

FORMATION

Introduction aux méthodes agiles & SCRUM

08/06/2023

Glodie TSHIMINI
contact@tshimini.fr

Chapitres

- I. Approches prédictives
- II. Approches agiles
- III. Framework SCRUM

Annexe

- [Manifeste agile](#)
- [Principes agiles](#)
- [SCRUM GUIDE 2020 EN](#)
- [GUIDE SCRUM 2020 FR](#)

INSTALLATIONS





GUIDE D'INSTALLATION DU LOGICIEL JIRA

Depuis le dépôt README.md

I. APPROCHES PRÉDICTIVES



Fonctionnement

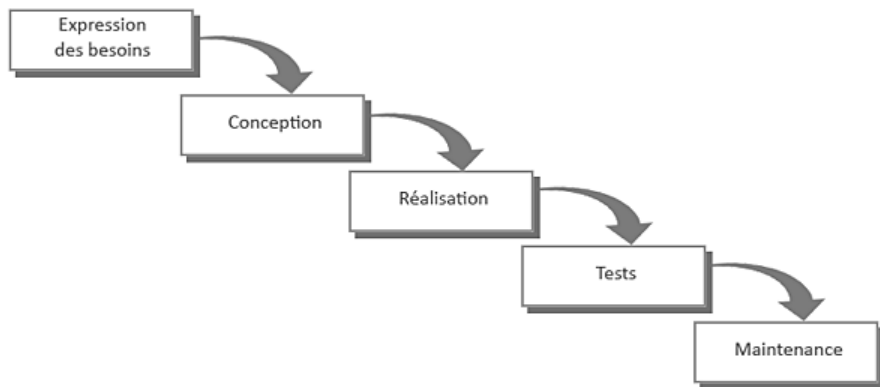
- Trouve ses origines dans le bâtiment
- Séquentielle
- Besoins spécifiés dans un cahier des charges
- Planification rigoureuse dans un diagramme de Gantt

Faiblesse

- Très peu d'interaction avec le client
- Peu flexible
- Peu d'adaptabilité

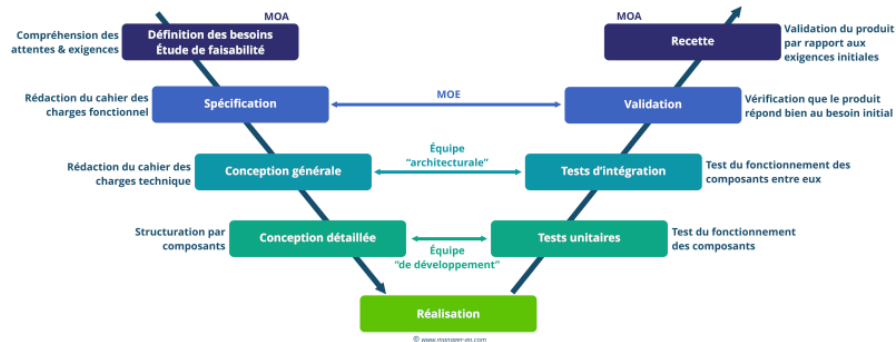
CYCLE EN CASCADE

Source image editions-eni.fr



- Apparition en 1956 Hebert D.Benington
- Une étape doit être fini à 100% avant de passer à l'étape suivante (linéaire)
- Planification rigoureuse et précise
- Besoin prévisible et stable dans le temps
- Repose sur spécifications fonctionnelles, techniques et globalement sur le cahier des charges
- Une mauvaise définition du besoin aura un grand impact sur la durée et le coût du projet
- Moins d'interaction entre les parties prenantes
- Intervention des utilisateurs finaux qu'à la fin

