



Agilité – principes – conséquences sur l'expertise

Journée de formation Cnejita – 2 décembre 2014

Préalable

1. Votre serviteur n'est pas un spécialiste de l'Agile et n'a pas conduit de projet en Agile
 - ▶ Les éléments abordés ici sont issus de la bibliographie et d'entretiens avec des spécialistes
2. C'est un document de recherche d'exploration de la problématique « Agile » en expertise
 - ▶ Le questionnement qui est posé ici n'est ni arrêté ni définitif
3. Les situations de référence sont « Agile » pures
 - ▶ Les cas réels combinent de l'agile et du classique, et la transposition immédiate des éléments abordés dans cette présentation n'est pas possible
 - ▶ Une analyse par cas d'espèce reste indispensable
4. Il y a plusieurs méthodes de développement Agile
 - ▶ Les plus connues sont Scrum et XP,
 - ▶ Cette présentation ne rentre pas dans les détails.

Méthodes Agiles

- ▶ Des méthodes de développement qui prennent le parti
 - ▶ d'alléger la trame documentaire des projets
 - ▶ pour réinvestir dans le travail des équipes de programmation
- ▶ pour faire face,
 - ▶ À des clients pressés,
 - ▶ Des systèmes sociotechniques plastiques, qui s'adaptent à chaque release,
 - ▶ Des environnements volatiles, marché, pratiques, etc.
- ▶ Les principes ont été formalisés et font l'objet d'une mobilisation dans les milieux du développement logiciel

Moins de papier, moins d'organisation

- ▶ Donc un référentiel contractuel plus léger
- ▶ Ouf !
 - ▶ Des dossiers d'expertises qui vont passer sous la barre des 3 cartons
- ▶ Ouf ? Pas si sur
 - ▶ Ca sera plutôt des nouveaux points de contrôle
- ▶ Aujourd'hui
 - ▶ Quelques clés pour aborder l'expertise
 - ▶ La contractualisation Agile avec Thomas Beaugrand
 - ▶ La mise en œuvre coté SSII avec Laurent Delvaux
 - ▶ La mise en œuvre coté client avec Yann Poles

Concepts

▶ L'inspiration du Lean

- ▶ Les 3 démons du Lean - L'inutile, la surcharge, l'irrégularité
- ▶ Effectuation theory, value stream - Piloter l'atteinte des résultats plutôt que planifier
- ▶ Kanban, Tirer les flux plutôt que pousser les flux
- ▶ ...Epicure, désirs naturels, désirs vains

▶ Un recentrage sur le développeur

- ▶ Replacer de l'entrepreneuriat dans les projets
- ▶ Le lien direct avec le client et les résultats, vs spécifications
- ▶ Ré-équilibrer les risques entre le client et le développeur
- ▶ Moins d'ISO/CMMI, plus d'opérationnel
- ▶ Travailler à la tâche plutôt qu'au lot

