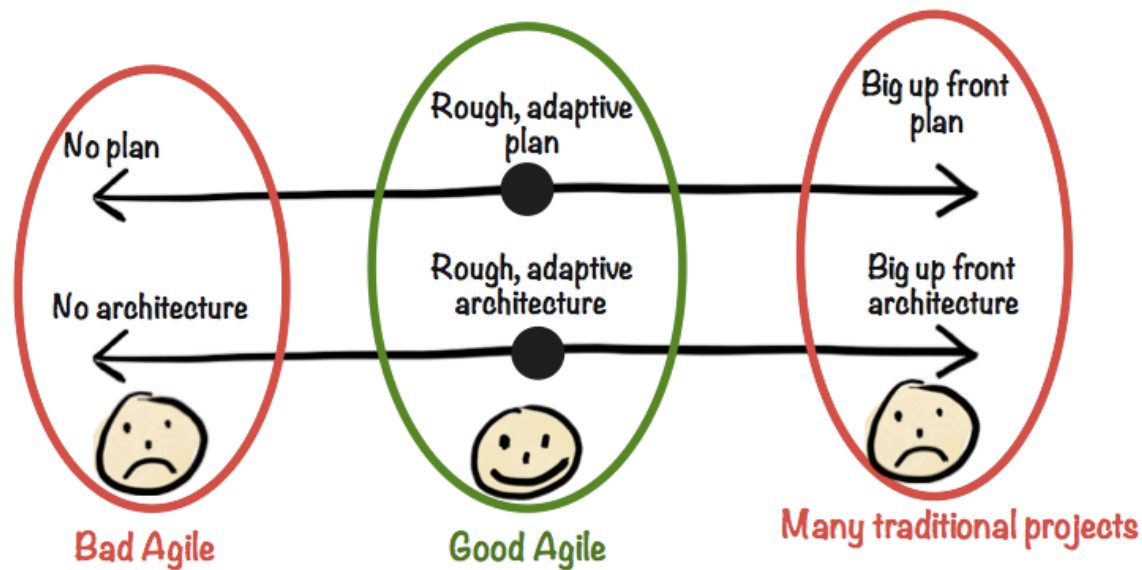


Don't go overboard with Agile!



Source : <http://blog.crisp.se/author/henrikkniberg>

Chap. 2 : Les méthodes agiles

Mots et concepts clés : Qu'est ce que l'agilité ?,
Itération, Sprint, Product Backlog, Product Owner,
User stories, Burndown

I. Le mouvement agile

A. Le manifeste agile

B. Quelques modèles phares

1. RAD
2. Scrum
3. XP



A. Manifeste pour le développement agile de logiciels

Source : <http://agilemanifesto.org/iso/fr/>



Nous découvrons comment mieux développer des logiciels par la pratique et en aidant les autres à le faire.

Ces expériences nous ont amenés à valoriser :

- Les individus et leurs interactions plus que les processus et les outils,
- Les logiciels 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.

Nous reconnaissons la valeur des seconds éléments, mais privilégions les premiers.



Principes sous-jacents au manifeste

Source : <http://agilemanifesto.org/iso/fr/>

Nous suivons ces principes:

- *Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.*
- *Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.*
- *Un logiciel opérationnel est la principale mesure d'avancement.*
- *Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.*
- *Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.*
- *La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.*
- *Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.*
- *Les processus Agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.*
- *Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité.*
- *La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.*
- *Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées.*
- *À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.*

(4)

