

Cours 3 : Les méthodes agiles de gestion de projet

Agile

Scrum

Gestion des risques

Agile

La méthodologie Agile

❑ Problèmes des méthodes classiques

- Cycle de développement long
 - ❑ Pas de versions intermédiaires livrées au client
- Le client est consulté uniquement au début de projet
- Difficile de modifier le projet en cours de développement
 - ❑ Figé par un contrat (rigide)
 - Les nouveaux besoins clients ne peuvent pas être intégrés
 - Les retards de développement ne sont pas admis
- Les développeurs ne sont pas toujours motivés
 - ❑ « Pourquoi je dois coder ce truc qui a mal été spécifié »
- Beaucoup de documentations pas toujours utiles

La méthodologie Agile

❑ **La méthodologie Agile**

- s'oppose à la méthodologie classique
- Elle se veut
 - ❑ plus souple plus adaptable → plus pragmatiques
 - ❑ Plus centré sur les besoins du client
 - impliquer le client et réagir rapidement à ses demandes
 - vise la satisfaction réelle du client en priorité aux termes d'un contrat de développement.

❑ **Méthode créée pour les projets de développement web et informatique.**

❑ **Aujourd'hui Généralisé à de nombreux types de projets, tous secteurs confondus.**

Agile Manifesto

- ❑ **Les individus et leurs interactions, plus que les processus et les outils**
- ❑ **Des produits opérationnels, plus qu'une documentation exhaustive**
- ❑ **La collaboration avec les clients, plus que la négociation contractuelle**
- ❑ **L'adaptation au changement plus que le suivi d'un plan**
 - Source : www.agilemanifesto.org

Méthodes Agiles

❑ **Les principales méthodes Agiles**

- ❑ Rapid application development (RAD), 1991
- ❑ Dynamic systems development method (DSDM) (1995),
- ❑ Adaptive software development (ASD)
- ❑ Feature Driven Development (FDD)
- ❑ **Scrum (1995) la plus utilisée**
- ❑ **Extreme programming (XP), (1999) la deuxième la plus utilisée**



Thomas Devogele

Bibliographie Scrum

- ❑ **Scrum 4e éd.- Le guide pratique de la méthode agile la plus populaire (2015) Aubry, Dunod**
- ❑ **Scrum, agilité & Rock'n Roll (Blog d'Aubry)**
 - <http://www.aubryconseil.com/>
- ❑ **Kanban et Scrum - tirer le meilleur des deux (Henrik Kniberg 2010)**
 - <http://henrik-kniberg.developpez.com/mattias-skarin/livre/scrum-kanban/?page=sommaire>
- ❑ **Guide Scrum 2013 (Auteur : [Jeff Sutherland](#), [Ken Schwabber](#))**
 - <http://wiki.ayeba.fr/Guide+Scrum+2013>
- ❑ **Introduction to Scrum - 7 Minutes (Vidéo)**
 - <https://www.youtube.com/watch?v=9TycLR0TqFA>

Définition

- ❑ **Scrum (prononcer scrueum) est un schéma d'organisation de développement de produits complexes.**
- ❑ **Il est défini par ses créateurs (Hirotaka Takeuchi et Ikujiro Nonaka 1986) comme un **cadre de travail** permettant de répondre à des problèmes **complexes** et **changeants**, tout en livrant de manière productive et créative des produits de la plus grande valeur possible.**

Pourquoi Scrum

□ Analogie au sport

- Méthode de type **Cycle de vie**
 - Analogie à une **Course de relais** avec des étapes bien définies
 - Spécifications des besoins,
 - Conception générale,
 - Conception détaillée,
 - Codage et tests unitaires,
 - Intégration des modules,
 - Intégration du logiciel,
 - Recette

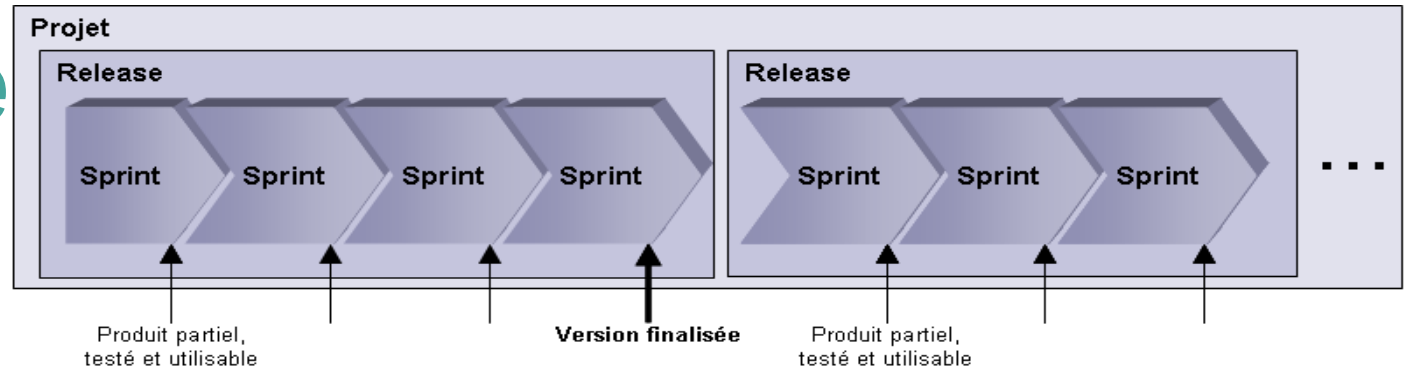


Pourquoi Scrum



- Méthode Scrum
 - Analogie à la **mêlée** du Rugby
 - une approche holistique comme au *rugby*— quand une équipe essaie d'avancer en restant unie, en se passant le ballon de main en main— peut mieux servir les exigences de compétitivité d'aujourd'hui.”
 - Taux de réussite des projets Scrum meilleur que pour les projets classiques

Caractéristique de Scrum



- ❑ Equipe responsable, en **auto-organisation**
- ❑ Avancement du produit par une série de « **sprints** » de 2 à 4 semaines
- ❑ Exigences définies comme des éléments d'une liste appelée « **backlog** de produit »
- ❑ Pas de prescription de pratiques d'ingénierie
- ❑ Utilisation de **règles génériques (cadre)** permettant de créer un environnement agile pour un projet

Méthode Scrum (agile)

❑ Objectifs

- Le **client** dans la boucle régulièrement
- Livrer des versions partielles très rapidement
 - ❑ Projet plus facilement adaptable
 - ❑ Seules 20% des fonctions d'un logiciel sont très souvent employées
- Des **équipes projets** dynamiques avec des objectifs rapides
 - ❑ Améliorer
 - l'organisation interne
 - La communication
 - La motivation

Un cadre

- ❑ **Pas de méthode**

- On fait ça puis ça

- ❑ **Pas d'outils**

- Plus de dialogue

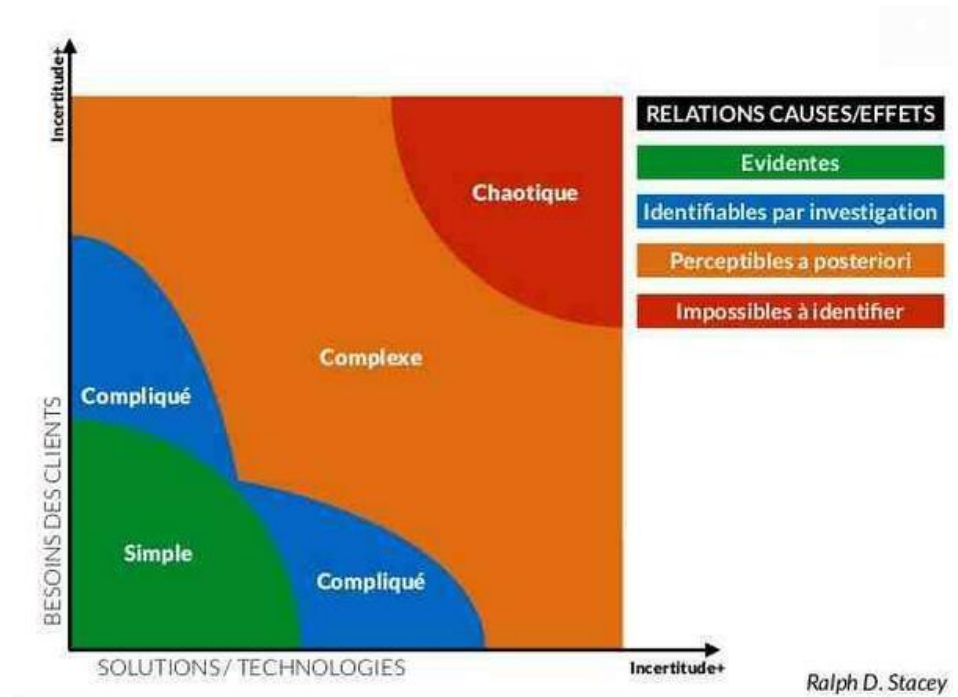
- ❑ **Peu de documentation**

- ❑ **Des règles à respecter pour**

- Produire
 - S'organiser
 - Livrer
 - Motiver l'ensemble des membres de l'équipe