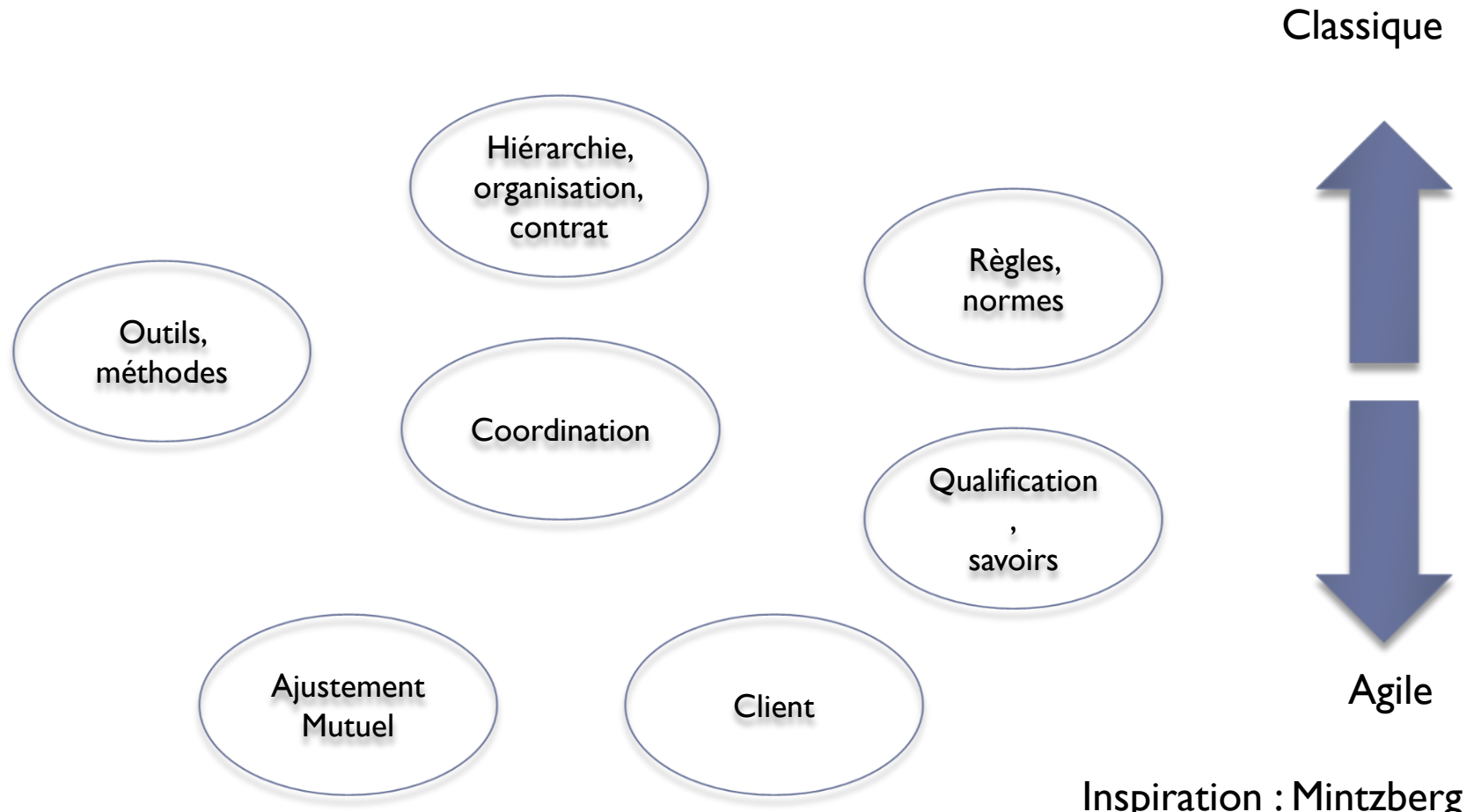
▶ Plusieurs cadres méthodologiques

- ▶ Scrum, XP pour les principales
- ▶ Les rituels associés

▶ 4 valeurs, 12 principes

- ▶ Le tetrapharmakon de l'AgileManifesto.org
 - ▶ Individus/interactions vs processus/outils
 - ▶ Logiciels opérationnels vs documentation exhaustive
 - ▶ Collaboration avec les clients vs négociation contractuelle
 - ▶ Adaptation au changement vs suivi d'un plan
- ▶ Beaucoup de contre-pieds pour l'expertise qui a pris l'habitude de s'appuyer sur les 2èmes termes (processus, documentation, contrats, plans)

Plutôt que moins de cadre, c'est des modalités de coordination différentes



Alléger les structures

Alléger l'armure pour privilégier la mobilité



Moins de papier
Moins de plans à long
terme



Plus de collaboration
Plus d'engagement
individuel

Aie, moins de cadre. Mais alors...?

- ▶ **Ca ressemble à CMMI niveau 0 absolu – le temps des héros**
 - ▶ Au niveau du sprint, il y a une capitalisation
 - ▶ Au niveau du sprint, il y a un contrôle qualité adossé à un environnement d'intégration continu (versionning, tests unitaires, règles de programmation)
 - ▶ Au niveau de la roadmap projet, il y a un suivi de la planification
- ▶ **Mais comment garantit-on un niveau de qualité ISO**
 - ▶ La qualité externe est garantie par une politique de collaboration client/fournisseur et des tests d'acceptation réguliers
 - ▶ Plutôt que par la congruence et l'exhaustivité d'un système documentaire
- ▶ **Le prestataire n'est plus contraint par le long terme**
 - ▶ Les objectifs se réévaluent Sprint par Sprint
 - ▶ L'engagement porte sur la capacité de production, la vitesse
- ▶ **Le programmeur n'est plus encadré**
 - ▶ Il est autant encadré qu'un joueur de rugby (engagement de l'équipe, rester en équipe !)
 - ▶ Points quotidiens
- ▶ **Encore beaucoup de contrepieds pour l'expertise**