Source image manageo-go

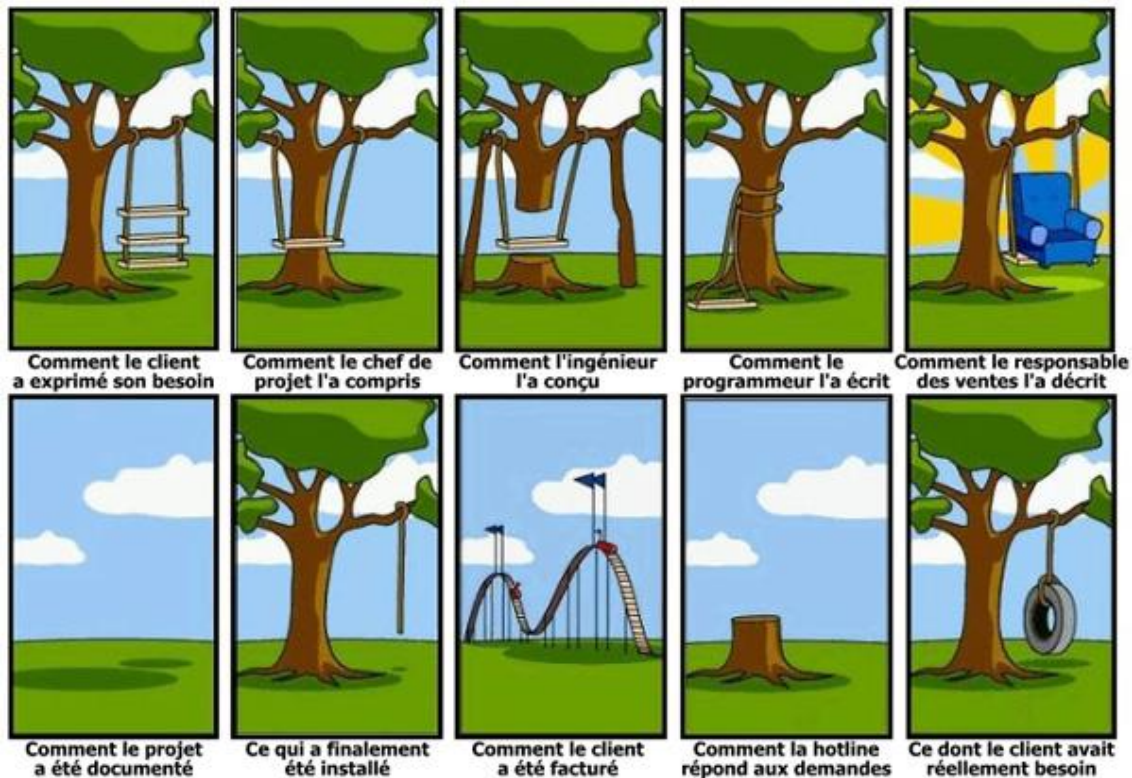


- Déclinaison de la méthode cascade
- Place plus importante des tests dans le cycle
- Une phrase descendante et ascendante qui peuvent s'effectuer en parallèle
- MOA
 - Maitrise d'ouvrage
 - Chargé de définir les besoins
- MOE
 - Maitrise d'œuvre
 - Chargé de transformer le besoin en solutions digitales (réalisation)

II. APPROCHES AGILES

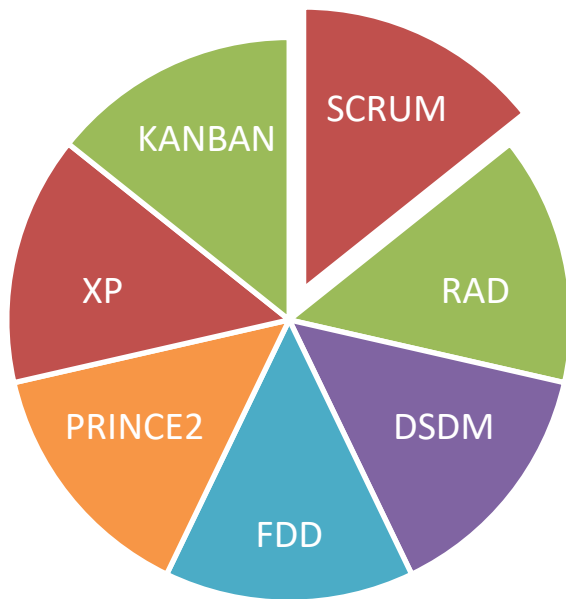


Vision d'un projet digitale



[Source image anyideas](https://anyideas.com)

Les quelques méthodes agiles



- RAD (Rapid Application Development)
 - Livrer rapidement les fonctionnalités à forte valeur pour un ROI rapide
- DSDM (Dynamic Software Development Method)
 - Collaboration étroite entre les utilisateurs finaux et les devs
- FDD (Feature Driven Development)
 - Réduire les risques avec des cycles de développement très court
- Prince2 (Projects In Controlled Environments)
 - Tient compte des facteurs d'environnement du projet susceptibles d'impacter sa réussite.

