

Les méthodes de conduite de projet d'entreprise

2020

Un point sur les méthodes les plus couramment utilisées en gestion de projet

- Ce dossier présente les **phases fondamentales** de la conduite de **projet** comme la **planification**, l'estimation des **coûts, délais** et besoins en **ressources**, ainsi que les principaux **outils** méthodologiques, diagrammes de **Gantt, Pert, Work Breakdown Structure**.
- Ce dossier consacre un chapitre particulier pour traiter des méthodes de type **agiles**, encore spécifiques aux projets **informatiques**.

Les phases du projet

- Exploration détaillée des principales **phases** et étapes de la conduite de projet, depuis **l'identification** du besoin jusqu'à la **mise en oeuvre**. **Conseils** et **recommandations** pour traiter la phase initiale de définition du projet.

1- Phasing

- Le déroulement classique d'un projet suit un **enchaînement logique de phases et d'étapes**:

- **Phase préalable et démarrage**
Etude d'opportunité et de viabilité

Cahier des Charges

- **Définition du projet**, périmètre, contexte, équipes, responsabilités, découpage fonctionnel (WBS, Work Breakdown Structure), étude d'impacts, identification des facteurs de risques et des facteurs d'échec.

- **Planification Ordonnancement**
Définition du plan d'exécution en termes de délais, de ressources et de responsabilité. Voir la phase planifier et ordonnancer détaillée.

- **Exécution**
Exécution et pilotage du projet, suivi du franchissement des jalons, gestion des dérives, importance de la réunion de travail projet.

- **Mise en exploitation**
Recette, vérification des conformités, mise en exploitation, formation, poursuite de l'accompagnement du changement.

- **Clôture**
Elaboration de la documentation dans une démarche de transfert de la connaissance. Gestion des évaluations du projet, préparation des évolutions...

1- Phasing

- Découpage en fonctions : **1er niveau**
- Définition du plan : **grandes lignes**
- Définition détaillée des **tâches**
- Composition des **équipes**
- Définition des **responsabilités**
- Estimation des **temps** de réalisation
- Génération du **PERT** (définition des dates)
- **Ajustement** jusqu'à concordance des **dates** (petite boucle)
- **Ajustement** avec la **disponibilité** des **ressources** (grande boucle)

Outils: WBS, GANTT, PERT, ...

1- Phasing

A- WBS

Work Breakdown Structure
du PMI

Pour maîtriser un projet =
découper en tâches
élémentaires évaluable et
réalisable.



1- Phasing

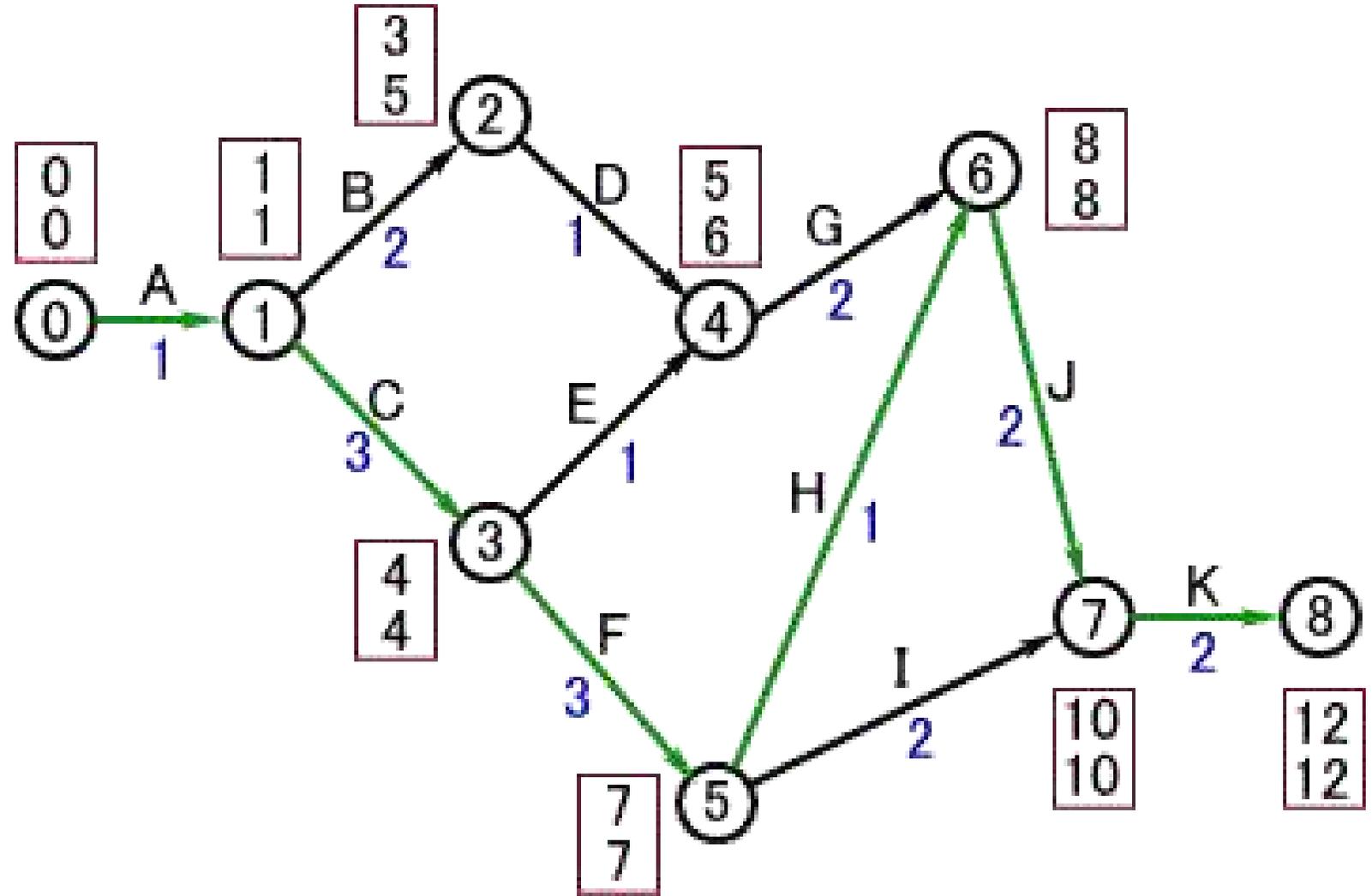
B- PERT

PERT = Program Evaluation and Review Technique

Marine US en 1957

Dates au plus tôt, dates au plus tard, chemin critique

Les taches anterieurs, les successeurs, la matrice d'anteriorité, de niveau.



1- Phasing

A- PERT

PERT = Program Evaluation and Review Technique

Marine US en 1957

Dates au plus tôt, dates au plus tard, chemin critique

Les taches anterieurs, les successeurs, la matrice d'antériorité, de niveau.

Exercice pédagogique TD : Préparer un repas.

1 Taches

- A : choisir le menu (30 min)
- B : acheter les ingrédients (90 min)
- C : préparer l'apéritif (30 min)
- D : nettoyer la table (10 min)
- E : mettre la table (10 min)
- F : préparer les ingrédients (30 min)
- G : cuisiner les plats (60 min)
- H : servir le repas (10 min)

2 Contraintes

- B doit être après A
- C et F doivent être après B
- E doit être après D
- G doit être après F
- H doit être après C, G et E

	Durée	antécédent
A	30	-
B	90	A
C	30	B
D	10	-
E	10	D
F	30	B
G	60	F
H	10	C-G-E

Dessiner le PERT

Exercice PERT

- **Exemple:**

<http://ressources.aunege.fr/nuxeo/site/esupversions/aa593e32-1ae0-4742-8f66-641ad89fd959/co/exemple1.html>

- **Application:**

<http://ressources.aunege.fr/nuxeo/site/esupversions/aa593e32-1ae0-4742-8f66-641ad89fd959/co/application2.html>

- **Exercice:**

http://ressources.aunege.fr/nuxeo/site/esupversions/aa593e32-1ae0-4742-8f66-641ad89fd959/co/exercice_1.html

1- Phasing

A- GANTT

Définition : le diagramme de GANTT, du nom de son inventeur, date du début du 20ème siècle.

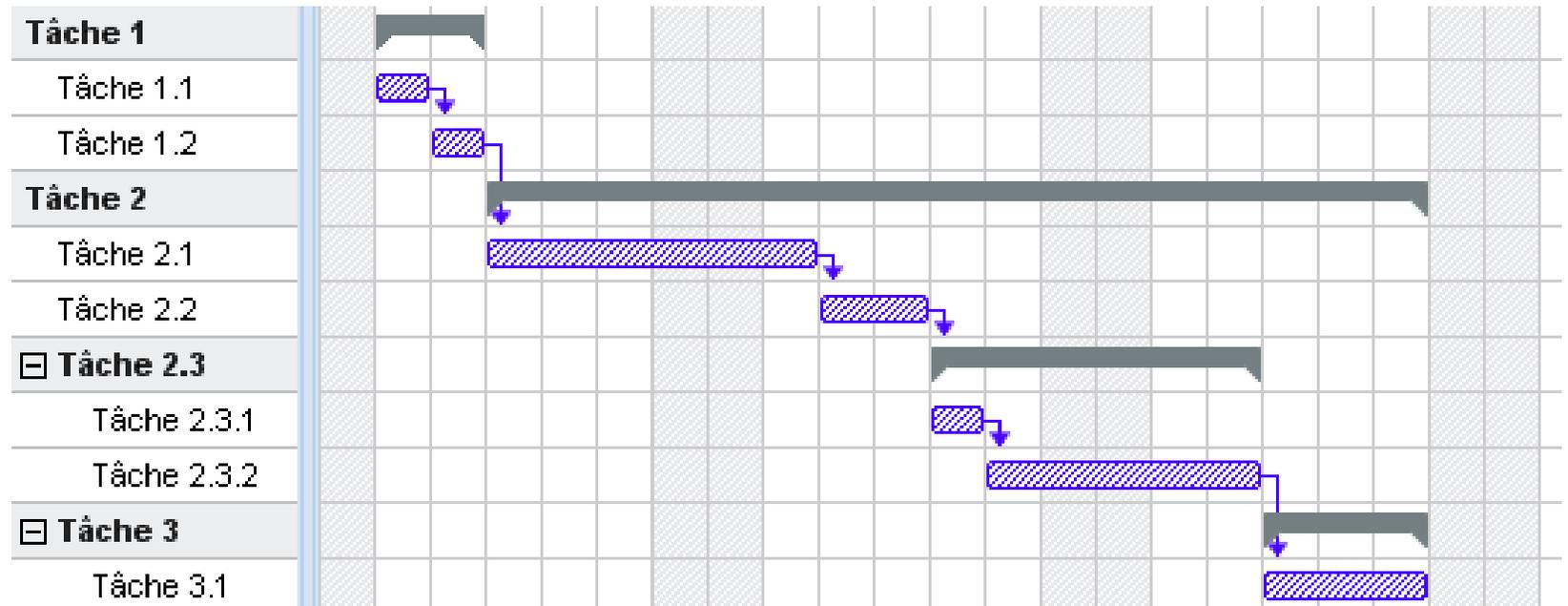
Ce type de diagramme est pratique pour la communication.

Aisément compréhensible par tout un chacun, il permet de se faire rapidement une idée précise autant du travail accompli que des tâches encore en attente de réalisation.

- 1890 - Karol Adamiecki
- 1905 - Henry Gantt - ingénieur et consultant en management
- Version Manuelle simple

Aujourd'hui:

- Liens de dépendance entre les tâches
- degré d'accomplissement
- ressources



1- Phasing

A- GANTT

Qu'est-ce
qu'un
diagramm
e de Gantt
?

- **Gestion de projet**
- **représenter visuellement l'état d'avancement**
- **Exemple**
 - **Eléments**
 - **Lignes = Tâches**
 - **Date de début**
 - **Date de fin**
 - **Colonnes = Temps**
 - **jour**
 - **Semaine**
 - **Mois**
 - **Trimestre**

1- Phasing

A- GANTT

Logiciels

- **Microsoft® Project**
- **Primavera Project Planner®**
- **MindView**

1- Phasing

A- GANTT

ETAPES

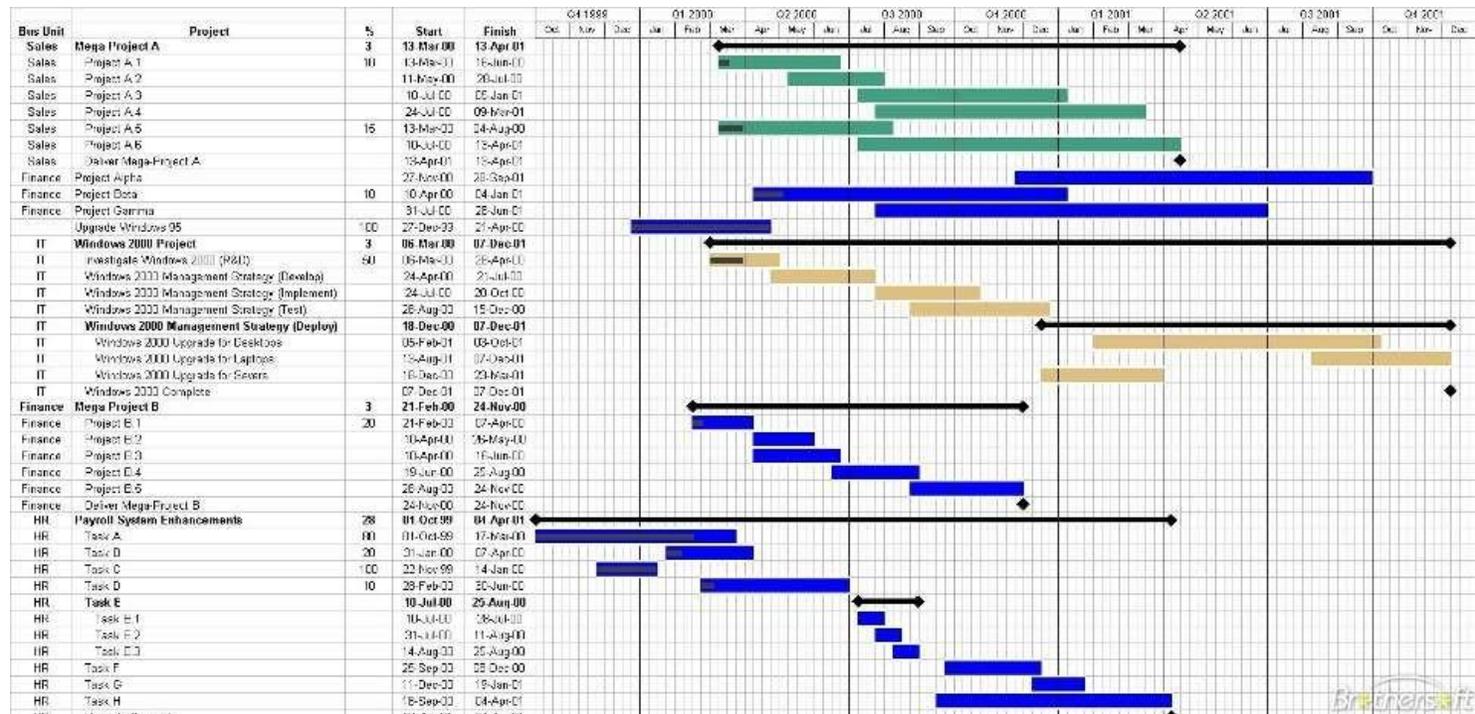
1. Définissez les **paramètres** du projet: date de **début**, date de **fin**
2. Définissez le **calendrier** du projet, nombre de **jours** de travail/**semaine**, le nombre **d'heures/jour**, les périodes **chômées** et ainsi de suite.
3. Saisissez ou modifiez les **noms** des **tâches** et leur durée.
4. Dressez la **liste** des **ressources** disponibles et **assignez** des **ressources** aux tâches.
5. Créez des **liaisons** entre les **tâches** pour indiquer la manière dont elles dépendent les unes des autres.
6. Définissez les **contraintes** de tâche nécessaires.
7. Apportez les dernières **modifications** au projet.
8. Roue de **Deming** = amélioration continue

1- Phasing

A- GANTT

ETAPES

- Ainsi en 1 seul coup d'Œil vous avez:
 - Tâches
 - Début et fin de chaque tâche et du projet
 - La durée des tâches
 - Le chevauchement éventuel



1- Phasing

A- GANTT

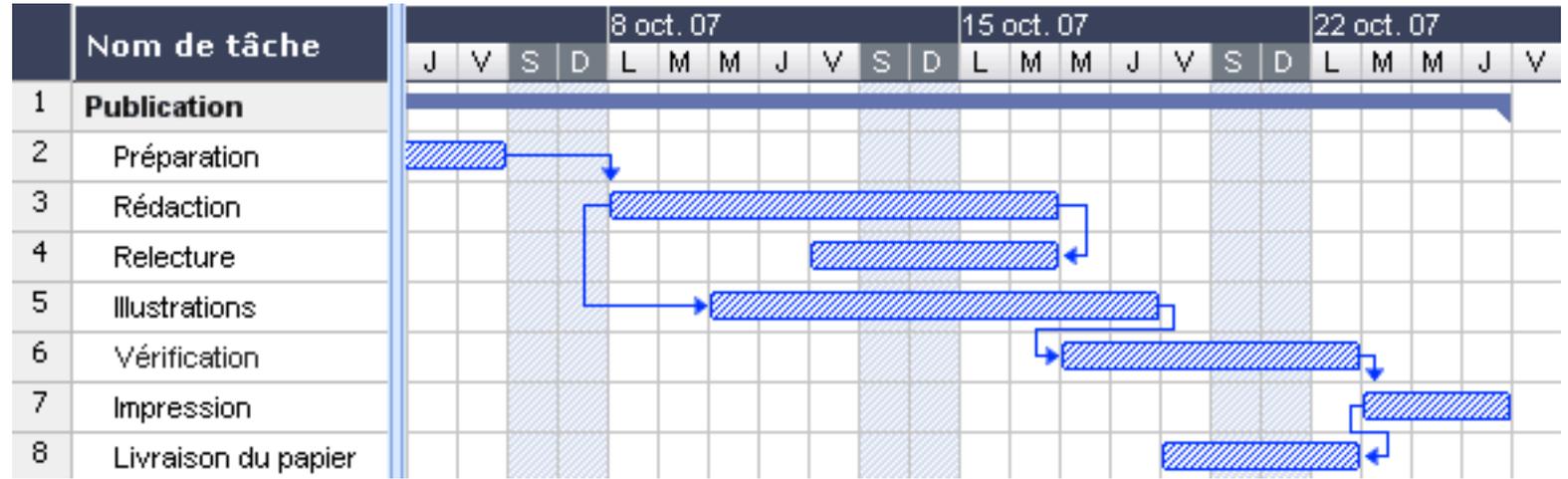
Liens

- La planification d'une tâche affecte les tâches qui y sont liées
 - **Avance** => recalcule toutes les tâches dépendantes => le projet gagne du temps
 - **Retard** => recalcule toutes les tâches dépendantes => le projet perd du temps

