

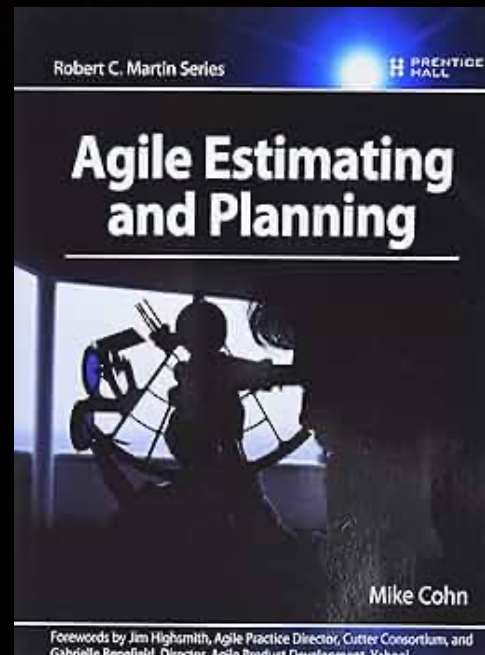
A wide-angle photograph of a long, straight asphalt road stretching towards the horizon. The road is flanked by dry, grassy fields and a few yellow posts. In the distance, a small car is visible on the road. The sky is a mix of dark and light tones, suggesting a sunset or sunrise. The overall mood is contemplative and forward-looking.

Agile

Estimation et Planification

Objectif du cours

Comprendre les notions de planification et d'estimation dans une approche Agile



Lien planification/estimation

« Lorsqu'on planifie on estime l'effort nécessaire »

**Pourquoi
planifier ?**



Question :

A quelle question essayons-nous de répondre lorsqu'on planifie ?

« Quand cela sera-t-il fini ? »

Affirmation

La philosophie Agile s'inscrit dans un cadre de développement à forte incertitude.
Donc la planification est difficile et souvent fausse

Question :

Néanmoins planifier reste essentiel, pourquoi ?

- Réduire les risques
- Réduire l'incertitude
- Permettre la prise de meilleures décisions
- Porteur d'informations
- Le client a le droit de savoir (cf : droits des clients)

Risques et incertitudes

Question :

Quel est le principal risque d'un projet informatique?

- Développer la mauvaise chose

Question :

Comment une planification/estimation Agile réduit se risque ?

- Les décisions d'investissement, de planification, de features sont périodiquement réévaluées (incrément + itération)

Prise de décisions

Question :

Comment planifier/estimer permet de prendre de meilleures décisions ?

- Un projet à 1m ou 4m ne sera pas étudié de la même façon
 - Réévaluer les besoins
 - Sera-t-il prêt à temp pour ne pas louper la « market windows »
- Beaucoup de décisions correspondent à un compromis entre le temps et le coût

Porteur d'informations

Question :

Que communique un plan ?

- Un plan ne garantit pas un ensemble de feature à une date exact et un coût spécifié
- Cependant, il communique une base d'attentes
 - Puis permet de convaincre sur la date, le coût les attentes

**C'est quoi un
bon plan Agile ?**



« A good plan is one that is sufficiently reliable that it can be used as the basis for making decisions about the product and the project.

Agile planning is **focused more on the planning than on the creation of a plan**, encourages change, results in plans that are easily changed, and is spread throughout the project. »

- On change le plan car nous avons appris/découvert quelque chose oublié au départ
- Ces découvertes affectent notre plan
- Nous avons donc besoin de rendre notre plan facilement modifiable