I. Le mouvement agile



II. Le recueil des besoins agile



III. La planification agile

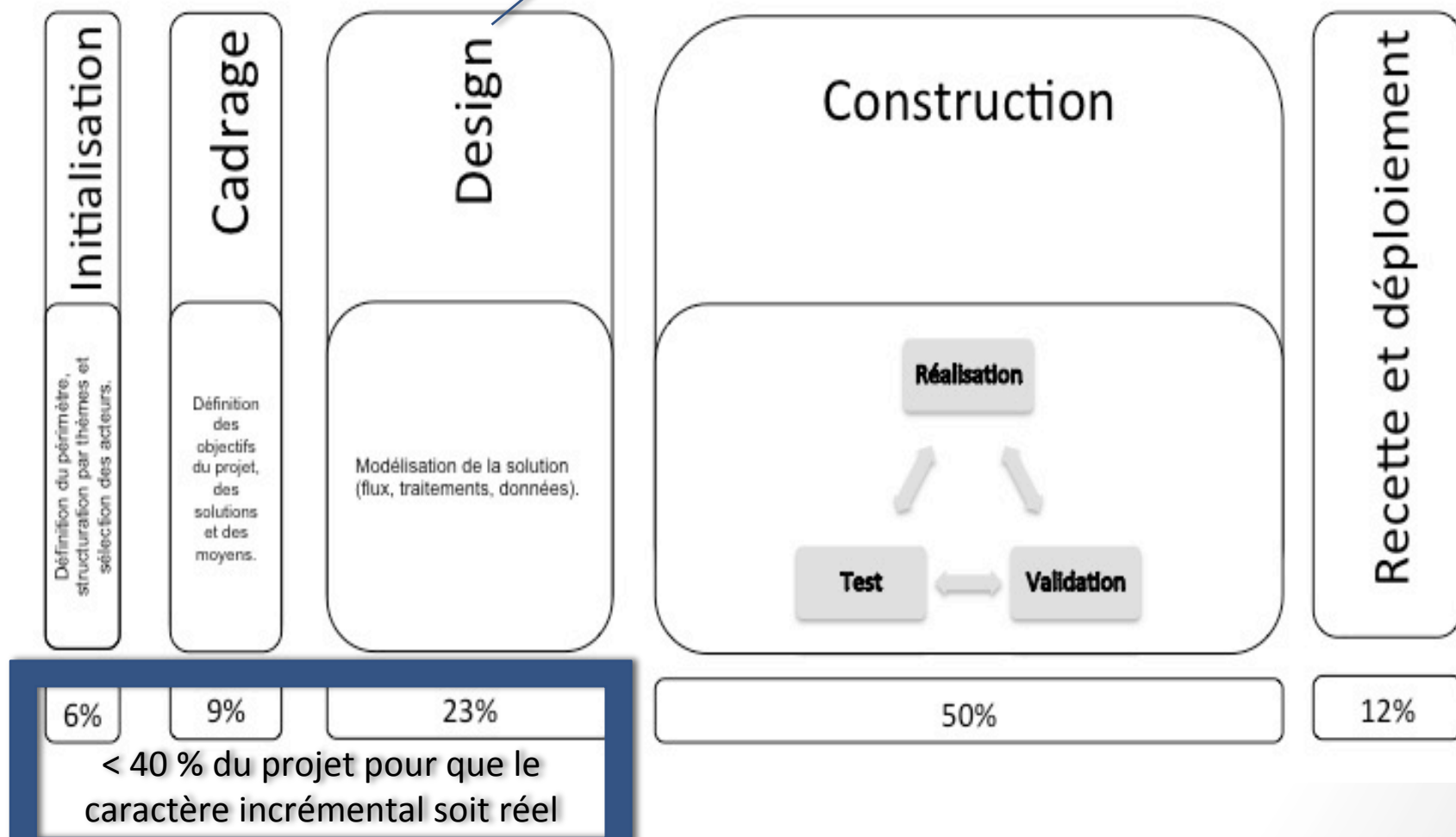


IV. Le pilotage agile

B. Quelques méthodes phare

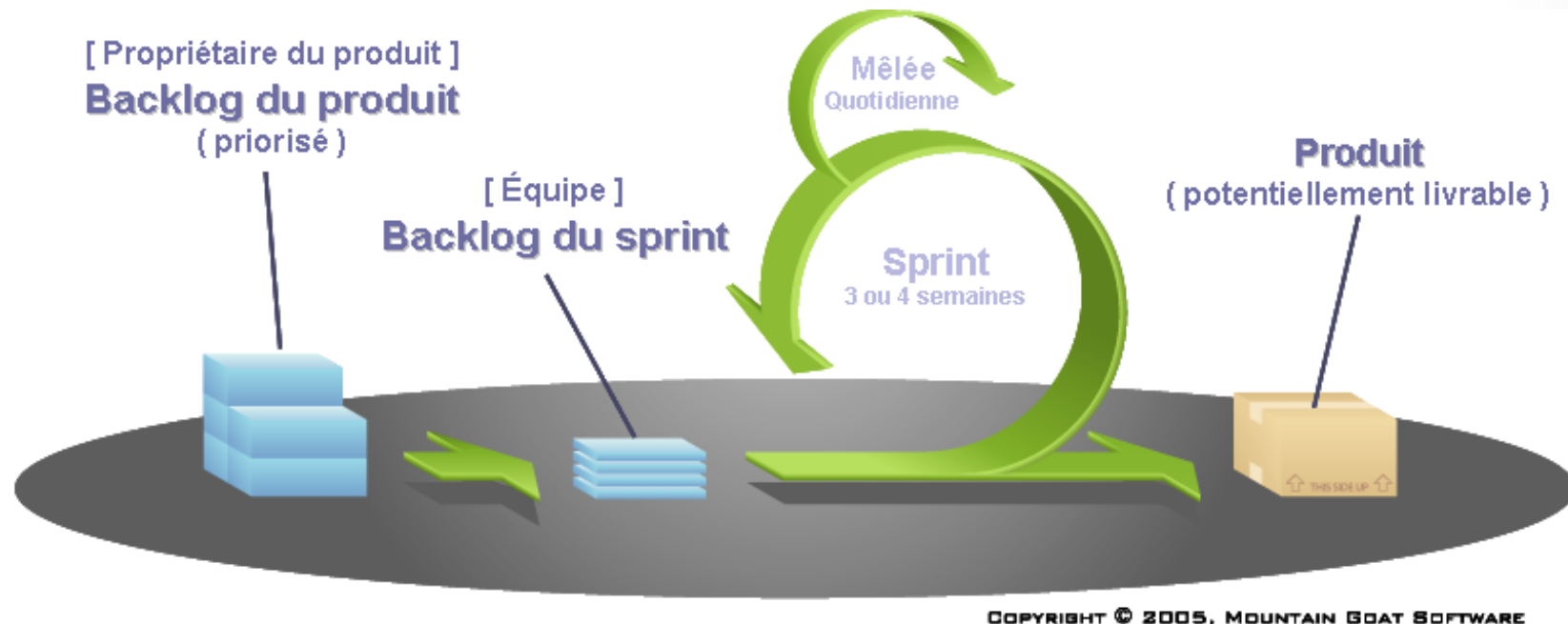
1. RAD :

Réalisé en commissions avec les utilisateurs pour favoriser les interactions avec le client



B. Quelques méthodes phare

2. Scrum :



I. Le mouvement agile



II. Le recueil des besoins agile



III. La planification agile



IV. Le pilotage agile

B. Quelques méthodes phare

3. XP : eXtrem Programming

- Des cycles courts comparables à Scrum
- Des pratiques d'ingénierie :
 - Développement par paire,
 - TDD (développement piloté par les tests),
 - Intégration continue,
 - Revues de code permanentes,
 - Conception au cours du projet (refactoring)
 - ...

Remarque : ces pratiques sont un bon complément à Scrum



II. Le recueil des besoins agile

1. Il faut avant tout partager une vision

(parce que les méthodes classiques font souvent oublier l'objectif principal)

- Un acteur clé le **Product Owner** (ou chef de projet utilisateur)
 - Il communique la vision, décrit les fonctionnalités, connaît les priorités, peut changer d'avis ;
 - Il collabore, est exigeant, soutient l'équipe et donne de la reconnaissance.

[10]

I. Le mouvement agile



II. Le recueil des besoins agile



III. La planification agile



IV. Le pilotage agile

2. Les besoins émergent petit à petit

- Pourquoi ?
 - Le marché est instable...
 - Le client découvre petit à petit l'outil qui se construit...
 - Les choix à opérer font évoluer les priorités...

- Comment ?

Démarche itérative de recueil-hiérarchisation-réalisation



3. La hiérarchisation des besoins

- « Développer les fonctionnalités par ordre de priorité »
 - Critères de priorisation :
 - Bénéfice attendu,
 - Coût de développement,
 - Opportunités d'apprentissage pour l'équipe,
 - Risques.
- ! C'est le product owner (et/ou les utilisateurs) qui priorise, pas l'équipe de développement !



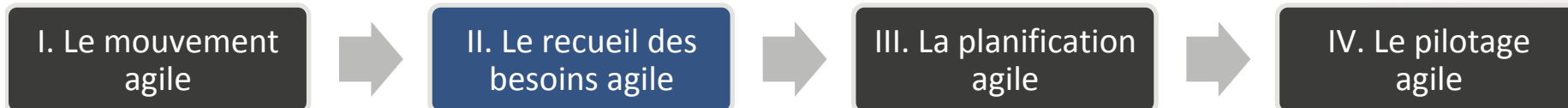
4. Quelques outils de recueil, formalisation et hiérarchisation du besoin

- Analyse [en largeur](#)
 - « Brainstorming »
 - « Benchmark »
- Analyse en profondeur et description détaillée *
 - Pour commencer : Interview ([technique des 9 cases](#)) et « Workshop »
 - Pour affiner avant de développer : Observation de l'utilisateur « in situ »
- Formalisation
 - Cas d'utilisation UML (cf. cours UML)
 - ou [User stories](#)
 - Les Cas d'utilisation ou les User Stories sont synthétisés dans le [product backlog](#)
- Hiérarchisation : [La matrice de hiérarchisation Risques/valeurs](#), la méthode [MoSCoW](#), [La matrice de Kano](#), Les poids relatifs de Wieggers...