1- Phasing

A- GANTT

LIAISON des tâches dans un diagramme de Gantt



- **Exemple:** impossible d'imprimer une brochure avant qu'elle n'ait été rédigée et relue
- Types
 - Fin à début
 - Début à Début
 - fin à fin
 - Début à fin
- Types d'enchaînements:
 - Tâches prédécesseurs
 - Tâches successeurs
- **Exemple**
 - La tâche de rédaction commence dès que la phase de préparation s'achève. Si cette dernière est retardée, le début de la tâche de rédaction est également repoussé. Il s'agit du type de liaison par défaut, à savoir Fin à Début.
 - La relecture commence quelque temps après le début de la rédaction, et se déroule en parallèle avec elle. Le rédacteur et le relecteur travaillent en équipe jusqu'à ce que le texte soit finalisé. Si la tâche d'écriture s'achève plus tard que prévu, la tâche de relecture est également repoussée de façon à ce que les deux tâches se terminent en même temps. Il s'agit d'une liaison de type Fin à Fin.
 - La création des illustrations est liée à la tâche de rédaction. Un décalage a été introduit de façon à donner un peu d'avance au rédacteur par rapport aux illustrateurs. Il n'en reste pas moins vrai que si la rédaction commence plus tard que prévu, la tâche de création des illustrations est également repoussée. Il s'agit d'une liaison de type Début à Début.
 - La tâche de vérification finale est liée à la création des illustrations, mais peut toutefois commencer avant la fin de cette tâche. Un chevauchement de deux jours a donc été défini. Si la création des illustrations est retardée ou prend plus de temps que prévu, la vérification finale sera également repoussée. Il s'agit à nouveau d'une liaison par défaut de type Fin à Début.
 - L'impression ne peut démarrer qu'une fois la brochure finalisée. Il s'agit encore d'une liaison de type Fin à Début.
 - Il est impératif que la livraison du papier aux imprimeurs soit terminée à la date prévue pour le début de l'impression. Cette tâche est toutefois indépendante des tâches précédentes. Par ailleurs, les imprimeurs ont un espace de stockage limité, et ne souhaitent pas réceptionner le papier avant d'en avoir besoin. La date de livraison du papier est donc régie par la date prévue pour le début de l'impression. Si cette dernière est repoussée pour une raison quelconque, la date à laquelle le papier doit être disponible est également repoussée. Il s'agit d'une liaison de type Début à Fin.

1- Phasing

A- GANTT

En cas de décalage

- **prolonger** une **liaison** vers **l'arrière** ou vers **l'avant** pour **avancer** ou **reculer** le début de la tâche **successeur**

1- Phasing

A- GANTT

contraintes

- **Contraintes = niveau de flexibilité** - en imposant des restrictions sur sa date de début ou sa date de fin - pensez à adjoindre une **note** ou un commentaire à la tâche pour **justifier** l'usage de la contrainte
 - **Début Au Plus Tôt Le (DPTO)** : La tâche, qu'elle soit liée ou non, ne peut pas commencer avant la date spécifiée. Elle peut cependant commencer après cette date.
 - **Début Au Plus Tard Le (DPTA)** : La tâche, qu'elle soit liée ou non, ne peut pas commencer après la date spécifiée. Elle peut cependant commencer avant cette date.
 - **Fin Au Plus Tôt Le (FPTO)** : La tâche, qu'elle soit liée ou non, ne peut pas s'achever avant la date spécifiée. Elle peut cependant s'achever après cette date.
 - **Fin Au Plus Tard Le (FPTA)** : La tâche, qu'elle soit liée ou non, ne peut pas s'achever après la date spécifiée. Elle peut cependant s'achever avant cette date.
 - **Doit Débuter Le (DDL)** : Cette contrainte rigide signifie que la tâche, qu'elle soit liée ou non, doit commencer à la date spécifiée. Même si la tâche prédécesseur se termine plus tôt que prévu, le logiciel de Gantt ne peut pas avancer la tâche pour tirer parti du temps gagné.
 - **Doit Finir Le (DFL)** : Cette contrainte rigide signifie que la tâche, qu'elle soit liée ou non, doit s'achever à la date spécifiée. Comme plus haut, même si la tâche prédécesseur se termine plus tôt que prévu, le logiciel de Gantt ne peut pas avancer la tâche pour tirer parti du temps gagné.
- 2 exceptions
 - Le Plus Tôt Possible (**LPTO**) = contrainte appliquée par défaut à toute nouvelle tâche lorsque vous planifiez votre projet à partir de sa date de début
 - Le Plus Tard Possible (**LPTA**) = C'est la contrainte appliquée par défaut à toute nouvelle tâche lorsque vous planifiez votre projet à partir de sa date de fin

conflits dépendance et contraintes

Principe = Contrainte prévaut sur la liaison

1- Phasing

A- GANTT

Ajout de ressources

- Personnes
- Hardware
- Software
- Locaux
- budgets
- ...

1- Phasing

A- GANTT

Optimisation

- Ajout de **notes** explicatives aux tâches.
- Affichage du **chemin critique** = tâches ou séries de tâches qui ne peuvent pas être repoussées sans que la date de fin du projet ne soit repoussée.
- Définition de **jalons** et **d'échéances** pour faire ressortir les dates clés.
- Attribution de **priorités** aux tâches.
- Affichage du **niveau de progression** de la tâche sur sa barre.

1- Phasing

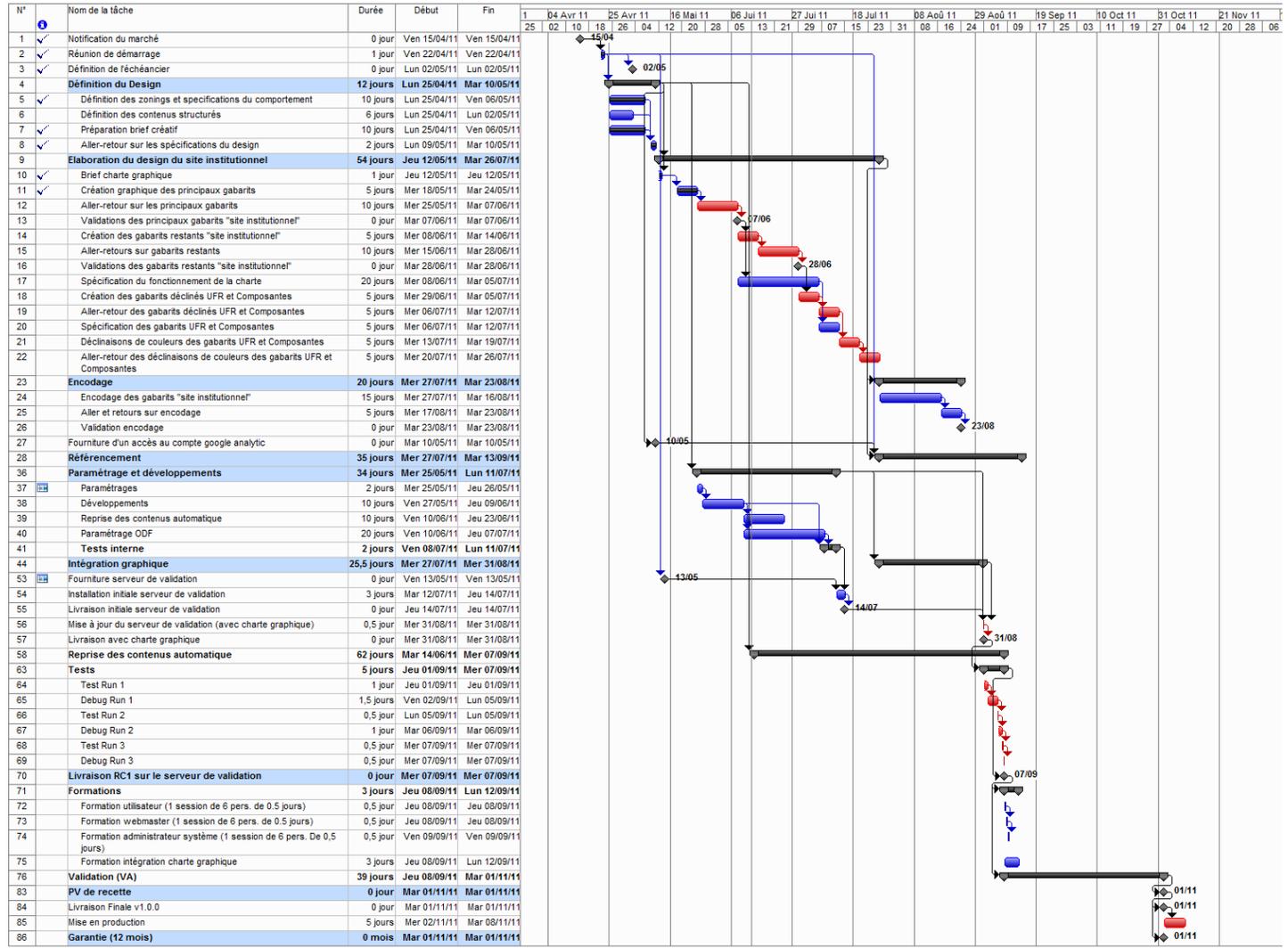
A- GANTT

Inspection et suivi

- Votre projet se déroule-t-il comme prévu ? **Vérifier date de fin**
- Certaines de vos tâches ont-elles **dépassé leur échéance** ? => repousser ou supprimer l'échéance, prolonger la tâche ou lui allouer davantage de ressources.
- Les **liaisons** établies entre les tâches sont-elles toutes **nécessaires** ?
- Voyez aussi s'il est possible d'effectuer certaines des **tâches en parallèle** afin de réduire la durée totale du projet.
- Toutes les **contraintes** définies ont-elles **l'effet voulu** ?
- Les tâches dotées de **contraintes** sont-elles **planifiées aux dates** voulues ?
- Existe t'il des **chevauchements** ou des délais **imprévus** entre les tâches liées ?

1- Phasing A- GANTT Exercice

<https://www.tomsplanner.fr/>



2- Estimation des temps

- **Estimer** la durée d'un projet est un peu un travail **d'acrobate**. En réalité, il n'existe **pas d'outils** ou de méthodes suffisamment fiables pour garantir un résultat précis.
- Bien sûr, les **impondérables** seront aussi là pour jouer les troubles-fêtes. Cela dit il faut bien fournir un délai. C'est pour cela que l'on parle **d'estimation**. La précision de celle-ci dépendra de la **marge d'erreur acceptable**
- C'est un travail **coopératif**. Pour l'estimation de chaque **module**, il existe au moins **un expert** : celui ou ceux qui devront le réaliser. Il semble évident qu'ils participent activement au chiffrage global.
- **En informatique on utilise parfois des méthodes ludiques comme le Planning Poker**

2- Estimation des temps

- L'estimation et la précision des timings dépend souvent:
- **I**: degré **d'innovation** du projet. Plus un projet est innovant plus l'expérience est réduite et donc moins de recul.
- **C**: **complexité**. Si on doit mettre en oeuvre de multiples technologies ou/et que de multiples domaines fonctionnels sont concernés et/ou nécessite de multiples intervenants
- **D**: **durée**: vive les projets courts !
- La Qualité de l'estimation est égal à l'inverse du produit : $(I \times C \times D)^2$ soit

Qualité de l'estimation = $1 / (I * C * D)^2$ 😊

3- Gérer le Projet Informatique

- Comment gérer un projet technologique ?
- Il est bon de connaître les spécificités du projet de mise en oeuvre des **technologies informatiques** pour le mener à son terme dans les meilleures conditions, qu'il s'agisse de pur **développement** ou de **l'installation** d'une solution système d'information de type business Intelligence, ERP, CRM, SCM, progiciel intégré, propriétaire ou open source ou solution SaaS, cloud...

3- Gérer le Projet Informatique

On commence par oublier la hiérarchie des cellules de pilotage...

Le déroulement classique d'un projet d'envergure s'appuie sur un pilotage bicéphale, **MOA-MOE**, hérité de l'univers de la construction et des travaux publics:

La MOE, **Maîtrise d'Oeuvre**, est chargée de la **réalisation** proprement dite et

la MOA, **Maîtrise d'Ouvrage**, est orientée sur les aspects **fonctionnels**, la **formalisation** du besoin et le **suivi** au cours du déroulement de l'adéquation en le réalisé et le désiré

Ce diagramme est expliqué dans
Le chef de projet efficace
(c) Eyrolles

(c) www.chef-de-projet.org



3- Gérer le Projet Informatique

- **2 erreurs à ne pas commettre:**
 - *Les **besoins sont clairement exprimés**, et toutes les parties prenantes, que ce soient les donneurs d'ordre, les développeurs ou les utilisateurs, ont parfaitement compris de quoi il en retournait.*
 - *Les **besoins n'évolueront pas** au cours du déroulement du projet*

3- Gérer le Projet Informatique

- On coopère, on mesure, on anticipe et on accompagne !
 - On coopère...
 - Un projet technologique est un projet **complexe** par définition. Il n'est **pas très judicieux** d'adopter un mode de **management directif** pour le mener à son terme. Il est très nettement **préférable** d'adopter un mode de gestion du projet plutôt **coopératif**.

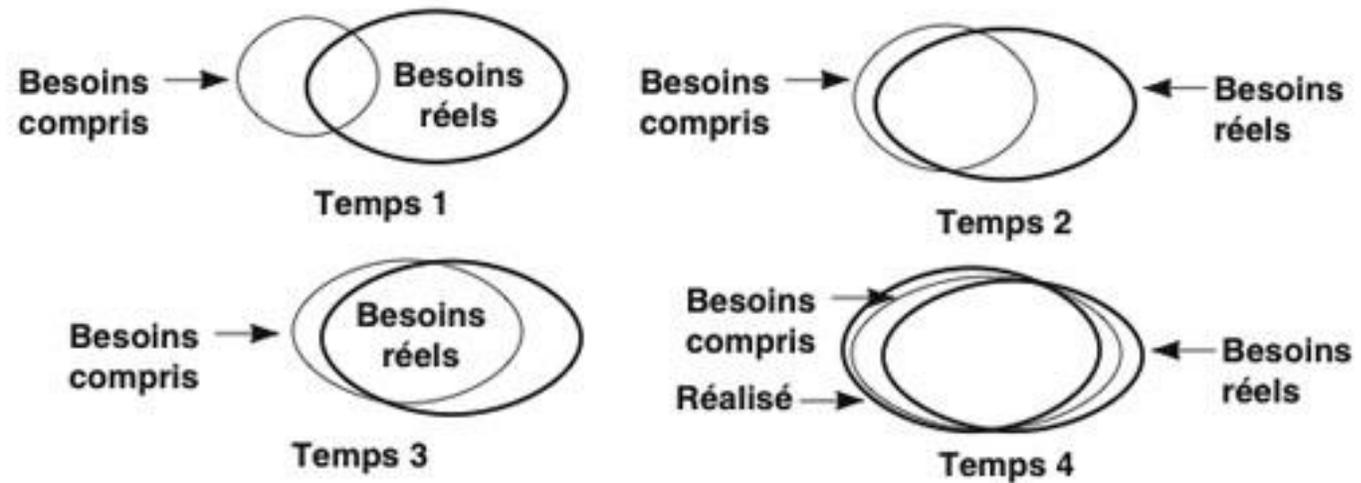


Figure 2.11. Un projet communicant avec méthode

(c) Eyrolles Les nouveaux tableaux de bord des managers

3- Gérer le Projet Informatique

- On coopère, on mesure, on anticipe et on accompagne !
 - On **mesure**...
 - On évitera de se laisser enfermer dans la spirale du reporting ou du "rendre compte à tout instant" pour bâtir un vrai tableau de bord de pilotage qui permettra à l'équipe de mieux s'organiser, d'anticiper les coups de colliers et de livrer un produit conforme aux exigences, quelles qu'elles soient (stratégiques, techniques, performances, ergonomiques...).
 - on **anticipe**...
 - La gestion des risques du projet n'est pas une formalité. Il est tout de même aberrant de négliger cette étape, et de se retrouver en plein coeur du projet aux prises avec un sinistre parfaitement prévisible...
 - on **accompagne**
 - Et on accompagne la mise en oeuvre du projet de **bout en bout**, sans omettre d'intégrer les coûts de la conduite du changement dans l'enveloppe budgétaire initiale...

4- Le PmBok Project Management Body of Knowledge

- Guide du corpus des connaissances en management de projet du **PMI Project Management Institute**. C'est une **référence exhaustive**, une démarche à connaître impérativement lorsque l'on envisage de maîtriser la conduite de projet d'envergure.