Un plan Agile est donc un plan facilement modifiable

**Pourquoi les
plans échouent ?**



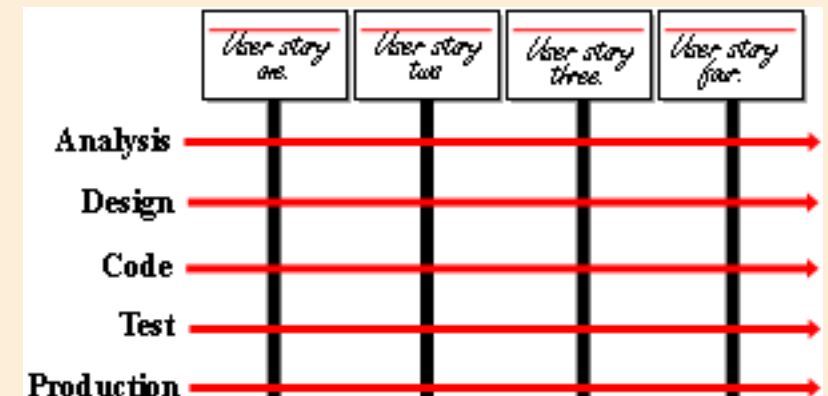
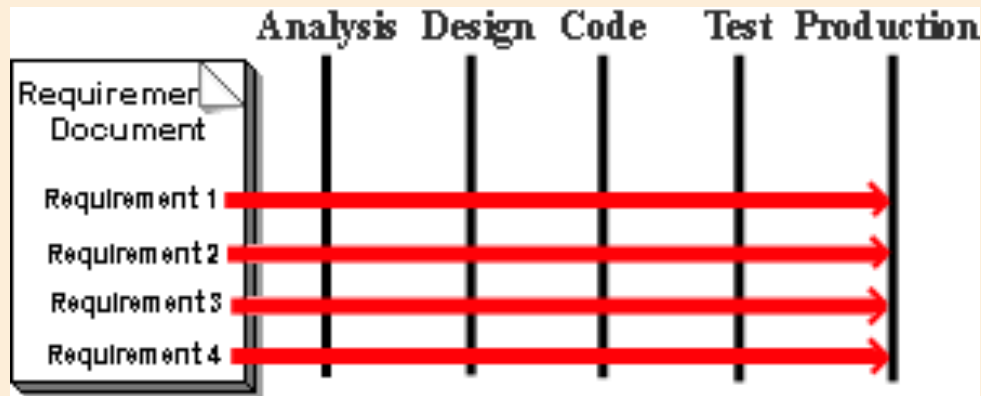
Question :

Pourquoi la planification traditionnelle (i.g. non Agile) échoue

- Planification par activités au lieu que par fonctionnalités
 - I.g. dépendances entre activités
- Multi-tâches
- Fonctionnalités ne sont pas développées par priorité
- Ignore l'incertitude
- Les estimations deviennent des engagements

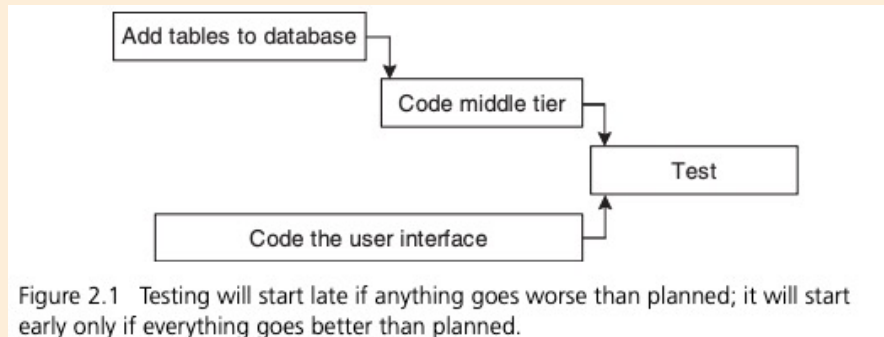
Planification par activités au lieu que par fonctionnalités

- Pour le client, ce sont les **fonctionnalités** qui comptent et la planification doit donc se concentrer sur ces **fonctionnalité**.



Planification par activités au lieu que par fonctionnalités

- De nombreuses tâches/activités dépendent les unes des autres de manière séquentielle. Si l'une d'entre elles est retardée, il est probable que la date de livraison « générale » soit affectée.



Multi-tâches

Database	API	User Interface
-----10 days-----	-----10 days-----	-----10 days-----

DB	API	UI	DB	API	UI
---5 days---	---5 days---	---5 days---	---5 days---	---5 days---	---5 days---

- La tâche relative à la base de données n'est achevée que 20 jours plus tard, au lieu des 10 jours prévus à l'origine. Même si la tâche relative à l'API a démarré tôt, elle s'est achevée avec cinq jours de retard