Matrice de Stacey

- ❑ **Simple, compliqué, complexe ou chaotique**
- ❑ **Pour bien travailler, il faut rendre les choses simples**
 - On part de choses incertaines
 - ❑ Besoins des clients
 - ❑ Technologies à employer
 - Petit à petit, on les clarifie
 - ❑ Décomposition des besoins
 - On réalise les choses une fois qu'elles ont été clarifiées et simplifiées



Le cycle des sprints

□ **Incrémental**

- Accroissement du produit à chaque étape
- Pas uniquement un produit à la fin

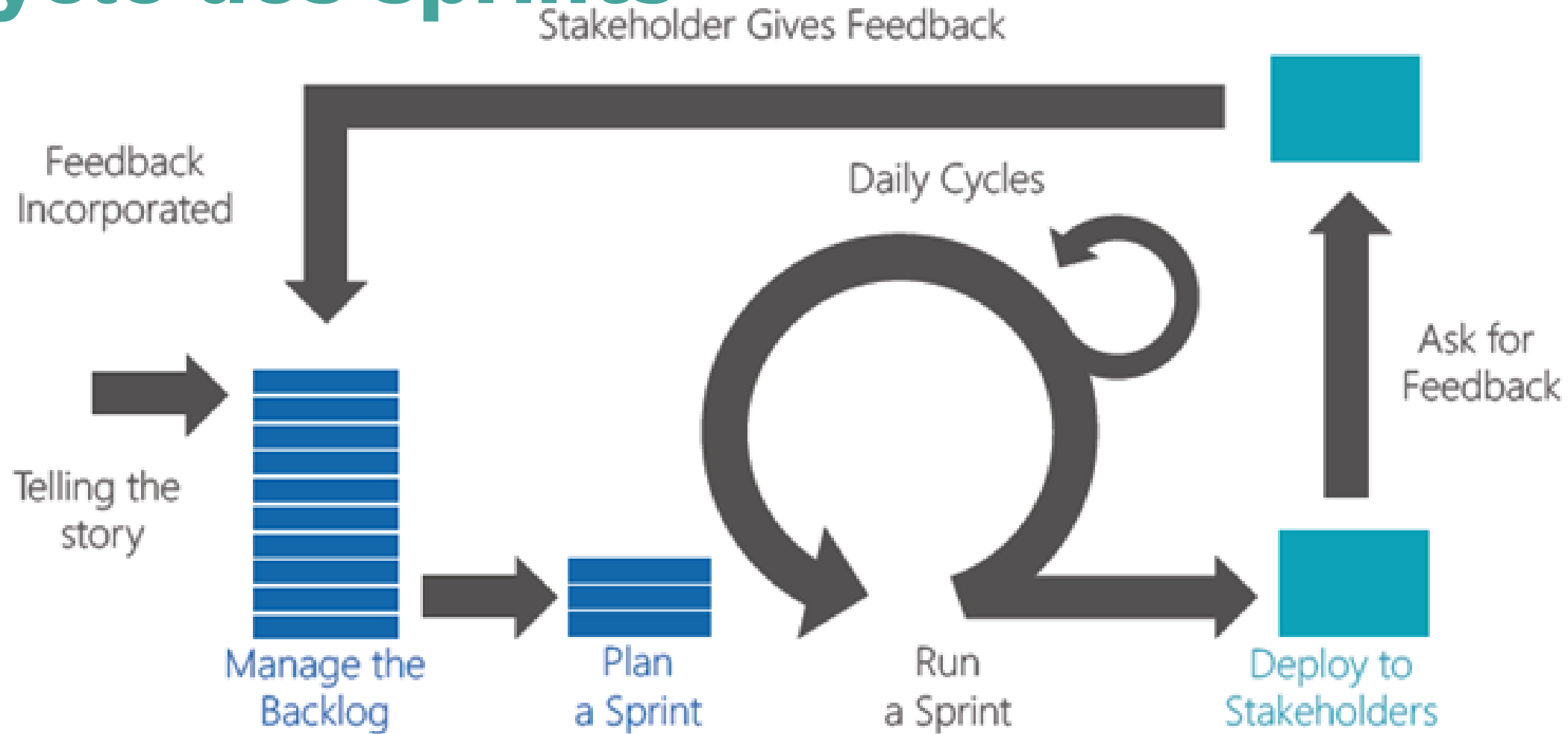
□ **Itératif**

- Les mêmes activités sont reproduites à chaque cycle
- De même durée (2 à 4 semaines)

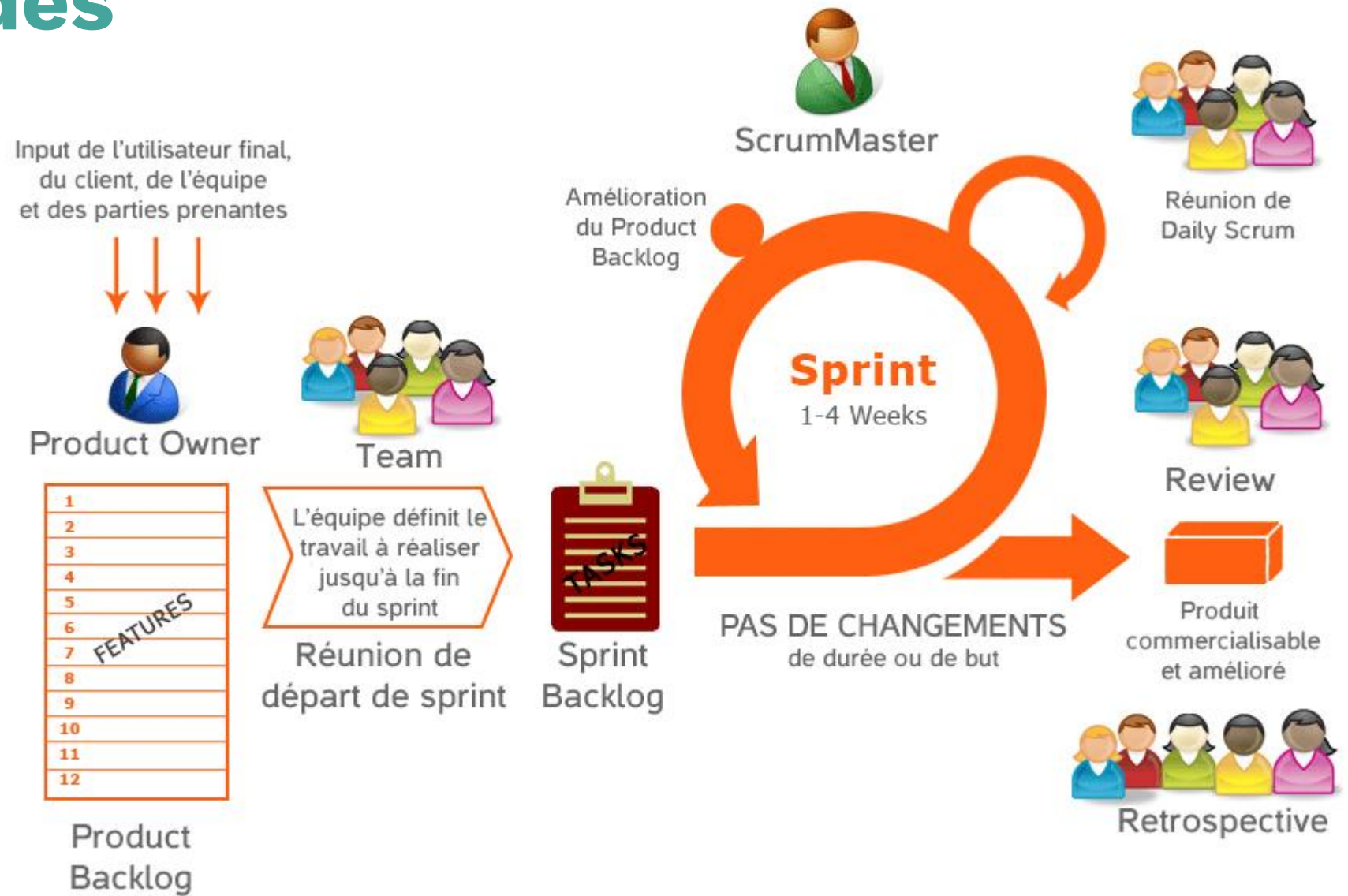
□ **Feedback**

- Pour chaque incrément
 - Un retour des parties prenantes (Stakeholders) qui est pris en compte pour le cycle suivant
 - Éviter les retours en arrière importants

Le cycle des sprints



Le cycle des sprints



Les activités du sprint

□ **Activités identiques pour chaque sprint**

- Spécification
- Architecture
- Codage
- Test

□ **Objectif**

- **Plutôt que de faire toute une discipline d'un coup, les équipes Scrum font un peu de tout, tout le temps.**