4- Le PmBok Project Management Body of Knowledge

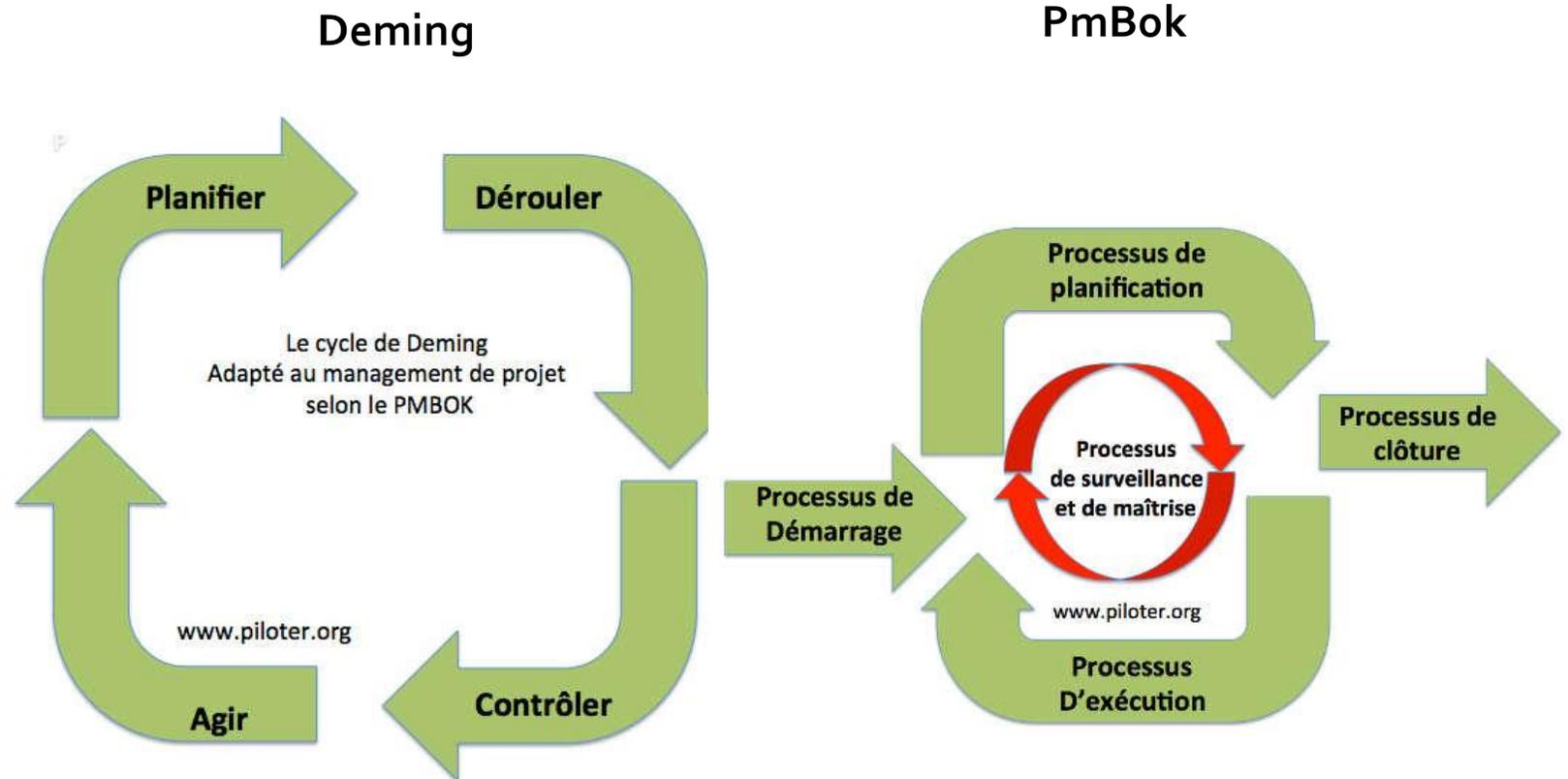
- La toute première version de ce guide remonte à 1987.
- La sixième édition date de Septembre 2017 en version anglaise.
- PMBOK Guide est une approche orientée **processus**.
 - Chaque processus est décrit en terme **d'entrées** (données, documents...),
 - de **sorties** (données, documents, produits....)
 - et **d'activités** (traitement des entrées, techniques, outils...).

4- Le Pmbok Project Management Body of Knowledge

- Les processus sont classés 5 groupes de base.
 - **Démarrer**
 - Ce processus précise les conditions et accorde l'autorisation de démarrage du projet ou de l'une de ses phases majeures.
 - **Planifier**
 - Ce groupe regroupe sans surprise l'ensemble des activités organisées en processus élémentaires indispensables pour assurer un enchaînement optimal des phases du projet dans les délais et ressources impartis selon les objectifs fixés.
 - **Exécuter**
 - Assurer la réalisation et coordonner les acteurs projets et les ressources disponibles selon le plan défini.
 - **Maîtriser et Surveiller**
 - Evaluer le travail réalisé, Assurer la gestion des coûts, Gérer les risques, entre autres.
 - C'est un groupe de processus particulièrement complet. Dans la terminologie un peu réductrice du Pmbok on utilise les termes de "maîtrise" et de "surveillance". Le terme de "pilotage" mériterait de remplacer celui de "surveillance", il couvre en effet une dimension "pro-active" plutôt qu'uniquement "réactive", la prévention en complément de la correction : Mesurer pour s'assurer que les objectifs seront atteints, détecter les variations et décider.
 - **Achever**
 - Ce sont toutes les actions de clôture du projet ou d'une de ses phases majeures pour formaliser l'acceptation du produit réalisé au terme du déroulement.

4- Le PmBok Project Management Body of Knowledge

- Plus précisément, Le PMBOK se base sur le cycle de Deming, le Plan, Do, Check, Act



5- Prince 2

- Prince2: **PR**ojects **IN** Controlled Environments
- Prince 2 est orientée **processus**. Elle est à vocation **généraliste**.
- La démarche propose son propre système de **certification**.
- Son formalisme et son rigorisme sont plutôt adaptés aux "**méga**" **projets**.

5- Prince 2

- Les processus supports
 - **DP Diriger le projet**
Comité de pilotage, prise de décision
 - **SU Commencer un projet**
Phase de préparation du projet, précision des termes du projet, formation des équipes
 - **IP Initier un projet**
Lancement proprement dit, préparation du cadre d'exécution, responsabilités, planning, contrôle qualité...
 - **CS Piloter une étape**
Le projet peut être découpé en étapes identifiables et finies : lancement, suivi de l'avancement, collecte d'infos, prise de décision, revue et reporting de l'étape.
 - **MSB Gérer le passage à l'étape suivante**
En fin d'étape, il s'agit de bien manager la poursuite du projet. Faut-il le poursuivre ? Recueil d'informations pour le comité de pilotage, préparation et lancement de l'étape suivante.
 - **MP Gérer la livraison des produits planifiés**
Les produits sont-ils réalisés et mis en oeuvre comme prévus ? Recette des produits.
 - **CP Clôture du projet**
Recette, évaluation, les objectifs ont-ils été atteints ? Libération des ressources, revue de fin de projet, documentation.

5- Prince 2

- Les **avantages...**
 - La structure de la démarche projet proposée par Prince2 est un excellent outil de **formation** et **d'initiation** à la **conduite** rationnelle de **projet**. Le **formalisme** permet de définir une **logique** et un **vocabulaire** commun facilitant les échanges; c'est aussi cela l'intérêt d'utiliser une méthode standard. Retenons aussi **l'orientation apprentissage continu**.
- Les **inconvénients...**
 - La **rationalisation** systématique camoufle les réelles subtilités de la conduite de projet. En pratique, la conduite de projet est bien plus **complexe**. Les **réalités** du **terrain**, les enjeux **humains**, les **technologies immatures**, les **exigences** de **délais** et de **budgets** ne seront résolues par la seule **obsession du formalisme**.

6- Panorama des méthodes agiles

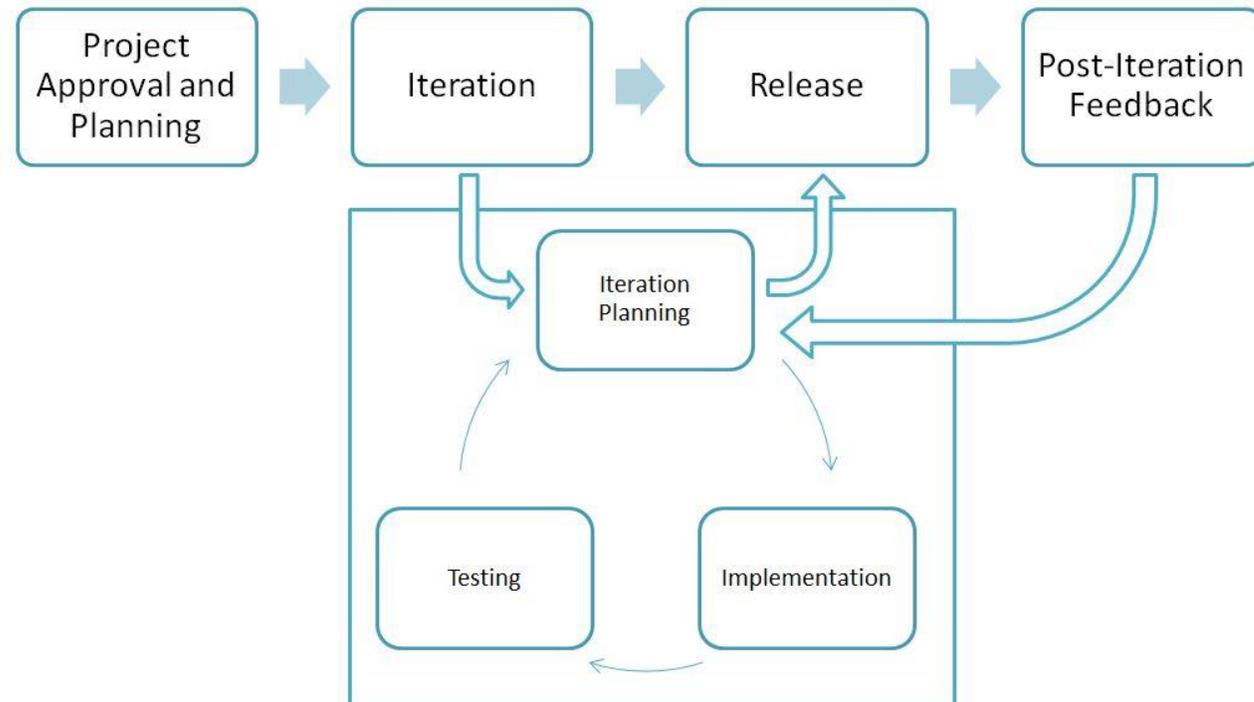
- Elles ont été conçues et développées à l'origine pour **résoudre** l'infernale question de **l'inadéquation** entre les **attentes** réelles des utilisateurs et le **produit** livré
- Les méthodes dites **agiles** proposent une réalisation **cyclique** du projet et instaurent un **dialogue continu** entre le **client** et les **équipes** de réalisation.

6- Panorama des méthodes agiles

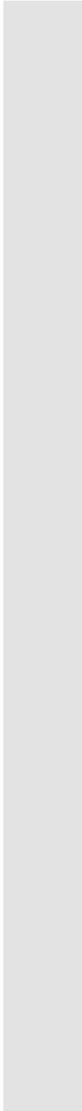
Projet classique



Projet agile



Découvrir l'Agilité

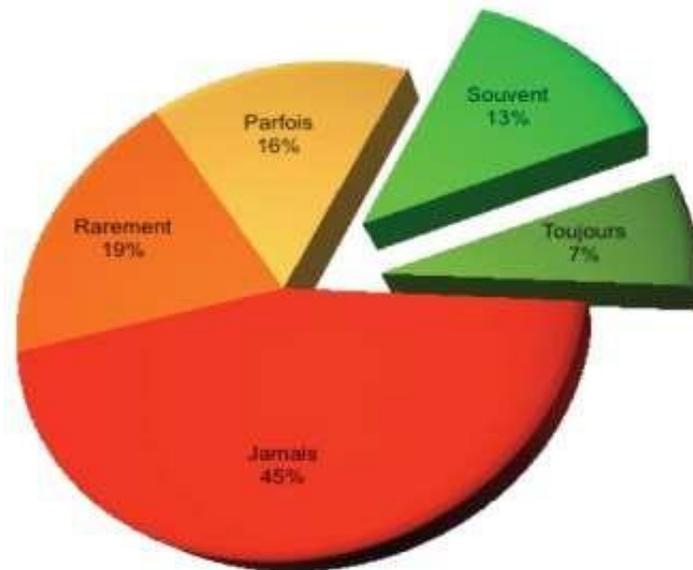


qu'est-ce que l'agilité ?

Limites des approches classiques

Taux de succès des projets informatiques en 2009 de 32%

- Source : enquête Standish Group sur 8000 projets



Fonctionnalités utilisées d'un SI en %

(source: Standish Group Study reported at XP2002 by Jim Johnson, chairman)

Peu de fonctionnalités développées réellement utilisées

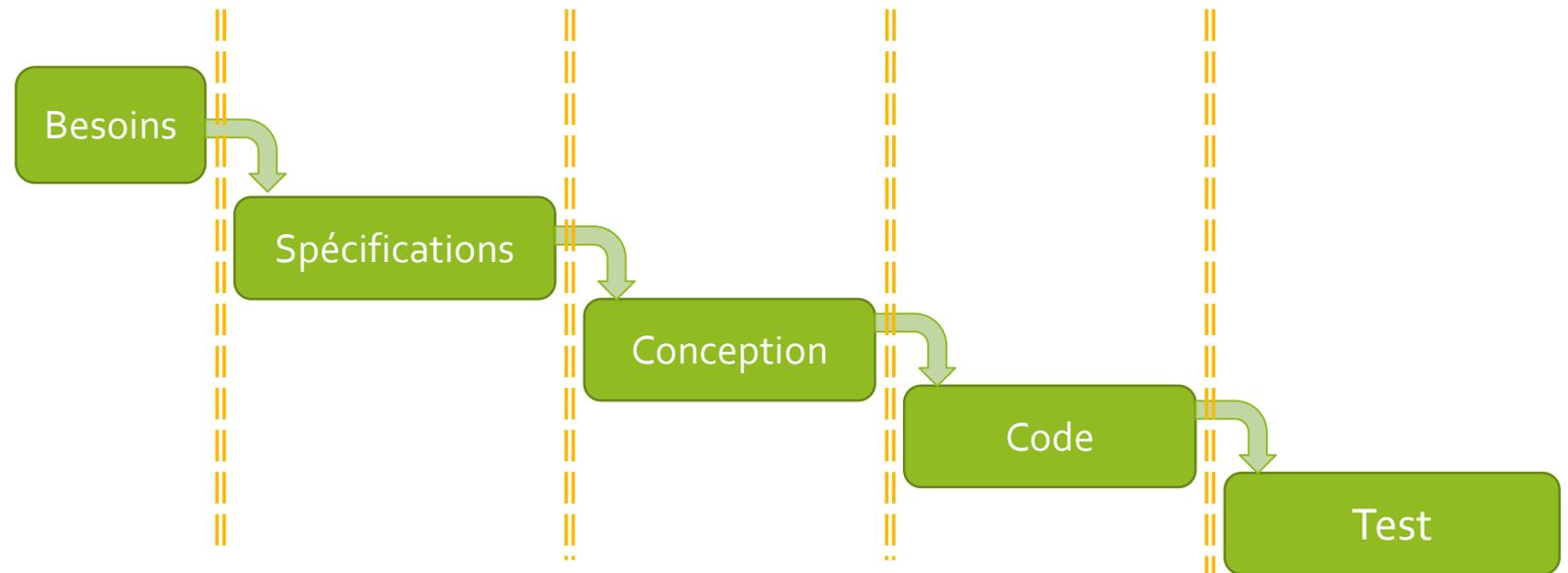
- 45% de fonctionnalités jamais utilisées

Agile

- Approche **réactive** et itérative d'organisation de travail
- Focalisée sur la **fonctionnalité** et **satisfaction** client
- Construit en **adéquation** avec les **capacités** et **limites humaines**

Pourquoi Agile ?