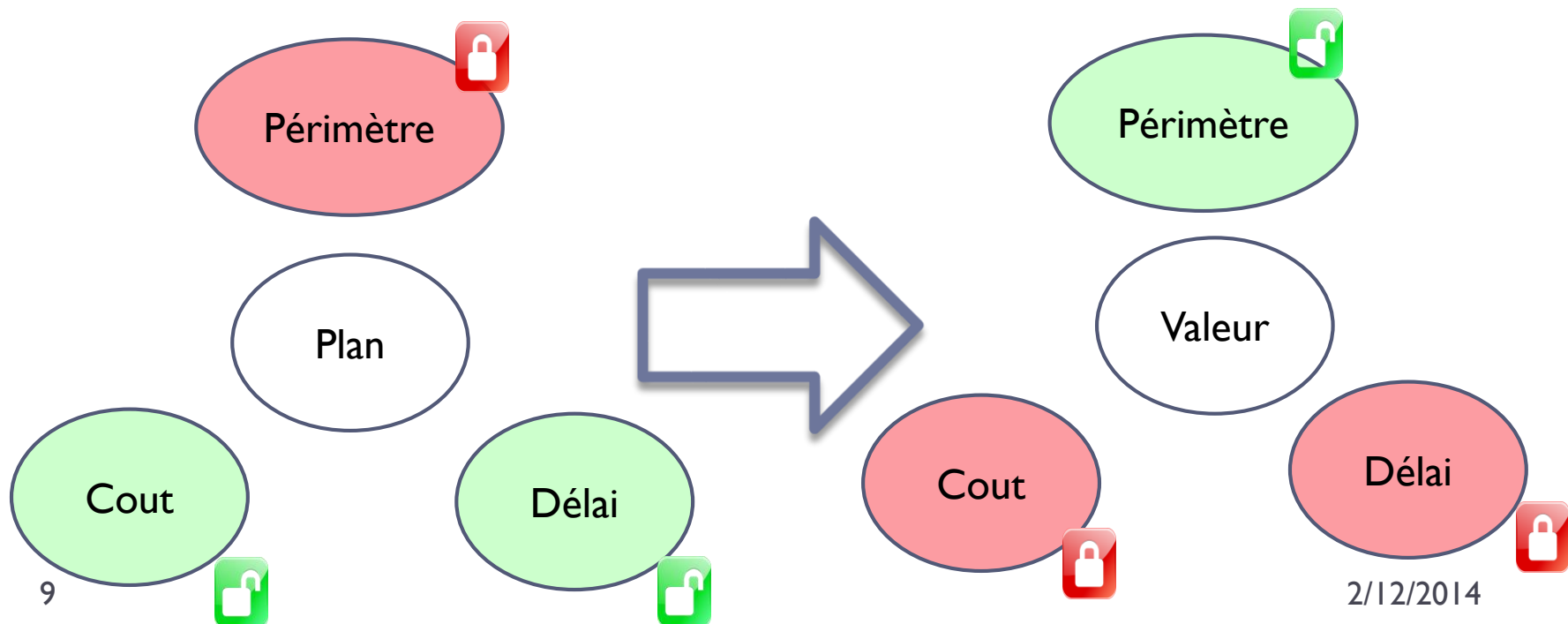
1^{er} contrepied – changement de verrous

Projet classique :

- le périmètre est figé,
- Coût et délais sont négociés
- Le plan cadence

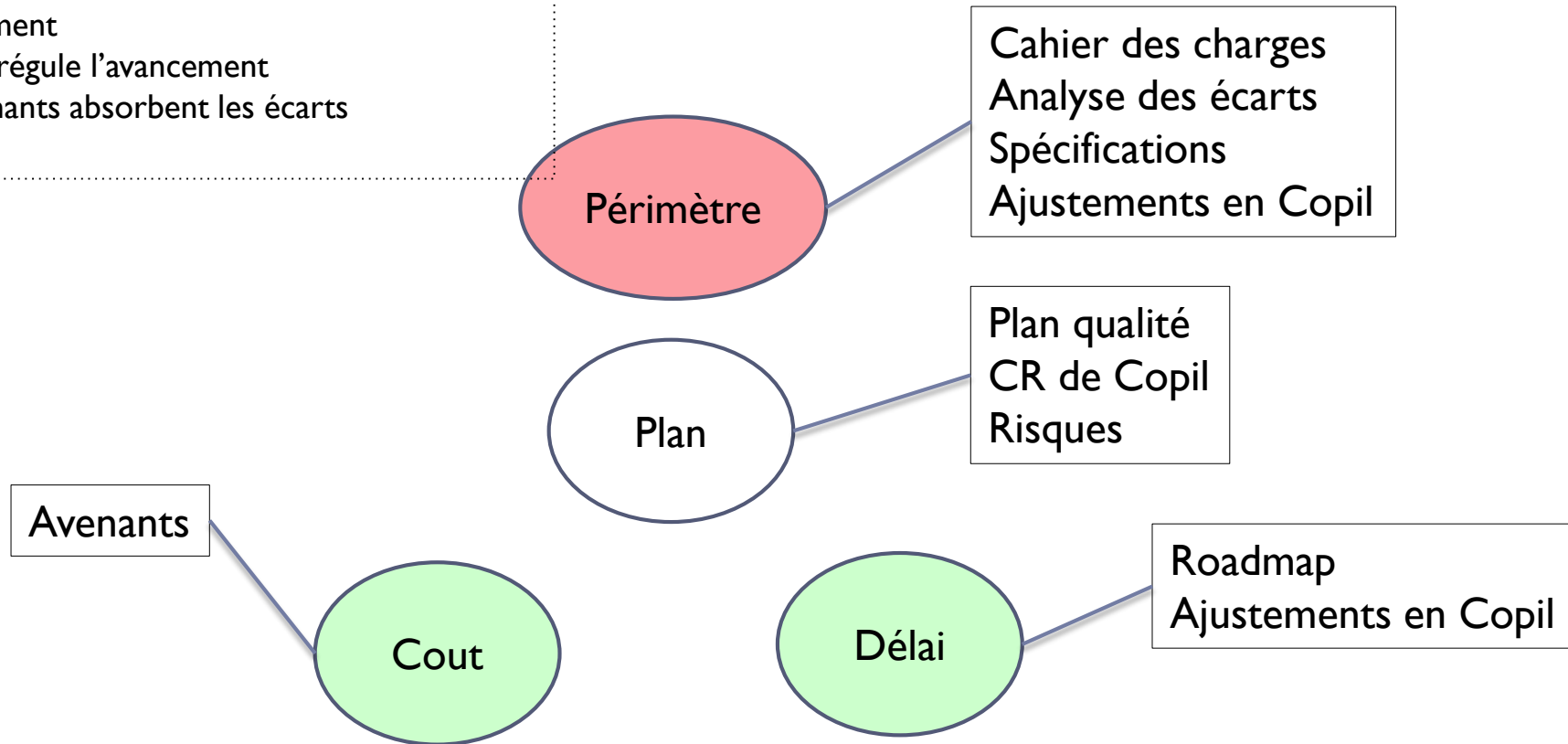
Projet Agile:

