

Kanban dans des projets Informatiques

Nicolas Duflot

nduflot@dawan.fr

Septembre 2022

Plus d'informations sur <http://www.dawan.fr>
Contactez notre service commercial au **0800.10.10.97** (appel gratuit depuis un poste fixe)

Introduction



Introduction : gestion de projet



Il existe deux types d'approche :

1 : L'approche prédictive.

Prévoir des phases séquentielles avec un engagement sur un planning précis de réalisation de projet.

2 : L'approche agile.

Construire un processus itératif et incrémental qui consiste à découper le projet en plusieurs étapes que l'on appelle itérations.



Introduction : l'approche prédictive



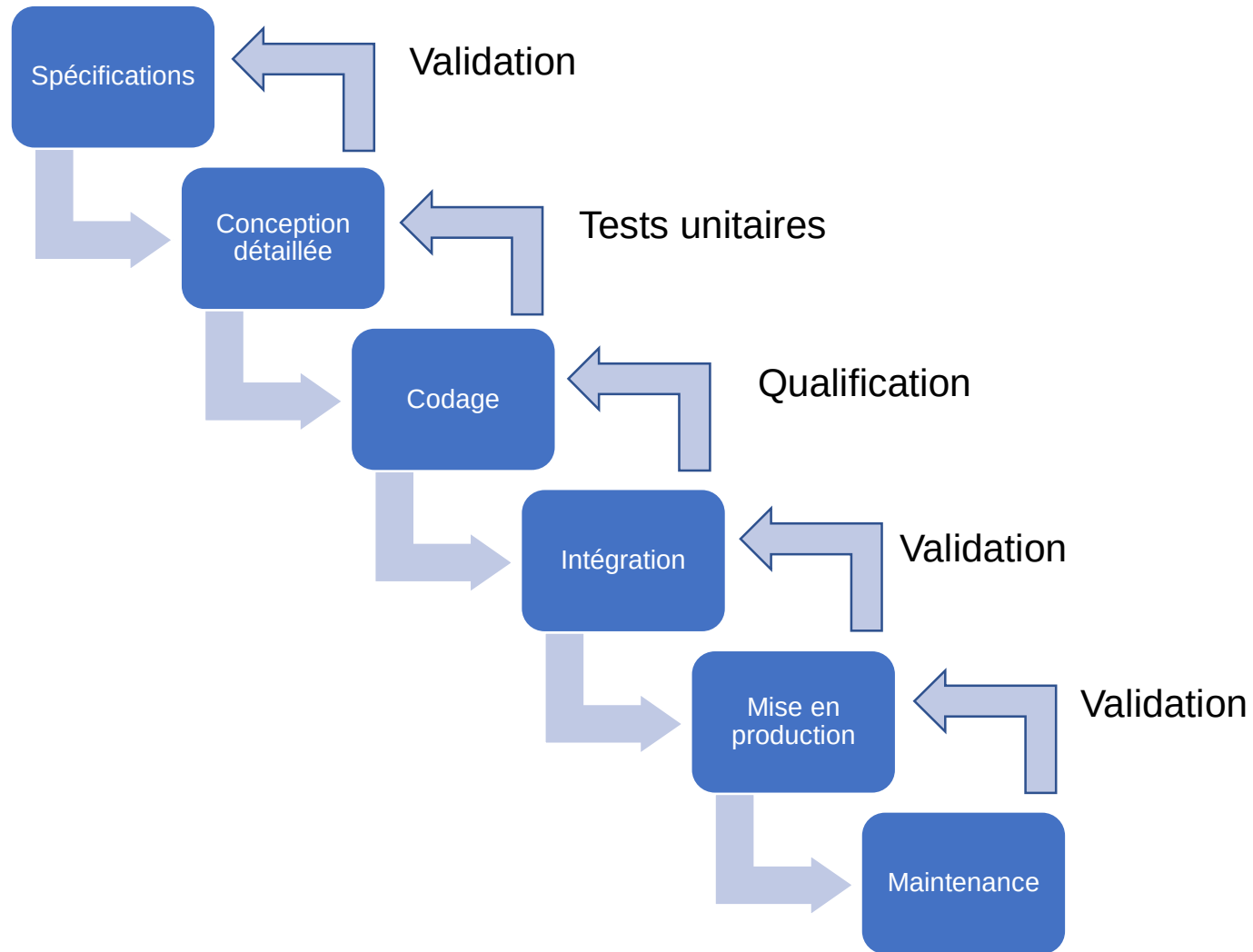
L'approche « classique » est aussi appelée **approche prédictive**.

Cette approche respecte généralement un cycle de vie prédéfini.

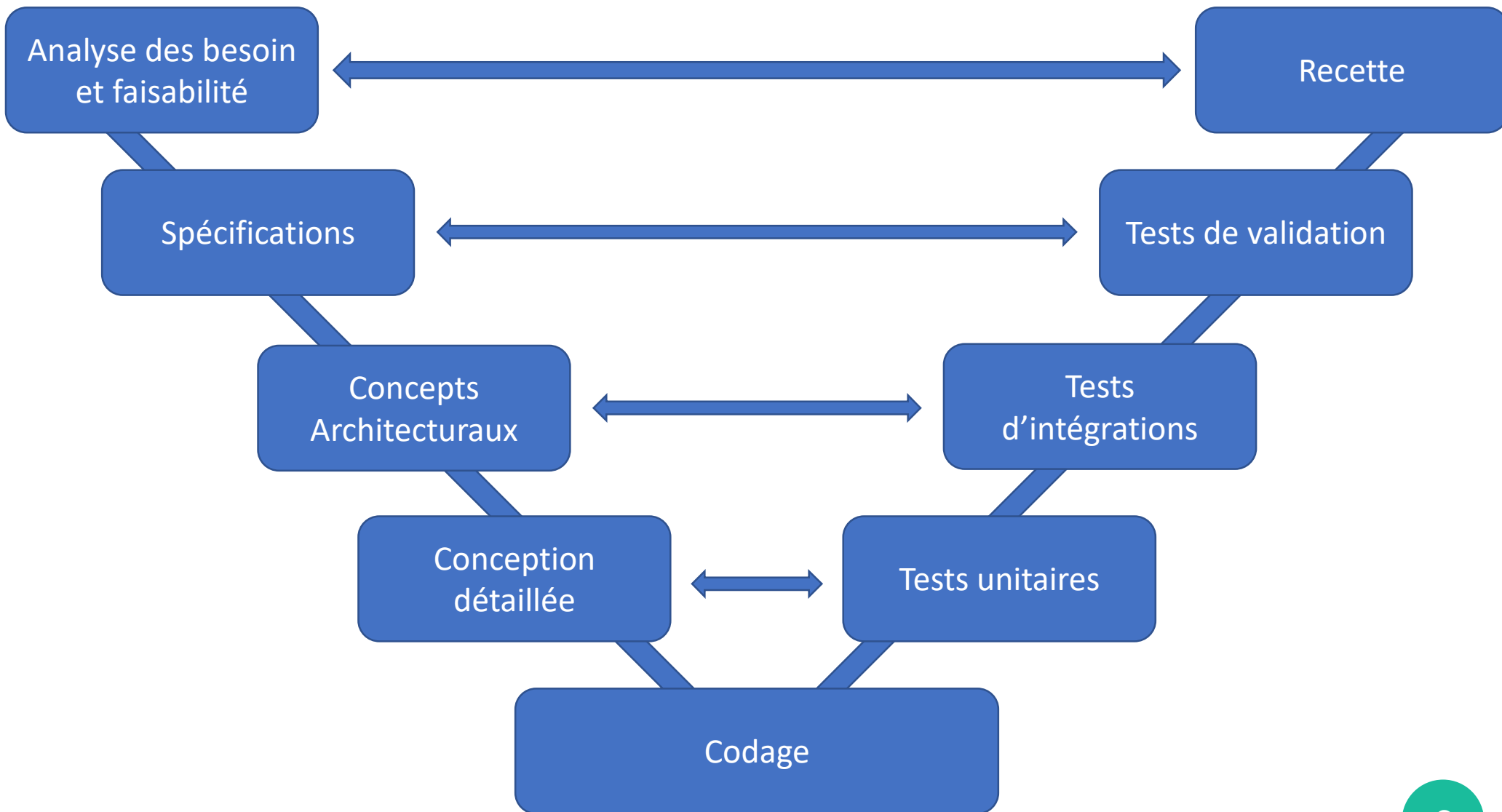
Les cycles de vie les plus fréquents sont :

- **Cycle en cascade**
- **Cycle en V**
- **Cycle en spirale**

Cycle en cascade



Cycle en V



Qualité / Coût / Délais... et le projet (1)



La gestion de projet repose sur 3 piliers : la **qualité** (en principe non négociable), les **coûts**, et les **délais** au centre desquels reposent le projet, constitués de **fonctionnalités**.



Qualité / Coût / Délais... et le projet (2)



Avec une approche prédictive, le facteur d'ajustement sera généralement l'un des piliers :

Les cycles de vie les plus fréquents sont :

- **Le respect des coûts et des délais au détriment de la qualité**
- **Le respect des coûts et de la qualité au détriment des délais**
- **Le respect des délais et de la qualité au détriment des coûts**

Qualité / Coût / Délais... et le projet (3)



Avec une approche agile, le facteur d'ajustement sera les fonctionnalités.

Les 3 piliers sont définis en début de projet.

Les fonctionnalités sont négociables.

Pourquoi l'agilité ?



Réponse à un malaise ?

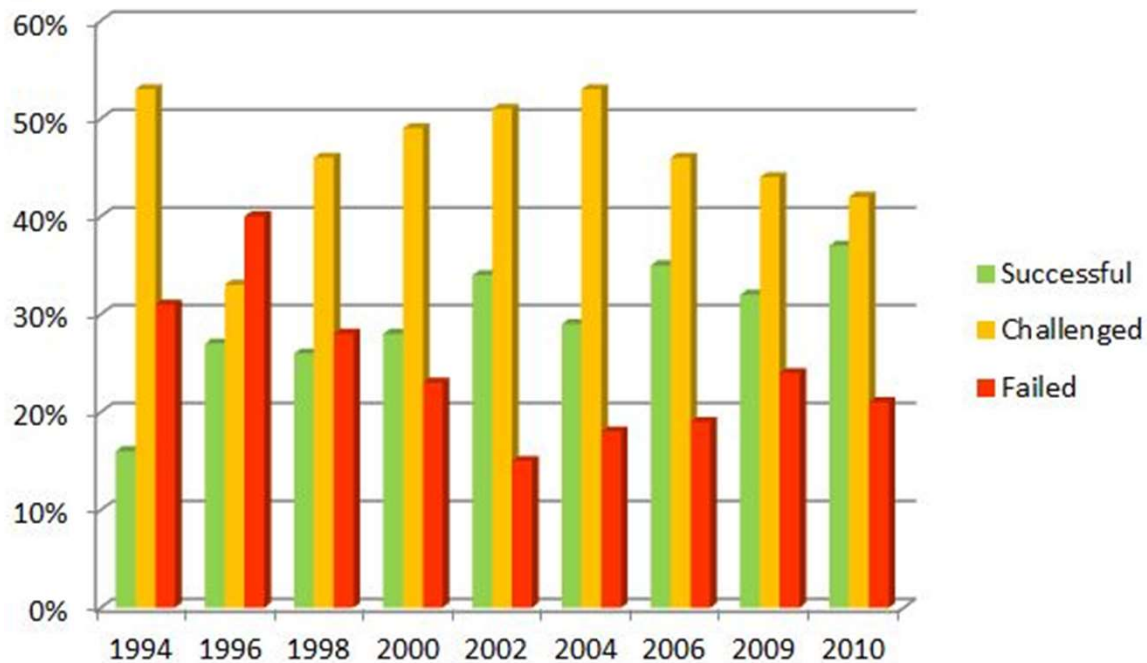


1994 : Rapport CHAOS (étude sur 8000 projets)