Multi-tâches

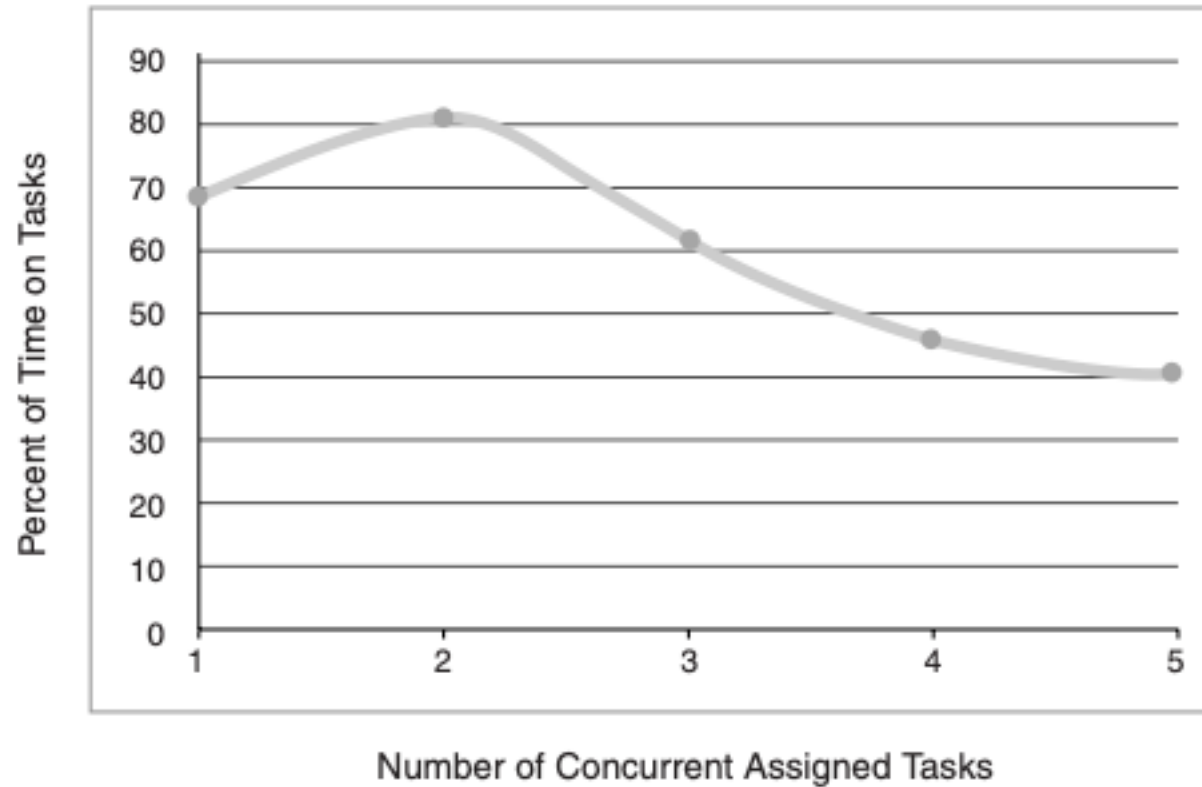


Figure 2.2 Effect of multitasking on productivity.

Fonctionnalités ne sont pas développées par priorité

- La planification traditionnelle suppose à tort que toutes les tâches seront accomplies.
- C'est pour cette raison qu'elle hiérarchise souvent les tâches en fonction de ce qui convient à l'équipe de développement.
- En cas de retard, l'équipe peut être contrainte d'abandonner des fonctionnalités qui ne sont peut-être pas celles qui apportent le moins de valeur à l'entreprise.

Ignore l'incertitude

- On prétend pouvoir assigner une estimation « précise » à un travail imprécis
- L'approche traditionnelle ne considère pas assez l'incertitude
- Le meilleur moyen de la gérer est d'adopter une approche incrémentale avec des itérations courtes

Estimation != Engagement



Question :

On s'engage sur quoi en Agile ?

- n'allons pas nous engager sur un nombre de stories à réaliser mais sur un objectif.
- nous engageons à être transparent avec les parties prenante.

Une planification Agile



Rappel :

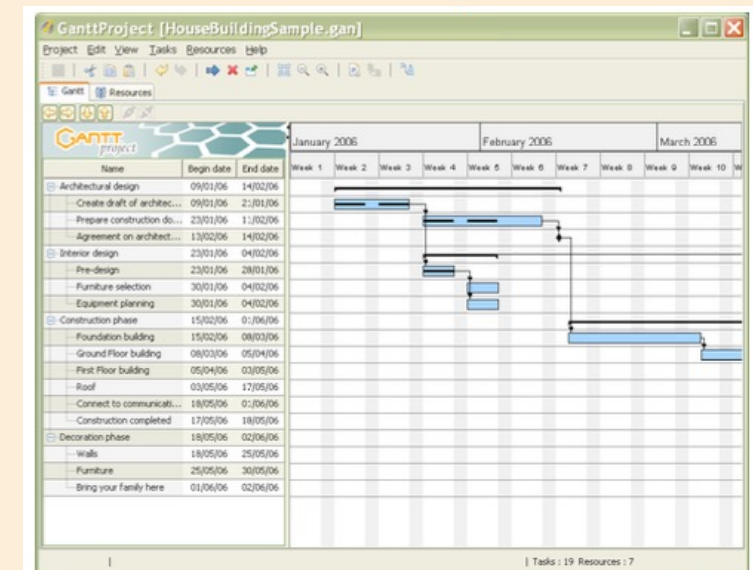
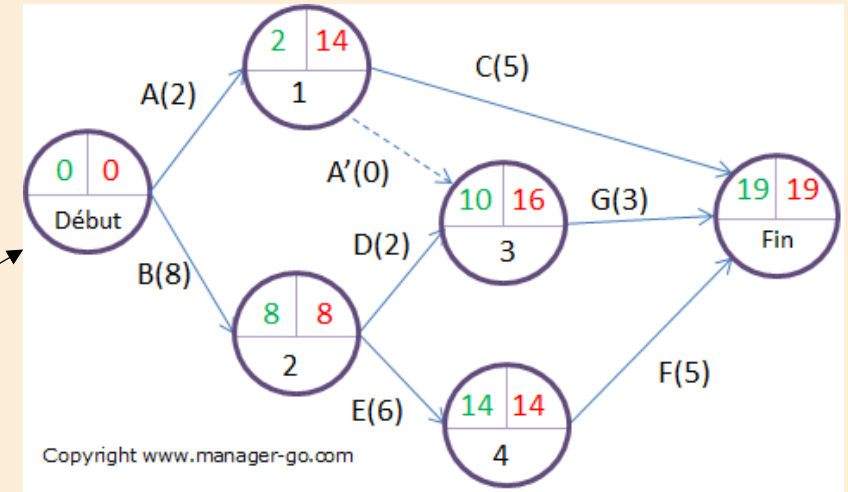
Question :