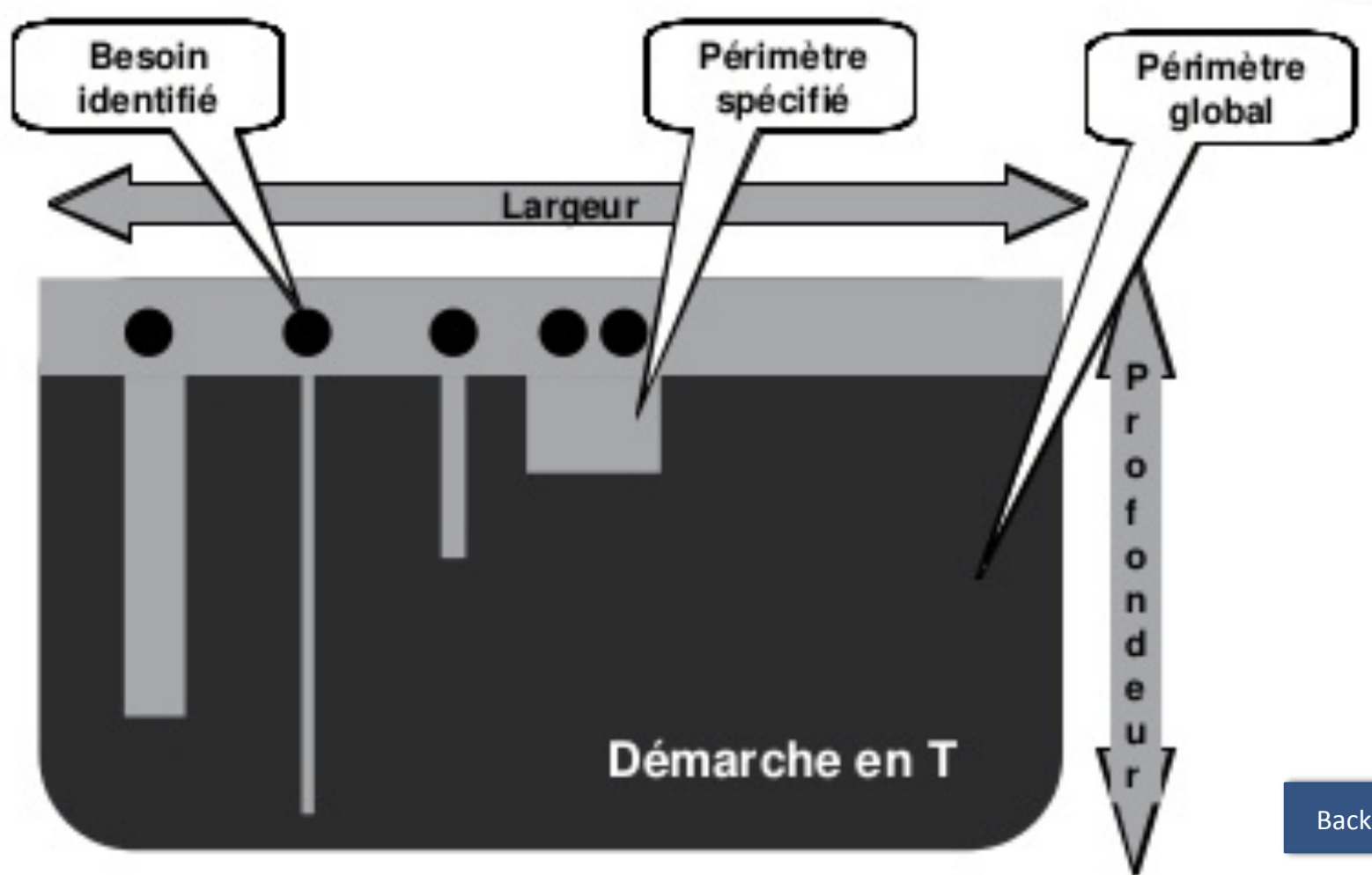
Le recueil des besoins, conclusion

- Le recueil des besoins se fait pas à pas,
- Le changement est accepté,
- Les besoins sont précisés au fil du projet,
- Les besoins sont consignés et priorisés sous forme de **USER STORIES** dans le **PRODUCT BACKLOG**.



Recueil des besoins : démarche en T

Source : gestion de projet agile, Véronique Messenger, Eyrolles 2013



[Back](#)

(15)

I. Le mouvement agile



II. Le recueil des besoins agile



III. La planification agile



IV. Le pilotage agile

Solution de Selling ou technique des 9 cases

[Back](#)

	Quel est le Problème ?	Qui est Impacté ?	Visualiser la solution
Question ouverte « Dites-moi » « Racontez-moi » « Et puis »	1	4	7
Contrôle Combien ? Quand ? Où ?	2	5	8
Confirmation « Si je comprends bien... si oui,... si non ... »	3	6	9

[\[16 \]](#)

I. Le mouvement agile



II. Le recueil des besoins agile



III. La planification agile



IV. Le pilotage agile

User story

- Exigence du système à développer,
- Formulée en 1 ou 2 phrases (par exemple sur un post-it),
- Dans le langage de l'utilisateur.
- Elle doit permettre :
 - d'estimer le coût de développement,
 - de réaliser la fonctionnalité au cours d'une itération.
- Pas de formalisme imposé mais...
 - En tant que <rôle>
 - Je veux que <fonctionnalité>
 - Afin de <bénéfice>

I. Le mouvement agile



II. Le recueil des besoins agile



III. La planification agile



IV. Le pilotage agile

Back

[17]

Exemple de product backlog

(Source : Agile product management with scrum, Roman Pichler)