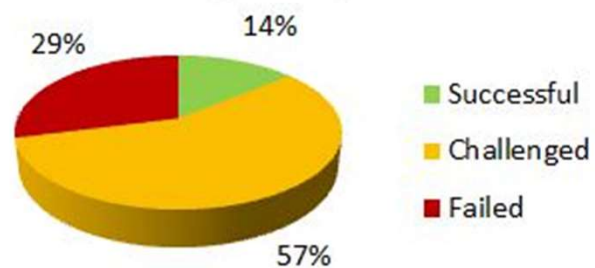
- 16% finalisés dans les temps et le budget prévus
- 32% interrompus en cours de route
- Le contexte économique difficile met une pression forte
- Rapports parfois difficiles entre la maîtrise d'ouvrage et la maîtrise d'œuvre
- Confusion entre des besoins estimés et des besoins réels
- Les guerres contractuelles font oublier l'objectif initial
- Chacun s'abrite derrière les modalités d'un forfait qui fige dans le marbre des spécifications incomplètes

<https://www.csus.edu/indiv/v/velianitis/161/chaosreport.pdf>

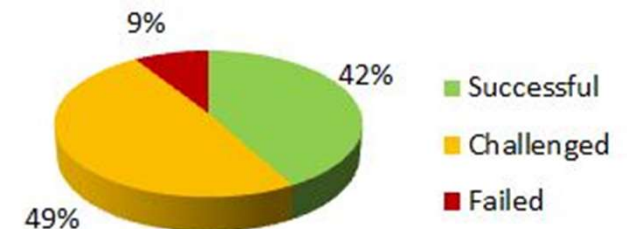
Constat



Cycle projet - méthode classique
(cascade)



Cycle projet - méthode agile



CHAOS 2015

Taille	Méthode	Succès	Finis mais...	Échec
Toutes tailles de projets	Agile	39%	52%	9%
	Cascade	11%	60%	29%
Larges	Agile	18%	59%	23%
	Cascade	3%	55%	42%
Moyens	Agile	27%	62%	11%
	Cascade	7%	68%	25%
Petits	Agile	58%	38%	4%
	Cascade	44%	45%	11%

<https://www.infoq.com/fr/articles/rapport-chaos-2015/>

Facteurs d'échec



- Le manque de communication à tout niveau
- Une mauvaise compréhension des besoins
- L'insuffisance de l'architecture
- L'absence de maturité des outils utilisés
- La mauvaise formation des personnes
- Le cadre contractuel inadapté
- L'insuffisance des tests
- L'absence d'une démarche Think Big – Start Small
- **L'effet tunnel**

Pourquoi choisir l'agilité ?



- Réduire le temps de mise sur le marché (Time to Market)
- Améliorer la qualité du produit
- Réduire les activités sans réelle valeur ajoutée pour l'utilisateur
- Avoir une meilleure prévisibilité
- Améliorer le contexte de travail

Critères d'éligibilité



Favorisants :

- Besoin rapide de mise à disposition du produit
- Imprévisibilité des besoins du client
- Nécessité de changements fréquents
- Besoin de visibilité du client sur l'avancement des développements
- Présence de l'utilisateur assurant un feedback immédiat

Défavorisants :

- Indisponibilité du client ou de l'utilisateur
- Dispersion géographique des ressources humaines
- Inertie des acteurs du projet ou refus des changements
- Gouvernance complexe de la DSI

Manifeste Agile



Origine



Mouvement né en 2001 aux USA

Réunion de 17 experts en développement logiciel

But : trouver un socle commun de valeurs et de bonnes pratiques

Résultats :

–écriture du Manifeste pour le développement Agile

<http://agilemanifesto.org/iso/fr/manifesto.html>

–création de l'**Agile Alliance** (association chargée de la promotion de l'agilité et du soutien aux équipes)

<http://www.agilealliance.org>

Manifeste Agile

Manifeste pour le développement Agile de logiciels

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
Des 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.

Les concepts



Les 6 concepts du manifeste Agile, qui doivent être considérés de façon interdépendante :

- Valeur
- Incertitude
- Client
- Individus
- Équipe
- Contexte

Les concepts



Vous ne pouvez créer de la valeur si vous ne collaborez pas avec le client, qui précisément définit ce qu'est sa valeur.

Vous ne pouvez attendre d'une équipe qu'elle soit performante si vous ne reconnaissez pas la contribution des individus qui la composent.

Vous ne pouvez maîtriser l'incertitude si vous ne prenez pas en considération la spécificité du contexte.

Les 13 principes : 1/13



Notre priorité est de satisfaire le client en lui livrant très tôt et régulièrement des versions opérationnelles de l'application, source de valeur.

- Développement itératif, livraison intermédiaire.
- Validation par le client et feedback.
- Conformité de la livraison aux attentes ou prise en compte des remarques et re-priorisation.

Les 13 principes : 2/13



Accepter le changement dans les exigences, même tard dans le cycle de vie, pour garantir la compétitivité du client.

- État d'esprit de l'équipe Agile.
- Capacité à comprendre et apprendre comment satisfaire encore mieux la demande.

Les 13 principes : 3/13



Livrer le plus souvent possible des versions opérationnelles de l'application, à une fréquence allant de 2 semaines à 2 mois.

- 1 version intermédiaire du produit final, visible et testable satisfait davantage le client qu'une documentation à valider.
- Il a la preuve tangible que le projet avance.
- Il peut exprimer son point de vue sur le résultat présenté

Les 13 principes : 4/13



Client et développeurs doivent coopérer quotidiennement tout au long du projet.

- Les relations conflictuelles ne font pas partie de l'esprit Agile.
- Les relations collaboratives et partenariales basées sur la confiance et le consensus sont préférables.
- Le client (ou son représentant) est accessible et disponible, totalement impliqué dans toutes les phases du projet.

Les 13 principes : 5/13



Construire des projets autour d'individus motivés. Leur donner l'environnement et le support dont ils ont besoin et leur faire confiance pour remplir leur mission.

- Le facteur clé du succès d'un projet est l'équipe.
- Tout obstacle à son bon fonctionnement devra être levé.
- Un changement, s'il s'avère nécessaire, sera apporté aux processus, aux outils, à l'environnement, à la composition de l'équipe.

Les 13 principes : 6/13



La méthode la plus efficace de communiquer des informations à une équipe et au sein de celle-ci reste la conversation en face à face.

- Par défaut, on privilégie l'oral à l'écrit, pour lever toute ambiguïté et favoriser la rapidité de la compréhension.
- Tout ne peut être formalisé par écrit, notamment la “connaissance tacite”, c'est à dire, “l'information informelle”, la culture du projet, détenues par chacun au sein d'une équipe.

Les 13 principes : 7/13



Le fonctionnement de l'application est le premier indicateur d'avancement du projet.

- Il n'existe pas d'autre indicateur plus pertinent que le pourcentage ou le nombre d'exigences satisfaites.
- On ne mesure pas un projet à la quantité de documents produits ou au nombre de lignes de code, non significatifs pour le client.

Les 13 principes : 8/13



Les méthodes agiles recommandent que le projet avance à un rythme soutenable.

- La qualité du travail fourni dépend du rythme du travail qui doit être adapté en fonction des spécificités du projet
- Le rythme doit être soutenu et soutenable sur la durée du projet.

Les 13 principes : 9/13



Sponsors, développeurs et utilisateurs devraient pouvoir maintenir un rythme constant indéfiniment.

- Ce rythme de travail est à déterminer par l'ensemble des membres de l'équipe et par le client, en fonction de la productivité de l'équipe et des priorités du client.
- Travailler en heures supplémentaires pour corriger des bogues ou des régressions n'apporte aucune valeur ajoutée.

Les 13 principes : 10/13



Porter une attention continue à l'excellence technique et à la conception améliore l'agilité.

- Maintenir un code propre, évolutif et performant est un objectif permanent de l'équipe.
- Il ne s'agit pas de produire du code non exploitable par les autres, ni du jetable.
- Cela évite d'enliser les développements ultérieurs, avec des modifications cassant un développement fragile, nécessitant des interventions à des endroits variés du code.

Les 13 principes : 11/13



La simplicité (art de maximiser la quantité de travail non fait) est essentielle.

- La simplicité garantit l'évolutivité du système.
- La complexité, au contraire coûte davantage et rend plus difficiles les évolutions inhérentes au développement incrémental.
- La conception doit ne comporter que des éléments utiles.

Les 13 principes : 12/13



Les meilleures architectures, spécifications et conceptions sont le fruit d'équipes qui s'auto organisent.

- Le chef de projet n'est pas celui qui attribue des tâches.
- L'équipe, elle-même, se responsabilise et définit ses travaux à réaliser.
- Le partage des tâches s'effectue sur la base du volontariat.

Les 13 principes : 13/13



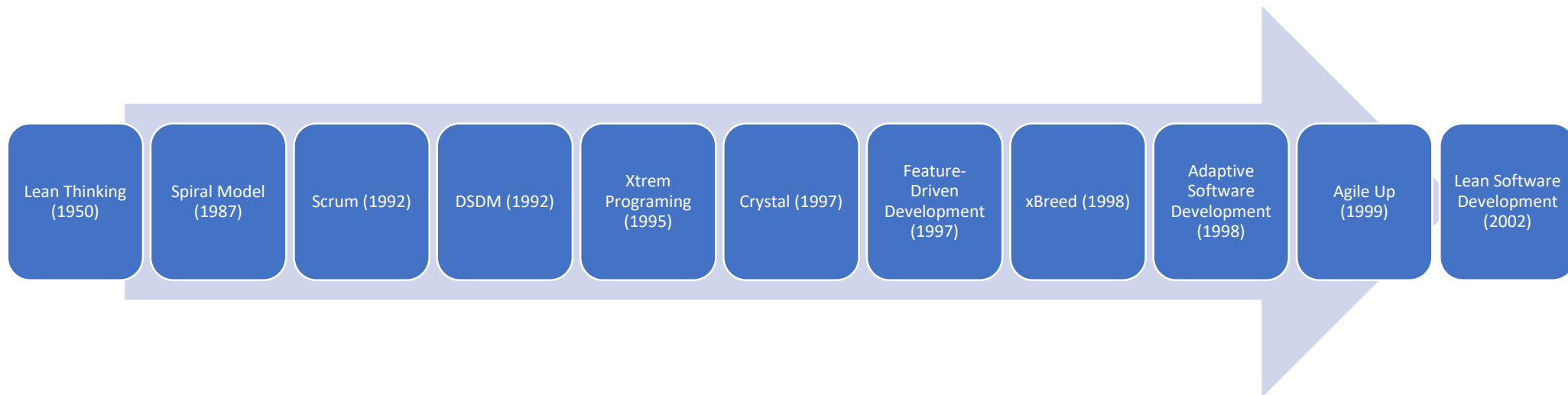
A intervalles réguliers, l'équipe s'interroge sur la manière de devenir encore plus efficace, puis ajuste son comportement en conséquence.

- L'environnement d'un projet n'est pas constant
- L'équipe s'interroge en permanence sur la façon d'améliorer son fonctionnement afin de s'adapter aux nouvelles conditions.
- C'est aussi l'acceptation du changement!

