été introduit par Eric Ries et fait partie de la méthodologie Lean Start Up.

Pour aller plus loin : [Vous connaissez le Minimum Viable Product \(MVP\) ? Vous allez adorer le Minimum Loveable Product](#)

OKR

Le but de l'OKR, Objective Key Results, est d'aligner l'ensemble des troupes vers les mêmes objectifs. De façon très simplifiée, on définit :

- Où on veut aller avec des objectifs ambitieux et inspirationnels comme aller sur la lune
- Comment on va y aller avec des key results qui se matérialisent par des indicateurs chiffrés et bornés dans le temps permettant d'apprécier la progression de l'équipe vers l'objectif, par exemple "avoir construit 3 réacteurs de la fusée"

Les meilleurs OKR sont orientés comportement utilisateur, comme par exemple : **Objectif : Augmenter l'engagement des utilisateurs pendant leur essai du produit**

Key Results :

- **80% des utilisateurs en essai complètent entièrement leur profil**
- **50% invitent un proche sous 7 jours**



Pain Points

Les pain points ou points de douleurs sont les obstacles auxquels est confronté l'utilisateur. Quand vous envisagez la création d'un nouveau produit ou service, il faut vous projeter dans un parcours utilisateur idéal. Celui-ci doit générer principalement une expérience positive en éliminant tous les points de friction, frustrations ou les difficultés ("pain points") de l'utilisateur face à l'utilisation du produit. Un exemple, faire la queue au cinéma est un élément irritant majeur.

Persona

Un Persona est une personne fictive permettant de représenter un groupe de clients ou d'utilisateurs. Le persona est généralement doté d'un prénom et de caractéristiques sociales et psychologiques types. Plusieurs personae peuvent être utilisés pour un même projet de développement.

Pivot

On dit qu'une start-up pivote quand elle décide de changer drastiquement de stratégie dans le but de d'adapter ses produits à la demande du marché. Il intervient après que l'entreprise se rende compte que son offre/produit ne rencontre pas le succès escompté et ne convient pas aux attentes du marché. Ce changement doit être rapide et impactant.

Quelques exemples de pivots pour mieux saisir le concept :

- Faire d'une feature d'un produit, le produit principal et simplifier l'offre
- Changement de plateforme, d'un logiciel vers une application, ou l'inverse
- Changer de modèle de monétisation, du freemium à des revenus grâce à la publicité

P

Poker Planning

Le Poker Planning est une technique permettant d'estimer collectivement l'effort ou la taille relative des user stories. Les participants à un poker planning seront amenés à estimer une User Story en la notant soit en taille (S/M/L/XL) ou plus fréquemment en utilisant des « Story points ». On peut pour cela utiliser des cartes qui présentent la suite de Fibonacci, où chaque nombre est la somme des deux précédents (1, 2, 3, 5, 8, 13, 20, 40, 100). Chaque « joueur » pose sa carte en même temps que les autres. Ainsi, on n'influence pas son voisin, en donnant son estimation après avoir entendu celle d'un autre membre de l'équipe projet.

L'estimation faite dans le Poker Planning aide le Product Owner à prioriser son Product Backlog et l'équipe à savoir quelle est sa vitesse de croisière en termes de capacité de shipping (sorties) de stories par sprint. Ce qui permet aussi de prévoir et d'adapter les «unusual sprints» : congés, jours fériés,... et pour aller plus loin, l'incontournable article [Joker Planning](#).

Product Backlog

Le Product Backlog est une liste ordonnée et priorisée des besoins du produit et de toutes les features viables, même si celui-ci est généralement sous forme de User Stories fonctionnelles, le product backlog doit comprendre également les bugs, les tâches techniques...

Prototype

Un prototype est une représentation simplifiée d'un produit dans le but de tester une idée auprès des consommateurs. La fidélité graphique du prototype peut être d'une qualité variable, mais le but derrière est toujours le même, à savoir tester une idée et s'assurer que le développement d'un produit ou d'une fonctionnalité se base sur des données, plutôt que sur des suppositions.

Un prototype peut être destiné à être testé par des utilisateurs réels, ce qui permet de récupérer des données et d'orienter les développements futurs. Il est donc bien plus abouti qu'un simple "Wireframe".

Refactorisation

Une refactorisation est une opération menée par l'équipe de développement consistant à replonger dans le code d'une fonctionnalité. Elle peut s'avérer nécessaire dans de nombreux cas (et chaque produit se retrouve confronté à ce besoin à un moment ou un autre). Il peut s'agir d'une nécessité de rendre le code plus lisible, ou encore de le décomplexifier suite à des changements fonctionnels qui ont aboutit à "tordre" ce qui avait été établi. Ou encore par exemple en cas de montée de version d'une technologie, remettre à jour de ce fait les conventions en place dans l'utilisation de certaines pratiques.