Comment une équipe Agile travaille ?

- En une seule équipe
- En itération courte
- En délivrant de la valeur à chaque itération
- Focus sur la valeur métier
- Inspecter et Adapter

Approche traditionnelle

1. Estimer le délais
2. Estimer le coûts (TJM)
3. Recenser les activités (lots)
4. Calculer la durées de chaque activités
5. Ordonner les activités
6. Etablir planning



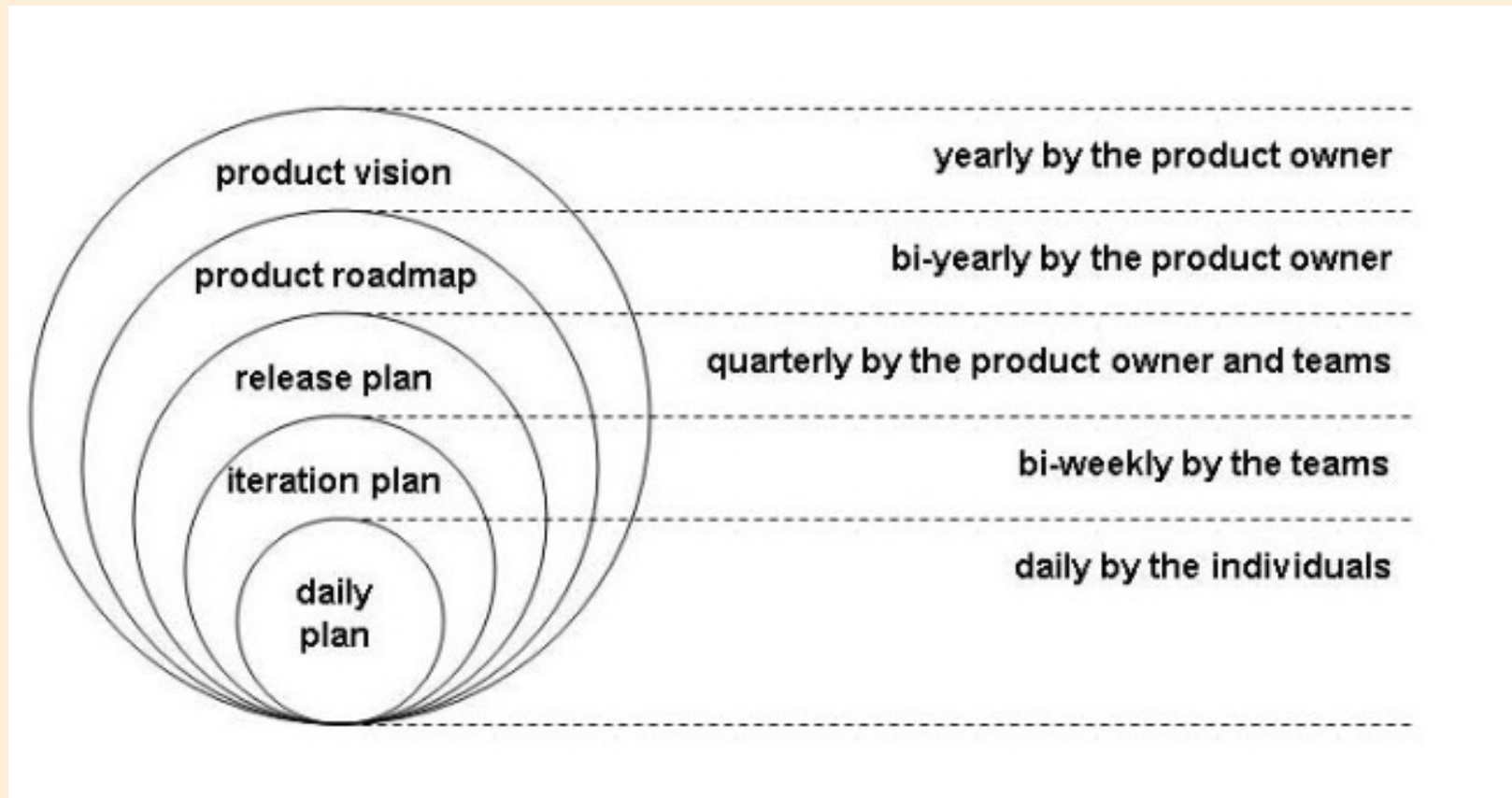
+/- Chef de projet quand il veut
parler avec le client