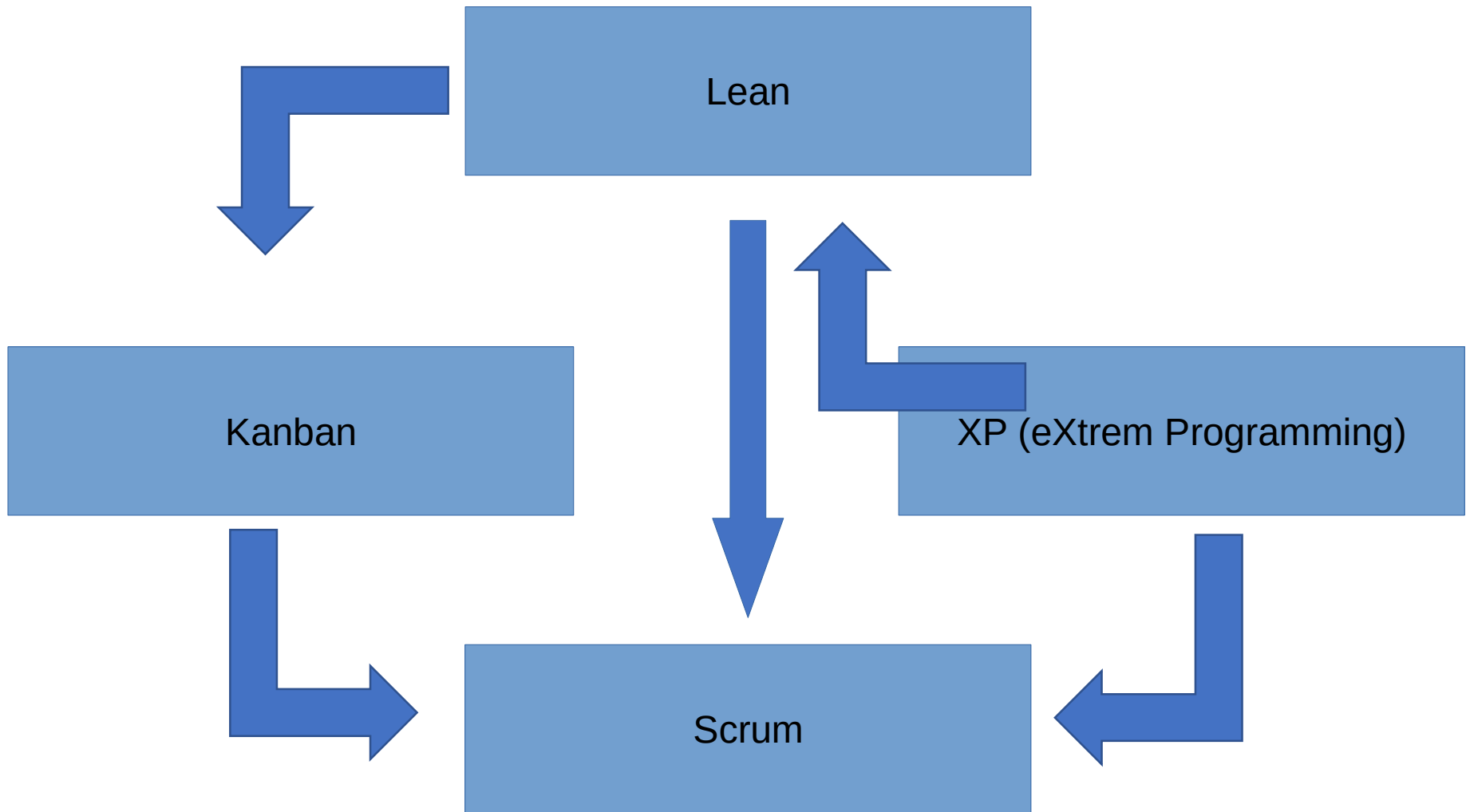
Méthodes Agile



Méthodes agiles



Méthodes agiles



Lean



Le lean management



Lean : maigre, dégraissé

Origine : Japon (Toyota), après la seconde guerre mondiale

Théorisé aux USA au début des années 1990

Objectif : recherche de la performance (productivité, qualité, coûts, délais) par l'amélioration continue et l'élimination des gaspillages (muda en japonais), afin d'améliorer la valeur globale pour le client.

Les 7 (ou 8) mudas



- Surproduction
- Attentes
- Transport
- Étapes inutiles
- Stocks
- Mouvements inutiles
- Corrections / retouches
- + un 8ème : la non-utilisation des ressources intellectuelles du personnel

Le lean software development

- Éliminer les gaspillages : éliminer ce qui n'a pas de valeur pour l'utilisateur
- Améliorer l'apprentissage
- Retarder l'engagement
- Livrer aussi vite que possible
- Donner le pouvoir à l'équipe
- Intégrer la qualité dès la conception
- Considérer le produit dans sa globalité

Lean : respect des personnes



N'inquiétez pas votre client

- Un client est quiconque consomme votre travail ou vos décisions.
- Analysez sans relâche la situation et changez pour arrêter de les inquiéter.
- Ne forcez pas les gens à réaliser un travail qui se traduit par du gaspillage.
- Ne leur donnez pas des anomalies.
- Ne leur imposez pas des vœux pieux.
- Ne les surchargez pas.

Les équipes et les individus développent leurs propres pratiques et améliorations

- Le management n'hésite pas à bousculer les gens pour qu'ils changent, à leur demander quoi améliorer, mais....
- Ce sont les gens qui apprennent à résoudre les problèmes et à étudier et à réfléchir aux processus, et qui...
- Décident d'eux-mêmes de la façon d'améliorer les choses.

« Développez d'abord les personnes, construisez ensuite les produits »

- Les managers ne sont pas directifs mais formateurs.
- Coachez de près les gens, pendant des années, en ingénierie et en résolution des problèmes.
- Formez les gens à l'analyse des causes racines et à rendre les problèmes visibles; ils découvrent ensuite comment améliorer les choses.

Les managers montrent l'exemple

Les managers comprennent les buts « éliminer le gaspillage » et « améliorer les choses en continu » et leurs actions et décisions vont dans ce sens ; les employés le voient.

« Respect des personnes »
en Lean

Construisez les partenariats

- Bâissez des relations pérennes basées sur la confiance.
- Aider les partenaires à s'améliorer et à préserver leur rentabilité.

Développez les équipes

- Équipes réellement stables de 5 à 6 personnes.
- Culture du travail en équipe et non pas d'un groupe d'individus.

Une méthode critiquée



Dans sa version « production » (en particulier dans le secteur automobile), le lean est parfois critiqué pour sa dégradation des conditions de travail (suppressions d'activités prévenant des troubles musculosquelettiques...)

Critiquée également pour favoriser des inégalités dans les formes d'emploi (main d'œuvre peu qualifiée flexible vs main d'œuvre qualifiée stable)

XP : eXtrem Programming

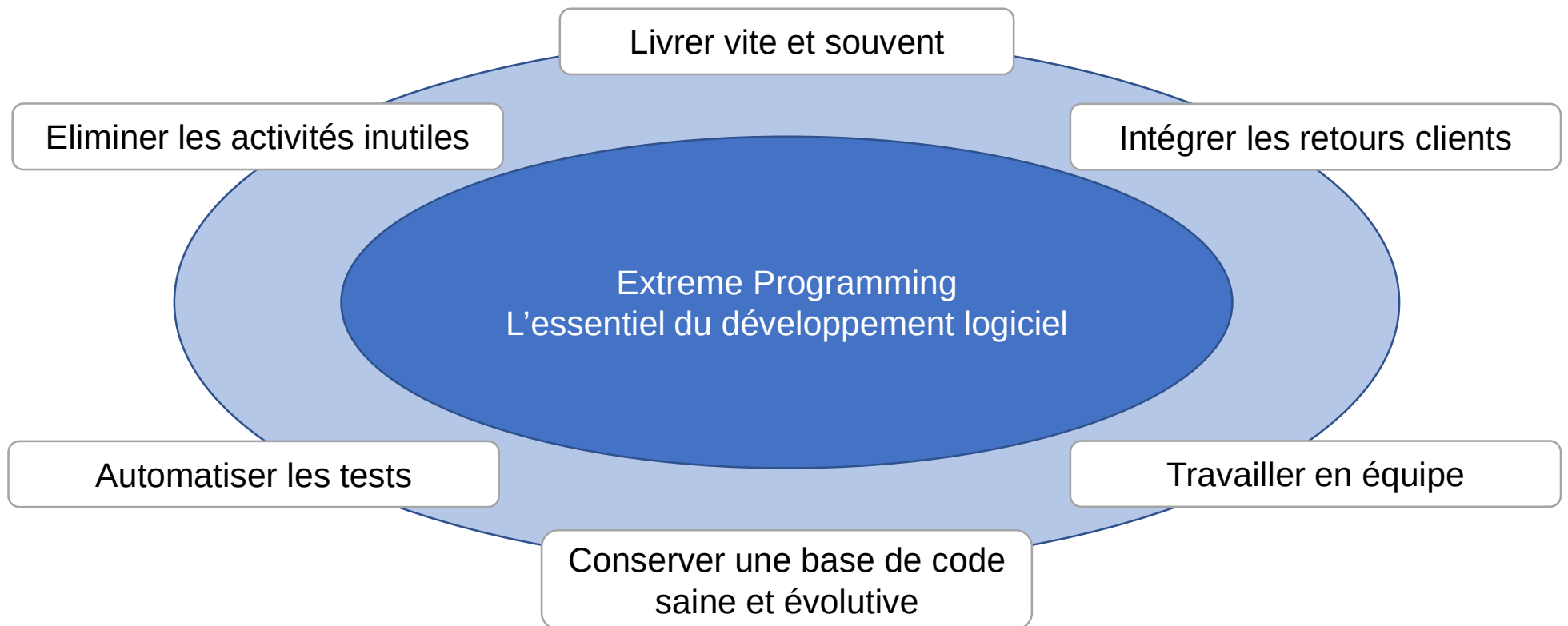


XP: principes



- Développement par module
- Qualité **NON NÉGOCIABLE**
- Client seul responsable du nombre de fonctionnalités (en fonction du temps et du budget)

XP: principes



XP: les 13 pratiques (1/2)



- Livraisons fréquentes
- Rythme constant et soutenable
- Client sur site
- Conception simple
- Règles de codage
- Équipe responsable du code
- Tests unitaires(TU) → Développement piloté par les tests (TDD)

XP: les 13 pratiques (2/2)



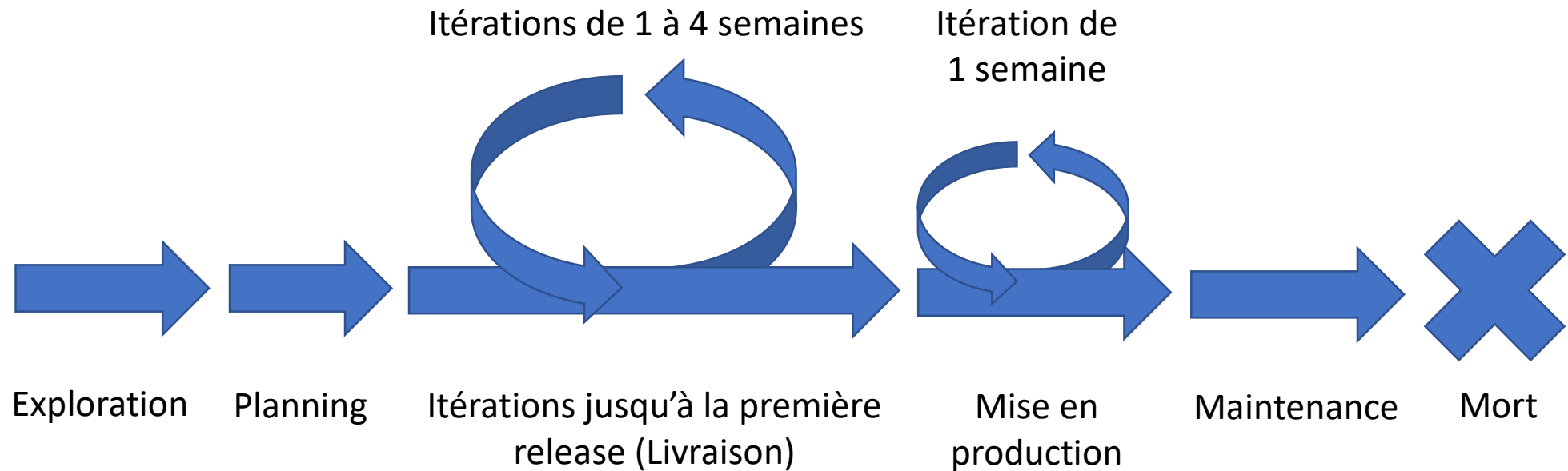
- .Tests de recette
- .Intégration continue
- .Refactoring de code
- .Programmation par binôme (Pair programming)
- .Planning Poker
- .Utilisation de métaphores et analogies