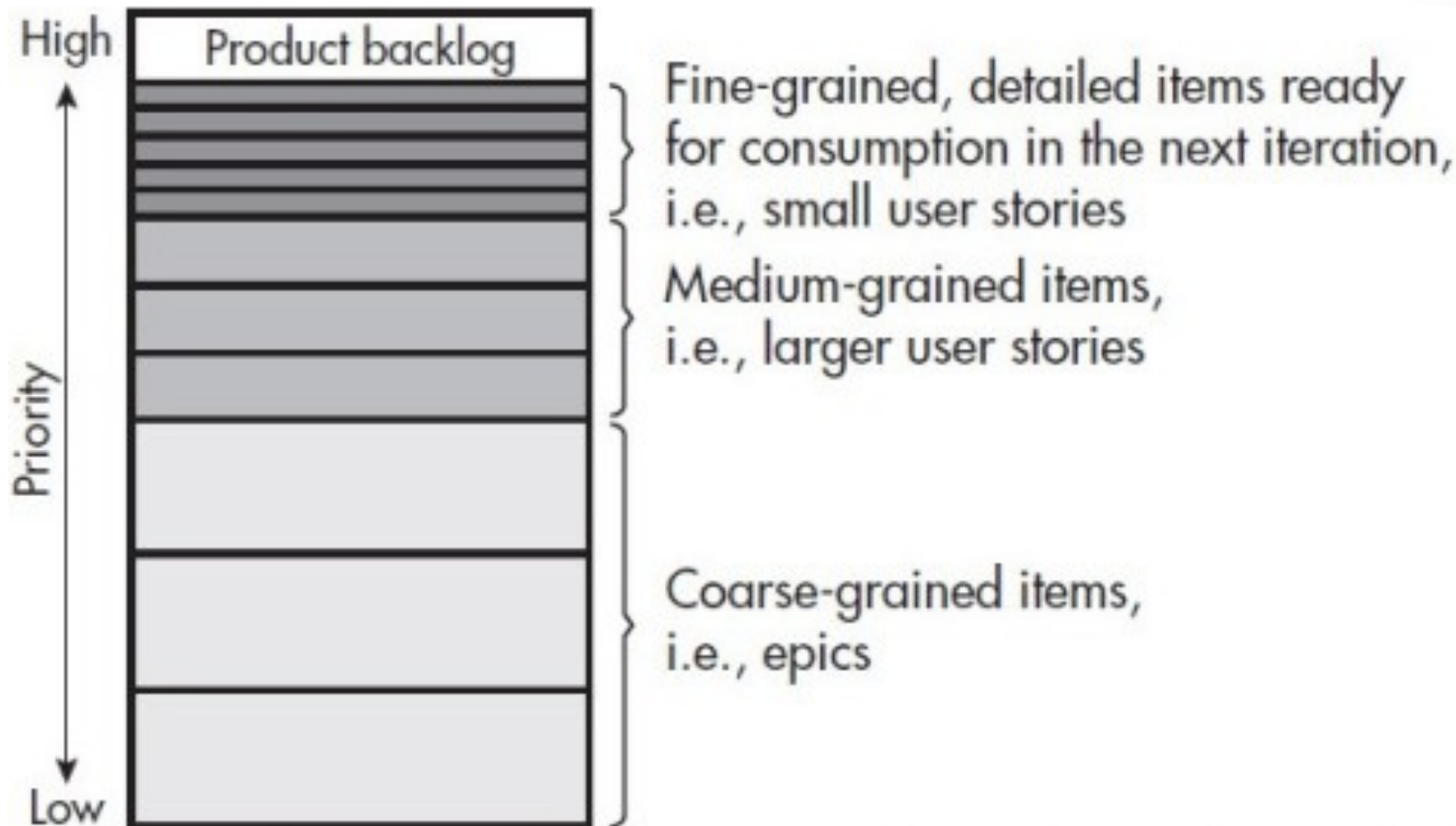


FIGURE 3.1 Product backlog prioritization determines the level of detail

I. Le mouvement agile



II. Le recueil des besoins agile



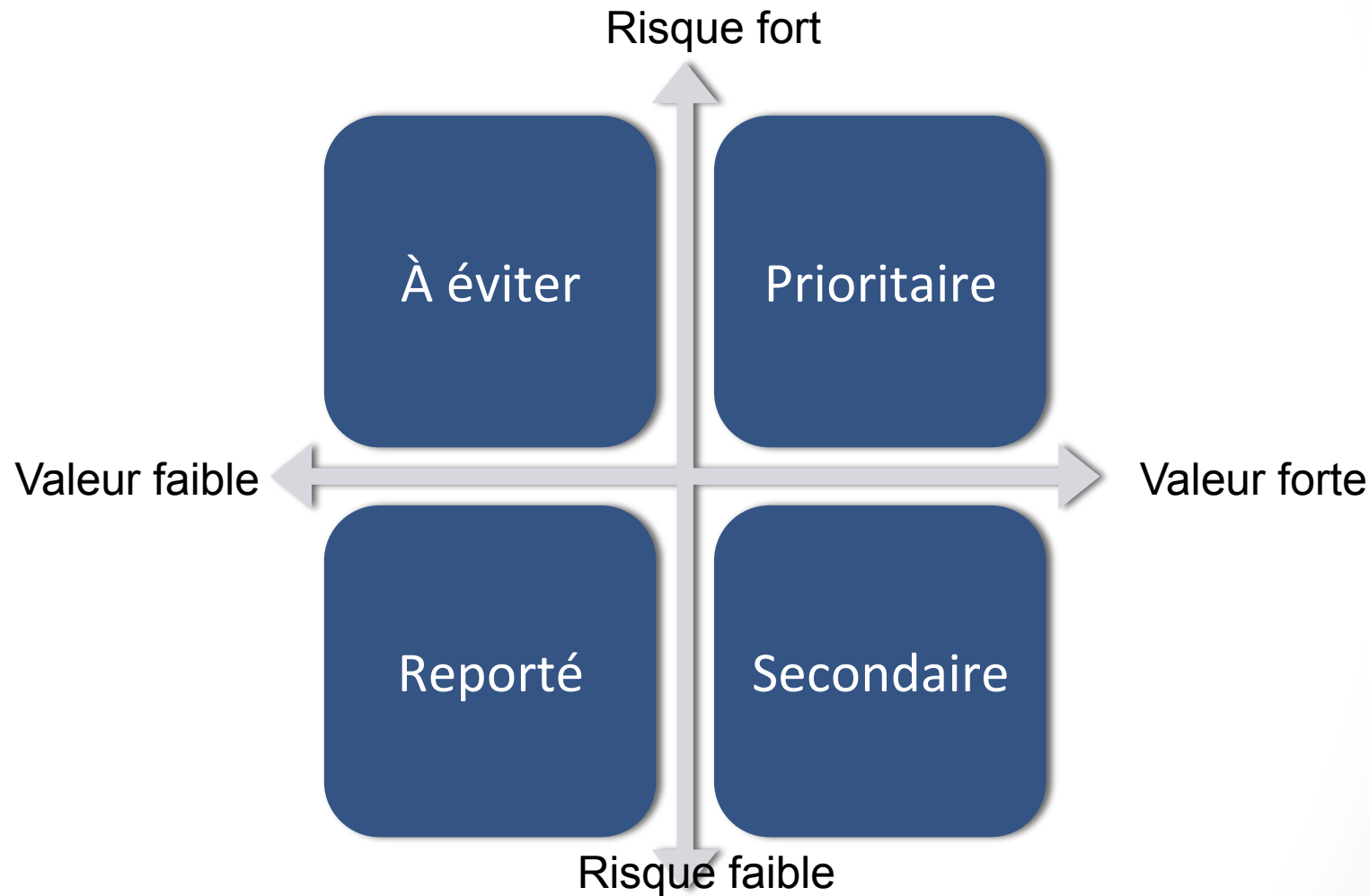
III. La planification agile



IV. Le pilotage agile

Matrice de hiérarchisation en fonction des risques et de la valeur

Source : A. Mike Cohn, Agile estimating and planning, Prentice Hall, 2004



I. Le mouvement agile



II. Le recueil des besoins agile



III. La planification agile



IV. Le pilotage agile

Back

(19)

MoSCoW

- Must have : fonctions indispensables
- Should have : fonctions souhaitables mais non prioritaires
- Could have : fonctions possibles
- Should have but won't have : fonctions éliminées

Back

(20)

I. Le mouvement agile



II. Le recueil des besoins agile



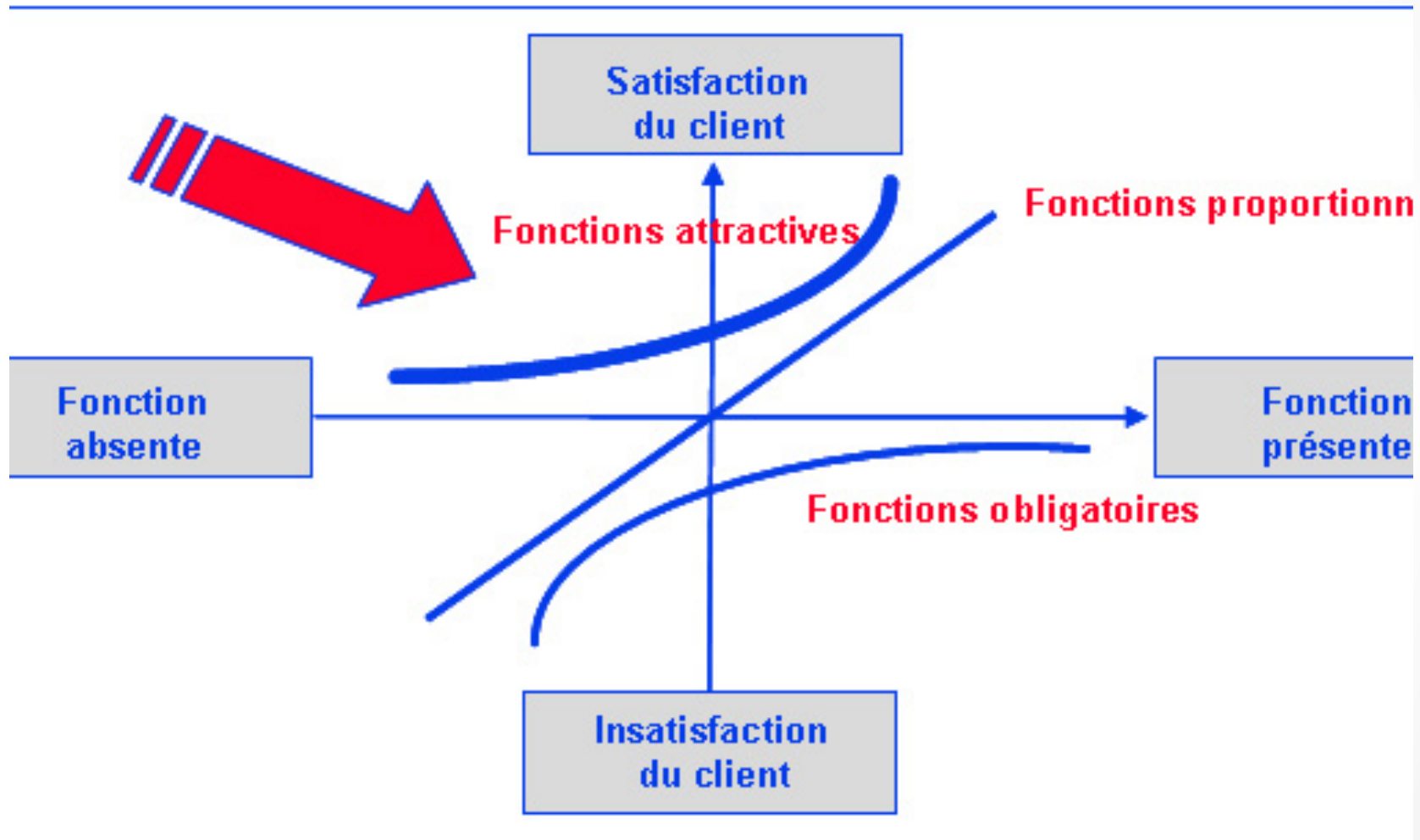
III. La planification agile



IV. Le pilotage agile

Matrice de Kano

Back



(21)