Les activités du sprint

□ Pour chaque sprint, règles des 100%

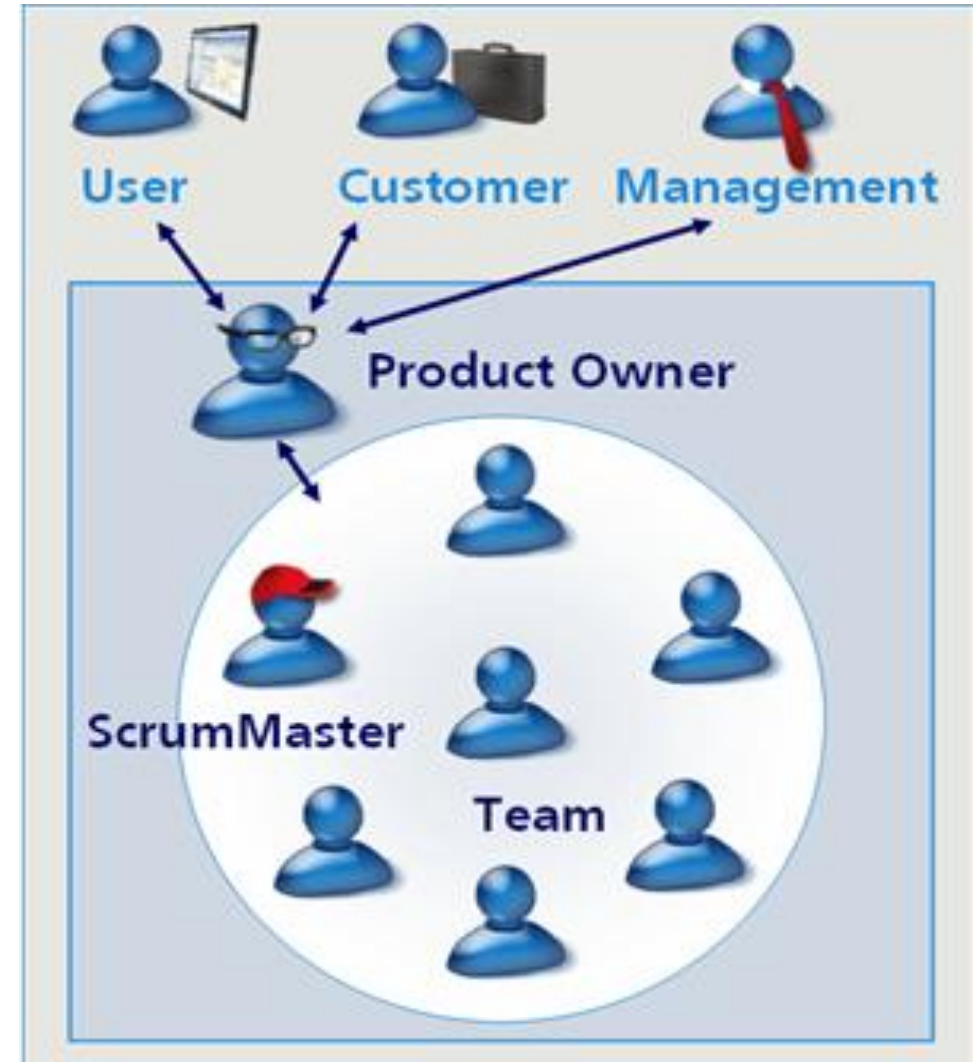
- Les exigences des parties prenantes sont connues
- 100% des spécifications détaillées avant de commencer à concevoir
- 100% de la conception avant de commencer le code
- 100% du code avant de commencer les tests

Acteurs et rôles

A solid teal-colored bar at the bottom of the slide, featuring a central V-shaped notch.

Acteurs et rôles

- ❑ **Product Owner**
- ❑ **Scrum master**
- ❑ **Equipe de développement**
- ❑ **Parties prenantes (client, marketing, direction)**



Product Owner



❑ Il doit définir les fonctionnalités du produit

- Choisir la date et le contenu de la release
- Responsable du retour sur investissement
- Définit les priorités dans le backlog en fonction de la valeur « métier »
- Ajuste les fonctionnalités et les priorités à chaque sprint si nécessaire
- Accepte ou rejette les résultats

Scrum master



□ Rôle

- Ce n'est pas un chef de projet hiérarchique
 - Il n'affecte pas les tâches
 - Il ne demande pas des comptes
 - Il ne rend pas des comptes individuellement
- C'est un **coach**
 - Il doit veiller à la bonne entente
 - Encourager, gérer les conflits humains
 - Élimine les obstacles
 - S'assure que l'équipe est complètement fonctionnelle et productive
 - Protège l'équipe des interférences extérieures

Équipe Scrum

- ❑ **De 5 à 10 personnes**
- ❑ **Regroupant tous les rôles**
 - Architecte, concepteur, développeur, spécialiste IHM, testeur, etc.
 - Les rôles peuvent changer
 - A plein temps sur le projet, de préférence
 - Exceptions possibles : des experts : administrateur, ...
 - L'équipe s'organise par elle-même
 - ❑ Répartition des tâches (Auto organisation)
- ❑ **La composition de l'équipe ne doit pas changer pendant un Sprint**

Partie Prenante (Stakeholder)

- ❑ **Clients**
- ❑ **Marketing**
- ❑ **Direction de l'entreprise (management, propriétaire)**
- ❑ **Fournisseur de services**
 - Invité pour les releases pour donner leur feedback

Le backlog

A solid teal horizontal bar spanning the width of the slide, featuring a central V-shaped notch.

Le backlog de produit

- ❑ **Pas de spécification détaillée en début de projet**
- ❑ **Mais une collecte des fonctions essentielles (les **features**) et les raffine progressivement**
 - Les features sont décomposées en **user stories**
 - ❑ Exemple :
 - Ajouter les **technical stories**
 - Exemple :
 - ❑ **sécurisation, archivage, optimisation, ajout d'une plate-forme...**
- ❑ **l'outil de collecte des stories s'appelle le **backlog de produit****

Objectif du backlog de produit

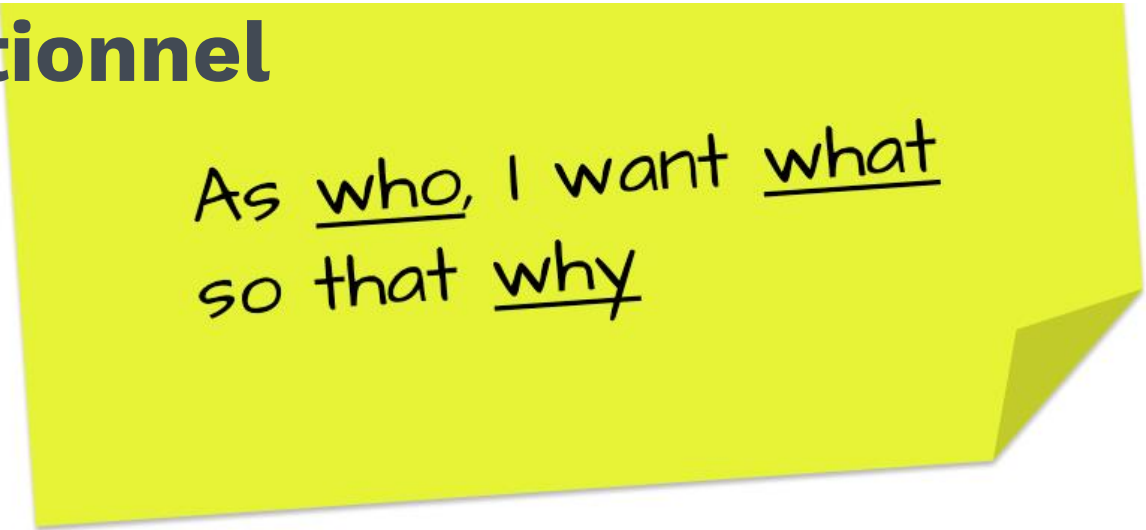
- ❑ Le backlog de produit est la liste des fonctionnalités attendues d'un produit détaillé à l'aide de stories.
- ❑ Il contient aussi tous les éléments qui vont nécessiter du travail pour l'équipe.
- ❑ Les éléments y sont classés par **priorité** ce qui permet de définir l'ordre de réalisation
- ❑ Les stories sont regrouper pour former des **releases**
 - Jalon majeur associé souvent à une nouvelle version