Le cycle XP



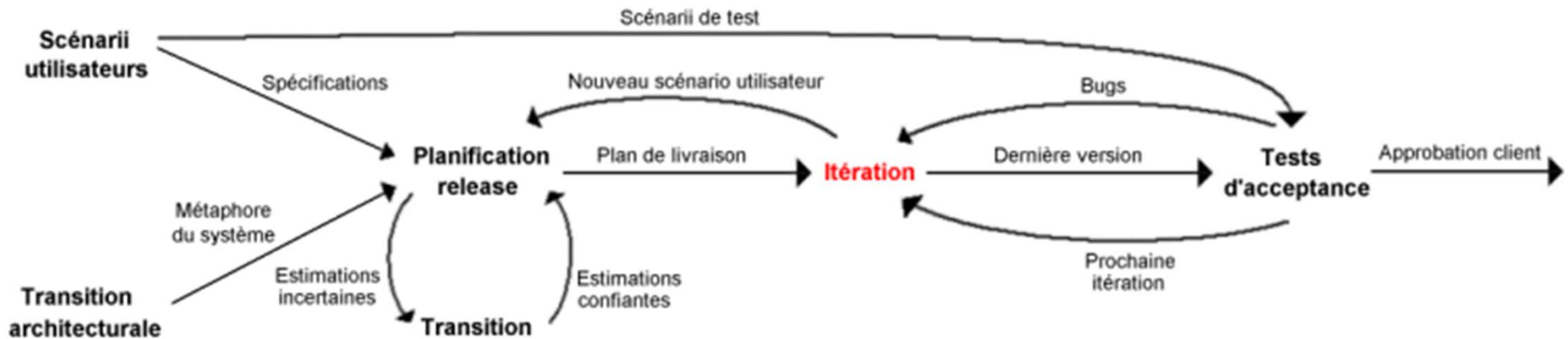
1. Le client rédige ses besoins (scénarios)
2. Les développeurs estiment la charge pour chaque scénario
3. Le client choisit les scénarios à intégrer dans la prochaine livraison
4. Chaque développeur prend la responsabilité d'une tâche
5. Chaque développeur se choisit un partenaire
6. Le binôme écrit les TU (Tests Unitaires)
7. Réorganisation du code existant puis implémentation de la nouvelle fonctionnalité
8. Intégration des nouveaux développements

Le cycle XP



Maintenance : ajouter des fonctionnalités → nouvelles releases qui utilisent le même processus que pour le release 1

Le cycle XP



SCRUM, Un mix des méthodes



De Lean à Scrum



- Adaptation
- Introspection
- Usage de post-its
- Qualité

De Kanban à Scrum



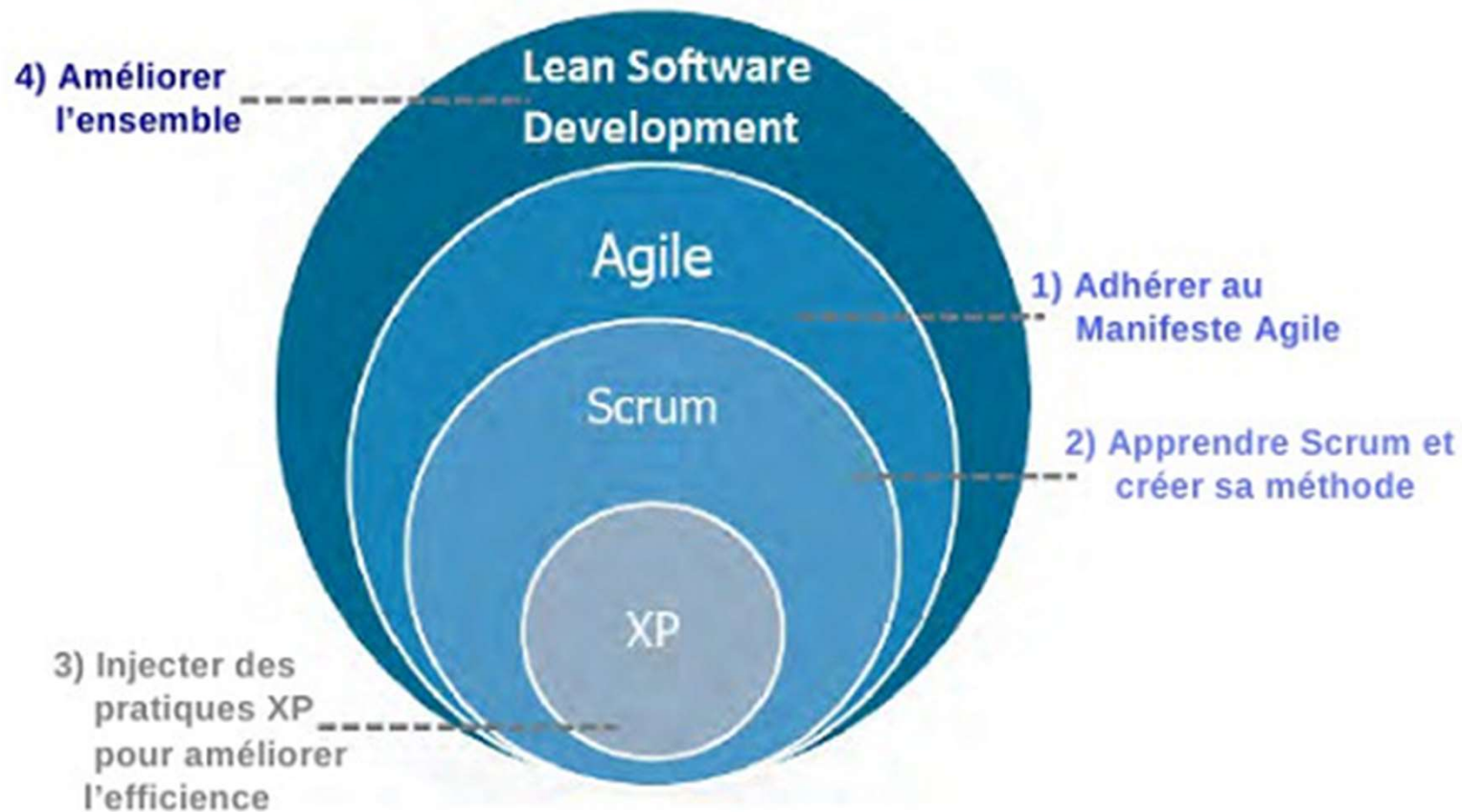
- Écriture des besoins sur une étiquette
- Visualisation du workflow sur un tableau
- Approche empirique

De XP à Scrum



- Livraisons fréquentes
- Rythme soutenable
- Client faisant partie intégrante du projet
- Équipe responsable
- Tests
- Planning Poker
- et à peu près tout le reste en option, mais recommandé

De XP à Scrum



Adopter une approche agile



- Dresser l'état des lieux
- Recenser les zones de dysfonctionnement
- Poser les bonnes questions
- Risques ou facteurs clés de réussite ?
- Fixer des objectifs réalistes
- Comment mesurer le succès de la démarche ?
- Quels sont les symptômes de l'échec de la démarche ?
- Initialiser la conduite du changement

Management Visuel

Le projet en « un coup d'œil »



Le projet en « Un coup d'œil » (1)



L'agilité s'appuie, de la préparation du projet à sa fin, sur des représentations visuelles des tâches, avancements, flux de travail, etc.

Les fonctionnalités sont souvent des fiches contenant son titre, sa fonction, son test, et elles sont déposées sur un tableau qui indique leur état d'avancement, la fiche passe d'un état à l'autre jusqu'à sa résolution.

L'avancement du projet est souvent représenté par des graphiques qui montrent le nombre de fonctionnalités prévues pour le projet à celles effectivement finies.



Le projet en « Un coup d'œil » (2)



Comme un projet agile ne commence pas sur un nombre défini et immuable de fonctionnalités, les outils sont adaptés à cette évolution

Par essence, un projet agile peut ne jamais connaître de fin, car des fonctionnalités nouvelles peuvent être demandées et des fonctionnalités prévues peuvent être jugées obsolètes donc retirées du projet.

Les outils de pilotage sont adaptés au court, moyen et long terme, selon la méthodologie adoptée (Scrum, Kanban, etc.)



Kanban



Kanban : Définition (1)



le Kanban (terme japonais signifiant “enseigne, panneau, étiquette”)

Il facilite la collaboration entre les membres de votre équipe agile. C'est aussi une méthode efficace pour surveiller l'amélioration continue du produit ou du service à développer.

Conceptualisé par Toyota en 1950, la méthode repose sur 4 principes :

- Commencer par ne rien changer (rôles, processus)
- Accepter les changements continus et évolutifs
- Respecter le processus actuel, les rôles, les responsabilités et les titres
- Leadership à tous les niveaux

Kanban : Définition (2)



	SCRUM	KANBAN
Cadence	Sprints réguliers, à durée fixe	Flux continu
Méthodologie de livraison	A l'issue de chaque sprint, si approuvé par le Product Owner	Livraison continue ou à la discrétion de l'équipe
Rôles	Product Owner, Scrum Master, Team Members	Aucun rôle existant. Certaines équipes sollicitent l'aide d'un coach Agile
Métriques clés	Vélocité	Durée de cycle
Philosophie du changement	Les équipes devraient s'efforcer de ne pas apporter de changements à la prévision du sprint durant celui-ci. Ce faisant, elles compromettraient les enseignements tirés de l'estimation.	Le changement peut avoir lieu à tout moment

Kanban : Concepts



Fondé sur un système à "flux tirés", tenant compte des demandes des consommateurs, et non à flux poussés.

L'objectif consiste à s'adapter en permanence au besoin du client, pour limiter le risque de surproduction et de gaspillage, mais aussi réduire les délais et les coûts.

Kanban prône la visualisation des flux de travail par le biais d'un tableau (dit tableau Kanban), permettant de prioriser et suivre l'état d'avancement des tâches à accomplir.

Kanban



Flux tiré : Produire et, donc approvisionner au prorata de la demande réelle.

Flux poussé : Chaque personne sur une chaine de production va pousser le travail qu'il fait à la personne suivante avec comme seule contrainte de ne prendre du travail que si il y en a de disponible pour lui.

Kanban : pratiques



La méthode s'appuie sur 5 grandes pratiques :

- Visualiser le workflow (tableau kanban)
- Limiter le nombre de tâches en cours
- Gérer le flux de travail (suivi, mesures, rapport)
- Rendre explicite les processus de gestion de travail
- S'améliorer collectivement

Kanban dans la gestion de projet



Kanban est donc une méthode qui a pour but de faire évoluer le projet, mais aussi les méthodes de production et l'équipe.

Le chef de projet devient un accompagnateur et un formateur sur la détection et résolution des blocages, ainsi qu'un validateur des optimisations proposées par l'équipe.

L'équipe devient donc plus qu'un groupe d'individus exécutants, mais une équipe travaillant et réfléchissant de concert.



Démarche Kanban

La démarche est donc de valoriser le travail fini, les blocages surmontés et les améliorations mises en places.

Les membres de l'équipe deviennent des forces de proposition, car les améliorations ne concernent pas que les méthodes, mais aussi le cadre de travail, les outils, le management, etc.

Il devient alors facile de suivre l'évolution d'un projet et de s'adapter à de nouvelles demandes.