I. Le mouvement agile



II. Le recueil des besoins agile



III. La planification agile

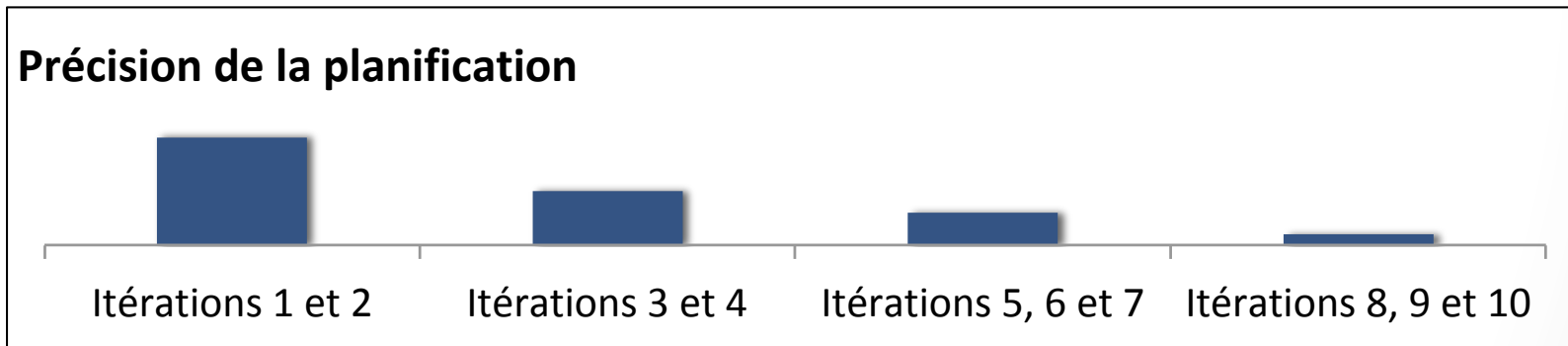


IV. Le pilotage agile

III. La planification agile

① Principe de base : Une planification précise à CT, grossière et incertaine à LT

- Les besoins changent et évoluent...
- Inutile de planifier à LT (au delà d'un mois l'incertitude est trop forte), on avance pas à pas, **on ne planifie avec précision que l'étape (l'itération) suivante.**



+ **À la fin de chaque étape on capitalise l'expérience** avant de planifier l'étape suivante.

(24)

I. Le mouvement agile



II. Le recueil des besoins agile



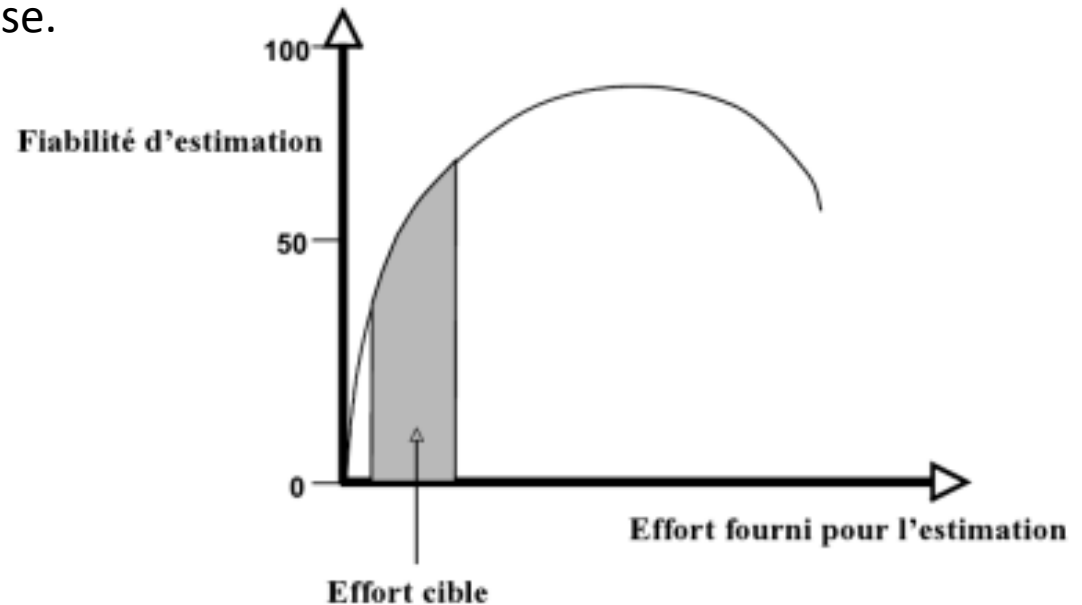
III. La planification agile



IV. Le pilotage agile

② Évaluation de l'enveloppe globale à partir du product backlog

- Pour démarrer le projet et contractualiser, sans gaspiller du temps à tenter de réaliser une estimation précise.