User story dans le backlog

- **Sous la forme de**

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

- **La partie afin de est optionnel**



As who, I want what
so that why

User story dans le backlog

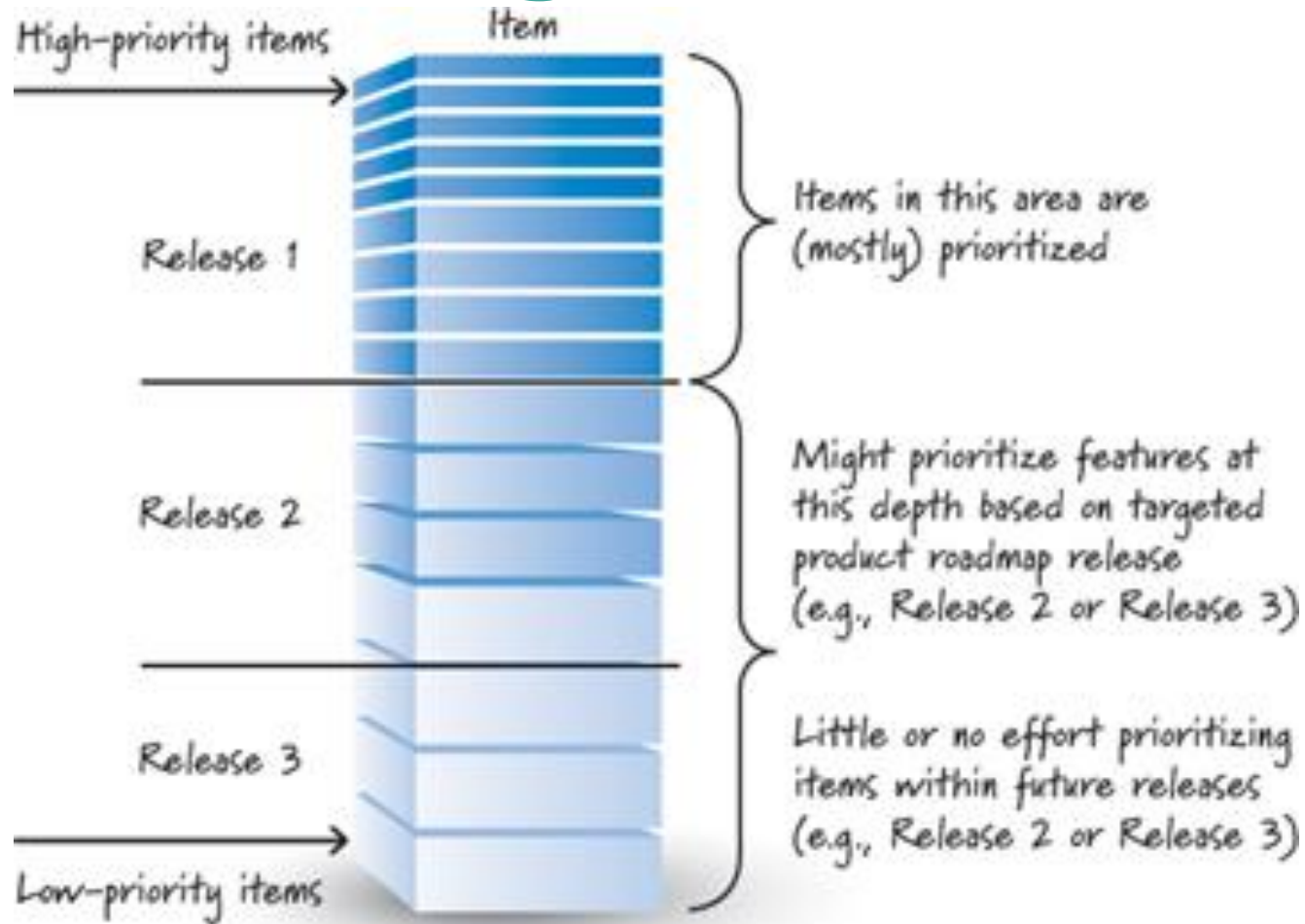
❑ **Exemple 1 : Recherche de clients**

- En tant qu'utilisateur,
- je veux pouvoir rechercher mes clients par leur prénom et leur nom de famille
- afin de les retrouver rapidement lorsque je reçois un appel de leur part

❑ **Exemple 2 : *Modification emploi du temps***

- En tant qu'utilisateur,
- je veux pouvoir modifier mes emplois du temps, mais pas ceux des autres utilisateurs

Backlog



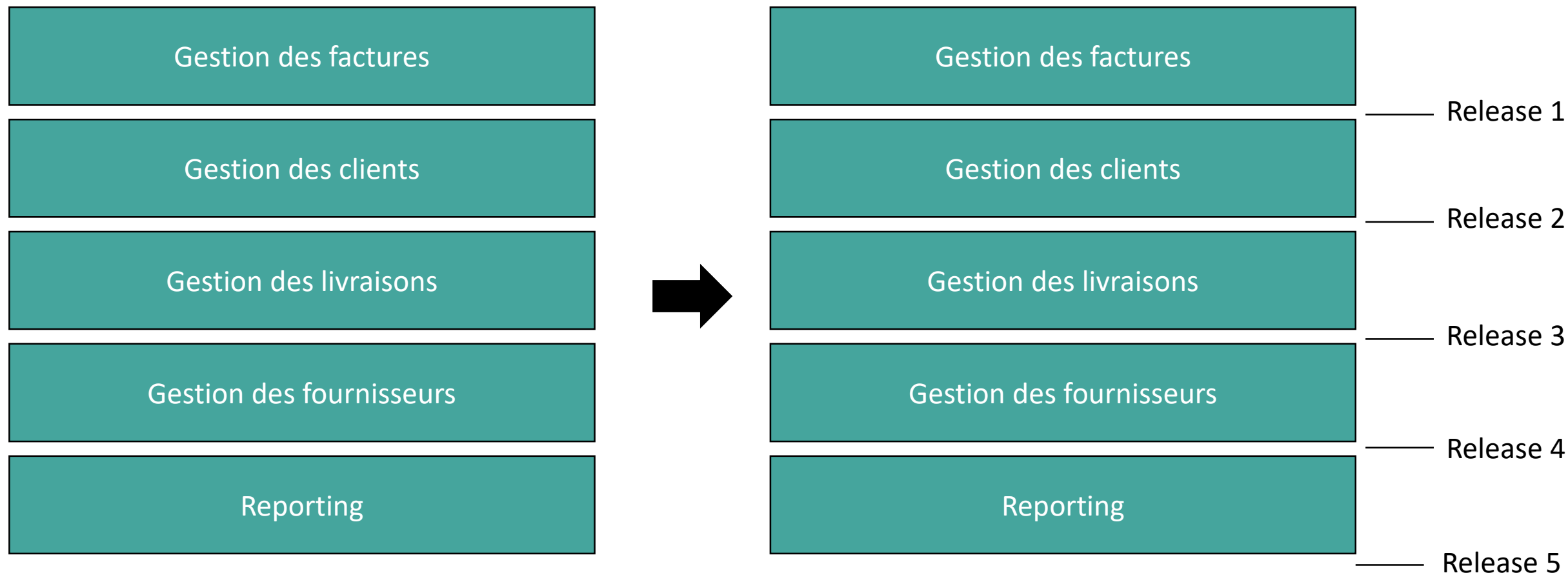
Gestion des factures

Gestion des clients

Gestion des fournisseurs

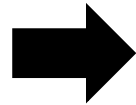
Reporting

Backlog ordonnancement & releases



Backlog détaillé de(s) première(s) user stories

Gestion des factures



Ajout d'une nouvelle facture
avec date et numéro

Ajout des coordonnées d'un client

Ajout des produits, quantité et prix

Calcule du montant HT, TVA et montant TTC

Exemple de backlog pour la gestion des réservation des chambres d'un hôtel

Story	Priorité
Un invité peut faire une réservation	3
En tant qu'invité, j'annule une réservation	4
En tant qu'invité, je change les dates d'une réservation	4
En tant qu'employé de l'hôtel, je produis les rapports de revenu par chambre	5
Améliorer la gestion des exceptions	6
	7
	8

À la fin des trois Stories vertes, je peux faire une release

Backlog de produit

- ❑ **Le backlog de produit est sous la responsabilité du Product Owner qui définit les priorités.**
- ❑ **Le backlog est élaboré avant le lancement (ou **sprint 0**).**
- ❑ **Il est utilisé pour :**
 - planifier des releases,
 - la planification des sprints
 - Définir les exigences
- ❑ **Il est **visible** par tout le monde.**

Vie du backlog

- ❑ **A une durée de vie longue**
- ❑ **Vivant, il n'est pas un document figé**
 - Des éléments sont ajoutés,
 - des éléments sont supprimés,
 - des éléments sont décomposés
 - les priorités ajustées.

Release

- ❑ Une release est une **séquence de sprints**
- ❑ À la fin il y a un produit partiel suffisamment intéressant pour le présenter aux clients, voir le déployer
- ❑ La **date** de la release est connue longtemps à l'avance, elle ne change pas
 - Cela favorise la motivation
 - En cas de problème, les tâches les moins importantes ne seront pas délivrées