- En réaction des problèmes avec des approches 'traditionnelles' :

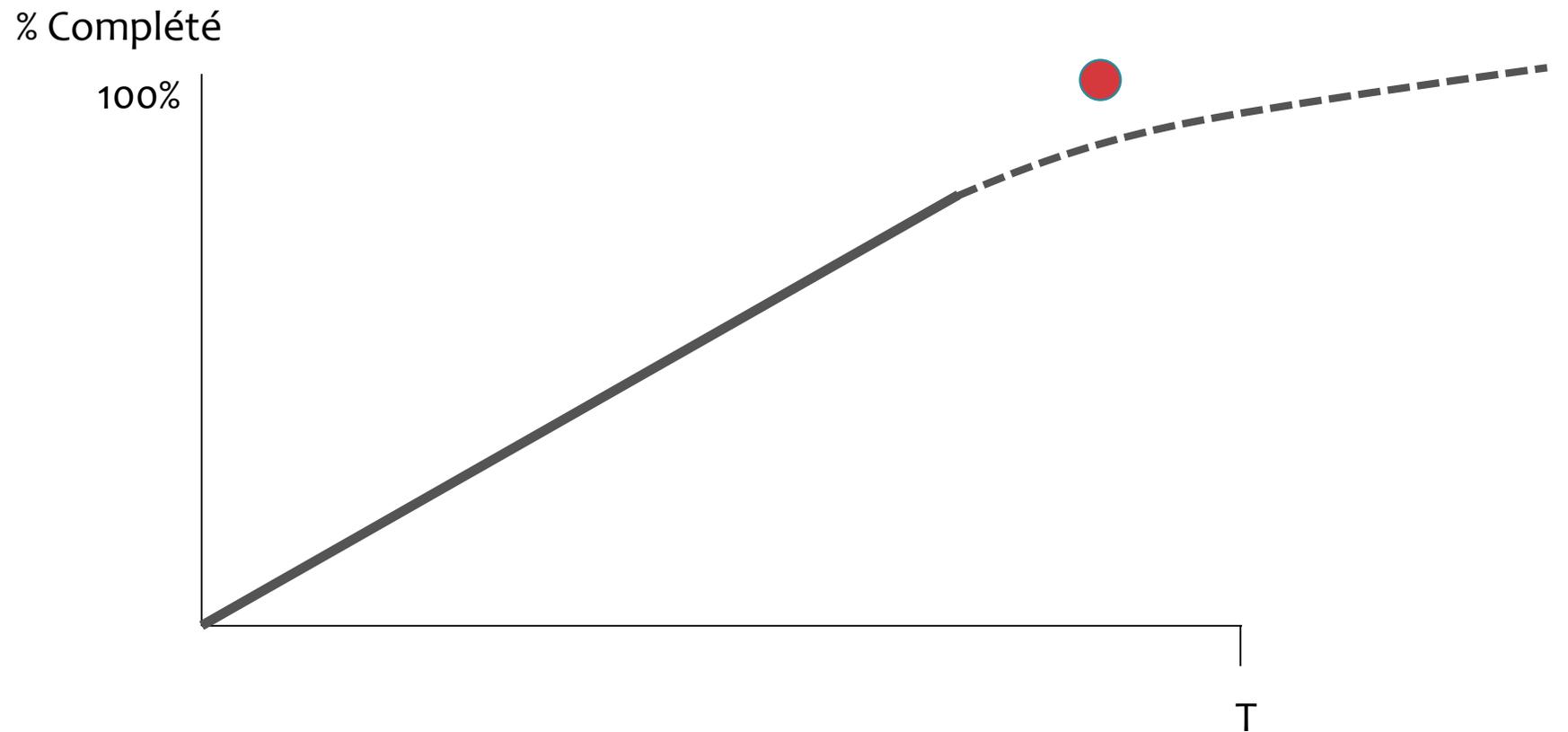


Les constats

- Les **meilleures idées** ne viennent **pas forcément** au **début** du projet
 - Il est plus facile de **construire** par **étape** que tout imaginer dès le début
- Les **besoins** peuvent **évoluer** pendant le projet
- Le **formalisme** n'est **pas naturel**
- **Chiffrages** et **Reste à Faire** sont **difficiles** à **évaluer**

Un projet informatique... la réalité

- On ne sait pas estimer la charge restante



Problèmes avec cascade

- Les méthodes **prédictives** fonctionnent bien, à **condition** d'avoir:
 - **Stabilité** et **prévisibilité**
 - **Communication** et **compréhension** parfaite
 - **Choix parfaits** dès le départ

➤ **Aucun humain!**

Agile : Un juste milieu

Très réactive

Réactivité

Peu réactive

Peu focalisé,
aucune maîtrise

Focalisation

Objectifs clairs

Absence de
méthode

Méthodes
prédictives

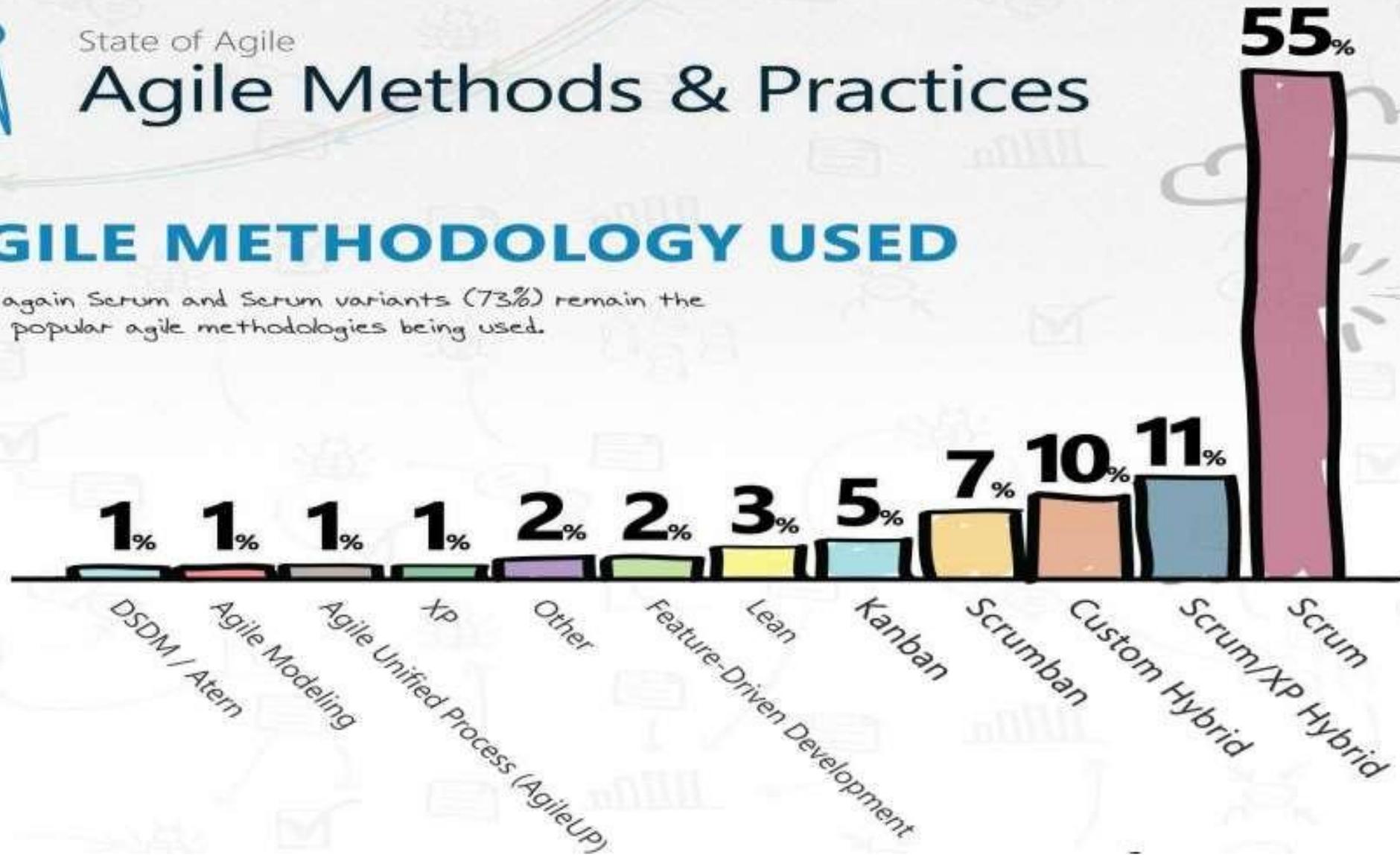


State of Agile

Agile Methods & Practices

AGILE METHODOLOGY USED

Once again Scrum and Scrum variants (73%) remain the most popular agile methodologies being used.



Agile : Une catégorie de méthodes

- 'Agile' regroupe plusieurs **méthodologies** :
 - **Scrum**
 - Extreme Programming (**XP**)
 - **DSDM**
 - **Crystal**
 - ...
- Notion officialisée en **2001** avec le Manifeste Agile

Le manifeste Agile

Personnes et interactions

Plutôt que

Processus et outils

Un produit opérationnel

Plutôt que

Documentation exhaustive

Collaboration avec le client

Plutôt que

Négociation d'un contrat

Adaptation au changement

Plutôt que

Suivi d'un plan

Le Manifeste Agile

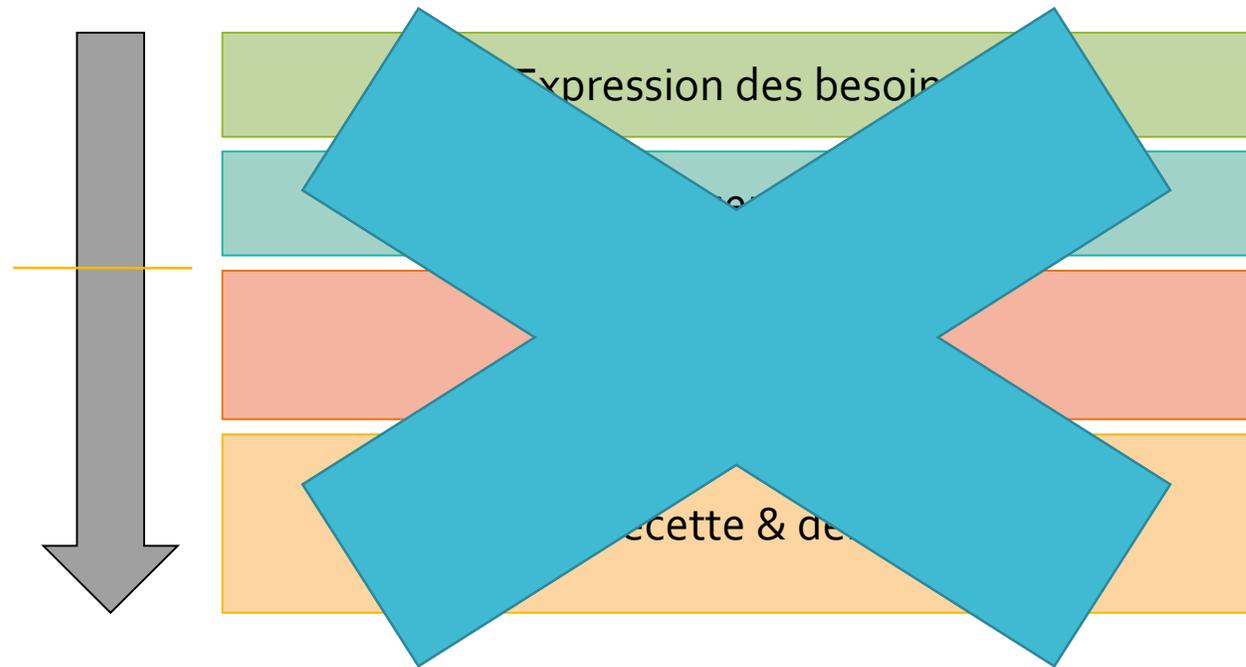
Libérer le génie humain

pour l'auto-organisation dans un contexte qu'il peut maîtriser :

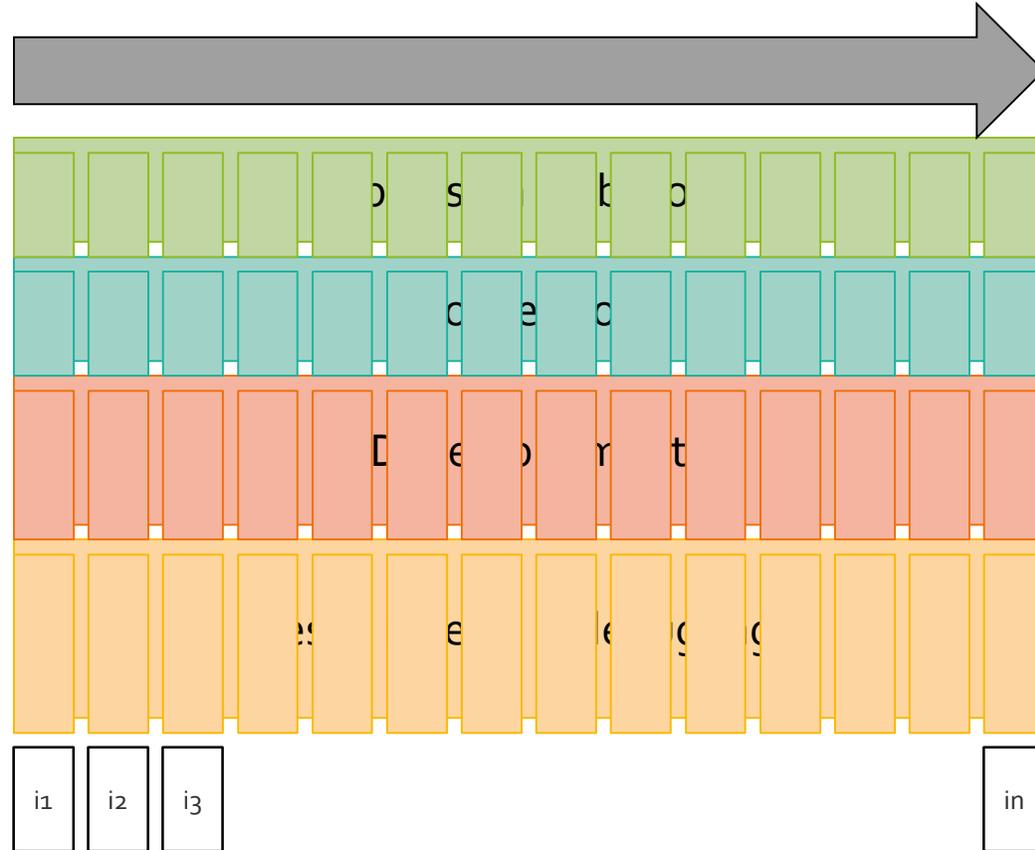
- La taille de l'équipe est limitée
- le domaine du problème est limité

- Petites équipes autogérées
- Portée fonctionnelle restreinte à un moment donné
- Garder un rythme de travail soutenable
- Avancement par itération

Le Manifeste Agile



Les solutions Agiles



Les solutions Agiles

- Toujours focalisées sur le produit final
 - Une vision commune pour l'équipe
 - la satisfaction du client
 - Découper le projet autrement
 - par fonctionnalité
 - Organiser en cycles de développement réduits
 - itérations

Les solutions Agiles

- Collaboration avec le client
 - Pourquoi on veut des contrats ?
 - *Instaurer la confiance autrement*
 - *Eviter les effets pervers d'un contrat*

Les solutions Agiles

- Adaptables
 - Réactives aux nouveaux besoins
 - Réceptives aux nouvelles solutions
- *Prendre les décisions définitives le plus tard possible*
- *De courtes itérations permettent de changer de direction sans laisser des éléments à moitié fait*

Agile : Planification

- L'estimation de charge est difficile, mais les courtes itérations nous aident
 - On est plus précis sur les petites tâches
 - Feedback très rapide
 - Plus facile à s'adapter face aux dérives, surprises

Scrum

Exemple de méthode
Agile

Issue du terme
"mêlée"
(Rugby) en
anglais

- Avancer ensemble vers un but commun



- QU'EST CE QU'UNE APPROCHE AGILE ?

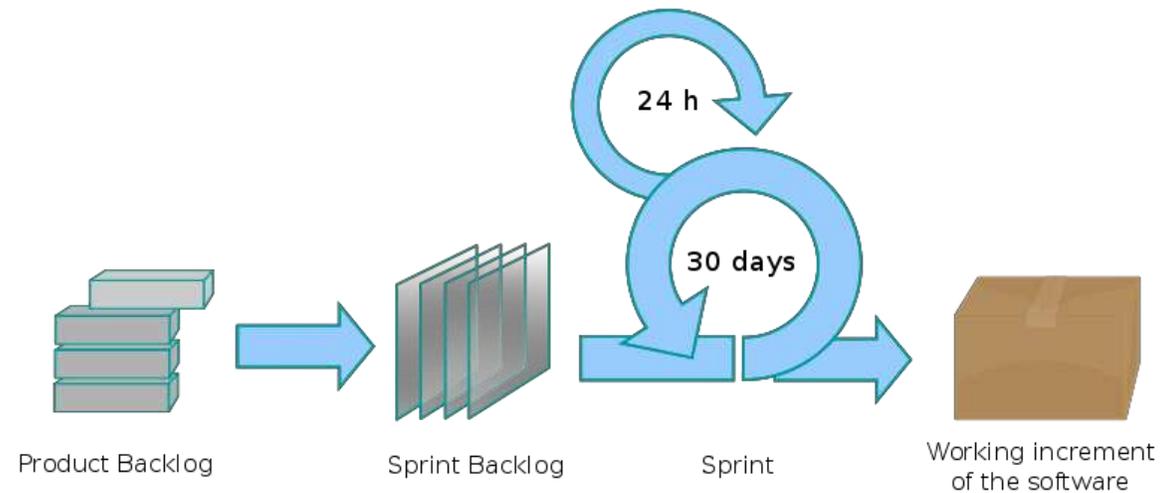
Le constat

- Incertitude inévitable
- Le besoin ne peut pas être complètement connu tant que les utilisateurs ne l'ont pas utilisé

Scrum : Diviser pour mieux maîtriser

- Diviser le périmètre
- Diviser le temps
- Diviser en équipes de tailles optimales

Scrum est un processus empirique PAR PHASES



Un **but unique** et **commun** :

Produire **au plus tôt** un maximum de **valeur métier**
par **incréments** de grande **qualité** et **industrialisés**