Approche Agile

Affirmation

Dans une démarche Agile on ne croit pas à la planification détaillée en amont

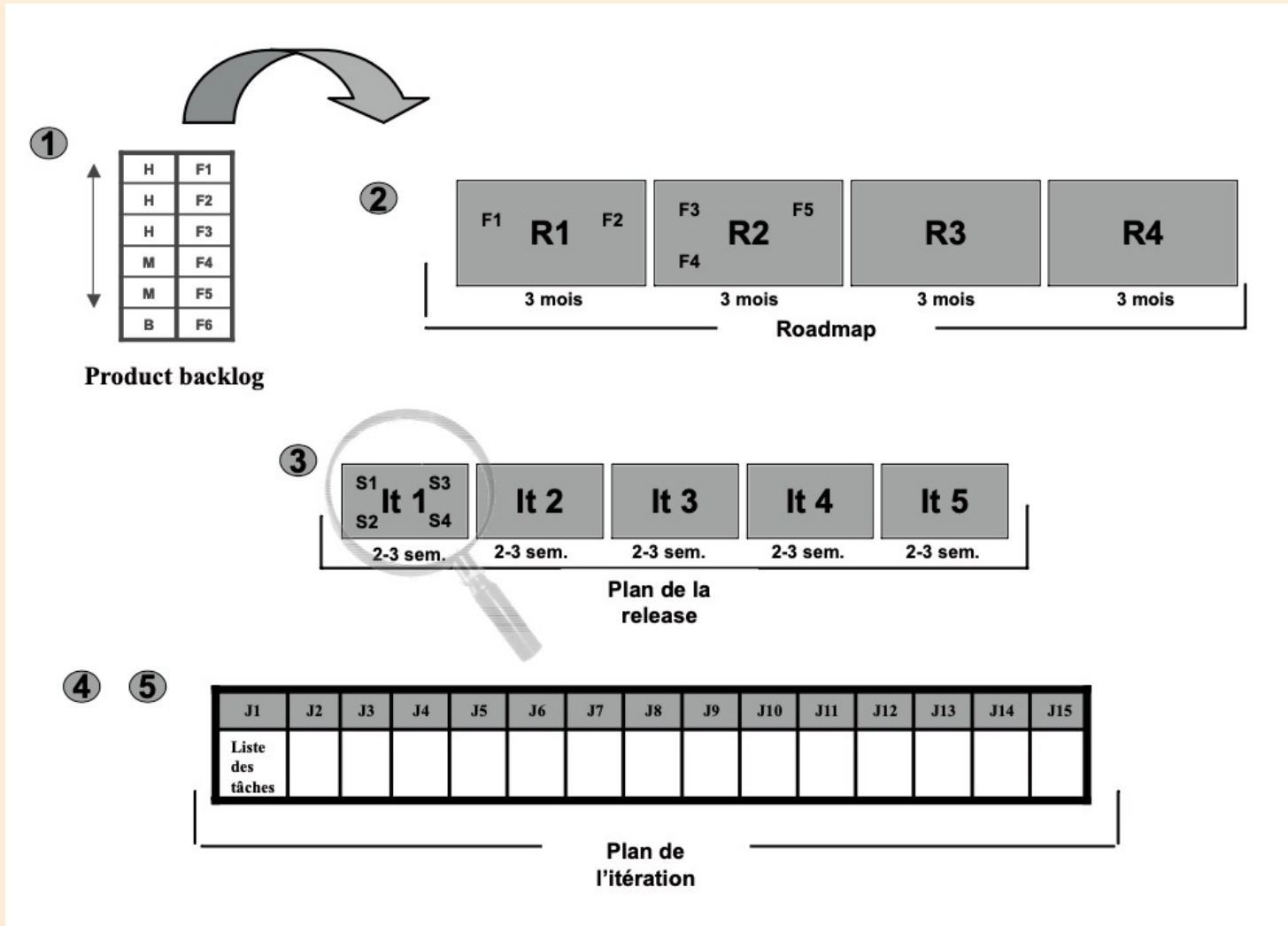
Approche Agile : planning onion



<https://web.archive.org/web/20161014123744/http://agileatlans.org/articles/item/five-levels-of-agile-planning>