Les moments clés

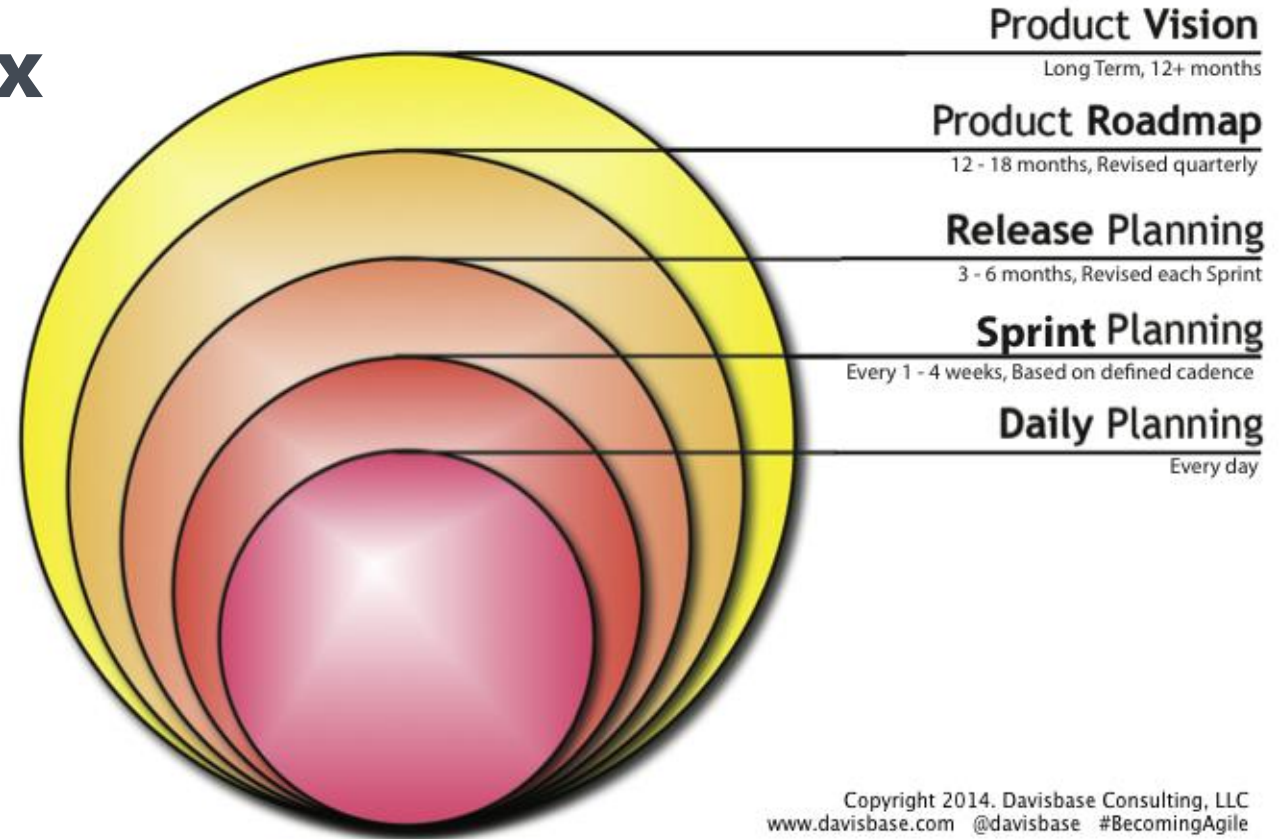
A solid teal horizontal bar spanning the width of the slide, featuring a central semi-circular notch that aligns with the text above.

Planifications et revues

□ À différents niveaux de détails

5 LEVELS OF PLANNING

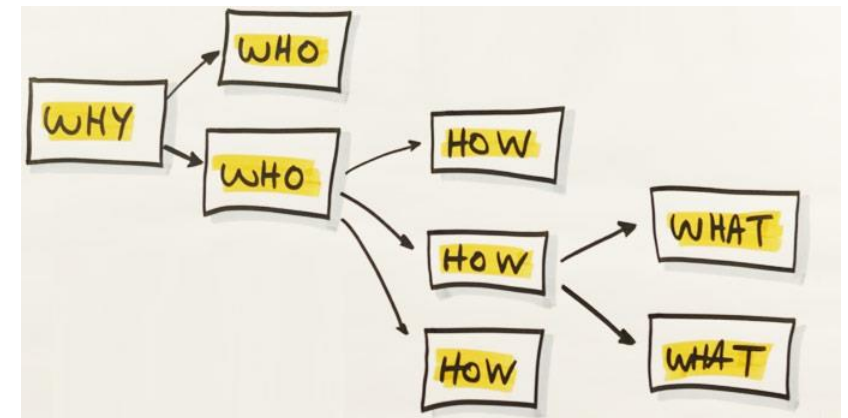
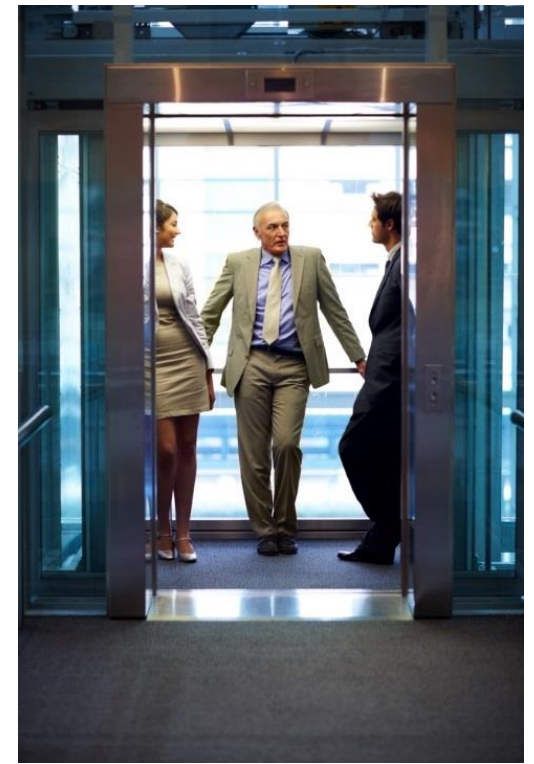
Always accurate, but at varying levels of precision.



Sprint 0

□ Bien définir la vision

- De manière simple
 - sous la forme de « **pitch** » ascenseur
 - 30 secondes
 - 7 points maximum
- En détaillant après la portée de la vision
 - Sous la forme **impact mapping**
 - Carte mentale qui répond à des questions



POINTS d'un pitch ascenseur

- ❑ **POUR** (public concerné par le produit)
- ❑ **QUI SOUHAITENT** (formulation du besoin des cibles)
- ❑ **NOTRE PRODUIT EST** (ce qu'est le produit)
- ❑ **QUI** (le bénéfice majeur, l'utilité de la solution)
- ❑ **A LA DIFFERENCE DE** (pratique actuelle, concurrence)
- ❑ **PERMET DE** (éléments différenciateurs majeurs)

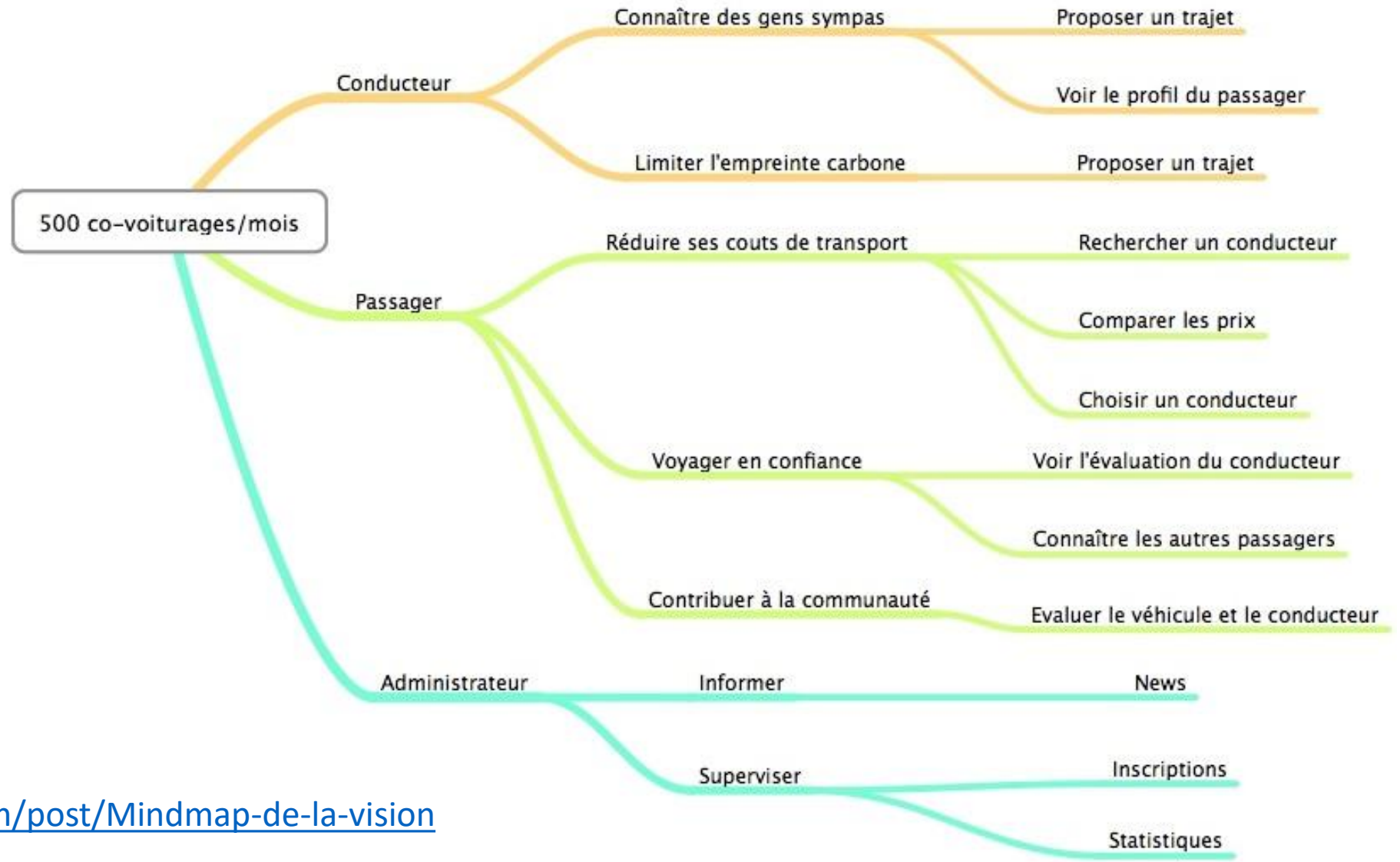
Exemple Smart Loire :