- Coût et délais sont fixés
- Le périmètre est adapté de sprint en sprint
- La production de valeur cadence



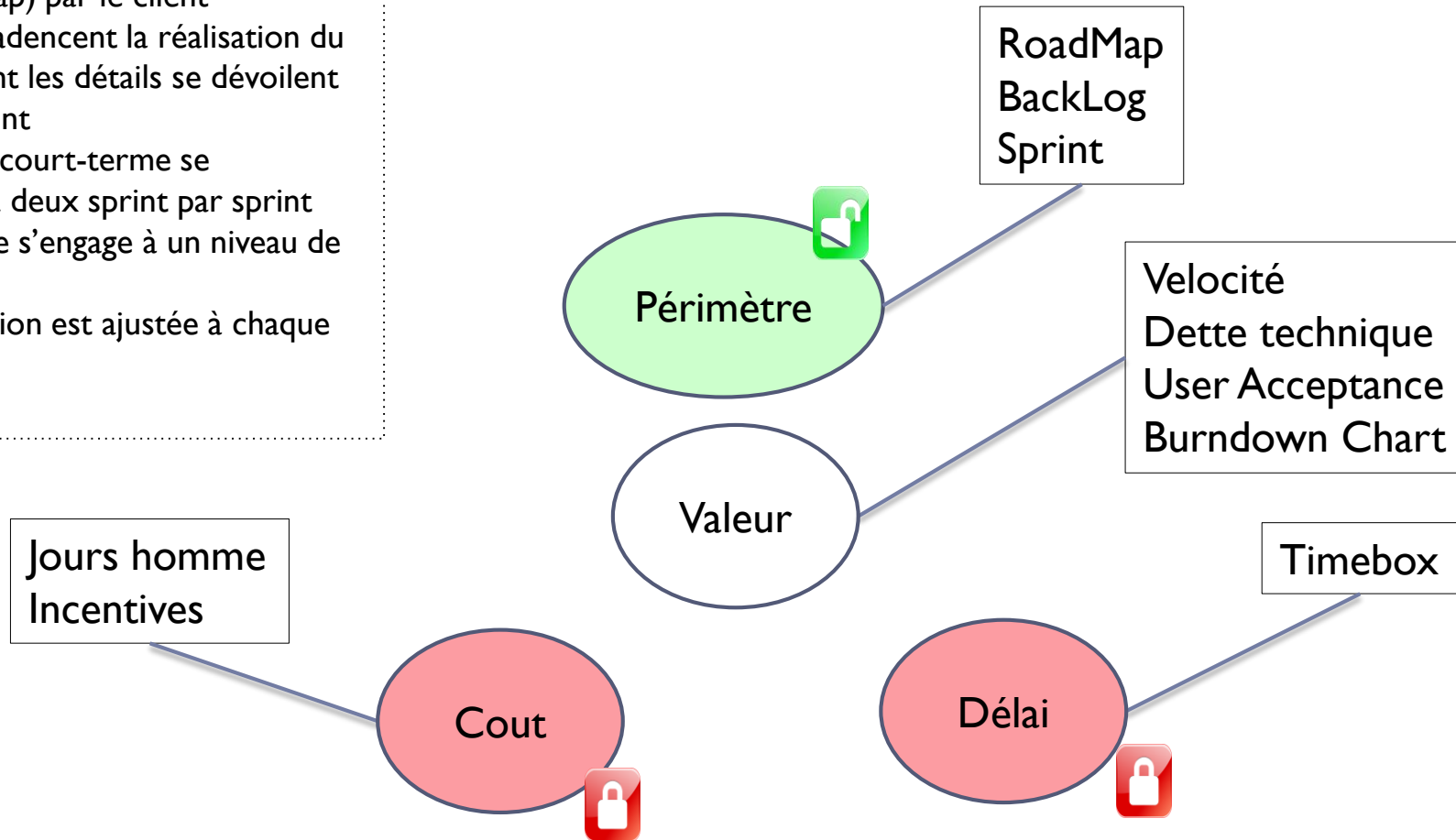
Cibles documentaires classiques

- Le périmètre est figé modulo une analyse des écarts réalisée au début du projet
- Le coût et le délai servent de variables d'ajustement
- Le plan régule l'avancement
- Les avenants absorbent les écarts



Cibles documentaires Agile

- Le périmètre est balisé dans ses grandes lignes (roadmap) par le client
- Les sprints cadencent la réalisation du périmètre dont les détails se dévoilent sprint par sprint
- Les objectifs court-terme se déterminent à deux sprint par sprint
- Le prestataire s'engage à un niveau de productivité
- La collaboration est ajustée à chaque sprint



Quelques repères

- ▶ Les mantras de l'agile – 4 valeurs, 12 principes ...
- ▶ Les user stories
- ▶ Le Product Owner, le scrum master
- ▶ La rétrospective
- ▶ Le backlog
- ▶ La vélocité
- ▶ La dette technique, les technical stories
- ▶ La project war room