KANBAN



CARACTÉRISTIQUES DU FRAMEWORK KANBAN

[Source image Ionis](#)



- Framework (cadre de travail) léger et visuel
- Facile à mettre en place
- Travail en flux continu
- Mise en place
 - Modélisation le flux de travail
 - Amélioration possible avec le Work In Process (WIP)
 - Nombre maximal des tâches dans une colonne

EXTREME PROGRAMMING



CARACTÉRISTIQUES DU FRAMEWORK KANBAN

Source image extremeprogramming.org



4 principes

1. Communication : les développeurs communiquent directement avec les utilisateurs finaux
 2. Simplicité: applique le principe *KISS (Keep It Simple and Stupid)* = choisir à chaque fois la solution la plus simple.
 3. Retour d'expérience : impliqué l'utilisateur dans les tests, priorisé les fonctionnalités, améliorer la qualité du code
 4. Livraison : fréquente et rapide
- Les bonnes pratiques de développement sont appliqués et poussées à l'extrême

4 VALEURS AGILES (MANIFESTE AGILE)

1. Les individus et les interactions, de préférence aux processus et aux outils.
2. Des solutions opérationnelles, de préférence à une documentation exhaustive.
3. La collaboration avec les clients, de préférence aux négociations contractuelles.
4. La réponse au changement, de préférence au respect d'un plan.

Précisément, même si les éléments à droite ont de la valeur, nous reconnaissons davantage de valeur dans les éléments à gauche.

LES 12 PRINCIPES AGILES

Knowledge
TRAIN

Les 12 principes Agile *

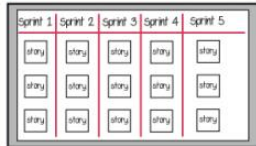
1 Satisfaire le client



2 Accueillir les changements



3 Livrer fréquemment



4 Travailler ensemble



5 Confiance et soutien



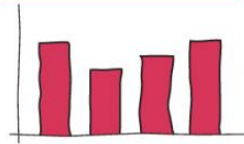
6 Dialogue en face à face



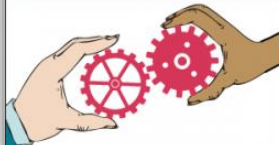
7 Logiciel opérationnel



8 Développement soutenable



9 Attention continue



10 Privilégier la simplicité



11 Équipes auto-organisées



12 Réfléchir et adapter



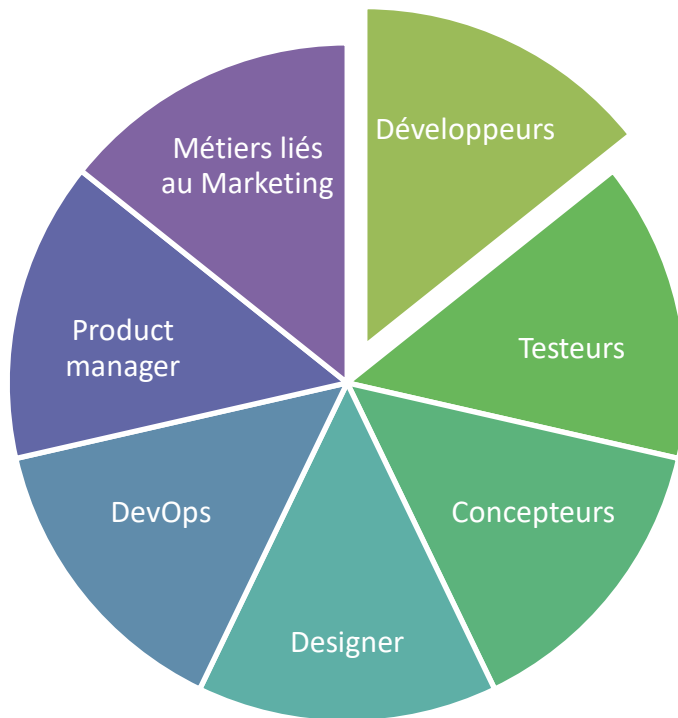
Copyright © 2016, Knowledge Train Limited. www.knowledgetrain.co.uk. *Quoted from www.agilemanifesto.org

[Source image wikiagile](https://www.wikiagile.com/)