Planifier en Agile

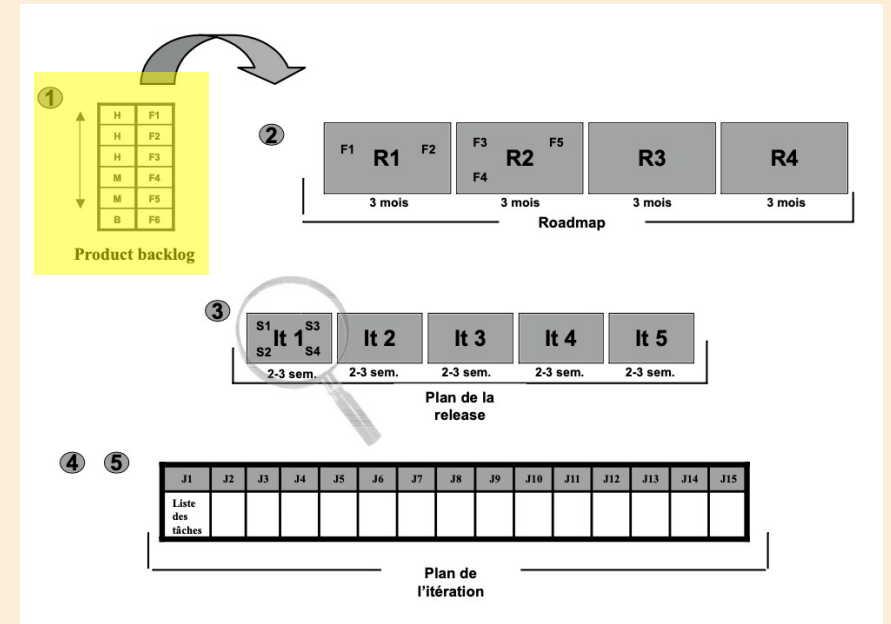
« On cherche à avoir un niveau de détail adapté à notre niveau de visibilité »



1- Vision du produit

Planification globale :

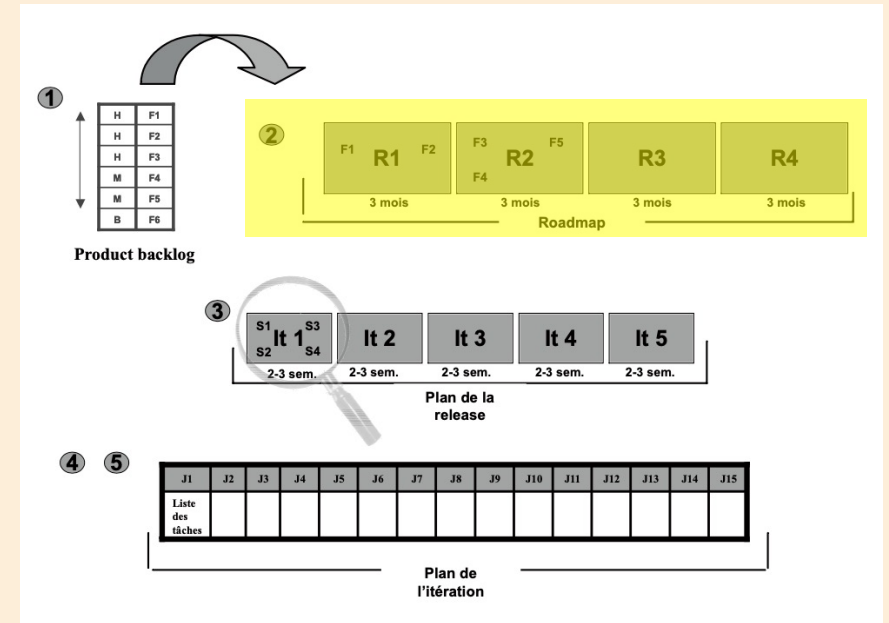
- Etablie via vision produit + PB
 - Avoir un PB priorisé par le client
 - Avoir une ligne d'arrivée (vision produit)



« Evaluer grossièrement l'importance et la taille des fonctionnalités »