Concevoir un système Kanban



Cadre du système



Kanban s'applique mieux quand :

- Les priorités changent souvent
- Les tâches peuvent mettre plus d'une ou deux semaines pour être achevées
- L'équipe a des difficultés pour évaluer estimations et / ou priorités
- L'équipe possède une forte culture d'amélioration en continue et d'auto-organisation
- L'équipe tend vers l'expertise concernant le TDD, l'intégration / livraison continue, etc.
- La priorité de l'équipe est la réactivité aux changements de besoins des clients
- L'équipe a tout d'abord des réticences sur l'adoption de l'agilité



Éléments de travail



Appliquer la méthode Kanban, c'est :

- Compartimenter le projet en petites portions ou features (items)
- Posséder un Kanban Backlog contenant les items
- Admettre que le Kanban Backlog n'est pas immuable
- Utiliser un Kanban Board
- Mettre en place des méthodes de suivi quotidiennes (Daily meeting, Burn Up Chart)
- Utiliser le TDD
- Faire de l'intégration et de la livraison continue



Items



Les items représentent les features du projets.

Ces features sont découpées en tâches (user story, bug, fonctionnalités, etc.) qu'on dénommera items.

Ces items on la particularité d'être définis simplement sur des cartes de petite taille qui seront mises dans le Kanban backlog et qui seront utilisées sur le Kanban Board.

Ces « items » sont rédigés sous la forme de cartes, faciles à lire.

Une « carte » comprend généralement le nom de l'item, sa fonction, sa description. Elle contient souvent sa priorité, la description de sa DoD (Definition Of Done) => les tests de validation client.

Si les story points sont utilisés pour la définition de la complexité, ils apparaissent ainsi que la valeur de risque si elle est aussi utilisé

Une carte comprend aussi des espaces dédiés, pour indiquer la date d'ajout au backlog, le nom de l'équipier qui la prendra en charge.

Il est possible de définir le type de l'item en donnant une couleur à la carte (vert = fonctionnalité, rouge = bug, jaune = Epic, etc.)



Kanban Backlog (1)



Le Kanban Backlog est tout simplement la liste des tâches, fonctionnalités, etc. qui sont nécessaires à la création du produit.

Le Kanban Backlog n'est pas immuable :

- Le Product Owner (le client) peut à loisir y ajouter ou en retirer de nouveaux items.
- Des items reconnus comme doublons peuvent en être enlevés
- Des items peuvent aussi y être ajoutés par l'équipe si elle considère qu'un autre item en dépend.

Dans un tableau Kanban, le backlog en est souvent la première colonne quand il y apparaît.



Kanban Board (1)



C'est l'outil majeur utilisé durant tout le projet

Ce tableau est un outil qui permet de déterminer le travail de l'équipe à l'instant T et il permet aussi d'avoir une vision en temps réel de l'avancement du projet.

Dans sa version la plus simple, il possède 3 colonnes :

To Do (A faire, le backlog), Doing (les tâches en cours par l'équipe, le nombre de carte dans cette colonne ne peut pas excéder le nombre de personnes dans l'équipe, Done (Les tâches finies et généralement validées et souvent intégrées dans le projet).

On peut ajouter d'autres colonnes :

Blocked (les tâches qui se trouvaient en Doing mais qu'un impondérable en empêche la réalisation), Testing (si on veut libérer la colonne To Do pour commencer de nouvelles tâches pour ne pas avoir à les mettre en Done alors qu'elles n'ont pas encore été validées), etc.

Il est important de limiter le nombre de colonnes pour garder l'aspect « d'un coup d'œil » de l'outil.



Kanban Board (2)



Il représente le Workflow

Se sont les colonnes qui le compose et leurs définitions qui vont représenter le workflow.

C'est donc un workflow assez linéaire avec peu (voir pas) de retour en arrière pour un item.

Il peut être décomposé en plusieurs tableaux

Il se peut que certains items aient besoin d'une phase d'analyse avant de partir en « développement », on peut alors créer un sous-tableau d'analyse avec par exemple les colonnes « Doing » et « Done ».

Certaines équipes préfèrent aussi séparer la validation dans un sous tableau, dans le même principe que l'analyse, avec aussi deux colonnes « Doing » et « Done ».

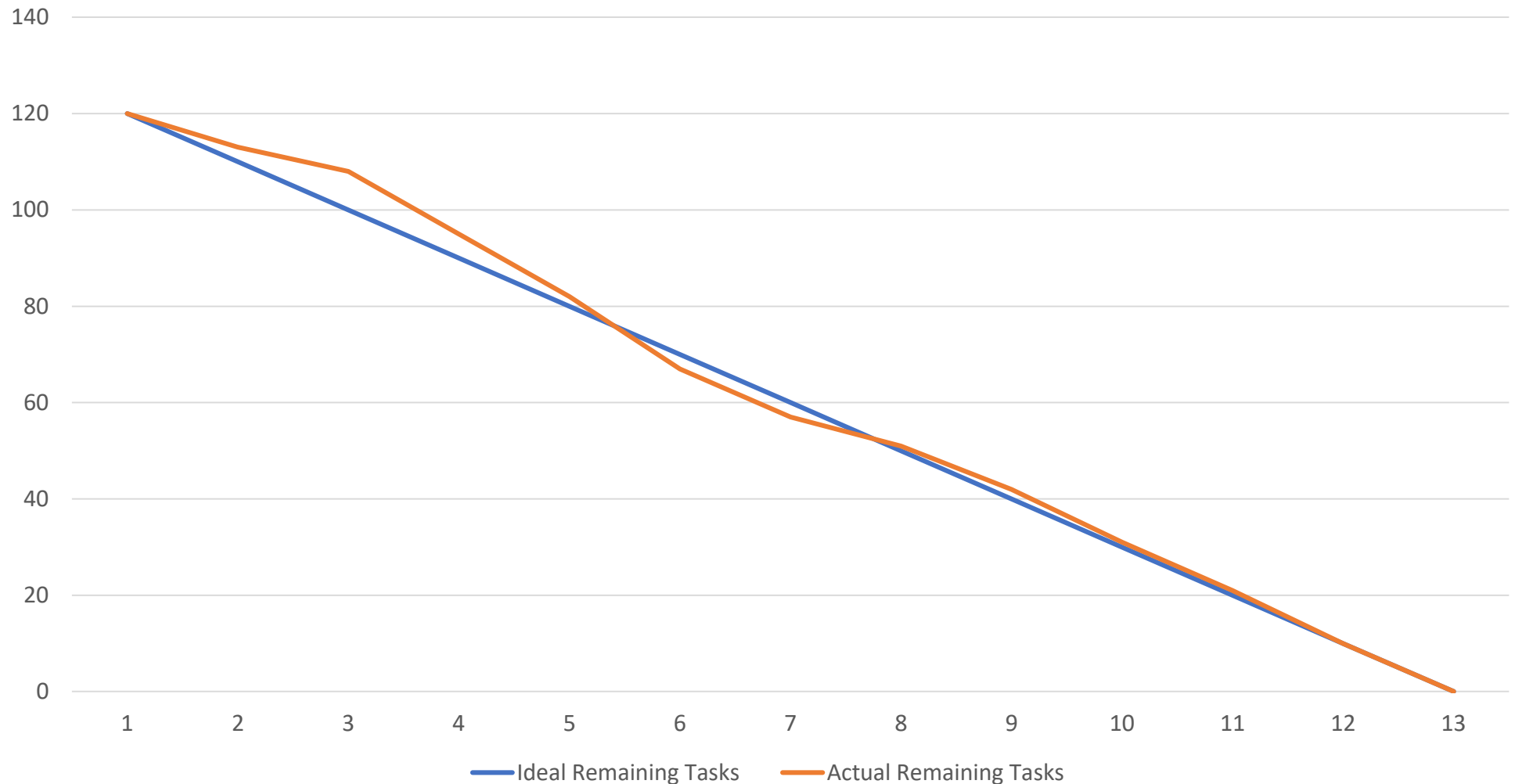
Ces choix dépendent de l'équipe (taille, façons de travailler), du chef de projet (suivi ultra détaillé ou non, sensible à la qualité, etc.).



Le Burn Down Chart, un outil de suivi (1)



Burn Down On Remaining tasks



Le Burn Down Chart, un outil de suivi (2)



Utilisation du Burn Down Chart

Dans la diapositive précédente, le graphique indique un nombre de tâches de départ (en Story Points, SP), pour arriver à 0.

Il existe deux courbes :

Ideal Remaining Tasks : le nombre idéal de tâches qu'il reste à faire à chaque jour d'avancement du projet, exprimé en SP

Actual Remaining Tasks : Le nombre réel de tâches qu'il reste à faire à chaque jour d'avancement du projet, exprimé en SP

Généralement il est utilisé avec la méthode SCRUM avec un graphique par itération (le nombre de SP et de jours est défini à chaque sprint) mais peut être adapté pour le Kanban.

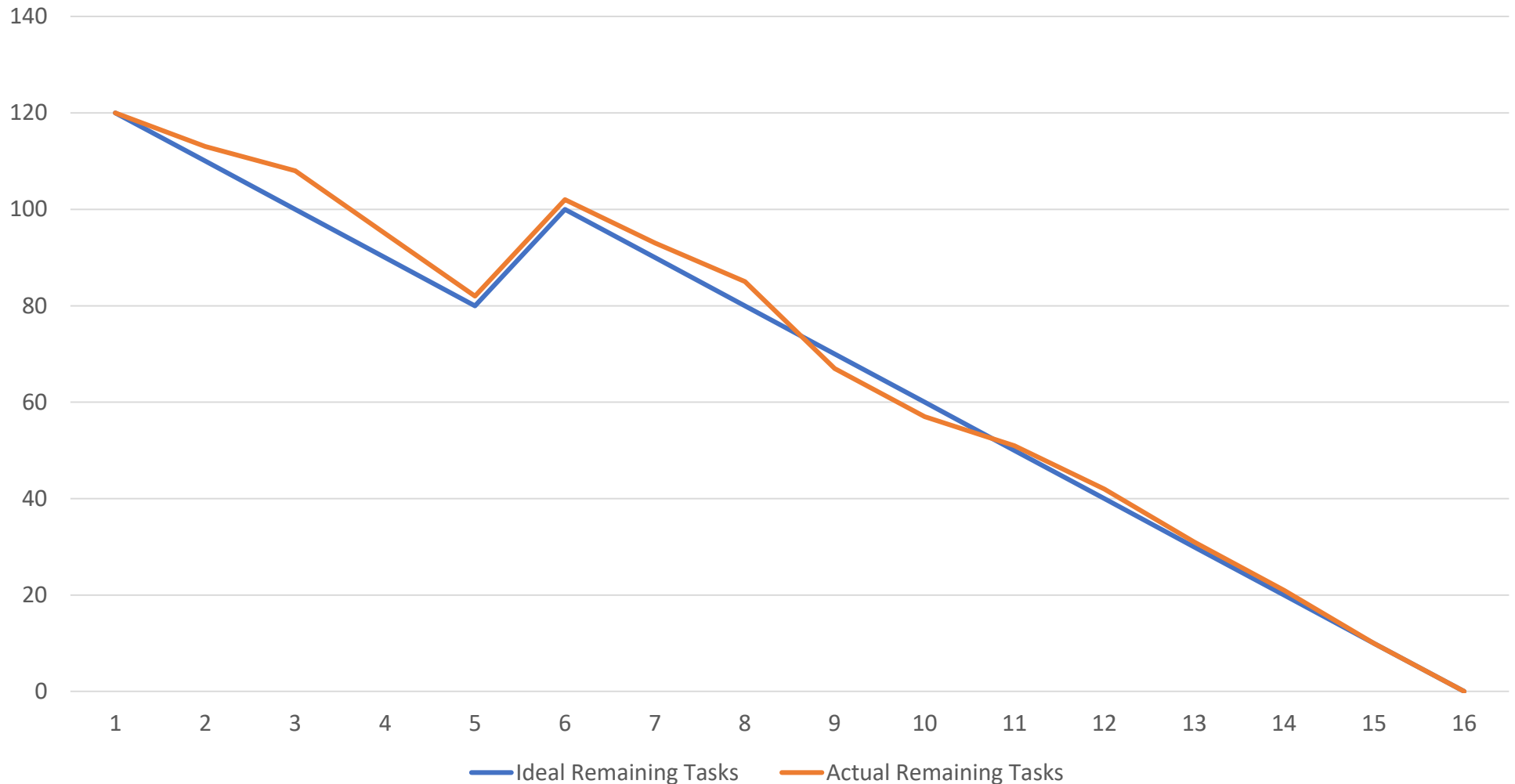
Dans le graphique suivant, 20 SP ont été ajouté au backlog, ce qui augmente aussi le réel restant de vingt SP