Fixons les idées

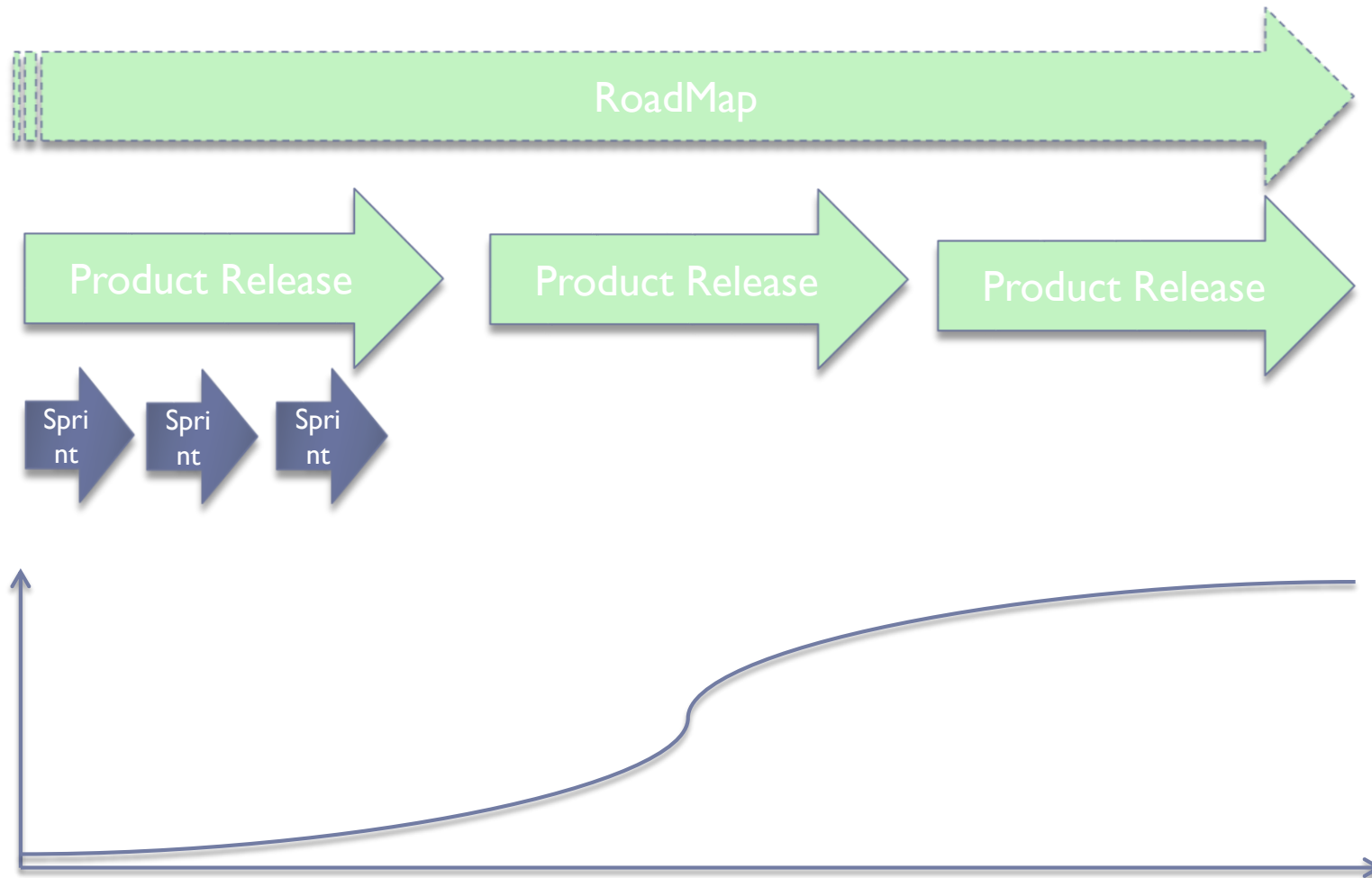
Video Kniberg

<https://www.youtube.com/watch?v=3qMpB-UH9kA>

Le partage du risque

- ▶ Agile : en filigrane de la méthode «développeur happy, client happy »
- ▶ Corolaires – le risque est transféré vers le client
 - ▶ Je travaille bien, si ca vient pas, c'est que ton problème est compliqué
 - ▶ Si ton problème est compliqué, c'est normal que tu paies
 - ▶ A chaque fois que je livre, je laisse un logiciel opérationnel, tu n'auras pas perdu ton argent
 - ▶ J'ai travaillé en toute transparence, tu avais les moyens d'intervenir – roadmap, priorités backlog,
 - ▶ Si j'ai mal travaillé, c'est que tu t'es mêlé de choses qui te regardaient pas (outils, développement,...) ou que tu m'as mal guidé (développement des US pendant les sprint)
- ▶ Les directions d'achats sont rarement ok avec ça
 - ▶ Aménagement des contrats pour transférer le risque vers l'équipe de développement
 - ▶ Introduction d'éléments de forfait, garantie de faisabilité
 - ▶ Equilibrage de l'agilité
 - ▶ SSII sécurisée par des règlements au fil de l'eau
 - ▶ Client sécurisé par la possibilité de rompre unilatéralement
 - ▶ Des mécanismes de régulation/d'incitation
 - ▶ Grande variabilité contractuelle dans le monde de l'Agile