Release/Version

Une release est une nouvelle version du produit, livrée aux utilisateurs. Elle est le fruit de plusieurs Sprints. Par habitude, on peut parler de release pour considérer la période de temps qui va du début du travail sur cette version jusqu'à sa livraison et qui passe par une série de sprints successifs. On parle également de Release dès que la nouvelle feature est disponible et communiquée aux utilisateurs.

Roadmap

La roadmap est l'itinéraire simplifié entre là où se trouve votre produit aujourd'hui (qu'il soit nouveau ou que vous envisagiez une n-ième itération) et la destination définie dans votre vision.

Il est primordial pour une Roadmap de contenir des objectifs chiffrés associés aux moyens à mettre en œuvre pour les atteindre, ainsi que les risques identifiés. Ne la confondez pas avec une liste sans fin de fonctionnalités prévues pour votre produit.



SCRUM

C'est un cadre méthodologique permettant de répondre à des problématiques complexes par des solutions adaptatives.

De manière très simple, Scrum demande d'avoir un Scrum Master qui a pour objectif de créer un environnement où :

- Le Product Owner ordonne le travail par la valeur, afin de répondre à des problématiques complexes de l'entreprise, dans un Product Backlog,
- L'équipe Scrum au complet transforme ces éléments de travail en un Incrément de valeur,
- L'équipe Scrum et les parties prenantes inspectent les résultats et s'ajustent pour le prochain Sprint,
- Et l'on répète cela.

Au cœur de Scrum se trouve le Sprint, qui est un temps court d'environ 1 à 2 semaines pour atteindre un objectif en équipe.

Le travail est planifié pendant le Sprint Planning avec un Objectif de Sprint, et un Sprint Backlog qui représente le travail à faire.

Puis le Daily Scrum (mêlée quotidienne) effectué quotidiennement, permet à l'équipe de développement de planifier leur journée.

Et enfin, la Revue de Sprint permet de présenter ce qui a été réalisé aux parties prenantes afin de prendre des retours utilisateurs, puis la Rétrospective de Sprint se fait avec l'équipe Scrum afin d'améliorer le fonctionnement de l'équipe, que ce soit côté processus, produit, technique, environnement de travail, etc...

Une Définition de Fini commune à l'équipe permet de rendre transparente la qualité.

De base, Scrum est incomplet, et demande à ajouter des techniques venant du monde du Product Management (Lean Startup), du Design (Design Thinking) et du développement (Software Craftsmanship) pour réellement être efficace.

S

SCRUMBAN

Le SCRUMBAN est un framework agile hybride qui mixe le Scrum et le Kanban. Cette méthodologie doit permettre aux équipes produit de structurer leur travail avec le Scrum en gardant les rôles majeurs et les cérémonies tout en gagnant en flexibilité et visibilité avec Kanban. L'établissement d'une cadence de livraison toutes les deux ou trois semaines par exemple pour marquer l'avancement et récupérer régulièrement du feedback sur le produit permet de sécuriser les développements au fur et à mesure du temps.

Pour en savoir plus : <https://leankit.com/learn/agile/what-is-scrumban/>.

Sprint

Un sprint est une itération déterminée dans le temps avec une date de début et une date de fin, pendant laquelle l'équipe Scrum va concevoir, développer et tester de nouvelles fonctionnalités.

La durée, fixée à l'avance pour chaque sprint, est généralement de 1 à 4 semaines. Au début du sprint sont définis : un objectif (sur lequel l'équipe s'engage et qui ne doit pas changer en cours de sprint), un sprint backlog ou périmètre de développement constitué de fonctionnalités (user story), correctif ou refactorisation ainsi qu'une Definition of Done (DoD).

Pendant un Sprint, le Sprint Backlog peut être revu par une négociation entre le Product Owner et le reste de l'équipe. Seuls l'objectif et la DoD sont fixés durant un Sprint.

Ainsi, lorsque l'équipe projet arrive à la conclusion d'un sprint, l'objectif doit être atteint et le Sprint Backlog écoulé en respectant les DoD définis. Un nouveau sprint peut dès lors commencer, et ainsi de suite jusqu'à l'aboutissement d'un produit fini répondant aux besoins utilisateurs puisse être livré.

Sprint Backlog

Le backlog de sprint est la liste des tâches et des user stories que l'équipe de développement doit traiter lors du sprint.

Sprint Planning

Réunion de travail entre le Product Owner, la Dev Team et s'ils sont présents les utilisateurs et/ou le client, consistant à rebalayer les users stories du prochain Sprint Backlog, les valider, ou changer les priorités avec d'autres users stories du Product Backlog prêtes à embarquer si nécessaire. C'est la cérémonie d'ouverture du sprint, essentielle pour bien cadrer avec toutes les parties prenantes l'objectif et le périmètre qui commence.