Le Burn Down Chart, un outil de suivi (3)



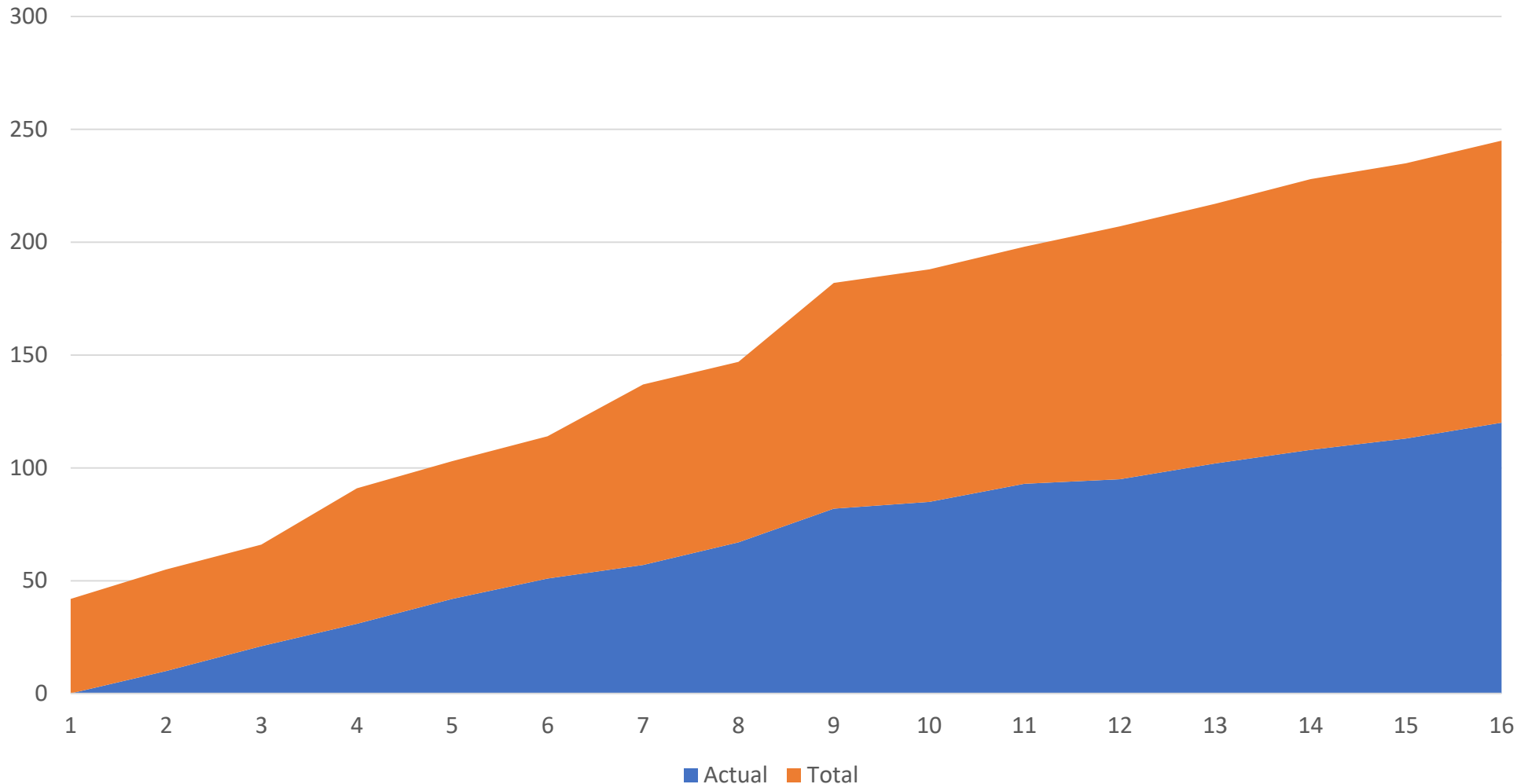
Burn Down On Remaining tasks day to day basis



Le Burn Up Chart, un autre outil de suivi (1)



Burn Up On Done SP and Total Project SPs Daily basis



Le Burn Up Chart, un autre outil de suivi (2)



Utilisation du Burn Up Chart

Dans la diapositive précédente, le graphique indique un nombre de tâches de départ (en Story Points, SP). Comme il s'agit de la méthode Kanban, il n'y a pas de sprint (durée) établi donc les SP du projets peuvent continuer d'augmenter ainsi que les SP réalisés au cours du projet.

Il existe deux courbes :

Total : Nombre total du SP du projet au jour J, ce nombre évolue au cours du projet et devient aussi un indicateur important sur la santé du projet.

Actual : Le nombre réel total réalisé sur le projet au jour J. Les différences d'évolution et les écarts qu'il a avec la courbe du total devient aussi un indicateur important sur la santé du projet.

Il peut être utilisé avec la méthode SCRUM, son grand intérêt est d'être un bon outil de suivi pour la méthode Kanban



Le Burn Up Chart, un autre outil de suivi (3)



Particularité

La courbe Actual ne pourra jamais redescendre, car tout ce qui est produit (SP) ne peut pas être retiré.

La courbe Total comme indicateur de santé

Cette courbe peut redescendre :

Nettoyage de backlog (Backlog grooming) :

il peut être parfois considéré comme une boîte à idée et une petite révision permet de retirer les items obsolètes ou qui ne seront jamais réalisés.

Ce genre de « nettoyage » arrive fréquemment quand la courbe Total monte beaucoup plus vite que l'Actual, phénomène fréquent où les demandes affluent plus vite que la capacité à produire.

Si la courbe stagne :

Il faut s'inquiéter de l'agilité du projet, le projet reste identique du début à la fin donc pas (ou peu) de feedback client (négatif).

La courbe monte d'un coup :

Pas alarmant mais peut soit signifier

- Affluence d'un gros nouveau lot d'items (pas optimal)
- Très peu de feedback régulier (négatif)



Test Driven Development (TDD)



Le TDD est une méthode qui permet d'assurer la qualité des créations du projet.

Chaque item possède une Definition Of Done (DoD = les tests qui assureront ou non sa validation).

Les développements sont donc faits pour réussir ces tests

Les tests sont systématiques (Test Unitaires sur chaque Item)

Le client valide ou non un item en suivant **RIGOREUSEMENT** le DoD (Qu'il a souvent écrit lui-même).

Il est donc très **IMPORTANT** que le DoD soit bien réfléchi à l'avance, il devient généralement le complément indispensable des user stories.



Intégration et / ou Livraison continue



Intégrer et / ou de livrer en continue permet de rassurer le client et la hiérarchie et de suivre l'évolution du projet.

L'intégration continue permet de tester facilement les nouveaux items avec ce qui a déjà été produit, ce qui améliore sa qualité.

L'intégration continue permet aussi de montrer régulièrement les avancées sur l'application au client entre les livraisons.

L'intégration continue permet de montrer la réactivité de l'équipe en intégrant toujours en premier les demandes à haute valeur ajoutée.

La livraison continue donne aux utilisateurs une impression constante de l'évolution du produit, donc de sa qualité, ce qui a pour effet d'augmenter leur confiance envers le fournisseur.



Le WorkFlow (1)



Le Kanban Board ne représente qu'une partie du Workflow, il existe quelques étapes supplémentaires.

Avant d'utiliser le tableau, il faut avoir une première étape de définition du besoin par le client.

Cette étape permet de créer le backlog qui ira alimenter le tableau Kanban.

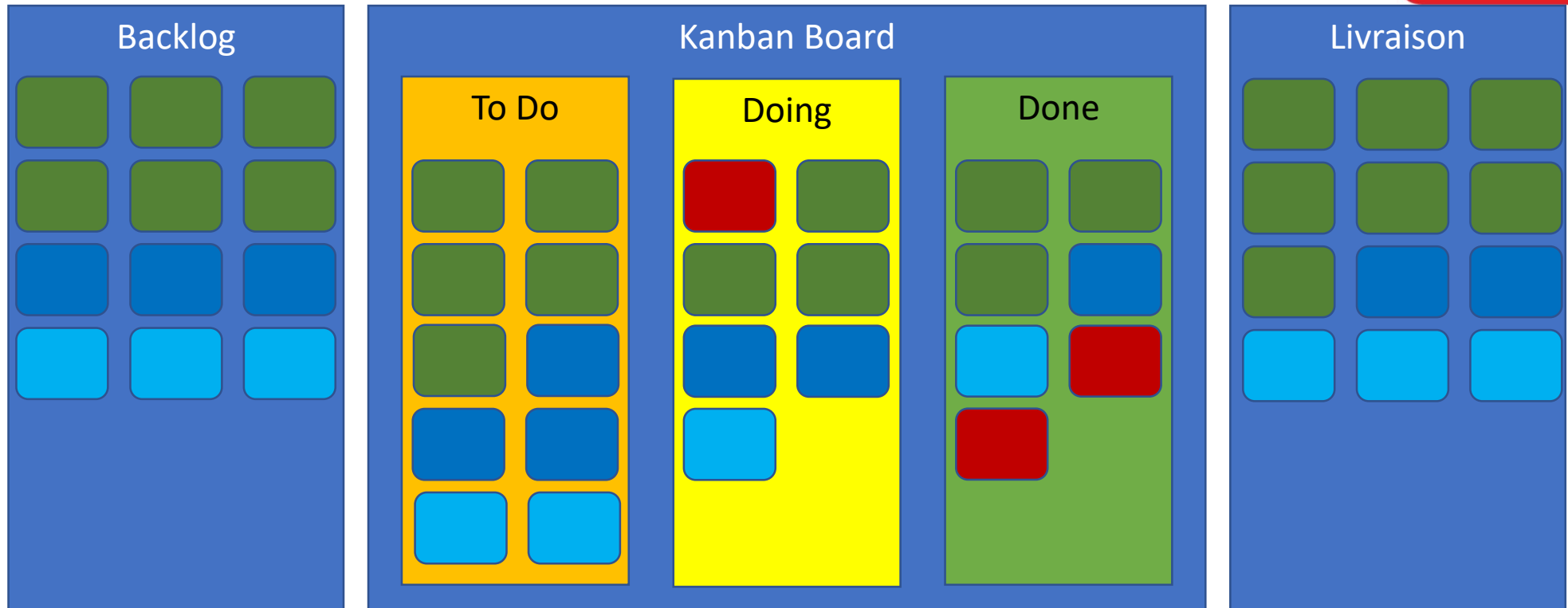
L'intégration continue permettra alors au client de donner rapidement un feedback sur ce qui a été fait, car il valide les items et aussi revoit régulièrement ses besoins. Il produit alors de nouveaux items qui alimentent le backlog.

C'est aussi lui qui va prioriser régulièrement les tâches restantes du backlog (selon ce qui a déjà été produit, selon les nouveaux items, etc.).

Le projet n'est déclaré fini que par le client, si il considère que la dernière livraison satisfaisait ses attentes, même s'il reste un certain nombre d'items dans le backlog.