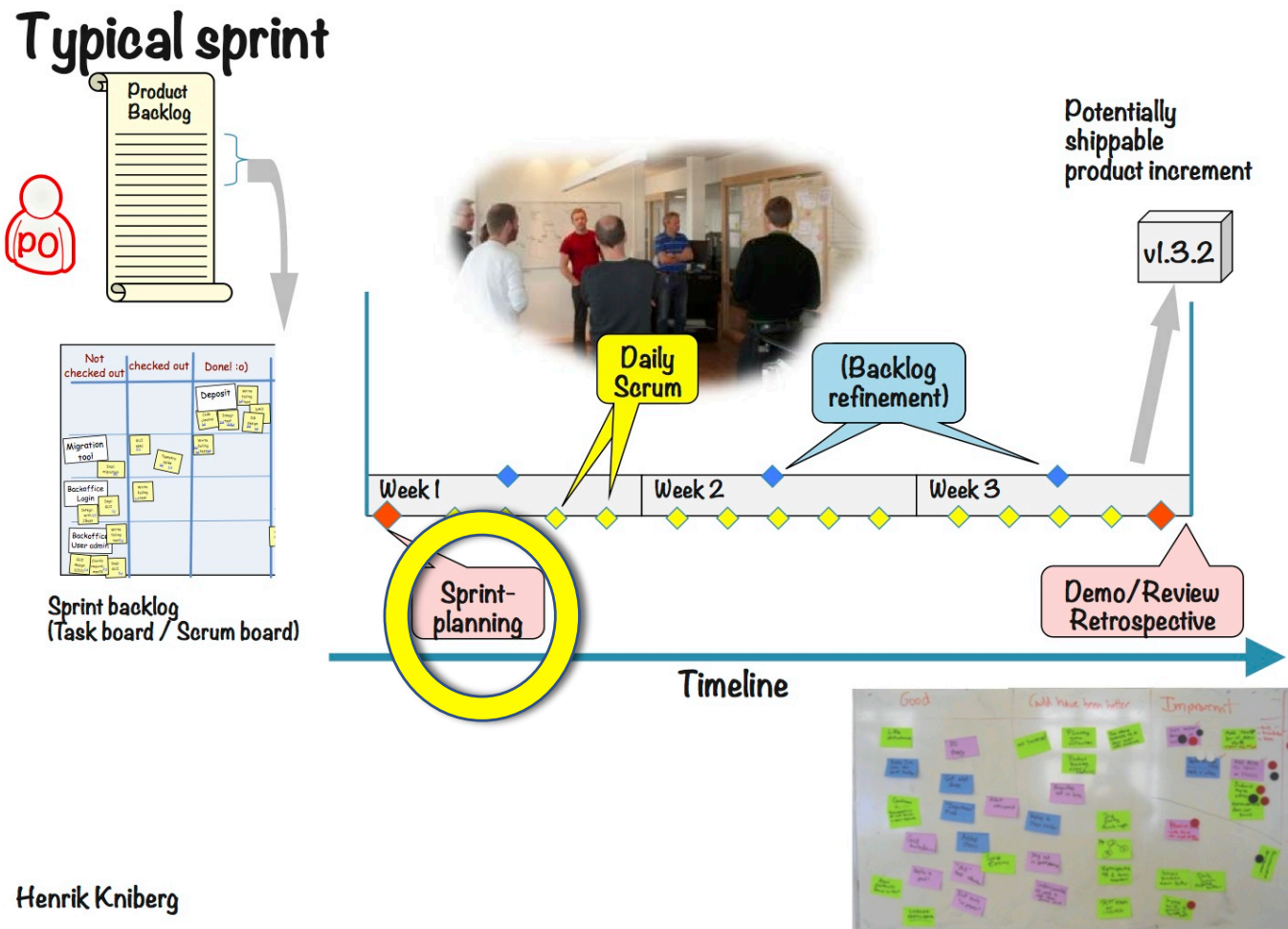


- Deux méthodes préconisées : La méthode Delphi (par analogie), La méthode des story points
- ! Le client doit savoir que cette enveloppe lui garantit un nombre d'itérations... mais pas la liste des fonctionnalités qu'il a fournies !

(27)

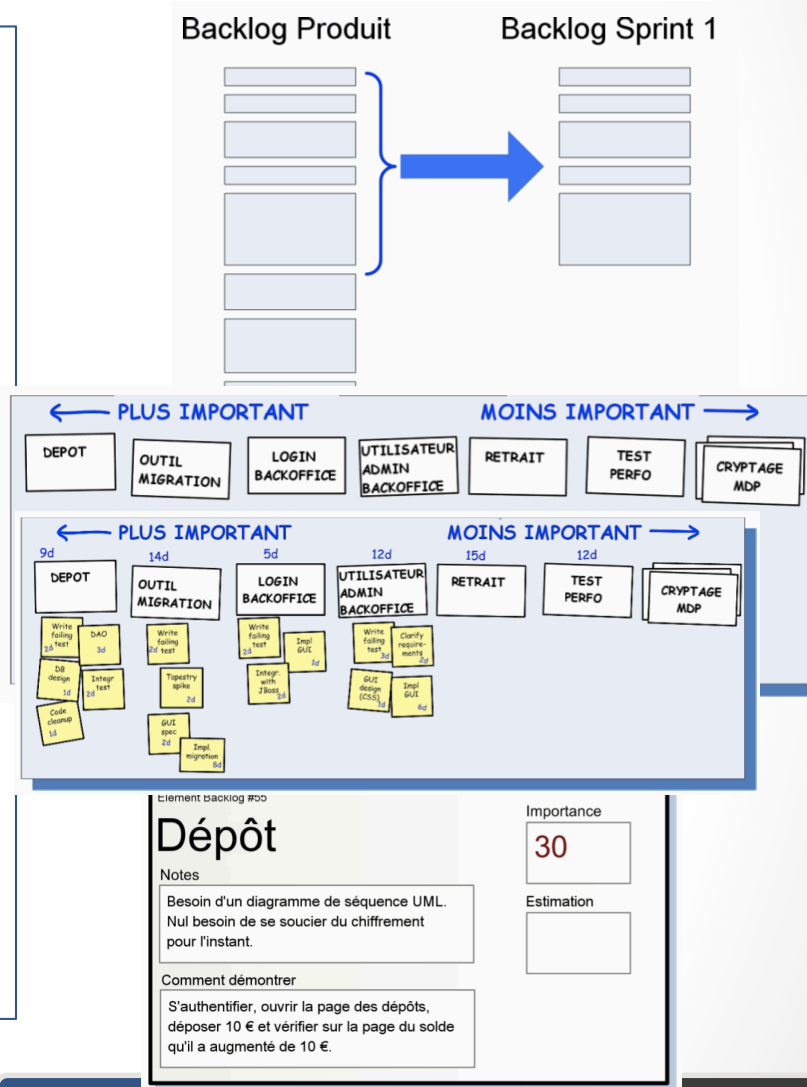


Les temps forts d'une itération scrum

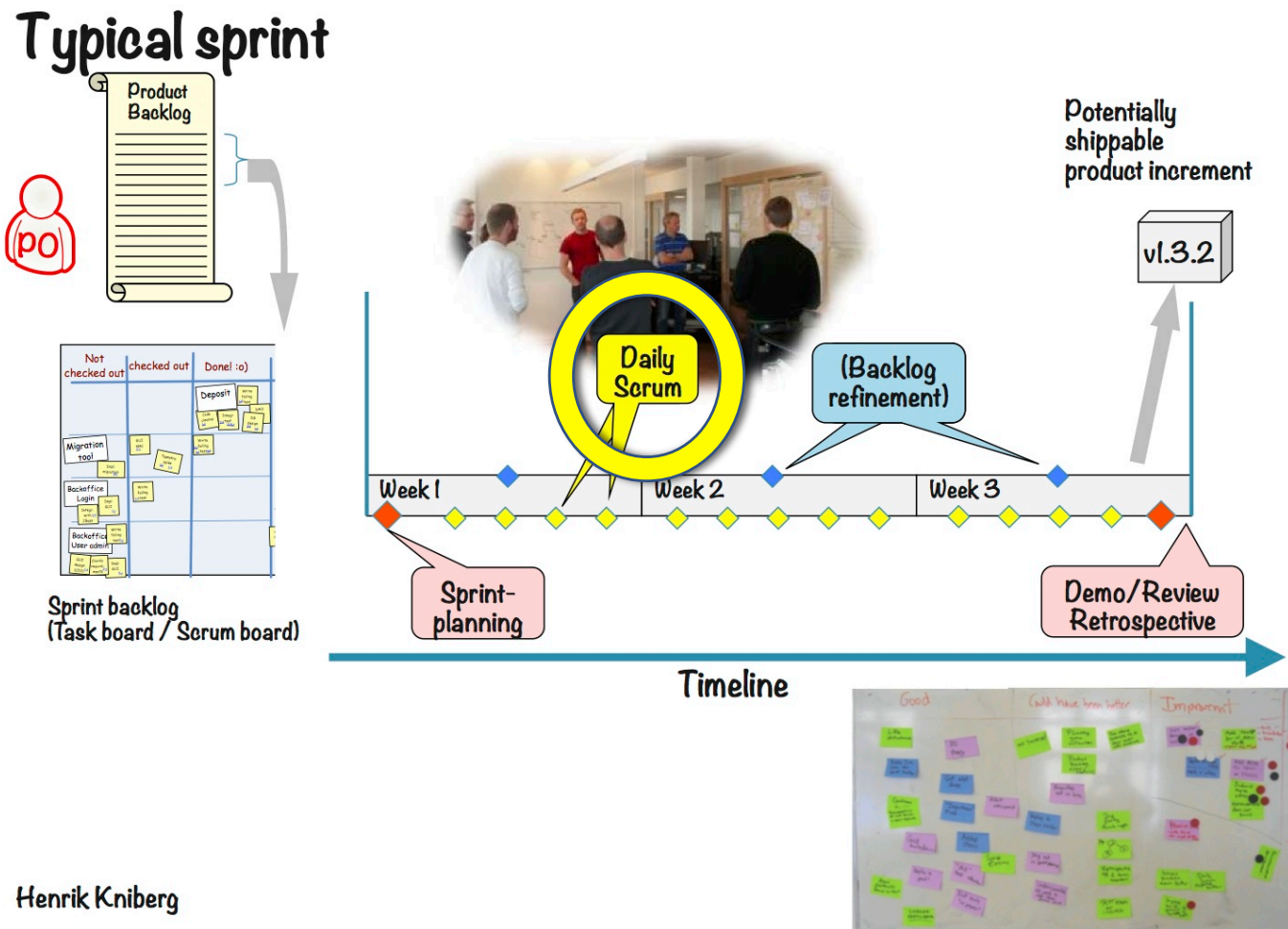


③ La planification de l'itération : le sprint planning meeting

- Réunion de l'équipe de développement avec le product owner (souvent avec des post-it et un tableau).
- Objectifs de la réunion :
 - priorisation et choix,
 - clarification et démonstration,
 - Découpage des users stories en activités, estimation du temps mais pas d'affectation.
- Fin de la réunion : lieu et heure de la mêlée quotidienne.