Scrum repose sur 3 pieds



Transparence



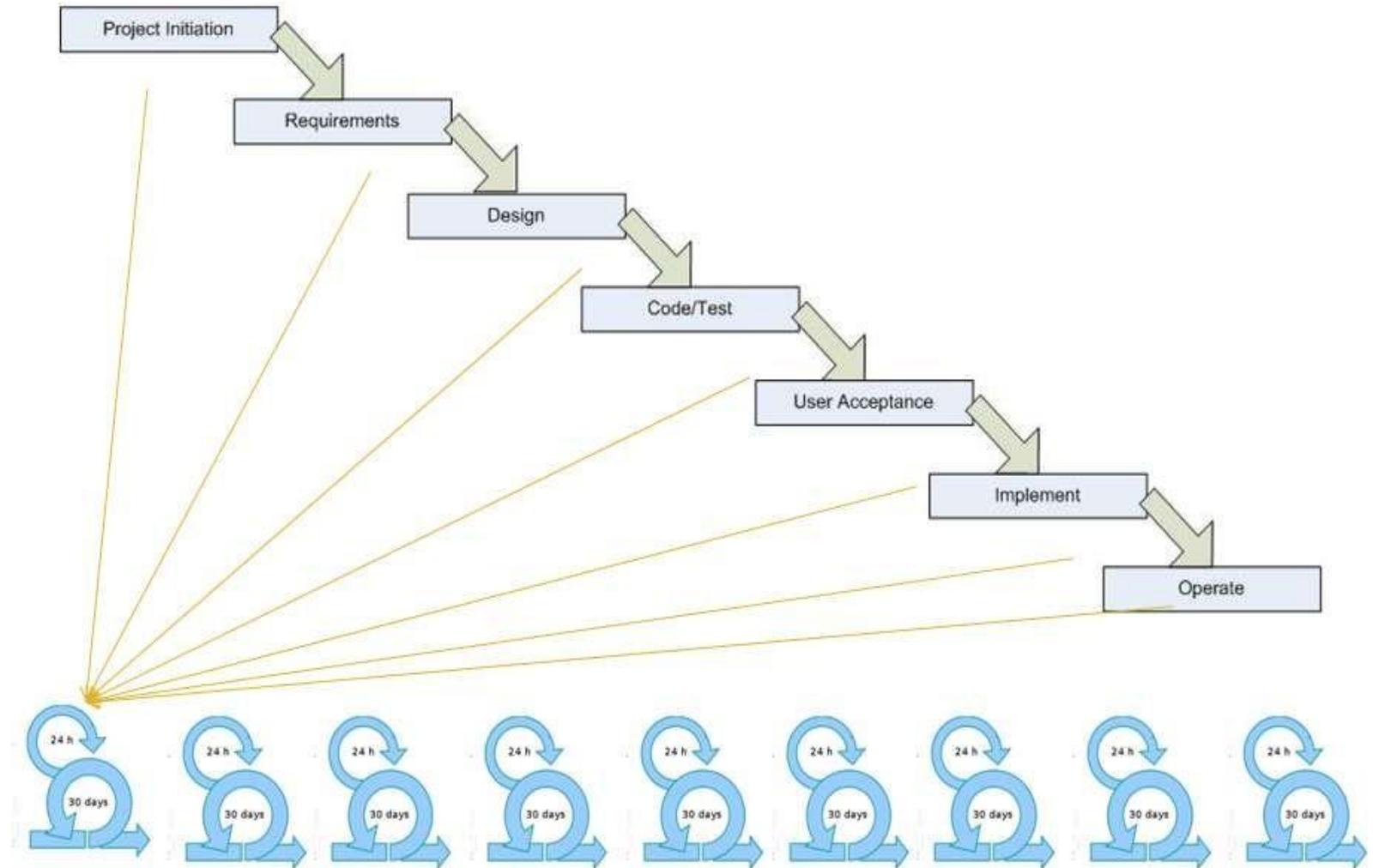
Inspection



Adaptation



Scrum vs Modèle en "cascade"





La Philosophie

L'Agile Manifesto

Manifeste pour le développement Agile de logiciels



Les individus et leurs interactions

- Des logiciels opérationnels
- La collaboration avec les clients
- L'adaptation au changement



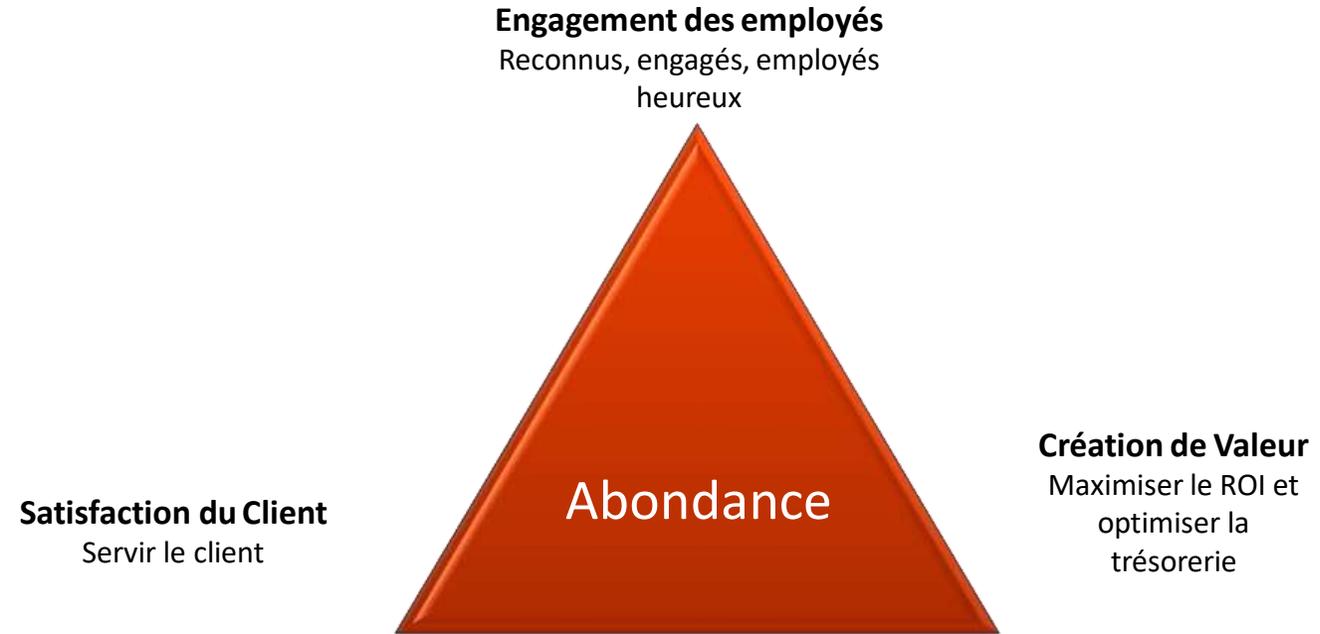
les processus et les outils

- une documentation exhaustive
- la négociation contractuelle
- le suivi d'un plan

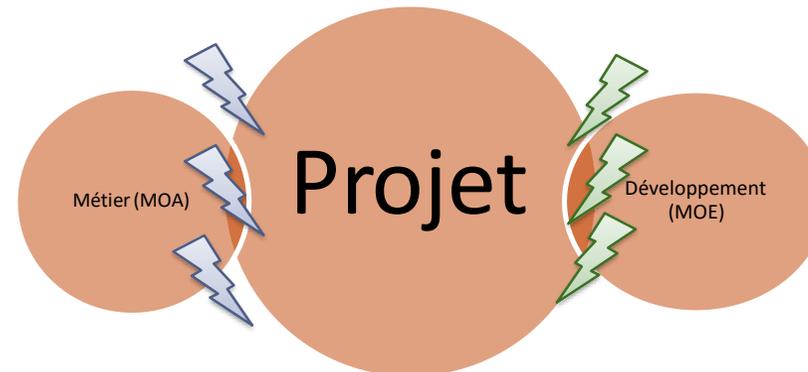
Principes sous-jacents au manifeste

Priorité	
1	satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.
2	Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.
3	Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.
4	Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.
5	Réalisez les projets avec des personnes motivées.
6	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.
7	Un logiciel opérationnel est la principale mesure d'avancement.
8	Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.
9	Une attention continue à l'excellence technique et à une bonne conception renforcent l'Agilité.
10	La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
11	Les meilleures architectures, spécifications et conceptions émergent d'équipes auto organisées.
12	À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

Le triangle magique



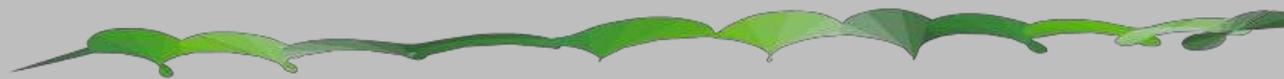
Le Problème



- Le métier et le développement sont souvent enfermés dans une relation malsaine.
- Les deux partenaires doivent changer pour améliorer la satisfaction client et la création de valeur



LE JARGON DE SCRUM



Le Jargon

Sprint:	Est une iteration
Backlog:	Est une liste de tâches ouvertes
Product Backlog:	Est une liste d'items ouverts pour livrer le produit
Sprint Backlog:	Est une liste de tâches ouvertes attribuées au Sprint
L'EQUIPE ou la TEAM:	C'est l'équipe de développement
La Scrum Team:	C'est l' EQUIPE + le ScrumMaster + le Product Owner
Estimation Meeting	C'est la réunion d' estimation
Sprint Planning Meeting	C'est la réunion de planification de Sprint
Daily Scrum ou Stand-up Meeting	C'est la réunion journalière de 15' où l'EQUIPE inspecte et adapte, coordonne son effort.
Sprint Review ou Revue de Sprint	C'est la réunion de fin de Sprint où tous les acteurs du projet se retrouvent pour inspecter les livrables du Sprint.
Rétrospective	C'est la réunion d'inspection et d'adaptation de la Scrum Team.

Équipes auto-gérées vs Organisation traditionnelle

Équipes auto-gérées

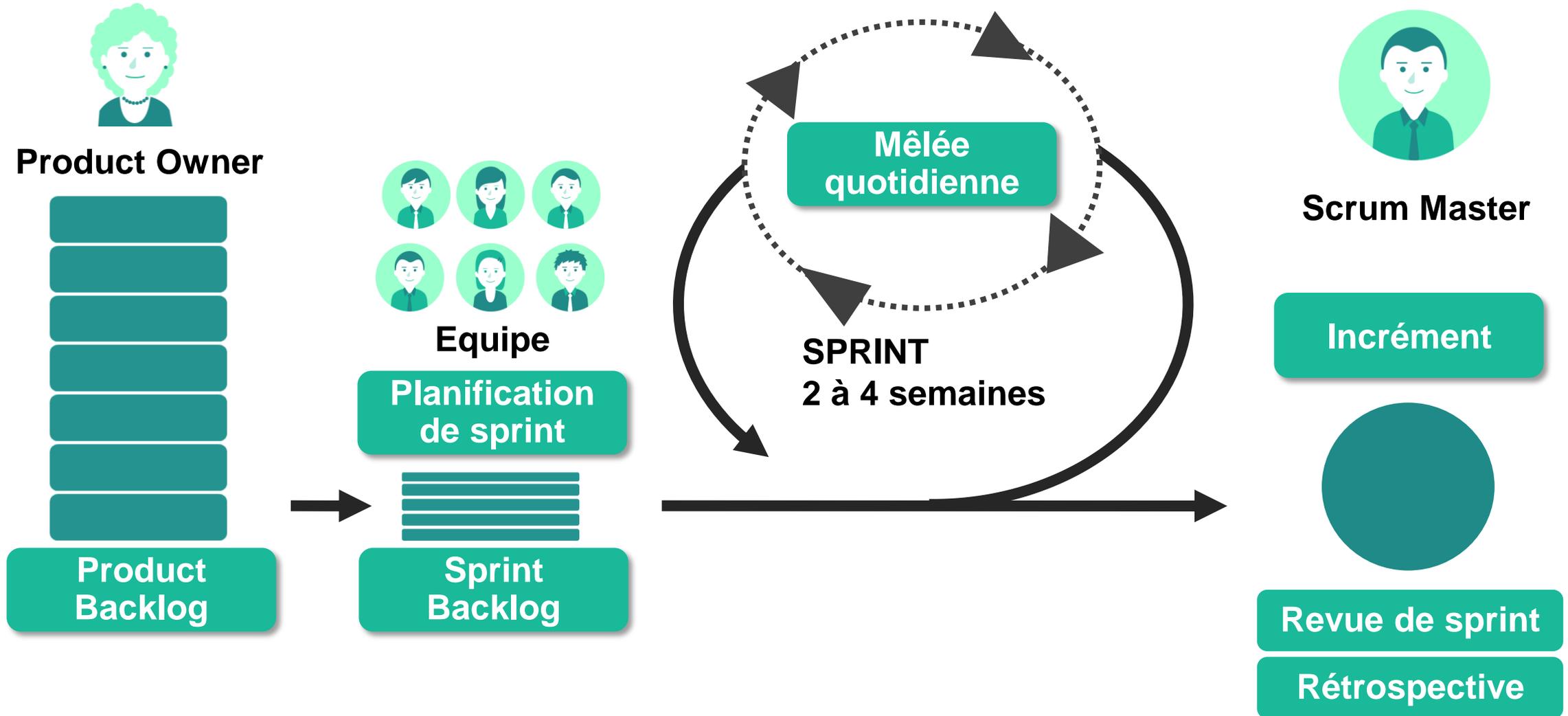


Organisation traditionnelle



Orientées client	Pilotée par le management
Force de travail multi-compétence	Force de travail constituée de spécialistes isolés
Peu de description de poste	Beaucoup de description de poste
Information largement partagée	Information limitée
Peu de niveau de management	De nombreux niveaux de management
Orientée Ensemble du Métier	Orientée fonction/département
Objectifs partagés	Objectifs séparés
D'apparence chaotique	D'apparence organisée
Emphatique sur l'hypothèse d'atteinte du résultat	Emphatique sur la résolution de problème
Très fort engagement des "producteurs"	Très fort engagement du Management
Améliorations continues	Améliorations incrémentales
Autorégulées	Contrôlées par le Management
Basées sur des valeurs et des principes	Basées sur les politiques et les procédures

RESUME DU FONCTIONNEMENT DE SCRUM



Focus sur Scrum

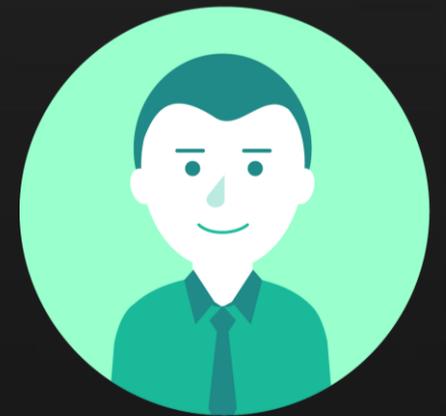
Le ScrumMaster

- Représente le management du projet
- Responsable de faire appliquer par l'équipe les valeurs et les pratiques de Scrum
- Son job est de faciliter la résolution des problèmes



Les qualités d'un bon Scrum Master

- ✓ Humble
- ✓ Collaboratif
- ✓ Engagé
- ✓ Omniscient
- ✓ Responsable

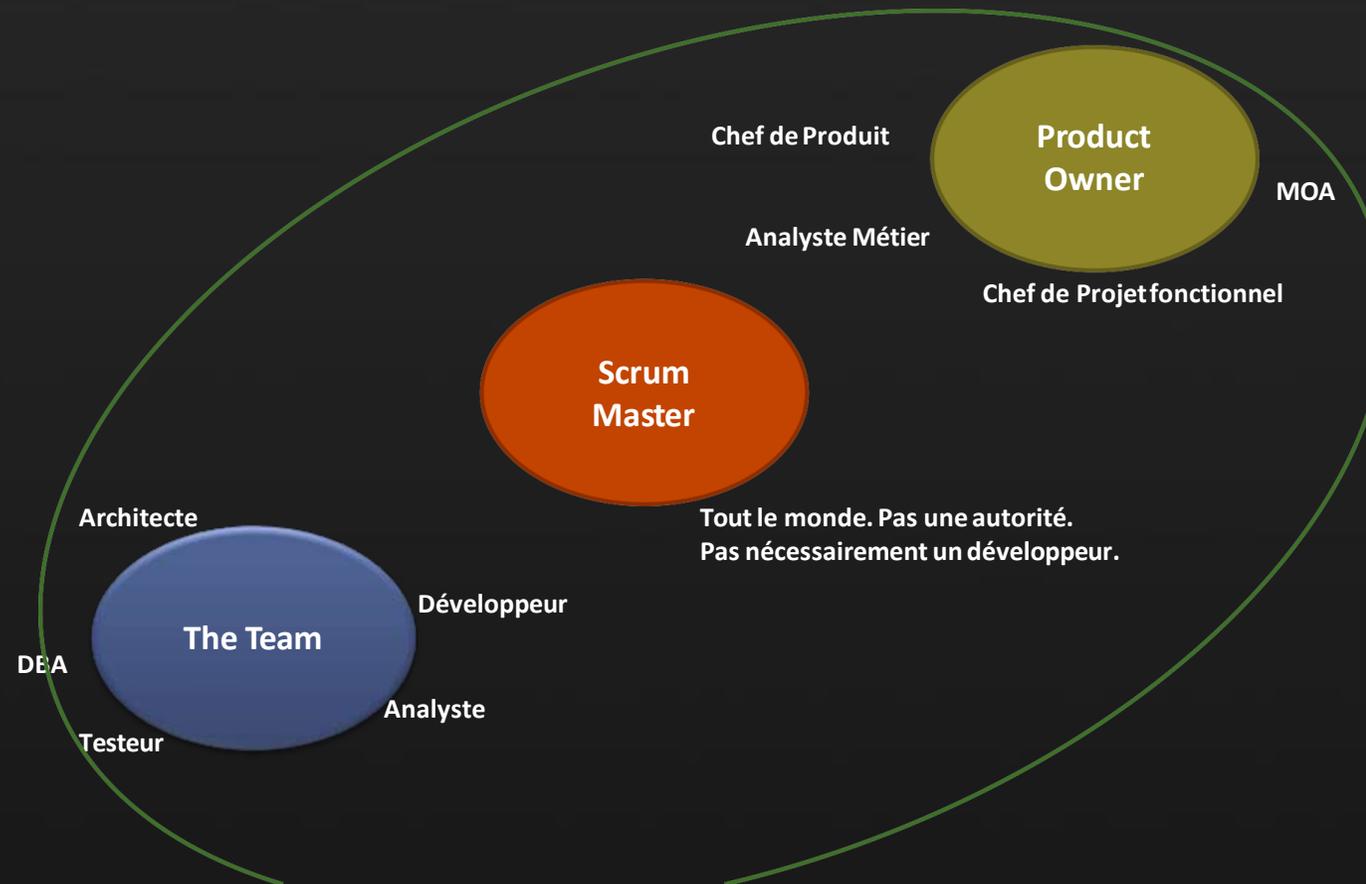


L'équipe Scrum

- Généralement composée de 5 à 10 personnes
- Regroupant toutes les fonctions nécessaires au développement
 - Architecte, Concepteur, Développeur, Spécialiste IHM, Testeur, etc.
- Membres de préférence à plein temps
 - Exceptions possibles (Administrateur, ...)
- L'équipe s'auto gère
 - Normalement pas de titre mais c'est rarement possible
- La composition ne doit changer pendant un Sprint

L'équipe Scrum

Constitution de l'Équipe



L'équipe Scrum

Comment optimiser le travail de l'Équipe...