L'approche par niveaux de grains successifs



La planification de la roadmap (grandes lignes)

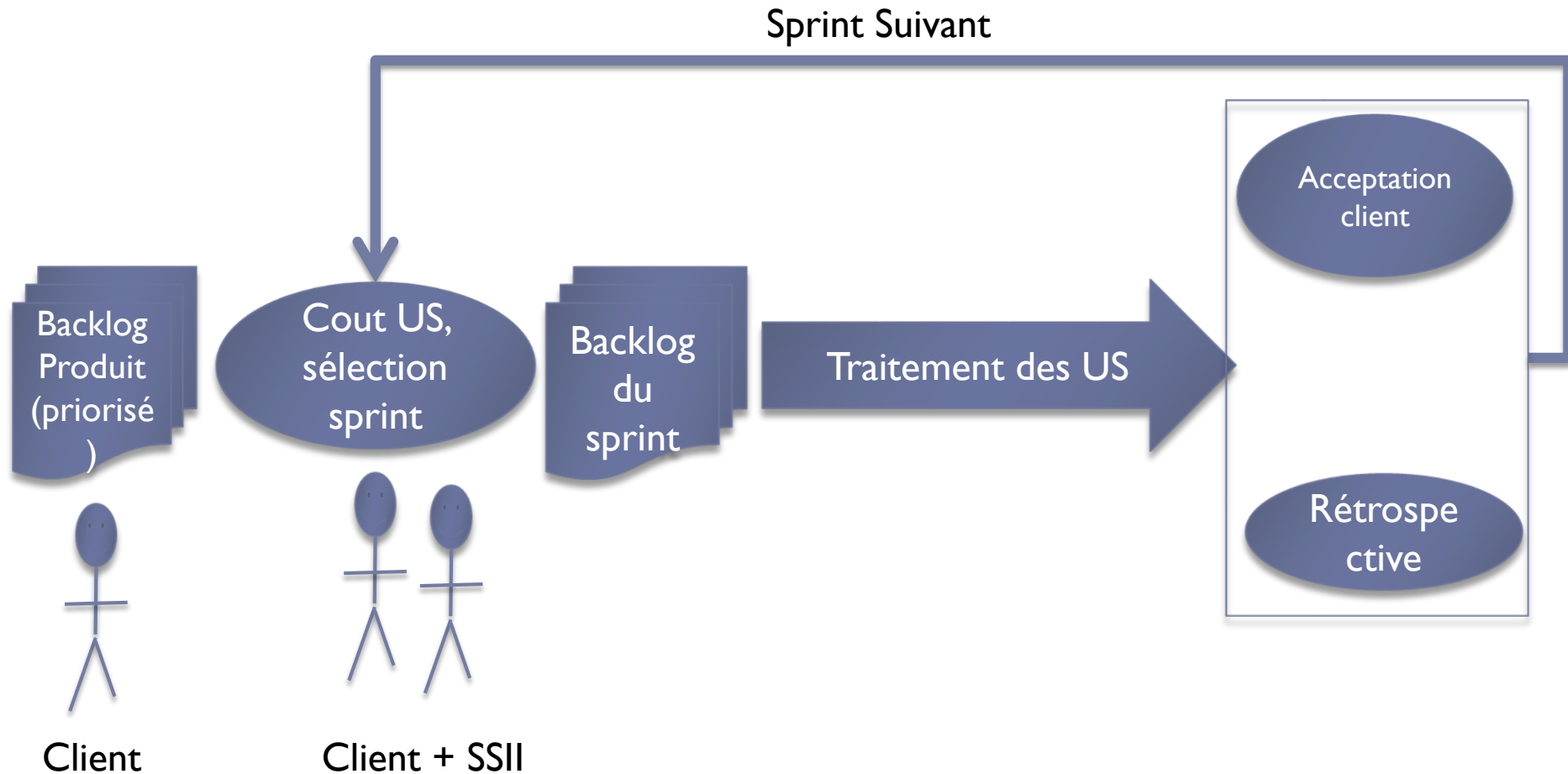
- ▶ **Nominal**
 - ▶ La roadmap se déroule conformément au calendrier initial
 - ▶ Les MEP (product release) ont lieu à dates fixes. Le contour des MEP correspond au contour défini par la roadmap
- ▶ **Aléas**
 - ▶ Les priorités utilisateur peuvent changer, périmètre, calendrier
 - ▶ La valeur cible peut être atteinte plus vite que prévue
- ▶ **Éléments d'agilité – Le plan peut être adapté à un changement de priorité**
 - ▶ Les principes
 - ▶ Le client peut revoir son plan quand il veut.
 - Le client fait sa roadmap, fixe ses priorités pour coller au mieux des attentes des utilisateurs, des avancées projet
 - Il détermine le contenu (grandes lignes) et le calendrier des MEP
 - ▶ Le fournisseur valide le réalisme de la roadmap
 - ▶ Effets de bord
 - ▶ Un projet peut s'arrêter plus tôt que prévu
 - CA en baisse pour le fournisseur
 - Economie pour le client
 - Comment on partage?
 - ▶ Modification peut impacter la productivité
 - Stabilité = courbe d'apprentissage, Instabilité = perte de productivité
 - Stabilité = planification architecture technique, instabilité= désordre architectural