Les temps forts d'une itération scrum



Henrik Kniberg

[30]

I. Le mouvement agile



II. Le recueil des besoins agile



III. La planification agile



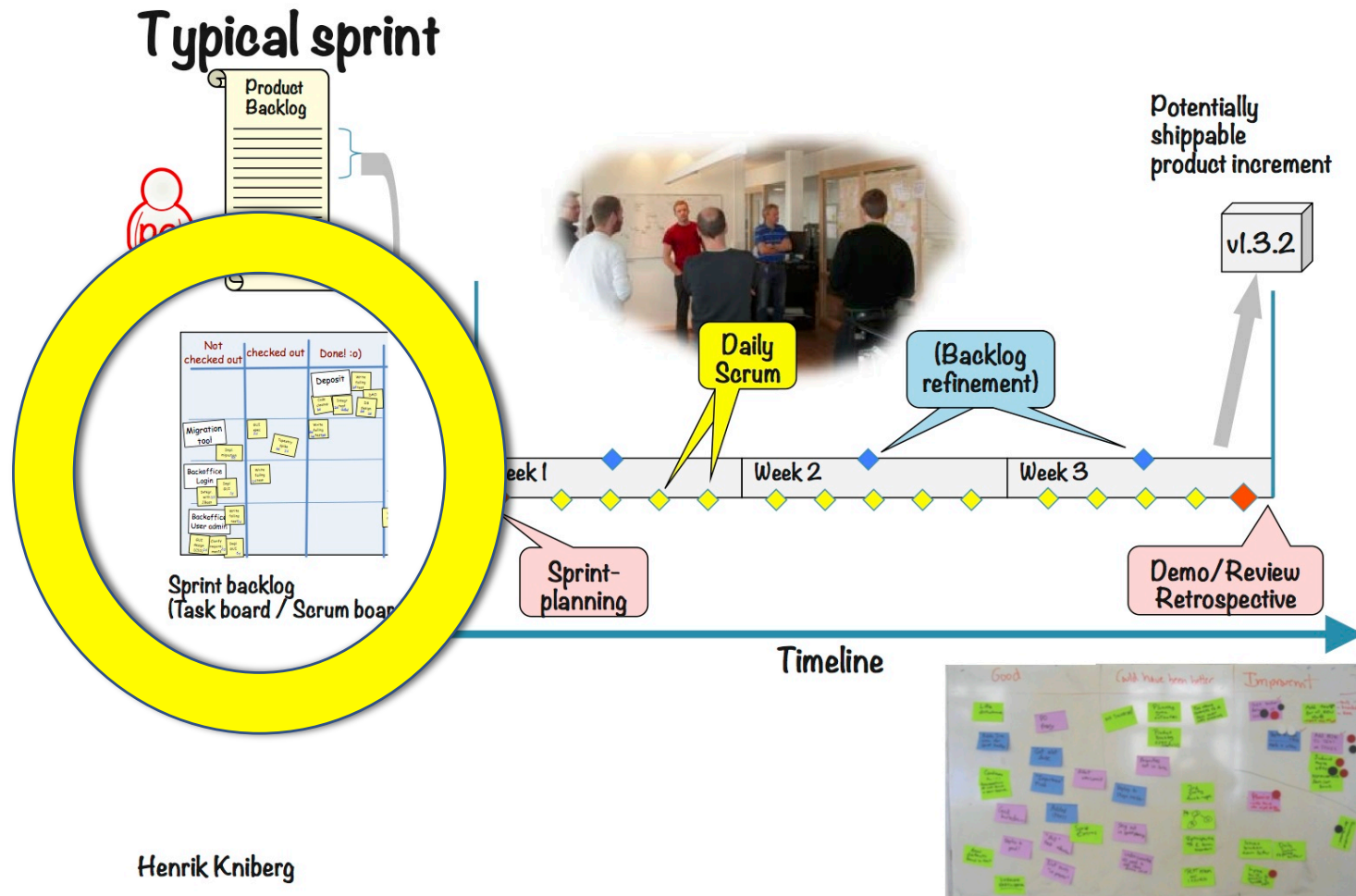
IV. Le pilotage agile

④ Le cycle quotidien : outil de planification et de pilotage

- 15 minutes en début de journée : « mêlée » ou « daily stand-up meeting »
- Chaque membre de l'équipe précise :
 - Ce qui a été fait la veille,
 - Ce qui va être fait aujourd'hui,
 - Les difficultés rencontrées.



Les temps forts d'une itération scrum



[32]

I. Le mouvement agile



II. Le recueil des besoins agile



III. La planification agile



IV. Le pilotage agile

IV. Le pilotage d'un projet agile

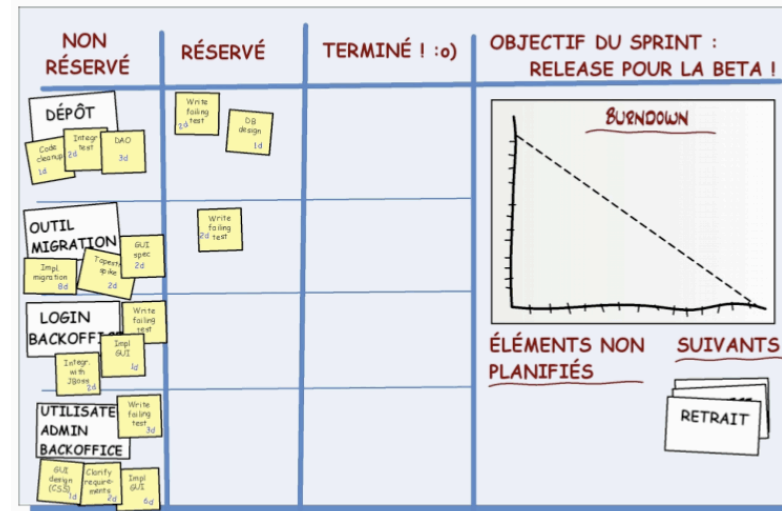
- S'appuie principalement :
 - Sur une relation très étroite avec le client (Product owner)
 - Et sur deux outils le Backlog du sprint et le Burndown
- Pendant l'itération :
 - La réunion quotidienne,
 - Le backlog,
 - Le Burndown :
 - courbe idéale de réalisation des user stories comparée à la courbe réelle
 - sera utilisé pendant l'itération par le scrum master (en collaboration avec les développeurs) pour piloter la réalisation.