- Créer une règle de vie de l'Équipe
- Ne jamais utiliser le "VOUS"
- Utiliser un "bâton de parole"
- Ne jamais être nominatif
- Garantir la Qualité
- Livrer
- Livrer
- Livrer
- Estimer
- Estimer
- Estimer
- S'engager
- S'autogérer
- S'organiser Elle-même

LE PRODUCT OWNER

- Il pilote le projet d'un point de vue métier = Chef de Projet Métier
- Il communique une vision claire du produit et définit ses caractéristiques
- Il accepte ou rejette le produit à la fin de chaque Sprint
- Il s'assure que l'Équipe se concentre sur les items du Backlog de plus forte valeur ajoutée
- Il a le même objectif que l'Équipe
- Il est responsable du Retour sur Investissement et des livraisons.
- Se concentre sur le retour sur investissement
- Construit et communique la vision
- Entretien le Product Backlog
- Rend compte de l'acceptance des livrables
- Établit et maintient le Plan de Livraison



Les Rôles Organisationnels



Le Client

Son rôle

Sa fonction

- Il demande le produit
- Il contracte l'organisation pour le développement de son produit
- Typiquement, il s'agit d'un responsable qui achète un développement de produit par un sous-traitant.
- Dans les projets internes, il s'agit principalement du sponsor au projet, c'est à dire la personne validant le projet et le budget.

Sa Mission

- Il commande le produit
- Il paye le développement du produit
- Il donne des feed-back et des révisions



Le Manager

Son rôle

Sa fonction

- Le management, la gestion, est primordial dans tout projet Scrum. Il permet à l'Équipe de constituer un environnement optimal pour le déroulement du projet Scrum.
- Le manager donne de la structure et de la stabilité.
- Il travaille de concert avec le ScrumMaster pour réorganiser l'organigramme de la structure et donner de la guidance si nécessaire.

Sa Mission

- Il s'assure que l'organisation puisse survivre en cas de défaillance.
- Il crée des règles et des lignes directrices.

L'Utilisateur Final



Sa fonction

- Ce rôle peut être joué par un grand nombre de personnes.
- L'Utilisateur final est celui qui connaît les besoins et avec cette connaissance, il définit le produit en disant à l'équipe ce dont il a besoin comme fonctionnalités.

Sa Mission

- Il connaît ses besoins et ses exigences
- Il donne son feed-back lors des revues
- Il participe au Sprint Planning 1

***COMMENT CES RÔLES TRAVAILLENT- ILS
ENSEMBLE?***



Client Final

Le Manager



Le product Owner

Rôles organisationnels



Le Scrum Master

Scrum Team Roles

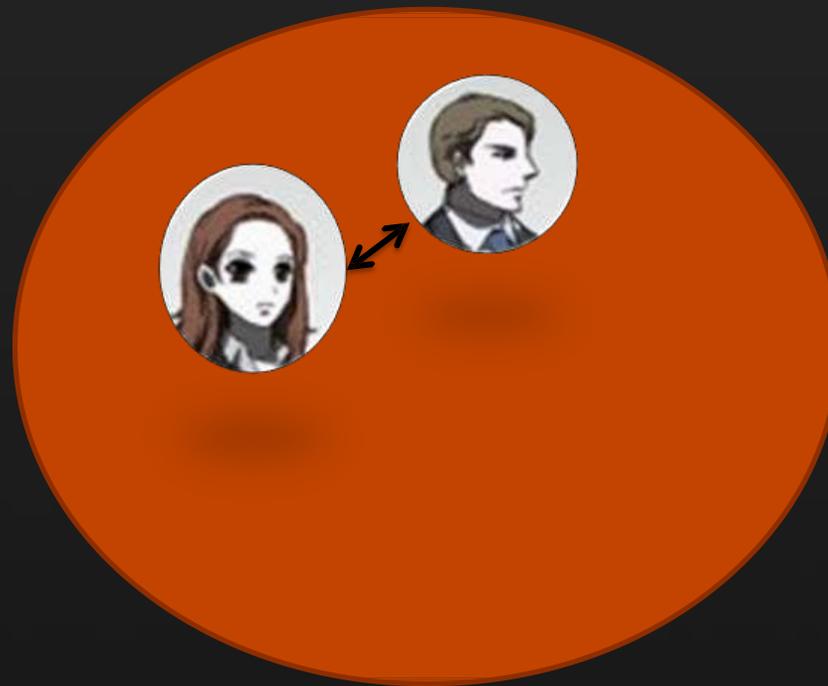


L'Equipe

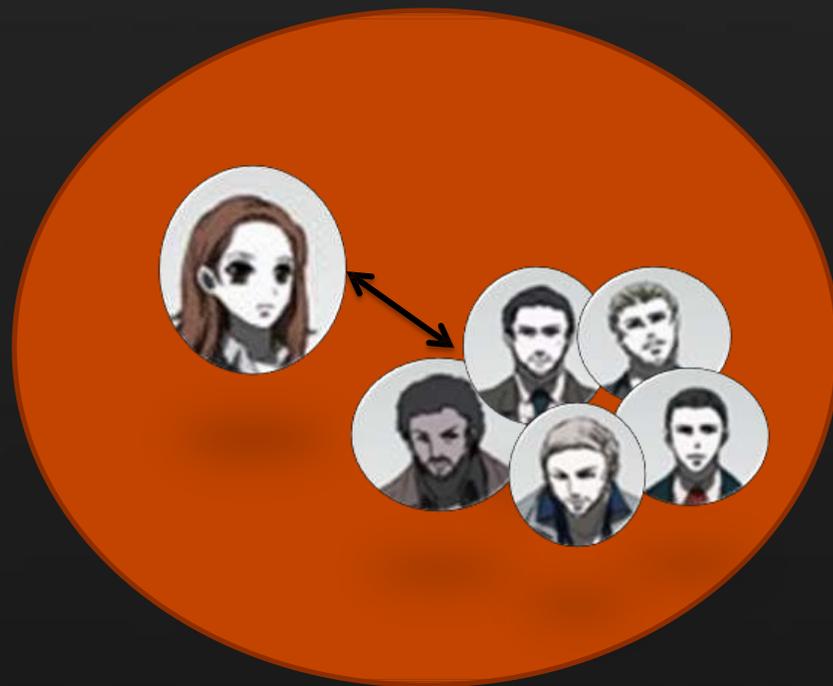


Utilisateur

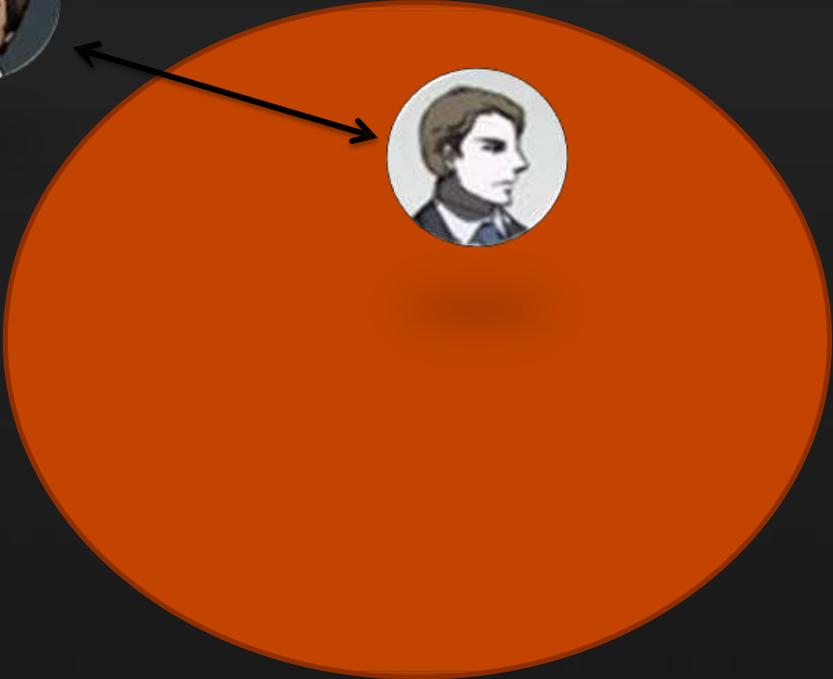
*Le ScrumMaster travaille avec
le Product Owner*



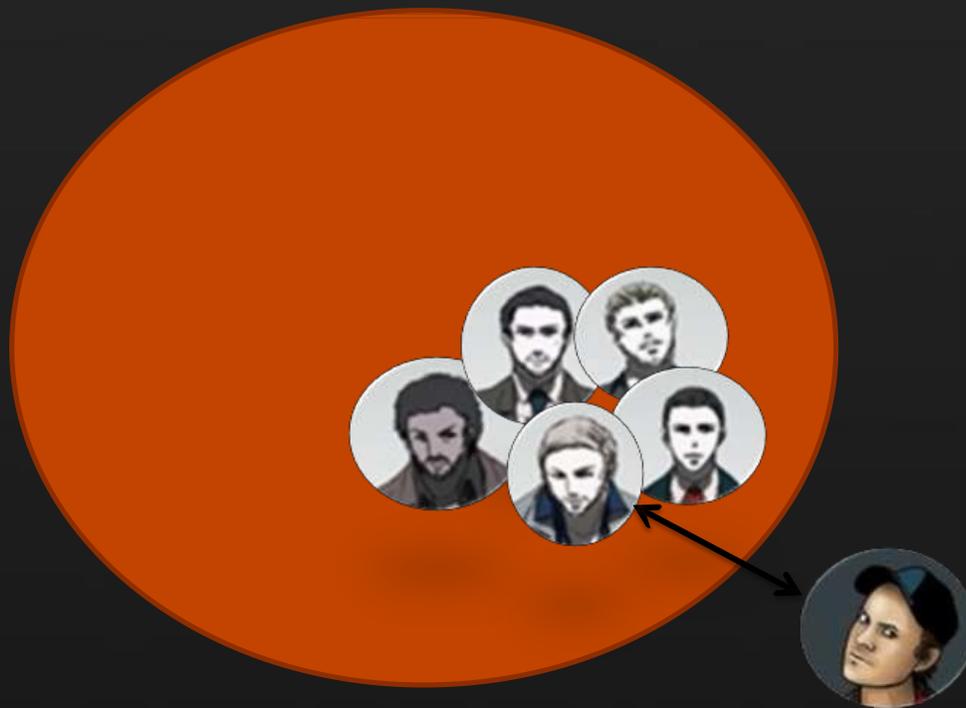
*Le ScrumMaster travaille
avec l'Equipe*



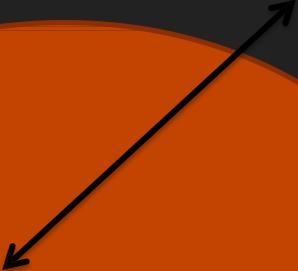
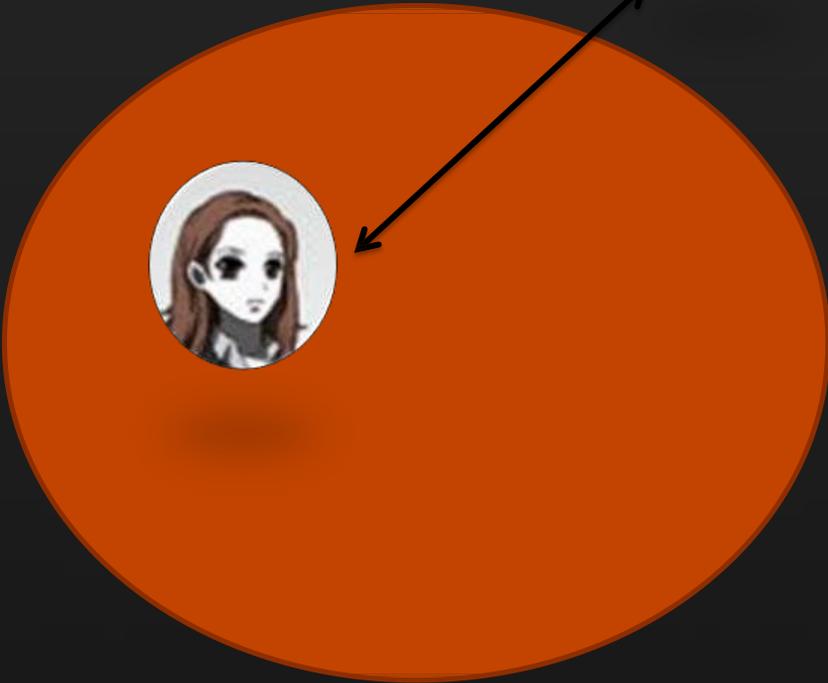
*Le Product Owner travaille
avec le client*



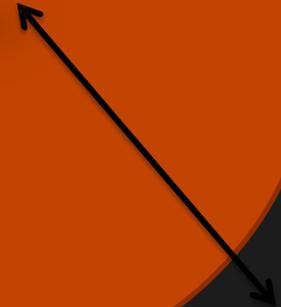
*L'Equipe travaille avec
l'utilisateur final*



***Le ScrumMaster travaille
avec le Manager***



***Le Product Owner a besoin
de connaître ce que le
marché (l'utilisateur final)
souhaite.***

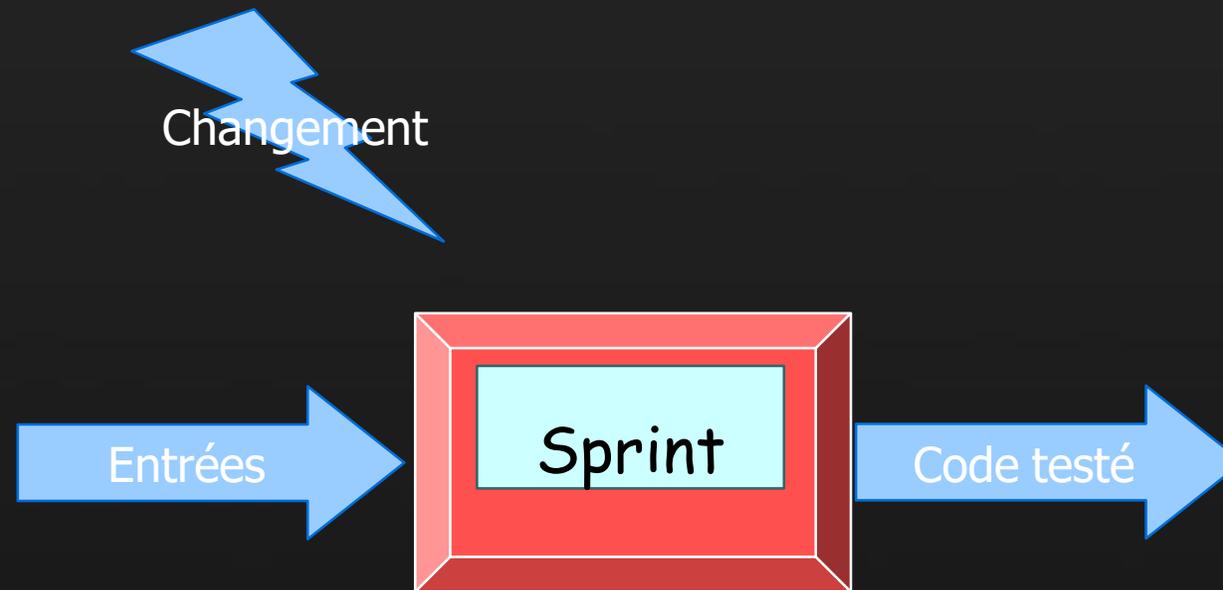


Sprints

- Les projets Scrum progressent à travers une série de Sprints
 - Equivalents aux itérations XP (Extreme Programming)
- La durée d'un Sprint est de 30 jours
 - +/- une semaine ou 2
 - Une durée constante apporte un meilleur rythme
- Le produit est conçu, codé et testé pendant le Sprint

Pas de changements pendant le sprint

- La durée des Sprints doit permettre de différer la prise en compte d'un changement jusqu'au prochain Sprint



Backlogs et Incrément

- Un **backlog** est une liste de fonctionnalités (**Story**) de plusieurs types : user, technical, defect
- Il existe deux backlogs :
 - **Product Backlog** : il recense les stories du projet, priorisées en fonction de la valeur métier que rapporte
 - **Sprint Backlog** : à partir des stories sélectionnées et détaillées par le Product Owner, les développeurs identifient les tâches unitaires qui les composent
- Le Product Backlog est géré par le Product Owner
- Le Sprint Backlog est géré :
 - par le Product Owner pour les aspects fonctionnels
 - par l'équipe pour les aspects techniques (découpage en tâches)
- **L'incrément** est la somme de toutes les fonctionnalités terminées pendant un sprint. Il s'additionne aux autres incréments

Exemple de Backlog



N°	Priorité	Item	Critère d'acceptation	Estimation	Release	Sprint	Statut
42	3	En tant qu'acheteur en ligne, je veux pouvoir supprimer un article de mon panier	L'article est supprimé du panier quand je clique sur « Supprimer ». Je peux voir que l'article ne fait plus parti de mon panier	5	2		Terminé
...		...					

Les fonctions peuvent être ajoutées, repriorisées et supprimées à tout moment. La préparation est une activité à temps partiel (pendant le Sprint) et elle ne doit pas prendre plus de 10% de la capacité de l'équipe.