LES EQUIPES AGILES



Quelques méthodes agiles



EXERCICE





EXERCICE 1 : AVANTAGES ET INCONVÉNIENTS DES APPROCHES TRADITIONNELLES ET AGILES

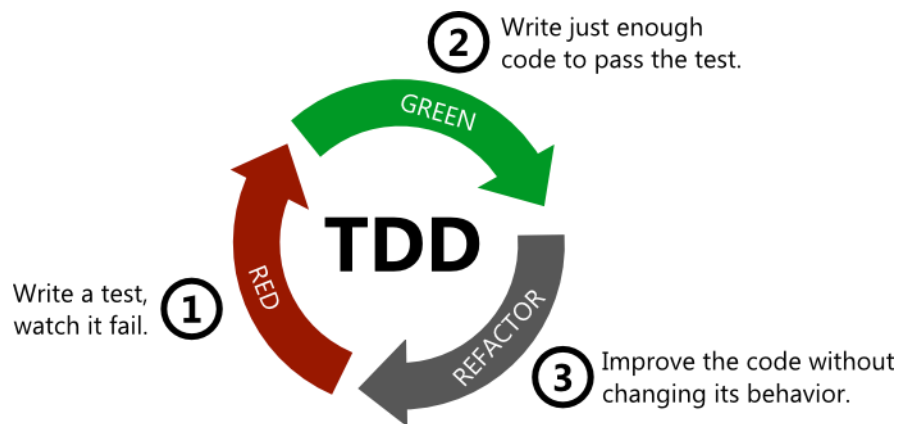
1-exercices/ex1-methodes.md

LES BONNES PRATIQUES DE DÉVELOPPEMENT



TEST DRIVEN DEVELOPMENT (TDD)

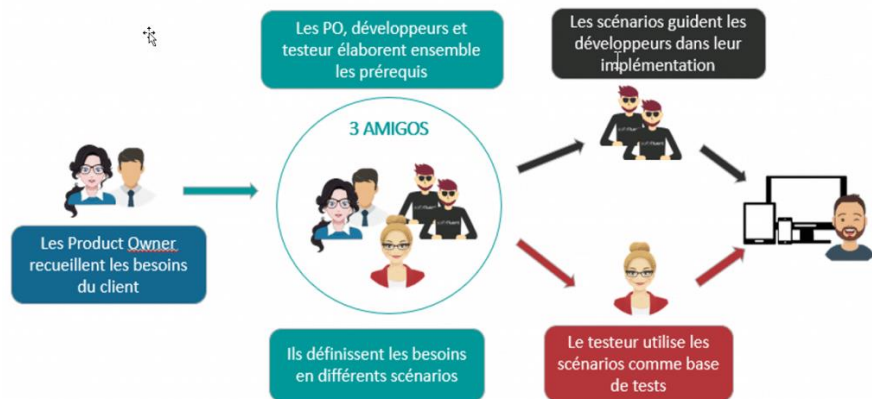
[Source image Radixweb](#)



- Développement piloté par les tests unitaires
- Écrire les tests en premier avant d'implémenter le code
- Réduit les bugs, mais ne les empêchent pas
- Résultat
 - Suite des tests de régression automatisée (on s'assure qu'une modification future de l'application n'induit pas d'erreurs sur l'existant)
 - Meilleure qualité du code

BEHAVIOUR DRIVEN DEVELOPMENT (BDD)