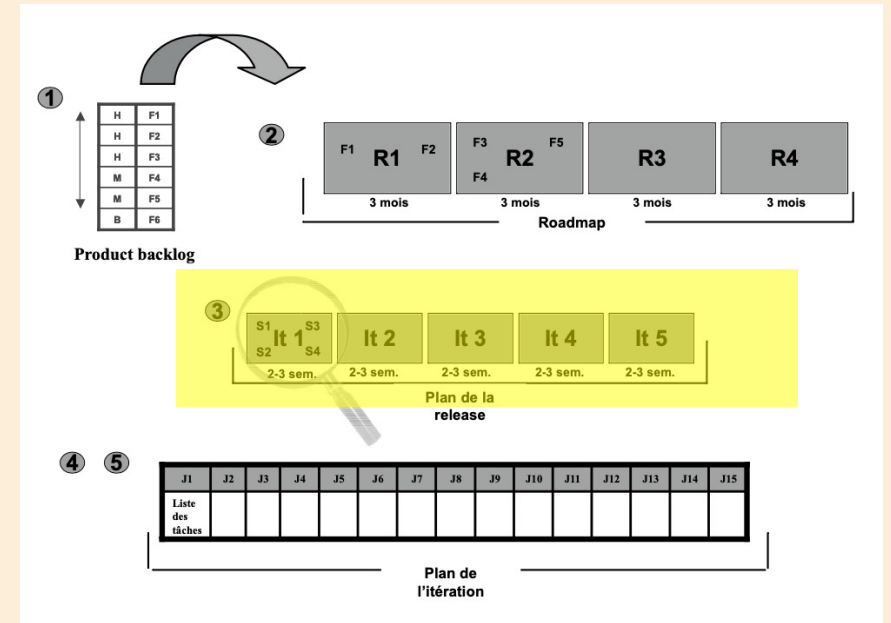
2- Roadmap

- Release livrer tous les n mois
- Les fonctionnalités sont positionnées dans la roadmap à titre indicatif (peut varier en fonction du PB)
- Les dates ne sont pas fixes, des facteurs externes peuvent venir les modifier



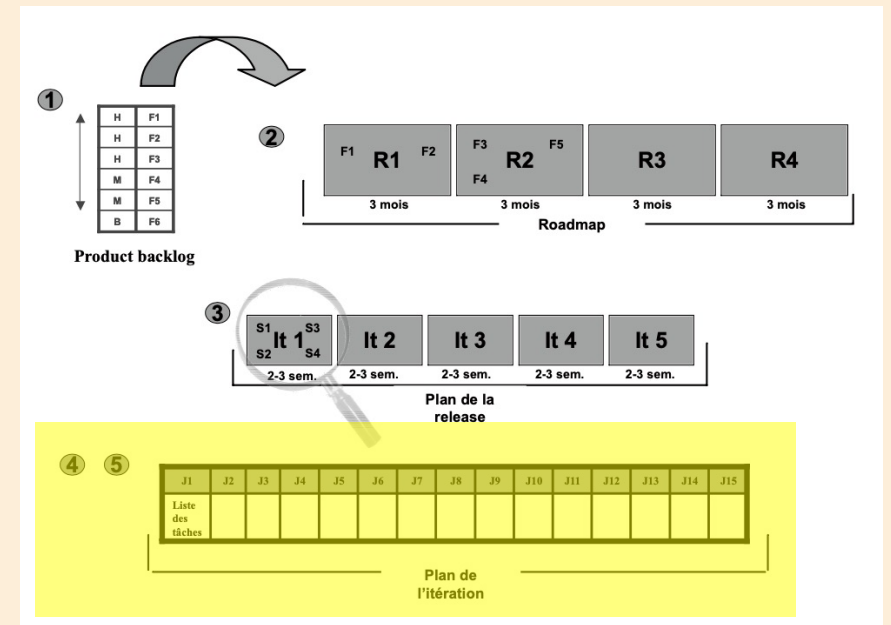
« L'approche Agile prévoit des livraisons régulières »

3- Plan de release



- Release date début et date de fin
 - Permet de définir un périmètre, les ressources
- A l'intérieur d'une release plusieurs itérations
- => On obtient une échéance et les fonctionnalités à dev

4- Plan de l'itération

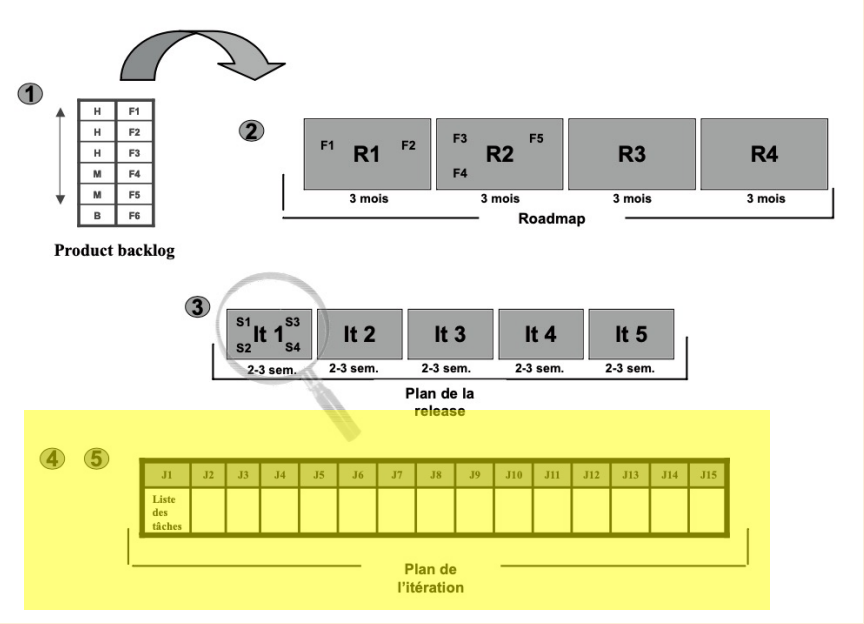


- Date de début et de fin de l'itération
- **SEULE l'itération qui débute est planifiée en détail**
- Lors du Sprint Planning lister les tâches techniques pour réaliser les US