Sprint Review

Lors du Sprint Review, l'équipe passe en revue le travail accompli, le travail planifié qui n'a pas été terminé et inspecte le produit livré à l'issue du sprint. Cette cérémonie peut également donner lieu à une démonstration du produit pour présenter aux parties prenantes les fonctionnalités développées au cours du Sprint afin de recueillir leurs feedbacks. Puis de faire le point sur l'avancement global du projet.

Story Map / Story Mapping

Le story mapping est une technique, mise en place au cœur des projets agiles, permettant de définir la vision d'un nouveau produit à son lancement, ou encore les nouveaux besoins utilisateurs macros d'un produit. Cet atelier très opérationnel permet d'ailleurs d'en ressortir avec un premier Backlog Product alimenté en Epics et larges user stories.

Son application consiste en 2 axes, l'horizontal représentation la chronologie des actions d'un utilisateur, et l'axe vertical les fonctionnalités des plus essentielles au moins essentielles.

Le story mapping sera toujours réalisé avant le développement du produit, ou des nouvelles features concernées.

Story Points

Les points attribués à une user story sont une unité de mesure pour exprimer une estimation de l'effort global qui sera nécessaire pour développer intégralement une User Story ou tâche du Backlog Produit. Une estimation en Story Points est classique en agilité au contraire de l'estimation jour/hommes et des délais de réalisation typiques de la gestion de projet. L'estimation en Story Points permet d'avoir une estimation «relative», où on estime des User Stories par rapport aux autres, plutôt que des estimations en temps, «absolues». Le livre «The Mythical Man-Month» l'explique bien : 9 femmes n'ont pas un bébé en un mois !

User Journey Map (ou Customer Journey Map)

Le User Journey Map est la représentation visuelle de toutes les interactions d'un client ou d'un prospect avec notre entreprise/service. Cet outil est important car il permet à tous les acteurs de l'entreprise d'avoir une meilleure vision du parcours utilisateur dans son ensemble et de son expérience lorsqu'ils interagissent avec notre entreprise.

User Story

Une user story est une description courte, synthétique et compréhensible par tous d'une fonctionnalité, d'un besoin précis. Une User Story, comme son nom l'indique, c'est avant tout une histoire qui se raconte, crée la discussion et amène l'équipe à confirmer sa compréhension du besoin.

C'est la retranscription écrite d'un besoin métier en une fonctionnalité ou partie d'une fonctionnalité qui est compréhensible facilement par toutes les parties prenantes d'un projet de par sa forme qui est normalisée (En tant que ... Je veux / Je souhaite ... Afin de ...).

Elle doit également respecter certains critères tels que l'INVEST (Independent, Negotiable, Valuable, Estimable, Small, Testable). Autrement dit, elle doit être indépendante vis à vis des autres US d'un backlog, elle doit pouvoir être remise en question, avoir une valeur business/métier, elle doit pouvoir être estimée en complexité par l'équipe de développement, elle doit être assez petite pour pouvoir être embarquée dans un sprint. Enfin, elle doit pouvoir être testée de bout en bout.

Une User Story est également toujours accompagnée de critères d'acceptance fonctionnels, si nécessaire de tâches techniques ajoutées par l'équipe de développement, ou encore potentiellement de tests d'acceptance.

Pour plus d'infos, vous pouvez consulter le super article [Écrire et découper ses users stories comme un ninja.](#)

U

User Test

Le produit est testé en conditions réelles par un panel d'utilisateurs représentatifs afin de comprendre exactement la manière dont ils se servent de l'application.

Le test utilisateur consiste à observer et interviewer les utilisateurs cibles d'un produit pour comprendre leurs objectifs réels et relever les difficultés d'utilisation.

Plusieurs ateliers existent pour réaliser un test utilisateur. Vous pouvez par exemple concevoir des scénarios de test avec des tâches à réaliser et des fiches d'observation par tâche (<https://medium.com/poool-stories/conception-ux-comment-organiser-un-test-utilisateurs-cd20f5b3e283>), ou encore mettre en place un atelier d'expérience mapping avec vos utilisateurs : (<https://www.ux-republic.com/template-experience-map/>). Ces approches ne sont pas exhaustives mais permettent déjà de se projeter.



V

Vélocité

Il s'agit de l'effort total accompli par l'équipe durant un sprint. Elle est obtenue de plusieurs manières : en additionnant le nombre de points des user stories terminées à la fin du sprint, ou bien tout simplement en additionnant le nombre de user stories terminées si elles valent toutes 1 (par exemple en NoEstimate). La vélocité peut être utilisée pour valider ou réviser la planification des prochains sprints tout en gardant à l'esprit que la vélocité peut fluctuer et n'est qu'une prévision instable, le but espéré étant surtout d'avoir une vélocité avec le moins de variation possible.

W

Wireframe

Un Wireframe est un prototype visuel dont l'objectif est de montrer la structure d'une page web et la disposition des différents éléments qui la compose, ainsi que leurs interactions. Les wireframes sont utilisés au tout début du processus de développement pour représenter les parcours utilisateurs.





Eleven Labs

www.eleven-labs.com