- ❑ **POUR** **touriste**
- ❑ **QUI SOUHAITE** obtenir un itinéraire optimisé qui s'adapte aux évènements
- ❑ **NOTRE PRODUIT EST** un ensemble de services web à intégrer dans des sites existants
- ❑ **QUI combine** les informations des acteurs du tourisme, propose un itinéraire optimisé en fonction de ses préférences
- ❑ **A LA DIFFERENCE DE** Tripadvisor qui ne propose pas des itinéraires, mais des listes et qui ne prend pas en compte les préférences utilisateur
- ❑ **PERMET DE** définir des millions d'itinéraires spécifiques adaptés à chaque touriste

Exemple Smart Loire (2) :

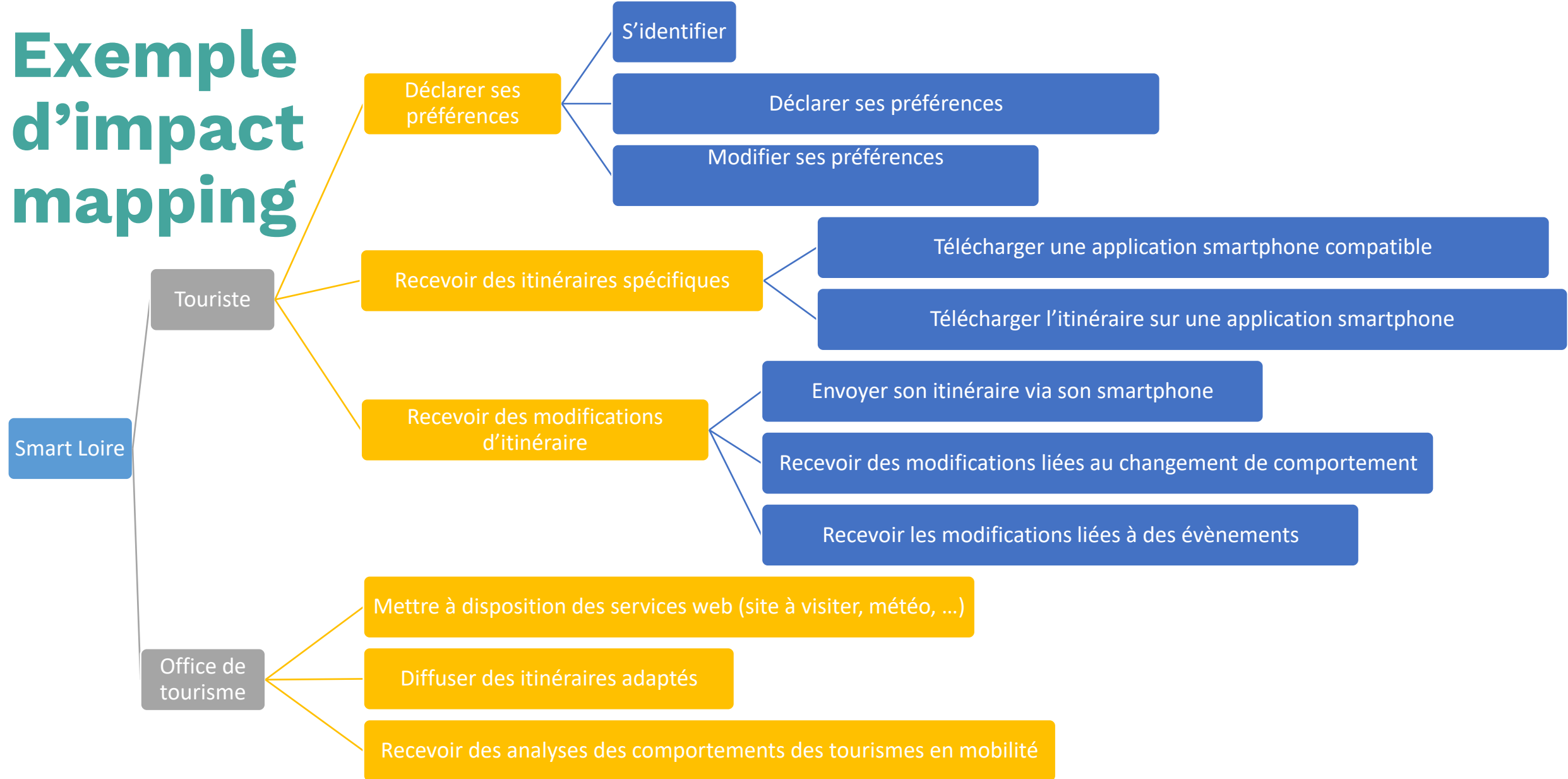
- ❑ **POUR acteur du tourisme** (office de tourisme)
- ❑ **QUI SOUHAITE** proposer des itinéraires variés aux touristes
- ❑ **NOTRE PRODUIT EST** un ensemble de services web à intégrer dans leurs sites existants (Internet et smartphone)
- ❑ **QUI combine** leurs informations et propose un itinéraire optimisé en fonction des préférences du client
- ❑ **A LA DIFFERENCE DE** leur site classique qui ne propose que des itinéraires prédéfinis
- ❑ **PERMET DE** définir avec le tourisme son itinéraire, leur transmettre et l'adapter en temps réel

Exemple d'impact mapping



<http://www.aubryconseil.com/post/Mindmap-de-la-vision>

Exemple d'impact mapping



Au début du sprint

□ Planification

- L'équipe choisit, à partir du backlog de produit, les éléments qu'elle **s'engage** à finir avant la fin du sprint.

□ Affinage

- des stories sont décomposées en **tâches**
- Objectif décomposer suffisamment de stories pour le sprint + 20% si l'équipe va plus vite
- La liste des tâches pour chaque story est créée
- Les tâches sont identifiées et estimées (**1-16 h**)
 - Collectivement, pas seulement par le Scrum Master
 - Si plus de 16h, il faut décomposer
- La conception de haut niveau est abordée

Au début du sprint

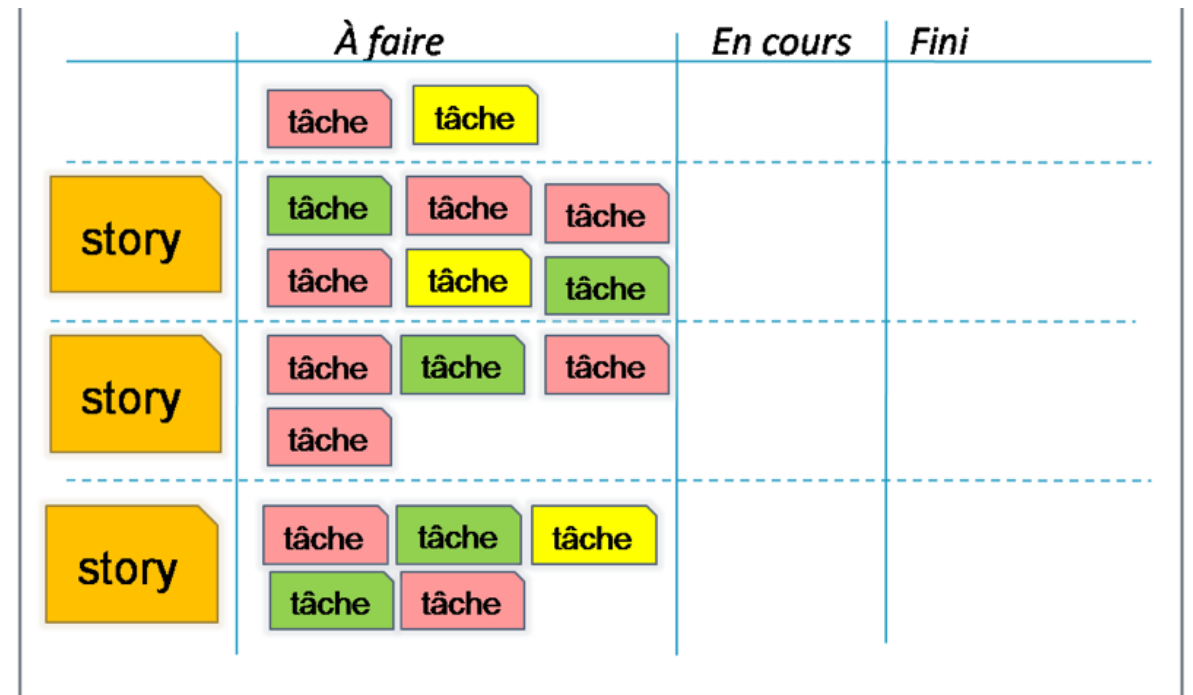
□ **Définition du fini** pour une story qui est prête

- Au minimal
 - démontré avec succès lors de la revue de sprint
- Ou plus loin, par exemple pour une user story :
 - codé en suivant le standard de codage
 - testé unitairement
 - tests d'acceptation passés avec succès dans un environnement de test
 - version disponible en français et en anglais
 - documentation marketing rédigée
 - documentation utilisateur rédigée

Au début du sprint

□ Tableau de suivi des tâches

- Il peut aussi y avoir des tâches du sprint précédent qui sont à revoir : **dette technique**



- Idée issue des **Kanbans**

Kanban

□ Kanban

- Étiquette en japonais
- Méthode issue de Toyota (1950)

□ Objectif

- Visualiser le flux de travail
 - workflow
- Juste à temps (just in time) → flux tendu
 - Plus collaboratif
 - Plus interactif
 - Dont client
 - Plus engageant
 - Plus motivant
 - Meilleure vision de l'avancement