Exemple de Backlog

Exemple simplifié de backlog de produit

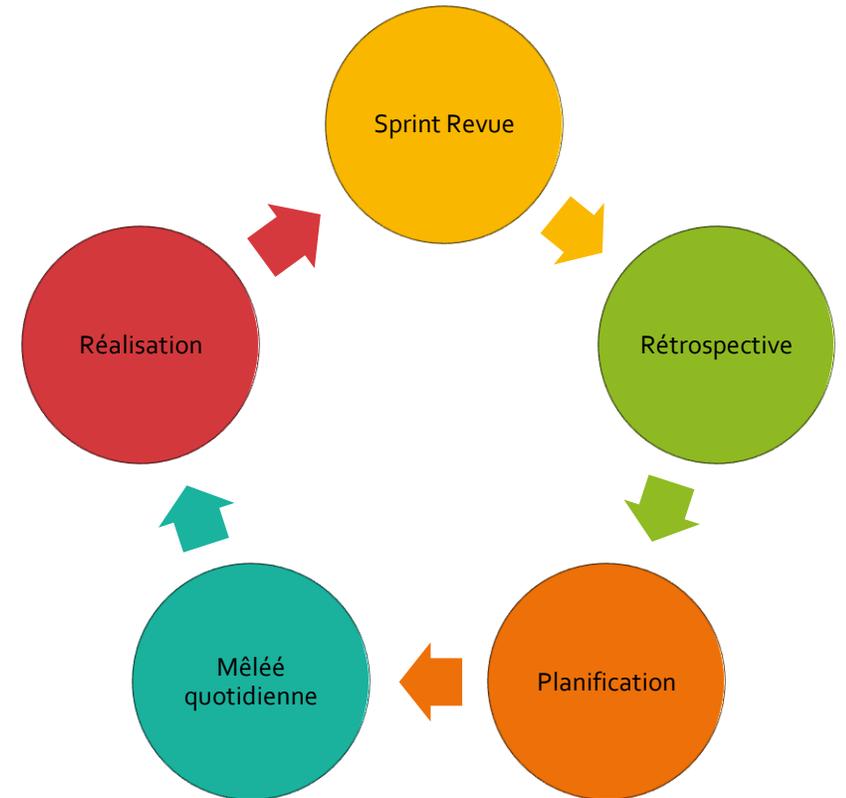
5 jours/homme : a une durée de 5 jours si le nombre de ressources qui travaillent en parallèle et à temps plein est égal à 1. A une durée de 10 jours si le nombre de ressources qui travaillent en parallèle et à mi-temps est égal à 1. A une durée de 1 jours si le nombre de ressources qui travaillent en parallèle et à temps plein est égal à 5

La colonne risque : indique le niveau de complexité technique

Scénario ou story	Priorité	Effort ou charge	Risque
En tant qu'administrateur, je dois pouvoir supprimer un usager du système sans le faire palnter	M	1	1
En tant que client, je veux que mon robot puisse trier les pièces selon leur forme.	M	2	1
En tant que client, je veux que mon robot se déplace entre le point A et le point B selon le chemin le plus court	M	10	3
En tant que candidat, je dois pouvoir soumettre mon CV par interne.	M	3	1

Sprint

- } Bloc de temps : de 1 semaine à 4 semaines
- } A la fin du bloc : Incrément produit « Terminé »
- } Durée constante
- } Contenu :
 - Réunion de planification,
 - Mêlées quotidiennes,
 - Période de réalisation,
 - Revue de Sprint,
 - Rétrospective.
- } Pendant le Sprint :
 - Aucun changement : équipe, objectifs et qualité,
 - Le contenu peut être renégocié entre PO et l'équipe
- } Annulation (objectif du sprint obsolète) : Responsabilité du PO



Réunion de planification



- } Toute l'équipe Scrum
- } Réunion de 8h pour un sprint d'un mois
 - } Deux parties :
 - Qu'est ce qui sera livré dans l'incrément résultant du prochain Sprint ?
 - Comment le travail nécessaire pour réaliser l'incrément sera-t-il accompli ?
 - Définir des tâches d'une journée ou moins



Mêlée Quotidienne

- } Bloc de temps : 15 minutes
- } Objectif : Synchroniser et planifier la journée
- } Tous les jours et uniquement l'équipe de réalisation
- } A la même heure et au même endroit
- } Questions à traiter par chacun des membres :
 - Ce qu'il a réalisé depuis la dernière réunion
 - Ce qu'il réalisera avant la prochaine réunion
 - Les difficultés qu'il rencontre

Le Scrum Master

- S'assure que la mêlée a lieu
- Aider l'équipe sur la tenue de la mêlée
- Veille à l'application des règles



La revue du Sprint



backlog si nécessaire.

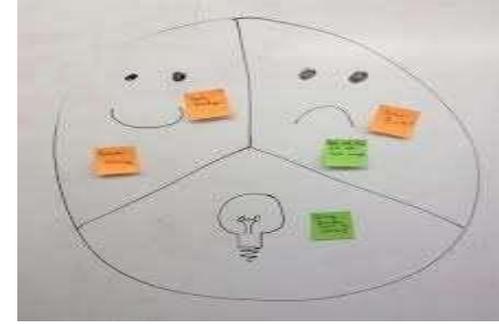
} Qui ? : Equipe Scrum et partie prenantes

} Points abordés :

- Le PO identifie ce qui a été « terminé » et le reste
- L'Equipe de réalisation discute de ce qui s'est bien déroulé et des problèmes rencontrés et comment ils ont été résolus
- L'équipe de réalisation démontre le travail et répond aux questions
- Le PO discute du backlog produit et détermine des dates probables d'achèvement
- L'ensemble du groupe convient de ce qu'il faut faire pour la suite



Rétrospective



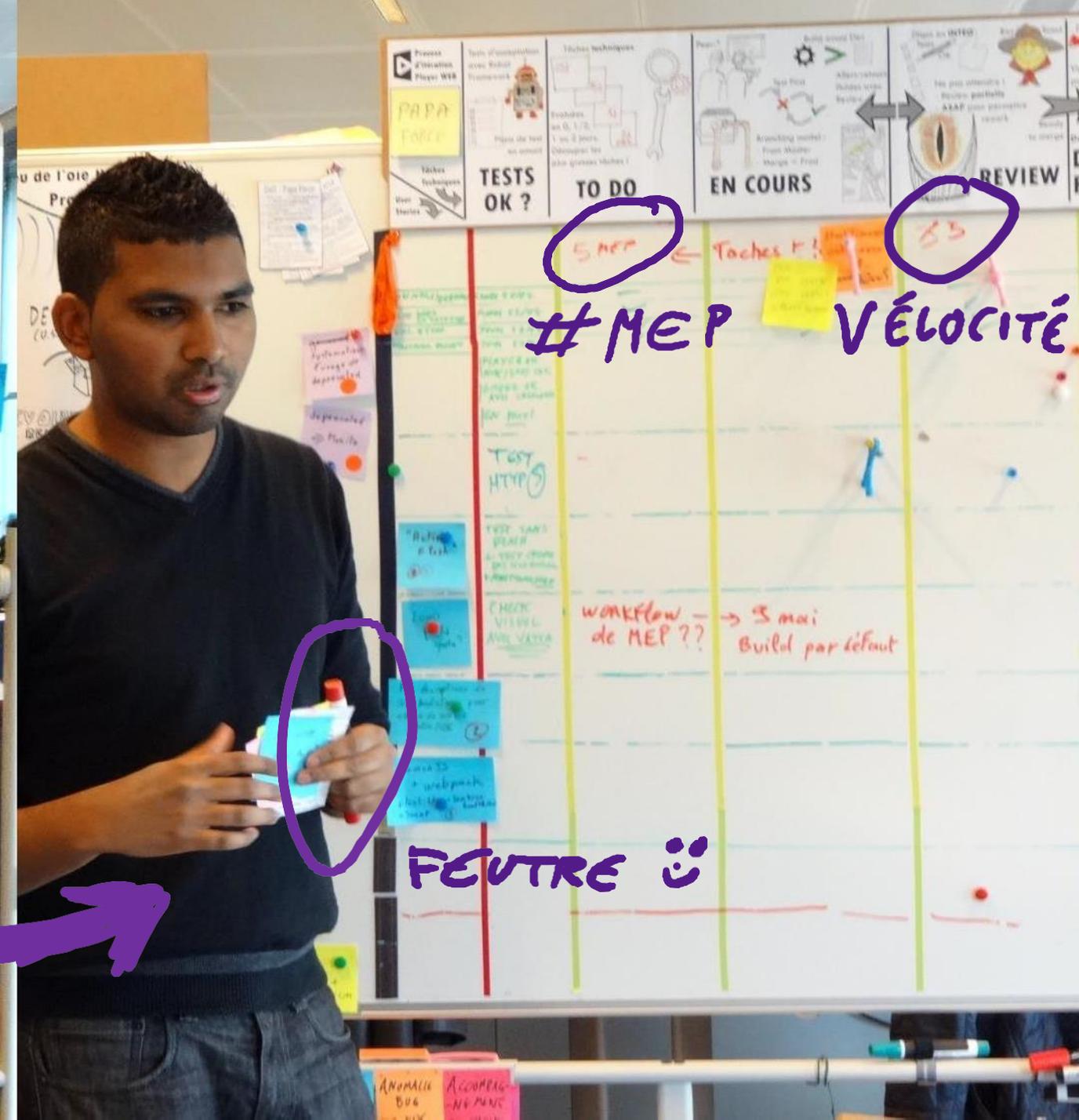
- } Bloc de temps : 3h pour un sprint de 4 semaines
- } Objectif : inspecter et créer un plan d'améliorations

- } Quand : Après la revue de sprint
- } Points abordés :
 - Inspecter la manière dont le dernier sprint s'est déroulé en ce qui concerne les personnes, les relations, les processus et les outils,
 - Identifier et ordonner les éléments majeurs qui se sont bien déroulés et les améliorations potentielles
 - Créer un plan pour améliorer les processus de travail de l'équipe Scrum





INFORMEL



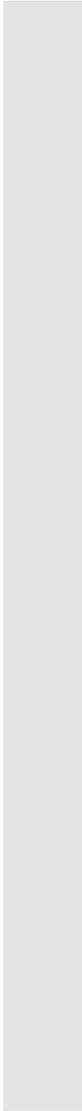
#MEP

VÉLOCITÉ

FENTRE ☺

Questions sur les réunions Scrum

- Pourquoi tous les jours ?
 - “Comment fait un projet pour avoir un an de retard ?”
 - “Un jour à la fois.”
 - Fred Brooks, The Mythical Man Month
- Est ce que les réunions Scrum peuvent être remplacées par des rapports d’activité envoyés par mail ?
 - Non
 - L’équipe entière possède une vision complète actualisée quotidiennement
 - Permet de créer de la pression poussant à faire ce qu’on a dit qu’on allait faire



ARTEFACTS

User Stories

- **En tant que** *[rôle Utilisateur]*
- **Je veux une** *[FONCTIONNALITE]*
- **Afin de** *[VALEUR AJOUTEE].*

User Stories

1. En tant qu'acheteur en ligne,
je veux pouvoir ajouter des items à mon panier supprimer les items afin de pouvoir n'acheter que ce dont je suis vraiment certain.
2. En tant que client,
je peux consulter la liste des factures émises.
3. En tant que client (du projet),
je peux consulter la liste de mes clients.
4. En tant que client,
je peux connaître le montant total des factures impayées.
5. ~~En tant que client, je veux, lorsque je clique sur le bouton «calculer» une ligne s'ajoute et on affiche sur cette ligne le montant total des factures impayées~~

Modèle

Sécurisation des Informations Personnelles

Je voudrais une application bureau que je puisse utiliser pour stocker toute mon information confidentielle tels que les numéros de série, les informations Carte de Crédit, les alias d'enregistrement sur les sites web, les mots de passe, etc. pour chaque item que je souhaite stocker, je dois définir le type de données (comme une date d'expiration). Bien entendu, le système devra être protégé par mot de passe et très sécurisé. Je souhaiterai effectuer des sauvegardes/restaurations online de sorte que je puisse récupérer mes informations à distance. Le produit

User Story Card

- Une brève description textuelle des exigences
 - + Risques
 - + critères d'acceptation



3 lignes de
descriptions

A yellow sticky note with a red pushpin at the top center. The text is arranged in two lines: '3 lignes de' and 'descriptions'.

Les bonnes Stories sont INVEST

I
indépendantes

N
négociables

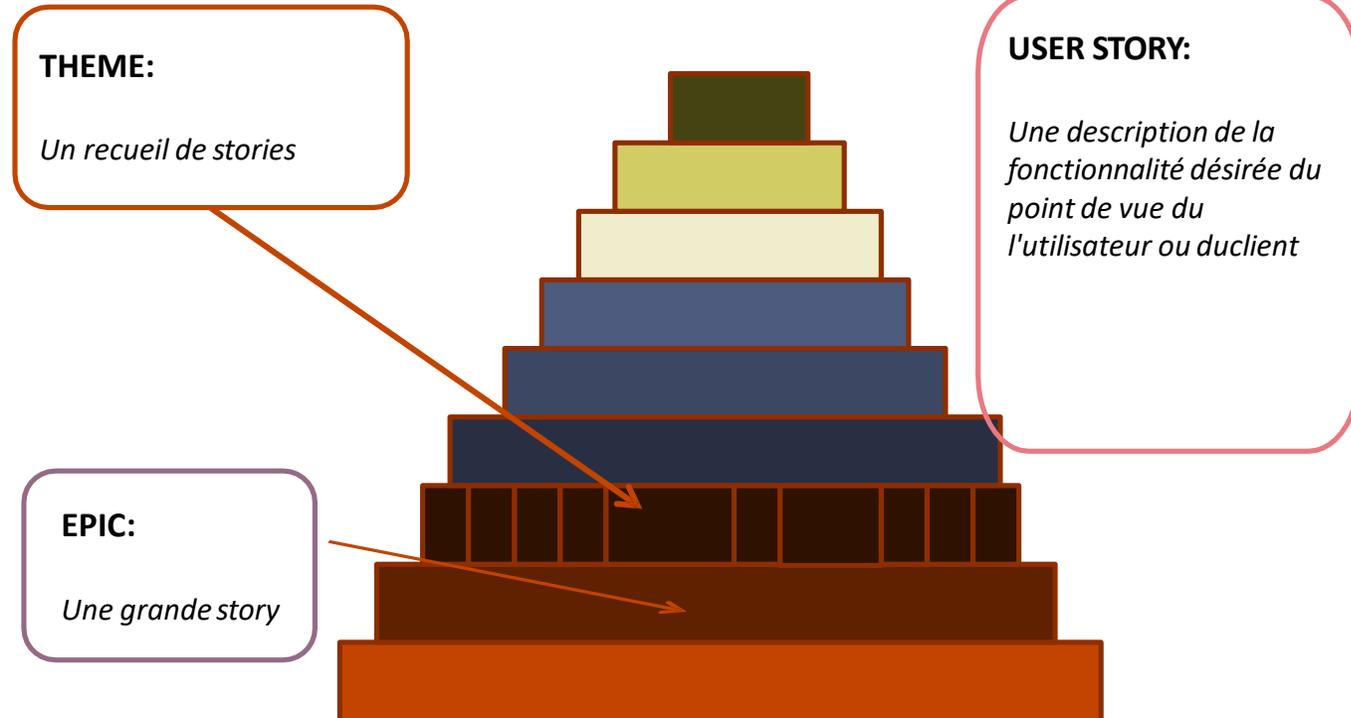
V
valorisables

E
estimables

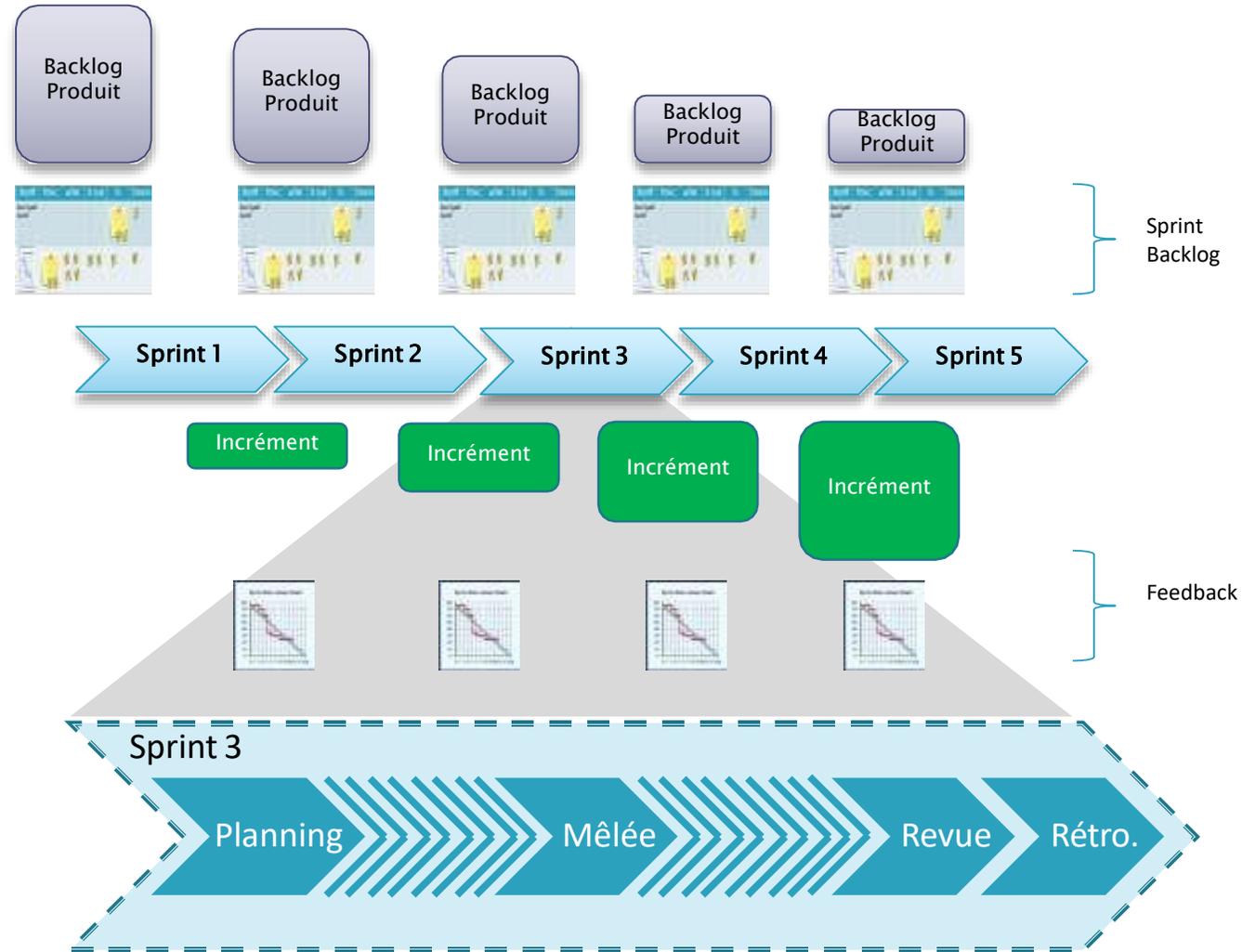
S
Sized to fit (à la
bonne taille)

T
testables

Stories, themes and epics



Synthèse



La certification « Scrum.org »

- } Nombre de questions : 80
- } Temps : 60 minutes
- } Où : En ligne sur le site www.Scrum.org
- } Coût : 100 \$
- } Condition d'obtention : 85 % de bonnes réponses

Limites de SCRUM

- La souplesse des exigences inquiète les nombreuses parties prenantes
- Un projet n'est jamais fini
- L'agilité peut ajouter de la pression à l'équipe
- Agilité = délais fluctuants = inquiétudes
- Gantt (à faire/en cours) ou burndown (reste à faire) ?
- Les niveaux d'implication sont souvent inégaux (investissement personnel) = tensions

Definition of Done / terminé



Exemple de Level of Done

**Pour
l'ÉQUIPE**

- Le Code est conforme aux normes
- Le Code est
 - ✓ Propre
 - ✓ Testé unitairement
 - ✓ Validé (checked in)
 - ✓ Intégré (Built)
 - ✓ Dispose d'une suite de test unitaire qui lui est appliquée.
- Pour arriver à cela, l'environnement de développement est constitué :
 - ✓ D'une bibliothèque de code source
 - ✓ De codes standards,
 - ✓ Build automatisé,
 - ✓ D'un environnement pour les tests unitaires.