« On dispose d'un Sprint Backlog »

5- Cycle quotidien



Le DAILY

Conséquence

Question :

Donc que vous permet cette approche ?

- **Lorsque votre client apprend quelque chose d'important sur ses besoins ou sur un avantage concurrentiel, vous établissez un nouveau plan.**

Question :

On parle toujours d'itérations, alors pourquoi faire des itérations ?

Agile teams work in short, timeboxed iterations that **deliver a working product** by the end of each iteration. The features developed in these iterations are **selected based on the priority** to the business. This ensures that the most important features are developed first. User stories are a common way for agile teams to express user needs. Agile teams **understand that a plan can rapidly become out of date**. Because of this, they adapt their plans as appropriate.

La meilleure façon de gérer l'incertitude est d'itérer

Estimations Agile



Pourquoi estimer ?

- Partager de la compréhension
 - Estimating is about creating a shared understanding of the requirements, and a shared understanding of the solution. [...] It's my opinion that creating shared understanding is the main reason we estimate
 - **L'effort fourni durant l'estimation est utile**
- Priorisation
- + les raisons de « Pourquoi planifier »

3 techniques courantes

1. Opinion de l'expert

2. Analogie

- Comparer une US à une autre US

3. Décomposant

- Découpant une US en plus petit et plus facilement estimable US

Planning Poker

Question :

Pourquoi le Planning Poker peut-être une bonne idée pour estimer ?

- Equipe cross-fonctionnal
 - Donc nous avons un comité d'experts réunis ensemble
- Devoir justifier ses estimations aux pairs (peers)

Quand ré-estimer ?

Story ID	Story	Estimate	
1	As a coach, I can enter the names and demographic information for all swimmers on my team.	3	✓
2	As a coach, I can define practice sessions.	3	✓
3	As a swimmer, I can see all of my times for a specific event.	3	
4	As a swimmer, I can update my demographics information.	3	

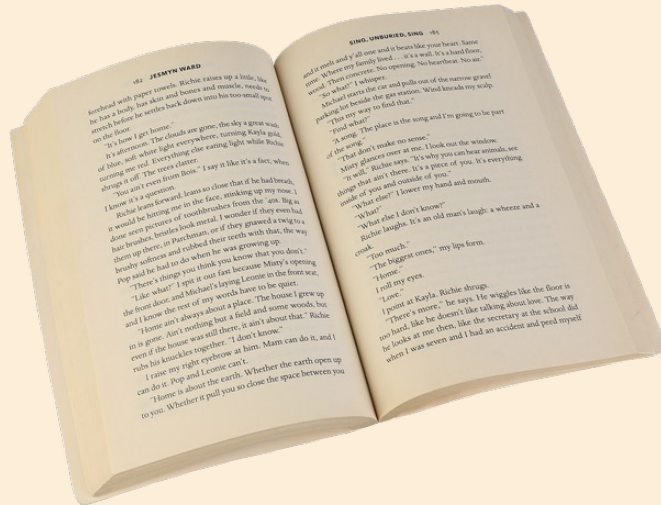
Quand ré-estimer ?

Story ID	Story	Estimate	
1	As a swimmer, I can see a line chart of my times for a particular event.	3	✓
2	As a coach, I can see a line chart showing the progress over the season of all of my swimmers in a particular event.	5	✓
3	As a swimmer, I can see a pie chart showing how many first, second, third, and lower places I've finished in.	3	
4	As a coach, I can see a text report showing each swimmer's best time in each event.	3	
5	As a coach, I can upload meet results from a file exported from the timing system used at the meet.	3	
6	As a coach, I can have the system recommend who should swim in each event subject to restrictions about how many events a swimmer can participate in.	5	✓

Conclusion



Agile estimating and planning succeed because estimates of size and duration are separated.



1. Vous allez estimer la taille du livre e.g. 600 pages
2. Ensuite vous allez estimer votre débit 1page/minute
=> 600 minutes (ou 10h)

Pour arriver à cette estimation de la durée vous avez commencez par estimer la taille

We start by estimating the size of a project's user stories in story points. Because story points are abstract and notional, they are pure estimates of size. We then estimate a rate of progress, which we call velocity.

Our estimates of size and velocity are then combined to arrive at an estimate of duration