Vie de la liste des tâches

□ **Chacun s'engage sur du travail qu'il choisit**

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

Exemple de tâches

- ❑ **Contexte : Site web touristique**

- La gestion des hôtels a été réalisée lors d'un sprint préalable

- ❑ **User story retenu pour être décomposé en tâches**

- En tant que touriste potentiel dans la région, je veux voir les photos des hôtels

- ❑ **Tâches identifiées**

- Coder la couche de persistance base de données (8 heures)
 - Coder l'IHM (4h)
 - Ecrire les tests (4h)
 - Coder la classe foo (6h)
 - Maj les tests de performance (4h)

- ❑ **Les membres de l'équipe doivent se les attribuer**

Planning poker

□ Définir collectivement

- Pour chaque tâche à réaliser lors du futur sprint
 - sa difficulté
 - Sa durée
 - Qui est le plus apte à réaliser la tâche

□ Comment

- À l'aide d'un jeu
 - Où chacun répond
 - Cache sa réponse
 - Toutes les réponses sont retournées en même temps
 - Des justifications



Mêlée quotidienne (Daily Scrum)

□ Cadre

- **15 minutes** en équipe et debout
- Sans ordinateur
- Avec un tableau blanc et des Post-it®
- Permet d'éviter l'organisation d'autres réunions



Mêlée quotidienne

- ❑ **Chacun répond à 3 questions**
 - **Qu'as-tu fait hier ?**
 - **Que vas-tu faire aujourd'hui ?**
 - **Y a-t-il un obstacle qui te freine ?**
- ❑ **Il ne s'agit pas de comptes rendus au Scrum Master**
- ❑ **Ce sont des engagements devant des pairs**
- ❑ **La liste des tâches peut être mise à jour**

Obstacle

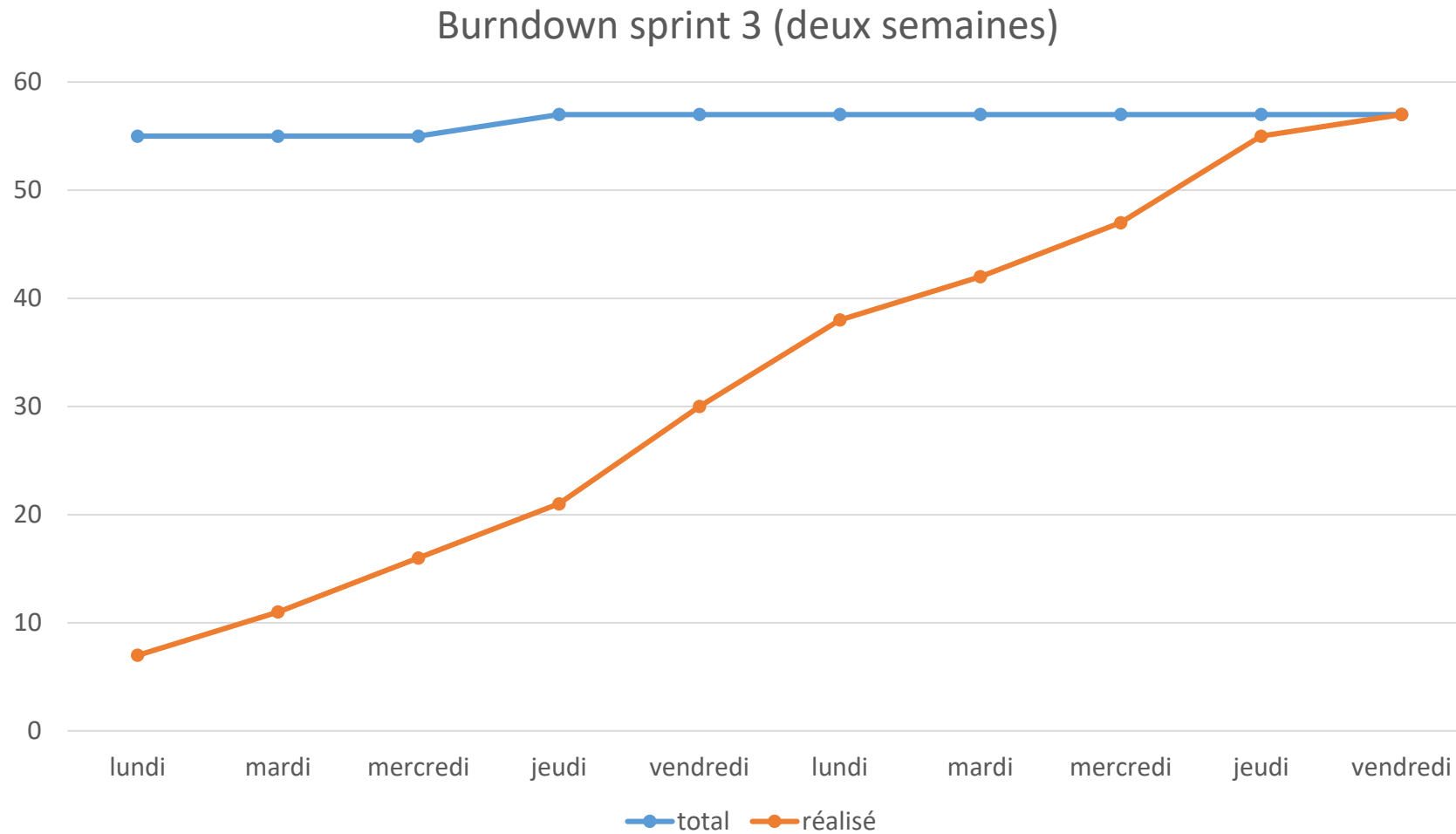
- ❑ **Fait concret qui bloque ou ralentit le bon déroulement d'une ou plusieurs tâches empêchant d'avancer à son rythme**
 - Exemple
 - ❑ Emploi d'une technologie mal maîtrisée
 - ❑ Un membre de l'équipe est malade
 - ❑ Un cas n'a pas été prévu
 - Exemple : nom de famille avec des apostrophes « Créac'h »
- ❑ **Ils doivent être identifiés**
 - Ajouter une **liste des obstacles** (à traiter, à résoudre) sur le tableau blanc
- ❑ **C'est le Scrum Master qui est chargé de trouver une solution**

Burndown de sprint

(graphique d'avancement)

- ❑ **est une représentation graphique de l'évolution de quantité de travail restante par rapport au temps sur une période de temps donnée**
 - En X : les jours du sprint
 - En Y : plusieurs choix possibles
 - ❑ le nombre de tâches restantes à réaliser
 - ❑ Le nombre d'heures restantes
 - Complexe
 - ❑ 2 courbes : nombre de tâches total du sprint et le nombre de tâches déjà réalisées
 - Plus positif
 - Permet de faire varier le nombre de tâches total
- **A dessiner sur le tableau blanc à côté du backlog et du tableau des tâches (Kanban)**

Burndown de sprint



Revue de sprint

- ❑ **L'équipe présente ce qu'elle a fait pendant le sprint (le vendredi en début d'après-midi par exemple)**
 - Se fait avec une **démonstration** des nouvelles **fonctionnalités qui ont été finies** ou de l'architecture
 - Informel
 - Préparation < 2h (le vendredi matin)
 - Pas de slides
 - Toute l'équipe participe
 - On invite du monde

Rétrospective du sprint

- ❑ **Réfléchir régulièrement à ce qui marche et ce qui ne marche pas**
 - Dure en général de 15 à 30 minutes
 - Fait à la fin de chaque sprint
 - Toute l'équipe participe
 - ❑ ScrumMaster
 - ❑ Product Owner
 - ❑ Equipe
 - ❑ Eventuellement clients et autres intervenants
- ❑ **Après la revue et une pause**
- ❑ **Toute l'équipe collecte du feedback et discute sur ce qu'elle aimerait :**
 - Commencer à faire
 - Arrêter de faire
 - Continuer à faire

Sur le tableau blanc

❑ Il faut au moins

- Le backlog
- Le tableau de l'avancement des tâches du sprint en cours
- La définition de prêt des stories en cours
- Burndown chart
- La liste des obstacles

❑ Il peut y avoir aussi d'autres outils, indicateurs

Avantages (1/2)

- ❑ **Cadre de production, méthodologies**
- ❑ **Augmente productivité et qualité**
- ❑ **Plus de souplesse et de créativité**
- ❑ **Résultat conforme aux attentes**
- ❑ **Pilotage au quotidien**
- ❑ **Règles définies clairement**

Avantages(2/2)

Copilotage avec le client

- Propice à une confiance réciproque
- Transparence sur l'avancement
- Adaptabilité

Equipe & responsabilité collective

- Pas d'individualisme → Améliorer la communication
- Amélioration permanente
- Chaque équipe a son lot de responsabilité

Inconvénients

- ❑ **Peu, voire pas, de documentation écrite**
- ❑ **Besoin d'une grande disponibilité de tous les acteurs**
- ❑ **Coûts (susceptibles de dériver)**
 - Pas facile de définir le coût au préalable et donc établir un devis
 - C'est aussi vrai a posteriori pour les méthodes classiques

Conclusion

Pour développer des logiciels, des systèmes d'information...
Besoin de méthodes et d'outils