Éléments de modération de l'agilité

Roadmap

- ▶ **Faisabilité technique de la nouvelle roadmap**
 - ▶ Découpage des US
 - ▶ Adaptabilité de l'architecture
 - ▶ Validation fournisseur
- ▶ **Compensation pour arrêt anticipé selon que**
 - ▶ Valeur cible atteinte plus vite que prévue
 - ▶ Valeur cible réduite et arrêt anticipé
 - ▶ Collaboration déficiente et arrêt anticipé
- ▶ **Renforcement, maintien prolongé**
 - ▶ Qui paie, à quel tarif?
- ▶ **Incitatifs pour encourager la stabilité des équipes, des plans**

La structure du sprint



Agilité au sein du sprint

- ▶ **Nominal**
 - ▶ Les US sélectionnées sont réalisées, testée puis la release du sprint est acceptée par le client
- ▶ **Aléas**
 - ▶ Il va manquer des US en sortie de sprint
 - ▶ Le backlog du sprint se révèle irréalisable dans les délais du sprint
 - Mauvais chiffrage (complexité US mal calibrée, descriptif insuffisant)
 - Mauvaise communication utilisateur/développeur
 - ▶ Des US n'ont pas été acceptées
 - ▶ Il manquait des comportements
 - ▶ La technique n'était pas au rendez vous
 - ▶ Le développeur n'a pas tenu compte des précisions apportées en cours de sprint par l'utilisateur
- ▶ **Éléments d'agilité – adapter dans l'instant, faire mieux au sprint suivant**
 - ▶ Revue du backlog «sprint » en cours de sprint
 - ▶ Report d'US au sprint suivant, split d'US, remplacement d'US par une autre
 - ▶ Acceptation de sous qualité technique pour tenir la US
 - ▶ Rétrospective
 - ▶ Point sur le fonctionnement du sprint – calibrage des US, collaboration utilisateurs, efficacité des développeurs, ...
 - ▶ Actions correctives au sprint suivant
 - ▶ Effets de bords
 - ▶ Création de dette technique
 - ▶ Perte de vélocité
 - ▶ Tensions au niveau des équipes

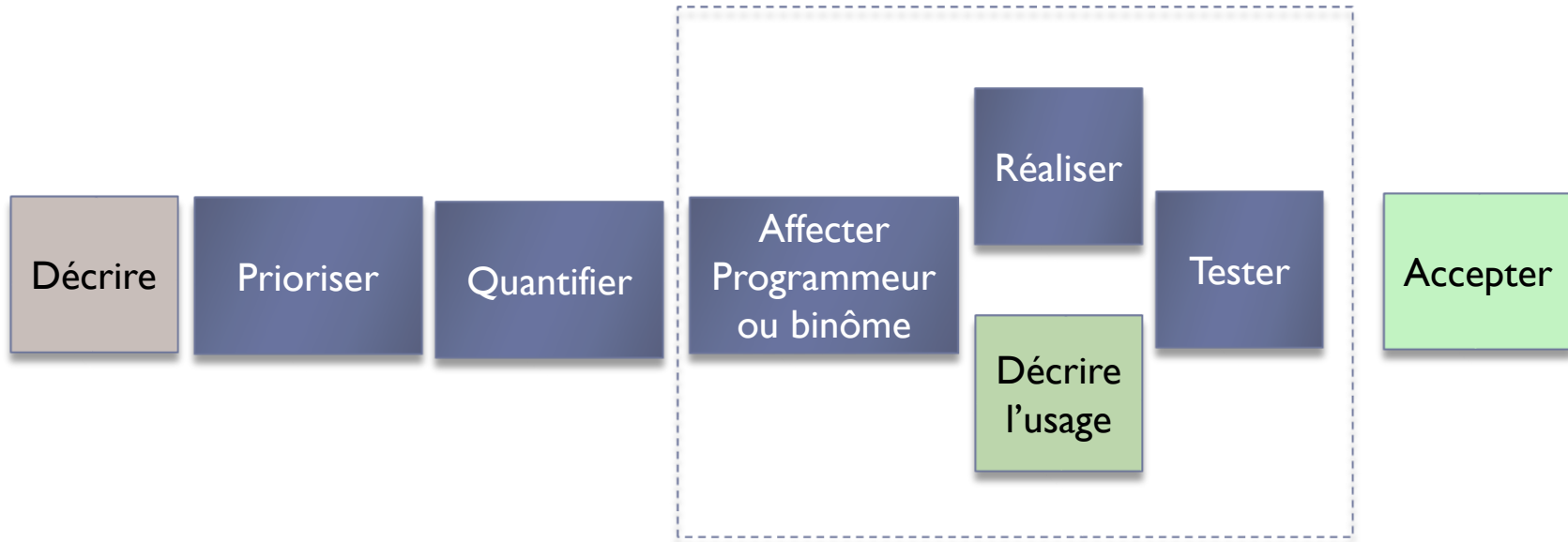
Éléments de modération de l'agilité dans les sprints