Definition of Done

Pour
SCRUM

- Une Story/Item est “done” lorsque l’équipe a atteint son Level of Done
- Le Sprint/Iteration est “done” lorsque
 - tous les items sont “done”
 - et que le Sprint atteint son objectif
 - et que les critères d’acceptation sont adressés.
- La Release est “done”
 - ✓ “done” pour l’intégration
 - ✓ “done” pour la production



Half done is **not** done





COMMENT IMPLEMENTER SCRUM?





Sponsor

Chercher un Sponsor le plus haut possible dans la hiérarchie permet de garantir la bonne mise en place du processus de changement.



Initier

Avancer seul sur un projet de changement est plus que risqué. Faites-vous aider par une ressource externe.



Enflammer

Allumez votre première balise
et enflamez ensuite le
développement.



Diversité

Allumez davantage de balises en diversifiant le nombre des acteurs de votre projet.



Promouvoir

Faites savoir et partagez votre expérience. Vous et votre équipe êtes les promoteurs de Scrum au sein de votre organisation. Allumez encore plus de balises.



Professionaliser

En adoptant une attitude « Scrum », vous engagez un processus d'amélioration continue et de montée en compétence de votre organisation. Créez un Centre de Compétence.



Etablir

Définir Scrum
comme option
officielle.



Consolider

Faites une transition vers
une Entreprise Agile



Intégrer

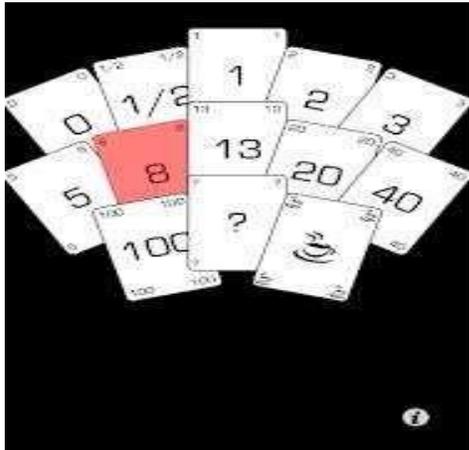
Construisez une
gouvernance IT.

Annexe : outils

- Planning Poker
- Le Feeling Board
- User Story
- Story Board
- Burndown Chart
- Maturité Scrum

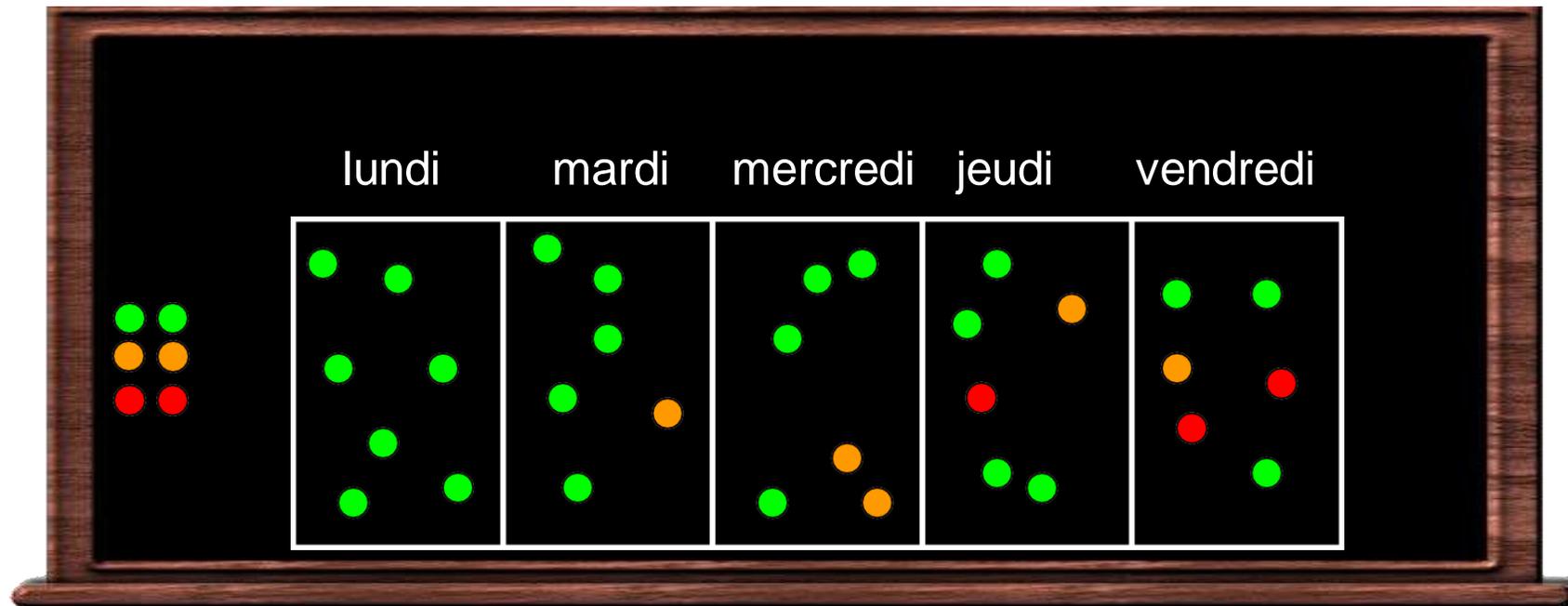
LE PLANNING POKER

- Le **planning poker** est une façon ludique de produire des estimations sur la complexité de fonctionnalités à développer . Cette pratique est surtout utilisée en informatique
- L'estimation serait meilleure parce que plusieurs personnes l'auront validée
 - Les participants s'installent autour d'une table, placés de façon que tout le monde puisse se voir.
 - Le responsable de produit explique à l'équipe un scénario utilisateur (user story).
 - Les participants posent des questions au responsable de produit, discutent du périmètre du scénario, évoquent les conditions de satisfaction qui permettront de le considérer comme "terminé".
 - Chacun des participants évalue la complexité de ce scénario, choisit la carte qui correspond à son estimation et la dépose, face vers le bas, sur la table devant lui.
 - Au signal du facilitateur, les cartes sont retournées en même temps.
 - S'il n'y a pas unanimité, la discussion reprend.
 - On répète le processus d'estimation jusqu'à l'obtention de l'unanimité.



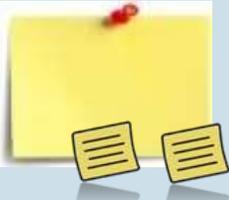
Performance sociale – Le feeling-board

- Une façon simple et efficace de savoir si l'équipe va bien
- Un tableau au mur, avec une case pour chaque jour du sprint
- Chaque soir, chaque membre de l'équipe met une gommette dans la case de la journée :
 - Gommette **verte** : j'ai passé une bonne journée
 - Gommette **orange** : journée moyenne
 - Gommette **rouge** : journée pénible



Story Board

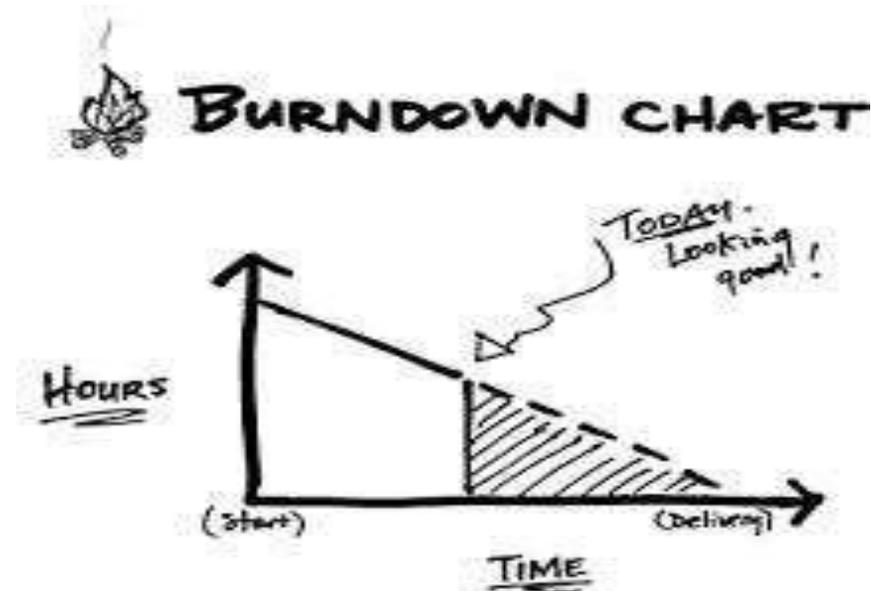
- Techniques pour la collaboration :
 - Management visuel sur les murs, ateliers collaboratifs
- Co-localisation des équipes de travail
- Equipes dédiées

Objectif	Stories	A faire	En cours	Finis	Obstacles
Ecrire l'objectif du sprint					
					

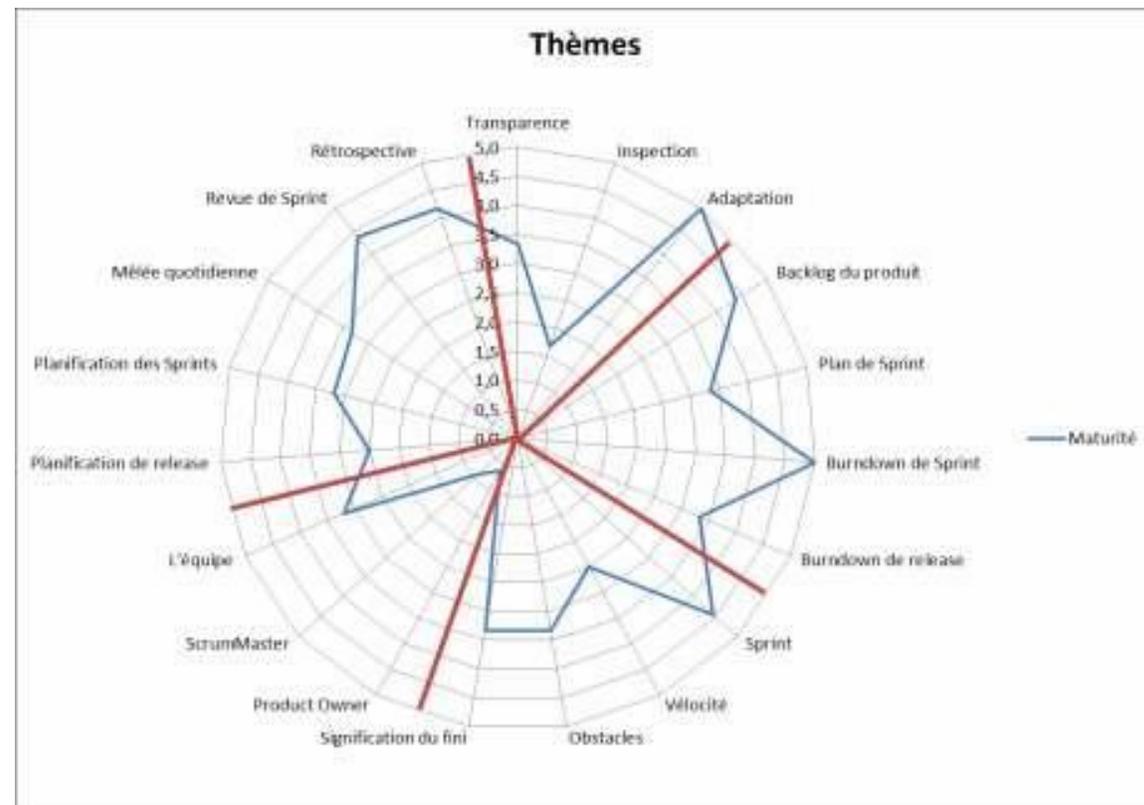
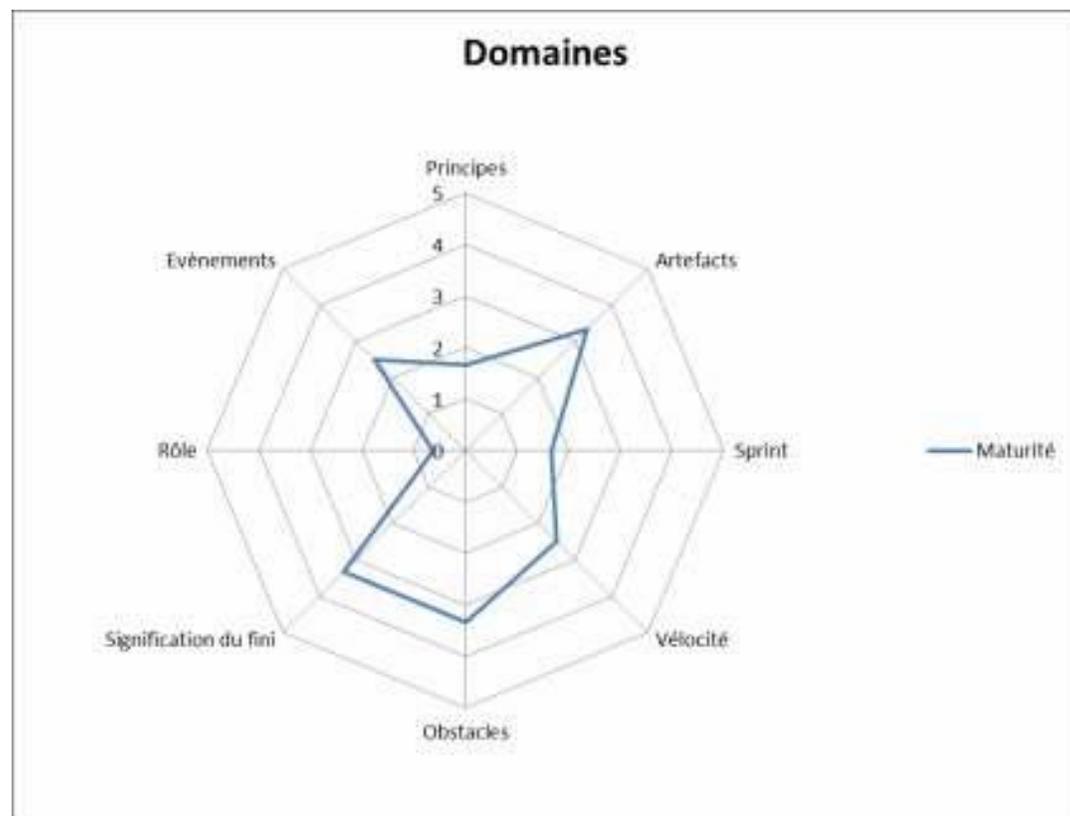
Tâches

Burndown Chart

- Le burndown chart montre la taille de ce qui reste à faire dans le backlog, sprint après sprint ou jour après jour



Maturité Scrum



Maturité Scrum

Domaines	Thèmes	Détails	OUI / NC
Principes	Transparence	L'équipe partage le même langage	O
		L'équipe partage la même définition des termes	N
		L'avancement du processus est visible	O
	Inspection	L'équipe passe en revue le Backlog et le plan du Sprint	N
		L'équipe passe en revue l'état d'avancement en fonction de l'objectif à atteindre	O
		les revues sont réalisées par une personne expérimentée et extérieur à l'équipe	N
	Adaptation	Des actions sont prises si les processus ont besoins d'être ajustés	O
		Des actions sont prises si le produit résultant sera inacceptable	O
	Artefacts	Backlog du produit	Un et un seul Backlog par produit
Le PO le met à jour très régulièrement			O
Il est toujours bien rangé par priorité			O
Tout le monde s'en sert			N
Il comprend bien des stories (pas des tâches)			O
Les stories techniques sont dans le Backlog			O
Les bugs sont dans le Bakclog			O
Chaque Story a ses tests d'acceptation			O
Plan de Sprint		Un et un seul plan par Sprint	O
		Il est facilement visible	O
		Il est mis à jour quotidiennement	N
		Une tâche fait en moyenne un jour	N
		Une tâche est liée à une Story	O
		Le reste à faire est estimé par l'équipe	O
Burndown de Sprint		Il est mis à jour tous les jours	O
		Il est affiché et visible	O
		Il sert à décider sur l'objectif du Sprint	O
Burndown de release		Il est mis à jour à chaque fin de Sprint	O
		Il est affiché et visible	N
		Il sert à décider sur l'objectif de la release	O