Le WorkFlow (2)



- Le backlog est alimenté par le client, tout au long du projet. Les items sont souvent rangés par fonctionnalité et priorité.
- Le client sélectionne une partie d'items à inclure dans le To Do (souvent par fonctionnalité complète et par priorité)
- Les bugs arrivent directement dans la To Do en priorité maximale et un item en cours doit être mis en pause tant que le bug n'est pas résolu
- Une tâche validé ou un bug résolu (et validé) passe dans la colonne Done.
- Les tâches passées de Done à Livraison font partie de la livraison (souvent par fonctionnalité finie)



Le WorkFlow (3)



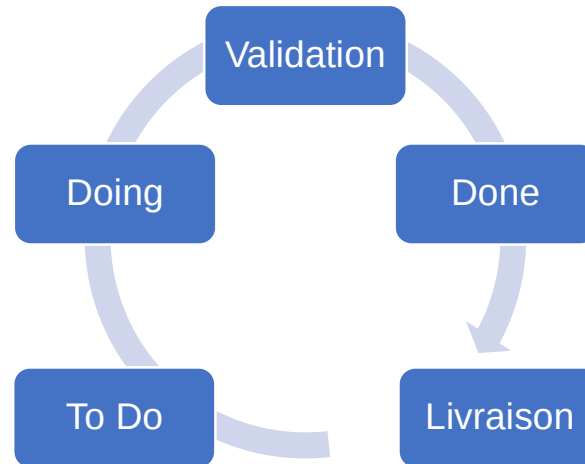
Définition des règles

En début de projet, le client et l'équipe définissent ensemble les règles qui permettent de passer d'une colonne du tableau à l'autre.

Ces règles ne sont pas immuables, car le but du Kanban est d'améliorer en continu le workflow, la qualité de l'équipe, la qualité du produit, selon les facteurs humain, architecturaux, logistiques, etc. ainsi que des retours des clients.



Visualisation du système



Début du projet

- Définition du besoin
- Préparation du backlog

Intégration continue

- Livraisons régulières
- Alimentation régulière du backlog

Fin du projet

- Le client est satisfait par les fonctionnalités livrées



Les limites du Kanban



Par nature, un projet Kanban peut ne jamais s'arrêter

Comme le développement est soumis aux priorités et aux retour des clients / utilisateurs, un projet Kanban peut ne jamais voir son backlog se vider.

S'il n'est pas régulièrement entretenu (nettoyage, alimentation), le backlog peut fausser le suivi du projet.

La durée de vie du projet est donc soumis à la volonté et au budget du client. Une durée et un coût de départ sont définis mais le client peut ou non demander des rallonges (fonctionnalités supplémentaires).

Le client peut, du jour au lendemain, partir à la concurrence avec son backlog et sa dernière livraison.

Le Kanban, s'il ne respecte pas les règles agile d'amélioration des équipes, peut cloisonner certains membres à des compétences spécifiques et limiter leur rôle au sein du projet, rappelons que le Kanban (et l'agile) compte sur l'augmentation des connaissances et des spécialités des membres des équipes.



Définitions des cadences



Contrairement à SCRUM, Kanban n'a pas de sprint défini

SCRUM calcule la vélocité (nombre de SP que peut produire l'équipe par sprint) pour avoir une certaine estimation de la durée d'un projet.

Ce sont le client et l'équipe qui conviennent du nombre de tâches à avoir dans le To Do Régulièrement.

Une pratique commune est de mettre une voir deux fonctionnalités (divisées en une série d'item) dans la colonne To Do. Cela donnera lieu à une itération du projet qui sera potentiellement livrable.

La cadence de livraison est aussi définie par le client et l'équipe : toutes les n itération ou quand un certain nombre de fonctionnalités finies rendent la livraison conséquente (patch, montée de version, options, modules supplémentaires, etc.)

L'intégration continue (fréquentes itérations) et la livraison continue assurent une satisfaction du client et généralement la qualité du produit.



Mise en œuvre du système



La méthode Kanban est un moyen de transition vers l'agilité

Contrairement à SCRUM, qui demande une adhésion dès le départ à 100% de l'équipe et du client aux principes de l'agilité, le Kanban permet une introduction en douceur.

Généralement, le début de la conversion se fait par l'utilisation du tableau, pour donner une vision générale du projet autant aux équipes, aux clients et à la hiérarchie.

La transformation des longues réunions hebdomadaires en réunion quotidienne courte d'équipe permet de visualiser rapidement les blocages et les problèmes sur le projet.

Ensuite, introduire l'intégration continue permet de voir le produit évoluer jusqu'à la réalisation du cahier des charges.

Elle permet aussi aux clients de voir rapidement si leurs spécifications correspondent à leur besoin véritable.

A partir de ce moment, il n'y a que quelques étapes pour passer en full Kanban (adaptation du cahier des charges en backlog, implication totale du client sur le projet, durée du projet donc monétisation du projet définie par la qualité des livraisons et la satisfaction du besoin client).



Suivi quotidien



Daily stand-up meeting, ou l'art de ne pas perdre sont temps

Souvent les projets classiques se trouvent sclérosés par des réunions qui finalement n'apporte que très rarement quelque chose au projet et à sa réalisation.

Dans l'agilité, il convient de faire, dès le début de la journée, un point rapide d'une quinzaine de minutes.

Chaque membre de l'équipe revient sur sa journée de projet précédente en répondant simplement à ces trois questions :

- Ce qu'il a fait la veille
- Les obstacles qu'il a rencontré
- Ce qu'il compte faire aujourd'hui



Étendre le Kanban



Diffusion



Le Kanban vient de la production industrielle.

Adapté aux projets, il n'utilise qu'une partie des principes utilisés en Industrie.

En Industrie, le principe Amont et Aval est représenté par le fournisseur précédent de tâches ou de matériel pour le faiseur suivant : le producteur des pièces qui fournis à l'atelier d'assemblage.

Le fournisseur fourni à la demande et ne produit que selon les demande, à flux tiré, pour limiter le stock

L'assemblage n'assemble qu'à la demande, à flux tiré aussi, pour limiter le stock

Les deux postes améliorent la qualité et diminuent le gâchis.



Kanban amont



En projet, l'amont devient le kanban backlog

Le kanban backlog, évolutif, représente l'amont. Il va alimenter régulièrement l'équipe (aval) selon la demande et la capacité de production de l'équipe.

Comme le but est de pouvoir réagir rapidement au changement (retour utilisateurs, nouvelles fonctionnalités, etc.) ce que donne l'amont à l'aval n'est pas forcément les demande les plus anciennes, il s'agit souvent des demandes les plus importantes et / ou les plus urgentes.



Kanban pour le Product Owner



En Kanban, il n'y a pas de rôle comme dans Scrum

Comme l'amont est représenté par le kanban backlog, la présence d'un PO n'est pas nécessaire. Néanmoins il est plus facile que les demandes ne viennent que d'une seule personne qui possède la connaissance métier nécessaire pour savoir les prioriser et alimenter l'équipe avec les demande les plus pertinentes.

Ce rôle est aisément occupable par un PO habitué à SCRUM.



Kanban Ops (1)



Peut-on utiliser Kanban dans une équipe Infra ?

Une équipe Infra va gérer les projets business et les projets IT, mais aussi le flux de changement (changement d'un PC, arrivée d'un collaborateur, achat serveur, etc.) et le flux d'incident.

Autant les flux projets peuvent être planifiés en sprint SCRUM, le flux de changement devrait être perçu comme planifiable mais il n'est pas facile (voir impossible) de le planifier en sprint SCRUM, mais en période courte (quelques jours).

Le flux d'incident, lui, n'est tout simplement pas planifiable.



Kanban Ops (2)



Utiliser différentes méthodes

Les projets Business et projets IT peuvent être gérés avec SCRUM

Le flux de changement (changement d'un PC, arrivée d'un collaborateur, achat serveur, etc.) et le flux d'incident peuvent eux être géré par Kanban.

La partie projet nécessite donc bien un PO, mais est-il nécessaire d'en avoir un pour la partie Kanban ?



Kanban Ops (3)



Un PO sur du Kanban ?

C'est tout d'abord les managers qui vont décider si la gestion des flux sera en auto-organisation (l'équipe) ou si elle désigne une personne en charge de gérer ces flux. Cette personne devra souvent expliquer ses décisions sur la priorisation des tâches de flux.

la priorisation devra être faite avec précision.

La priorité est généralement établie selon le modèle suivant :



Kanban Ops (4)

Priorisation des tâches flux (change & incidents)

Selon les tâches reçue, une priorité normale subira une augmentation selon les critères suivants.

Type de tâche	Type d'augmentation	niveau d'augmentation	Définition	Intérêt
Urgence	Immédiate	Brutale	Anomalie bloquante en production	Pouvoir gérer les impacts (limites)
Standard	Linéaire par rapport au temps de traversée moyen	Normale	La majorité des fonctionnalités des projets	Pouvoir gérer la vitesse (capacité, flux, cadence)
Date Fixe	A partir d'une certaine date	Brutale ou non	Les tâches ayant des dates limite d'exécution	Pouvoir gérer la conformité (engagement, périmètre)
Intangible	Variant	Variant	Il n'est pas visible à plusieurs magnitudes du temps de traversée. Typiquement pour les évolutions techniques, migration, remaniement, montée de version, etc.	Pouvoir gérer le long terme

Un blocage de paiement en prod par exemple, deviendra une urgence immédiate qui passe en premier dans le flux des demandes.

La mise en place de l'inscription à la newsletter pour la fête de la musique aura une priorité basse jusqu'à début juin ou elle augmentera chaque jour.



Kanban Portfolio (1)



Qu'est-ce qu'un portfolio Kanban ?

C'est le portefeuille projets de l'organisation, du département, de l'équipe, représenté de façon visuelle, qui va *aligner* tout le monde, *communiquer* de façon égale et directe à tous, pousser à *gouverner*, à *s'améliorer*.

- Montrer la réalité et que la réalité
- Adapter la capacité pour favoriser valeur et apprentissage concernant votre stratégie.
- Adapter au mieux à la capacité et impliquer avec un flux tiré

A faire généralement quand il n'y en a pas, quand on a besoin de gouvernance à cause de l'amplitude de l'activité...



Kanban Portfolio (2)



Mise en place

Un portfolio s'utilise quasiment de la même façon qu'un tableau Kanban, à cela que les items deviennent des fiches projet sur lesquelles chacun travaille.

On commence par les classer en « A faire », « En cours », « Fini / En prod ».

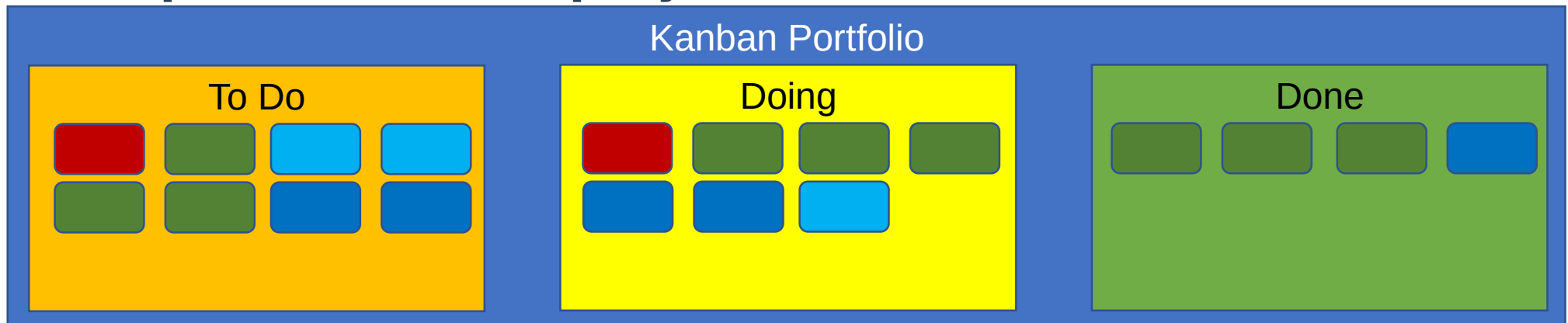
Il s'agit du premier tri, il convient ensuite d'affiner la granularité, par exemple en utilisant les colonnes « Idées / demandes », « Sélection », « Étude / préparation », « réalisation », « Validation », « déploiement », « En prod / Live ».