- ▶ Pilotage des sprints
 - ▶ Rétrospective, et actions d'amélioration sur la formulation, classification, chiffrage des US
 - ▶ Suivi de la vélocité, des burndown charts
- ▶ Incitatifs/pénalités sur le report
 - ▶ Pur agile – report à charge du client (logique régie)
 - ▶ Contrat hybride peuvent laisser une part à charge fournisseur
- ▶ Incitatifs/pénalités sur le traitement de la dette technique
 - ▶ Pur agile – report à charge du client (logique régie)
 - ▶ Contrat hybride peuvent laisser une part à charge fournisseur

Dettes techniques

- ▶ Elle représente les non qualités techniques accumulées dans le code pour privilégier la production de valeur
- ▶ Cause
 - ▶ Urgence projet imposée par le client
 - ▶ Rattrapage d'une mauvaise estimation du backlog du sprint
 - ▶ Conséquence naturelle d'une conception incrémentale
- ▶ L'effacement de la dette technique
 - ▶ Travaux de remise au propre (commentaires, paramètres, nommage, tests unitaires,...)
 - ▶ Travaux de refactoring
 - ▶ Dette technique se paie au détriment de la production de valeur
- ▶ Attention au surendettement !
 - ▶ Accumulation des non qualités
 - ▶ Big ball of mud
- ▶ Qui paie ?

Le traitement de la User Story



User Story

▶ Nominal

- ▶ La US est priorisée et chiffrée avant le sprint
- ▶ La réunion de planification du sprint retient la US dans le sprint
 - ▶ Thèmes de la MEP en cours, ratio V/C,...
- ▶ Un programmeur ou un binôme de programmeur est affecté
- ▶ Les utilisateurs assistent le programmeur pour détailler l'US et la rédaction des spécifications
- ▶ Les évolutions introduite par la US sont testées au fil de l'eau (option TDD – les tests unitaires sont développés avant de coder)
- ▶ L'US est validée lors de la revue d'acceptation

▶ Aléas et éléments d'agilité

- ▶ US pas réalisable dans les temps du sprint
 - ▶ Si détectée assez tôt reordonnancement dans le sprint suivant ou split ou dette technique
- ▶ US pas finalisée dans les temps du sprint
 - ▶ Traiter la US dans le sprint suivant (qui n'aura que le reste à faire à produire)

▶ Éléments de modération

- ▶ Incitatif sur le report de US (le sprint est vu comme un forfait)
- ▶ Impact sur la vélocité
- ▶ Retrospective!

User story - caractéristiques

- ▶ Un support de communication et de pilotage plutôt qu'une spécification
- ▶ Centré sur l'utilisateur, le service attendu plutôt que la fonction
 - ▶ Le demandeur, la demande, le but visé
- ▶ Associée au conditions d'acceptation
- ▶ Format compact
 - ▶ bristol pour post-it
 - ▶ Recto/verso – user story/conditions d'acceptation
 - ▶ Alternatives électroniques
- ▶ Critères I.N.V.E.S.T.

Plateau projet – project war room

- ▶ L'accent est mis sur la collaboration
- ▶ Elle est mise en scène par les rituels (comitologie de l'agile) et l'organisation de l'espace
- ▶ Un plateau projet
 - ▶ Physique plutôt que virtuel
 - ▶ Regroupe tous les participants
 - ▶ Les indicateurs et plan projets sont exposés (radiateur projet)
 - ▶ La présence d'un représentant des utilisateurs
- ▶ Travail à la tâche (user story) affecté à des binômes
- ▶ Lieu privilégié de l'exercice de l'ajustement mutuel
 - ▶ « **L'ajustement mutuel** : La réalisation du travail par le biais d'une communication informelle (par exemple deux ouvriers qui communiquent à l'oral). » (http://fr.wikipedia.org/wiki/Henry_Mintzberg)
 - ▶ Réguliers
 - ▶ Morning (ou afternoon) meeting
 - ▶ Clarification sur les US
 - ▶ Gestion des points durs
 - ▶ Périodiques
 - ▶ Démonstrations
 - ▶ Rétrospectives

Ce qui peut changer pour l'expert

- ▶ La régulation de la relation change de polarité
 - ▶ Le périmètre devient négociable
 - ▶ Le respect du rythme et de la vélocité est prioritaire sur le périmètre
 - ▶ Le rôle plus actif du client dans la planification
- ▶ Un cadre théorique qui relaxe les contrôles et transfert le risque sur le client
 - ▶ Enrichissement/surcharge du cadre agile avec des mécanismes incitatifs pour stabiliser/encourager la coopération
 - ▶ Contrats hybrides pour associer le fournisseur au risque
 - ▶ Pas de contrat standard - des dispositifs à passer en revue au cas par cas
- ▶ Les points de contrôle changent
 - ▶ Roadmap, sprint, retrospective, backlog, burndown, vélocité, dette technique,...
 - ▶ La sur-qualité n'est plus une vertu
- ▶ Importance de la coopération, de l'ajustement mutuel
 - ▶ Grilles de lecture du « facteur humain » - cf colloque 2014
 - ▶ Plateau projet, animation, indicateurs