❑ Assez rigides

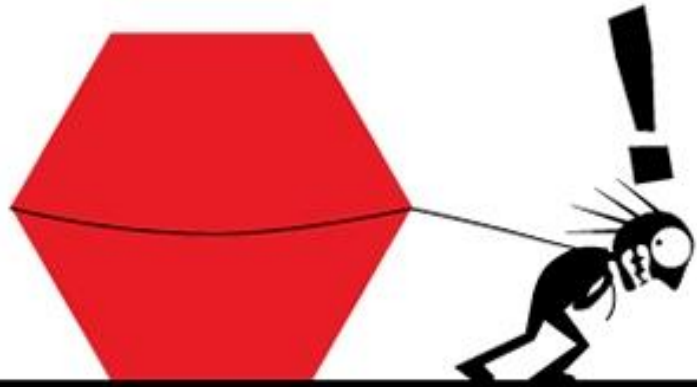
- Cycle en **V**
 - ❑ Étape bien définie
 - ❑ Répartition des tâches et planification définie
 - ❑ Problème de réadaptation
 - ❑ Motivation humaine qui peut être assez faible
 - ❑ Résultat après une longue période qui peut ne pas convenir aux clients

❑ Assez souple

- Modèle **Scrum**
 - ❑ Le client est dans la boucle
 - ❑ 1^{er} résultat rapide
 - ❑ Motivations humaines meilleures
 - ❑ Planification aléatoire
 - ❑ Projet « jamais fini »

Conclusion

THE WATERFALL PROCESS



*'This project has got so big,
I'm not sure I'll be able to deliver it!'*

THE AGILE PROCESS



*'It's so much better delivering this
project in bite-sized sections'*

<https://blog.ganttpro.com/en/waterfall-vs-agile-with-advantages-and-disadvantages/>

THE AGILE SCRUM PROCESS





Gestion des risques

Gestion des risques

❑ **Les projets comportent des risques**

▪ 5 classes de risques

- ❑ Organisationnel
- ❑ Technique
- ❑ Financier
- ❑ Humain
- ❑ Juridique

➤ Il est indispensable de les gérer

Exemple de risques très fréquents

- ❑ **mauvaise expression du besoin,**
- ❑ **défaillance ou indisponibilité d'une ressource,**
- ❑ **dérapage financier et calendaire,**
- ❑ **spécification non tenue.**

Gestion des risques

□ 5 étapes dans la gestion des risques:

- L'identification et la caractérisation des risques
 - Quel(s) est/sont les **points faible(s)** de mon projet?
- L'évaluation et la hiérarchisation des risques
- Le traitement des risques.
- Le suivi et le contrôle des risques.
- La capitalisation et la documentation des risques.



L'identification et la caractérisation des risques

- ❑ **Quel(s) est/sont les points faible(s) de mon projet?**
 - Utilisation de questionnaire
 - ❑ Exemple : http://www.cvr-it.com/Samples/XRisk_Assessment_Questionnaire_Template.pdf

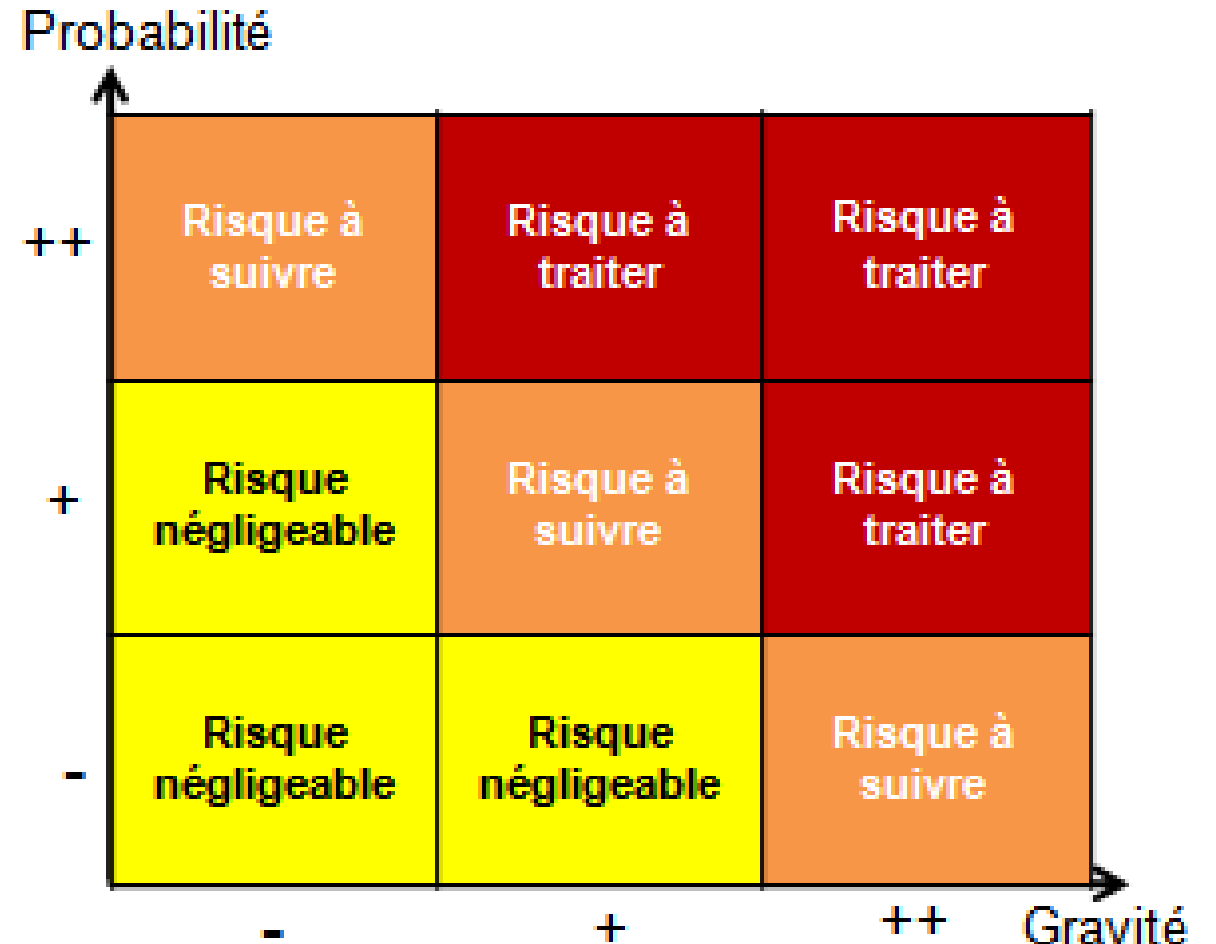
Risques projets



L'évaluation et la hiérarchisation des risques

□ Parmi les risques identifiés,

- Les **qualifier** en termes de
 - **Probabilité** d'apparition
 - **Gravité** sur les objectifs du projet.
- Les **hiérarchiser**



N°	Risques & Facteur de risques	Probabilité	Gravité	Déterminabilité	Criticité	Impacts	Nature du risque	Action de réduction des risques
1	CONTEXTE							
	Facteur de risque : Incertitude sur la réglementation Risque : Ouvrage réalisé non conforme dans un avenir proche	2	2	1	1	Risque coûts supplémentaires.	Coût	
	Facteur de risque : Le début des travaux est conditionné par une tâche qui est du ressort d'un autre projet Risque : Retard du projet	3	3	1	2	Un projet prioritaire en cours de réalisation a pris du retard Si le retard dépasse un mois, les projets risquent de se chevaucher	Coût	
	Facteur de risque : Le début des travaux est conditionné par une livraison Risque : Retard du projet	3	3	2	3	Pressions et pénalités de retard fortes sur le fournisseur	Délai	
	Facteur de risque : L'expérience montre que les sondages du sous-sol sont insuffisants pour garantir la qualité du sous-sol Risque : Découverte d'un sol pollué	2	4	1	2	Allongement de la durée des travaux	Délai	
	Facteur de risque : Les plans du sous-sol sont incomplets Risque : Découverte d'une canalisation de gaz ou d'un fourreau électrique en service.	1	3	1	1	Allongement de la durée des travaux.	Délai	
	Facteur de risque : zone inondable en cas de forte averse Risque : Subir une inondation durant les travaux	2	4	1	2	Retard des travaux et commande de matériels inutilisables Coût supplémentaire	Délai Coût	

Le traitement des risques

- ❑ **Définir et mettre en œuvre les dispositions appropriées pour les ramener à un niveau acceptable**
 - définir des réponses types et de mettre en œuvre, risque par risque,
 - ❑ Exemple
 - Si Riri est malade
 - alors Loulou qui est aussi compétent pour cette tâche le remplacera
 - Si telle technologie est risquée
 - La tester à l'aide de prototype sur les fonctions les plus importantes et les plus complexes
 - Chercher une technologie de remplacement

Le suivi et la documentation des risques

□ **Le suivi et le contrôle des risques**

- Un risque évolue dans le temps
 - Loulou démissionne
 - Riri devient indispensable car il est le seul à maîtriser telle technologie qui a été retenue
 - → former Fifi à cette technologie devient urgent

□ **La capitalisation et la documentation des risques**

- portfolio des stratégies pour remédier à des classes de risques

Gestion des risques

- ❑ **Ne vous mettez pas dans des situations ingérables.**
 - Former
 - Tester
 - Suivre le planning
 - Prendre des marges
 - → faire diminuer le risque