Exemples de backlog

(source : <http://wiki.ayeba.fr/Le+nouveau+backlog+du+nouveau+sprint>)

PBI	Todo	In Progress	Done
			 
	   		
	 		
	  		



I. Le mouvement agile



II. Le recueil des besoins agile

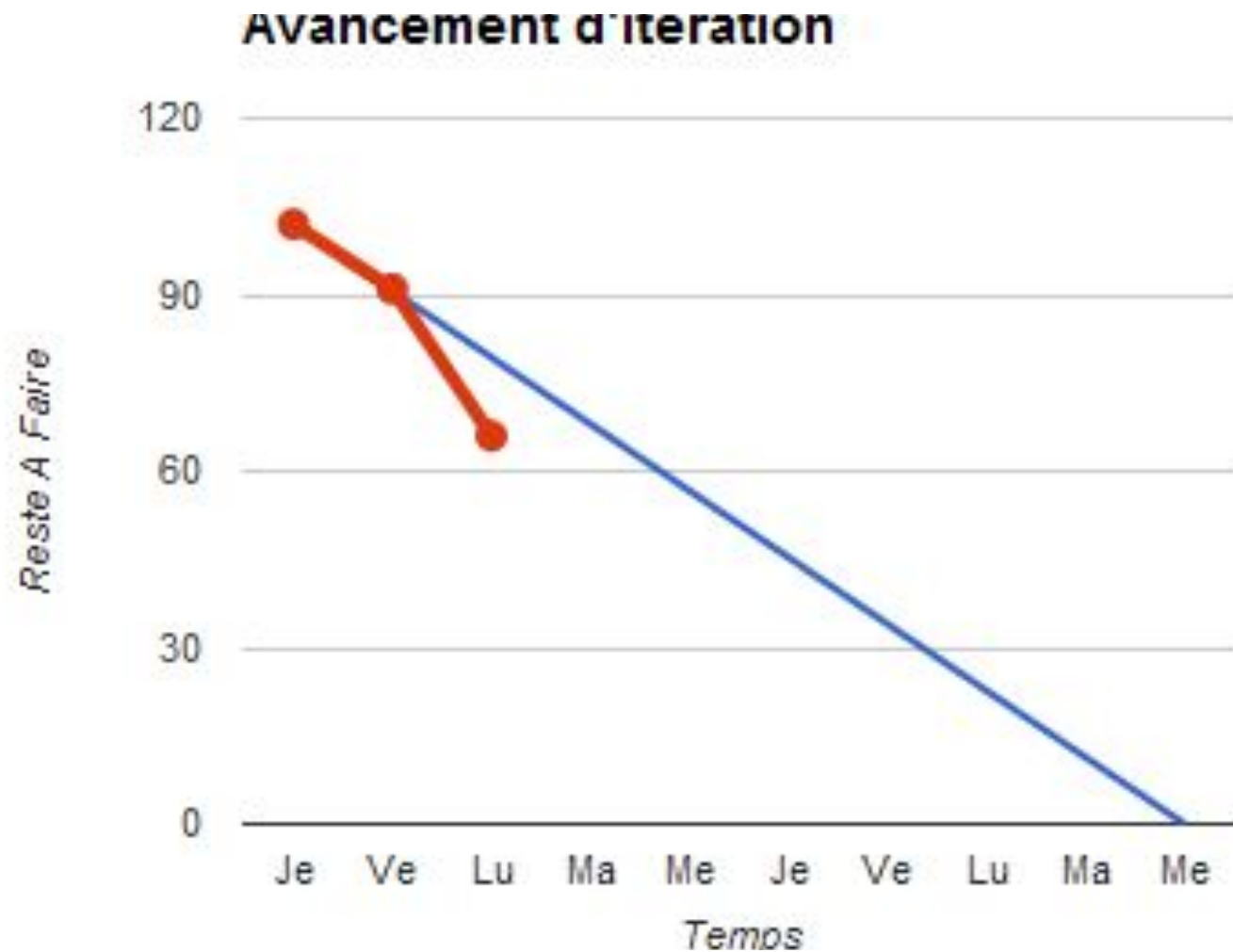


III. La planification agile



IV. Le pilotage agile

Exemple de Burndown



I. Le mouvement agile



II. Le recueil des besoins agile



III. La planification agile

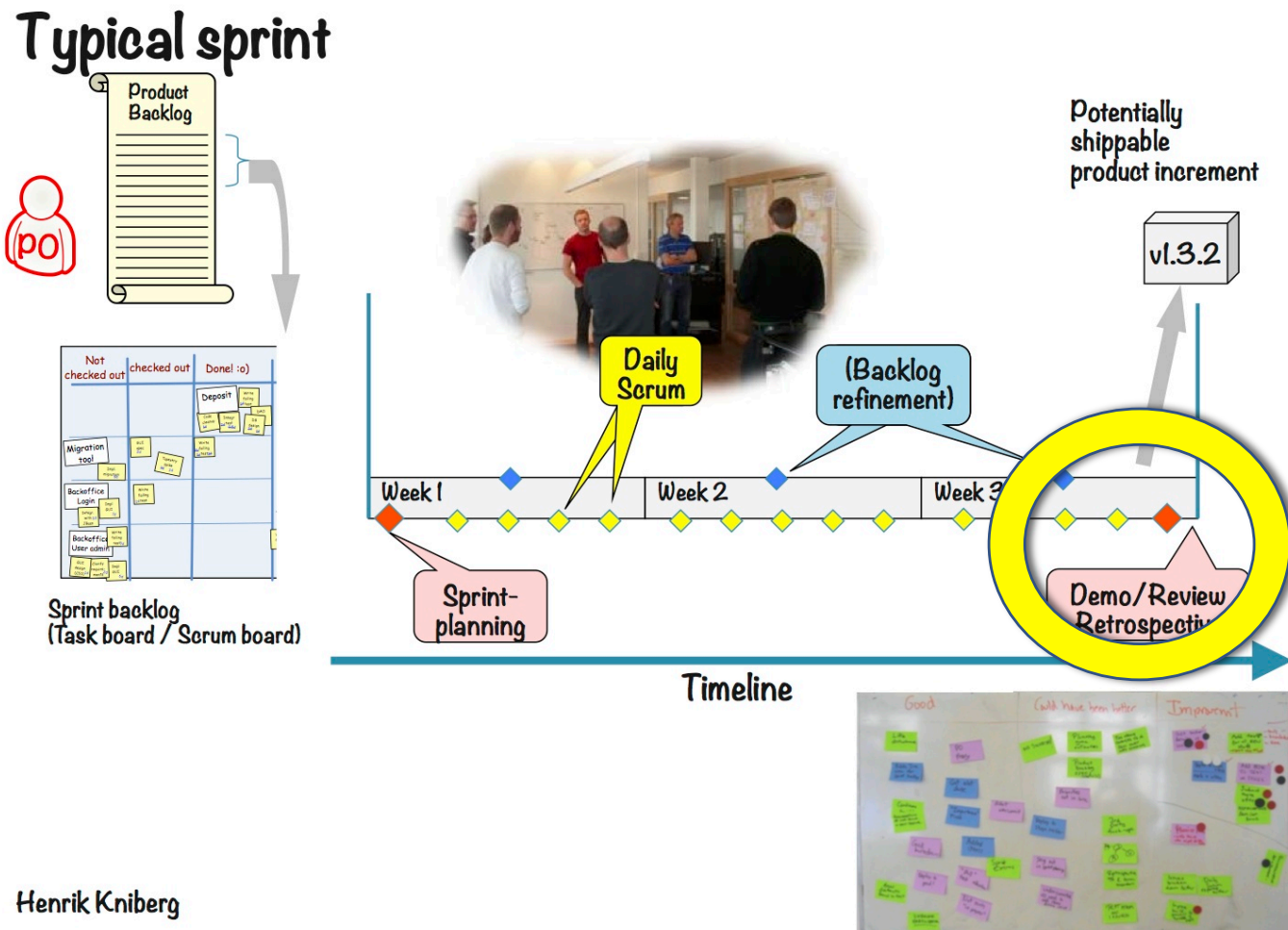


IV. Le pilotage agile

(35)

Les temps forts d'une itération scrum :

Ne pas négliger la rétrospective, outil fondamental de pilotage de l'équipe



Henrik Kniberg