Le classement des projets dans ces colonnes doit être la réalité (pas les envies ou projection) sinon le portfolio ne sert à rien.

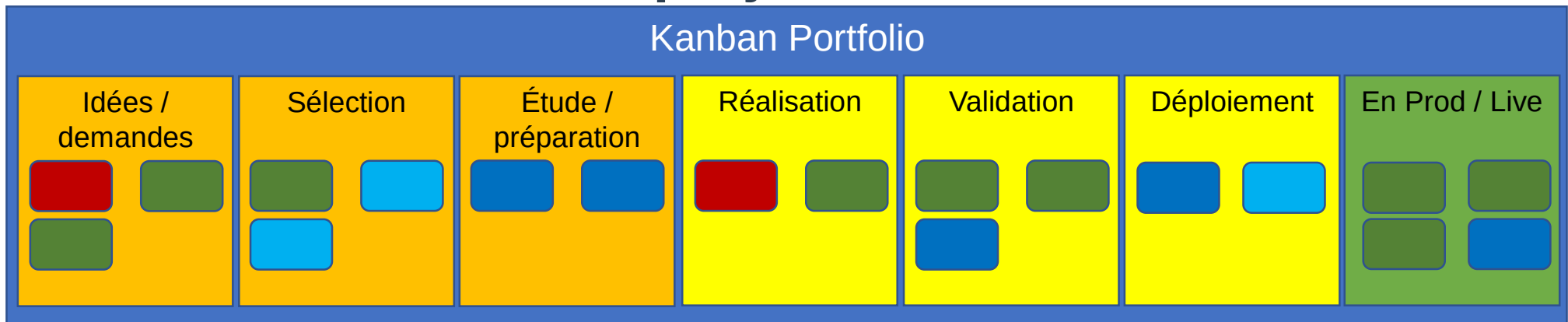


Kanban Portfolio (3)

Tri « primaire » des projets



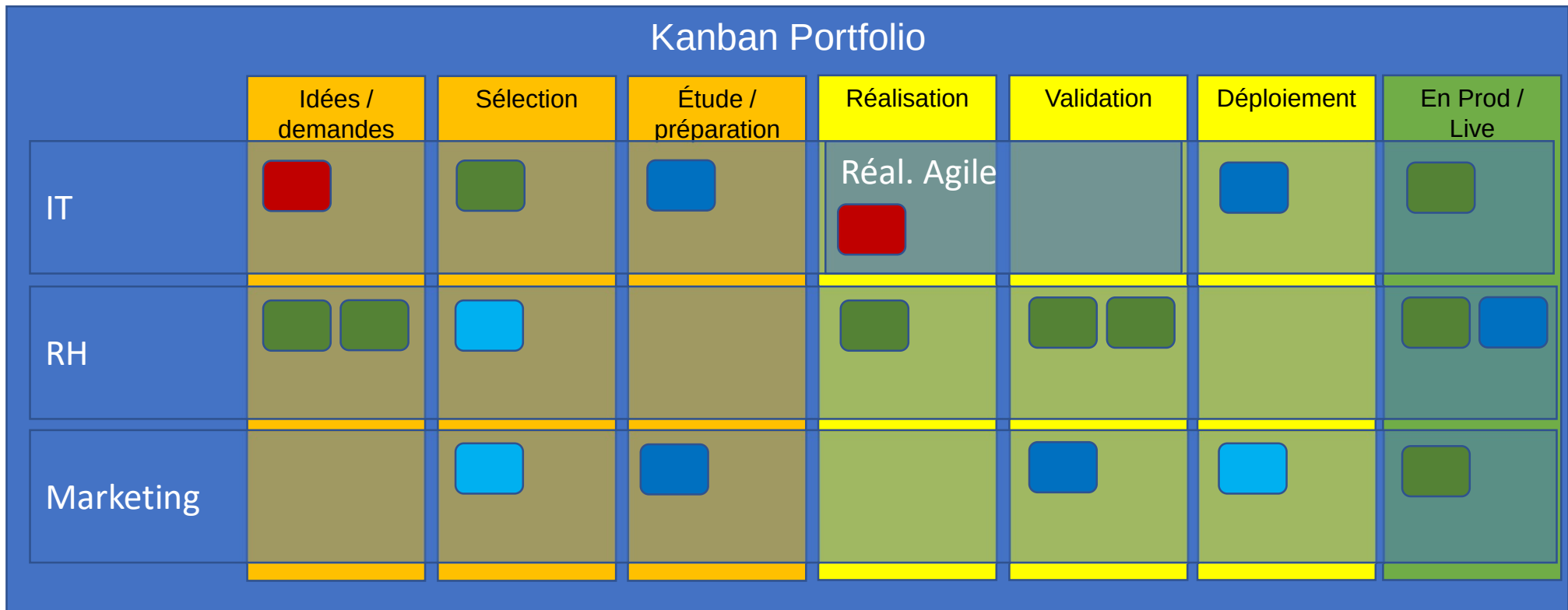
Tri « secondaire » des projets



Kanban Portfolio (4)

Démarrer la gouvernance

Ensuite, on va pouvoir séparer les projets dans chaque colonne avec des lignes horizontales, pour séparer par exemple des gammes de projet, de produite, entre le marketing, les RH, etc.



Kanban Portfolio (5)



Des règles claires

Les règles claires permettent à chacun de s'investir au mieux. On précise souvent, mais uniquement si c'est la réalité, à qui appartient la colonne, qui en est responsable, qui peut en manipuler les éléments.

Une ou plusieurs personnes, un groupe, département, etc.

Rien que la réalité.

Chacun peut s'accaparer l'information, se projeter, s'investir.

Du recul

On obtient un tableau qui nous indique où travaille qui, les densités, les vides, l'équilibre entre les différents domaines.

On voit les dynamiques, les inertie, les anticipations, etc.



Kanban Portfolio (6)



Exemple de fiche projet

Titre				Version	Type : Legal, Innovation, Stratégique, Evolution, Tactique	S
Description					Briques métiers / Techniques clefs	
Acteurs				NB	Effort	
					Valeur	
Dépendances					Deadline officielle	Coût estimé
Idées	Sélection	Etude / préparation	Réalisation	Validation	Déploiement	Validation



En savoir plus sur le Kanban Portfolio :

<https://pablopernot.fr/2016/01/portfolio-projets-kanban-partie-1/>

<https://pablopernot.fr/2016/02/portfolio-projets-kanban-partie-2/>



Organisation du travail



Par défaut, il n'y a pas de Product Owner, c'est l'équipe avec (ou non) le chef de projet qui vont choisir les tâches selon leurs priorités déterminés.

Le chef de projet peut avoir aussi le rôle de Product Owner, pour construire le backlog des tâches de type datées, Linéaires, redondantes, etc.

C'est aussi lui qui pourra interrompre la colonne « To Do » avec les demandes urgentes, les bugs, les tâches qui se rapprochent trop près de la deadline.

Comme vu précédemment, si l'équipe et / ou l'entreprise n'a jamais eu d'expérience avec l'agilité, il convient d'apporter les changements au fur et à mesure, et commencer par le Kanban board est souvent la voie la plus simple.



Étudier et améliorer le système



Saturation des systèmes



Globalement & Localement

En Kanban, il peut y avoir une saturation globale du système s'il n'y a pas de limites imposées globalement et localement.

Globalement, faire une « digestion » de toutes les demandes sur un projet : les demandes les moins pertinentes et ayant le moins de succès (c'est-à-dire une personne sur une trentaine la veut absolument par exemple), empêcher les interruptions tierce (un boss qui débarque avec ses propres items et les dépose directement dans la To Do ou la doing), etc.

Localement, c'est d'abord imposer des limites dans la To Do, un nombre max d'items autres que Bug dans la Doing, surveiller les validations, etc.



Systeme variable



Kanban évolue avec la vie du projet

Une réévaluation des items de la kanban backlog, de nouveaux items, un arrivant sur le marche oblige à créer de nouveaux items pour mettre le produit à niveau, de fait ces items deviennent prioritaire, etc.

Tous de type d'éléments font que le kanban backlog sera toujours en évolution et que le projet peu se terminer en étant complètement différent de l'idée de départ :

Le produit colle aux attentes des clients / utilisateurs



Gestion des limites



Les limites évoluent avec le projet et l'entreprise

Un nouveau membre dans l'équipe, un changement dans le système de validation, des nouveaux outils, un nouveau lot d'items, un lot d'items urgents remplacent l'actuelle To DO, etc.

Il faut, à chaque changement majeur, prendre un peu le temps de retailer les limites, sinon on risque une saturation ou une perte d'optimisation du système.

Les limites sont les premiers facteurs à vérifier quand on rencontre une saturation du système.



Gestion des délais de résolution de blocages



Mais le délais d'un item, ça n'est pas Kanban ?

Généralement, dans le kanban, mis à part les items ayant une date limite de réalisation, il n'y a pas de limite à l'exécution d'une tâche, ainsi qu'il n'y a normalement pas de limite de temps sur la présence d'une tâche dans le Doing.

Il convient de quand même mettre un délais maximum de résolution aux items présents dans la Doing (sauf items datés), ce délais correspondant au mieux à la priorité déterminé vue précédemment (Urgence, Standard, Intangible, etc.)

Selon la priorité, le délais commence à l'arrivée de la tâche dans le backlog (bug bloquant par exemple qui est transféré directement dans la Doing) ou à l'arrivée de la tâche dans la To Do (une fonctionnalité non prioritaire par exemple) etc.

Avoir un délais sur tous les items permet aussi d'avoir une idée de la vélocité de l'équipe (plus souvent utilisé en SCRUM), de sa réactivité et permet d'identifier des blocages divers (difficulté avec une technologie, problèmes d'infra, limites qui ne sont plus adéquates, etc.) pour trouver rapidement des solutions efficaces et pérennes.



Impacts globaux des changements locaux



A l'extérieur de l'équipe projet

Le projet en kanban est lancé, il tourne, l'équipe livre régulièrement des versions contenant des évolutions, de nouvelles fonctionnalités, des montées de version, etc.

Si un changement local influe sur la qualité des items finis, sature les systèmes, au niveau global il y a des risques de perte de confiance, d'impression de non avancement du produit, absence d'évolution, etc.

Il faut donc surveiller l'impact à l'extérieur de l'équipe à chaque changement local, c'est un indicateur qui permet de vérifier que ce changement est bénéfique ou non au projet, et généralement l'impact est très rapide donc il est très souvent possible de rétablir le tir rapidement en ajustant le système de l'équipe.



Évaluation du système



Suivi quotidien avec les outils

Le kanban board, les burn-down et burn-up charts, les retours utilisateurs, la fréquence d'évolution du backlog, le stand-up meeting, les présentations des itérations / livraisons, etc.

En analysant quotidiennement dans la globalités ces indicateurs (qui sont tous compréhensibles « d'un seul coup d'œil », il est facile de piloter les changements adéquat pour régler le système de façon optimale.

Il ne faut pas se reposer sur ses lauriers une fois qu'un semblant de stabilité apparaît : l'agilité est justement une méthode qui évolue avec les changements et les impondérables, il ne faut jamais arrêter de surveiller les indicateurs au risque de fin de vie ou de gros risque sur l'agilité et la longévité du projet.



Performances du système, réglages, cycles d'amélioration



Amélioration continue

La surveillance des indicateurs, le re-taillage des limites suite à des changements significatifs locaux, changement des méthodes de travail, rituels, environnement de travail, etc. sont tous les réglages à faire en continue afin d'améliorer continuellement le système.

Il n'existe pas à proprement parler de cycle d'amélioration, parce qu'au départ le kanban vise tout simplement l'amélioration en continu.