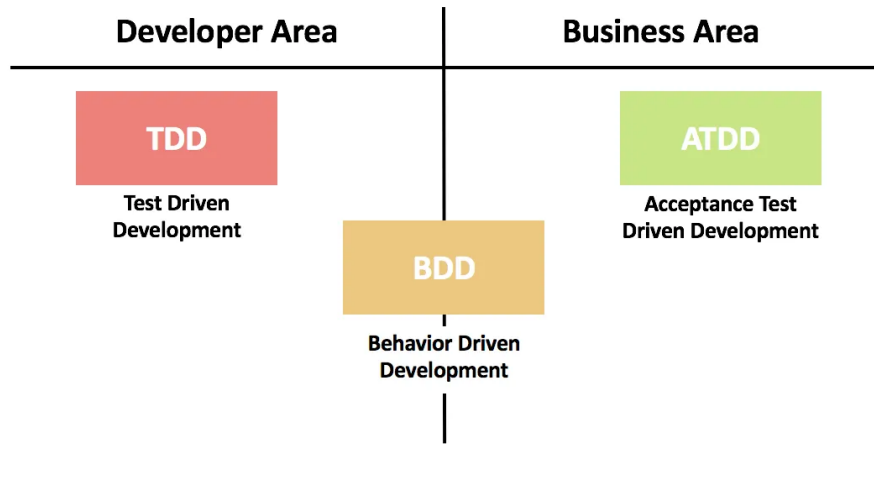
Source image Softfluent



- Développement piloté par le comportement des utilisateurs finaux
- Des scénarios de cas d'utilisation sont rédigés en langage naturel
- Ces scénarios de cas d'utilisation servent de base d'abord à rédiger des tests puis à développer
- Utilisation du langage naturel compris par tous
- Le BDD est généralement combiné au TDD

DIFFÉRENCES TDD, BDD, ATDD

[Source image Widyan Ivan](#)



[Source image Damia Mutlu](#)

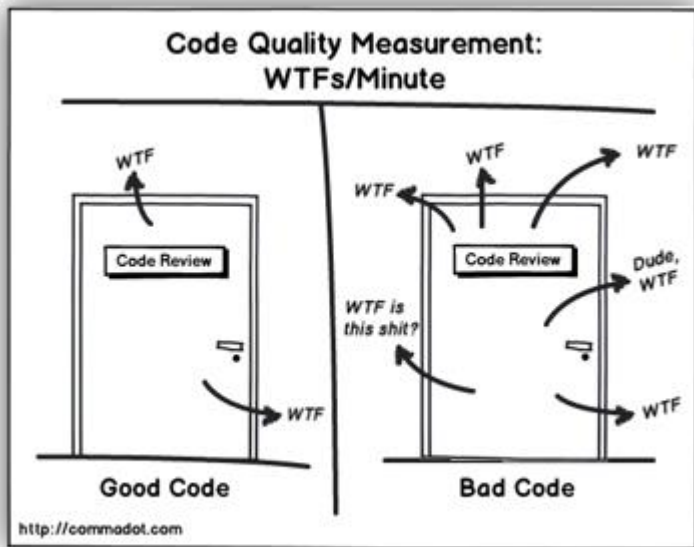
Parameters	TDD	BDD	ATDD
Definition	TDD is an approach of development that focuses on implementation of features	BDD is an approach of development that focuses on system behaviors.	ATDD is a technique that focuses defining accurate requirements.
Participants	Developers	Developers, Customers, QAs	Developers , Customers, QAs
Language	Testing language of used programming language	Simple native language	Simple native language
Main Focus	Unit test	Understanding requirements	Defining acceptance criteria.

Bonnes pratiques de développement

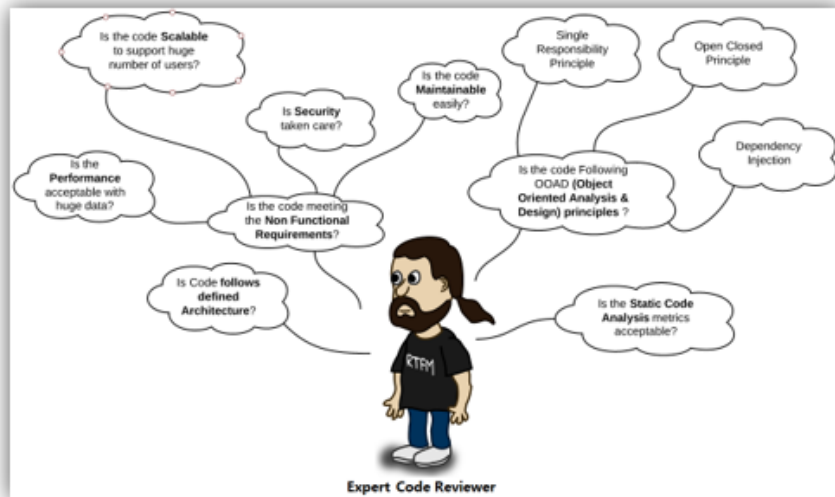
- CI/CD (Continuous Integration / Continuous Delivery ou Deployment)
 - Automatisation de l'intégration du code
 - Automatisation des livraisons en production
 - Delivery = action humaine
 - Deployment = sans action humaine
- Pair programming
 - Développement en binôme
 - 1 Driver (tactique)
 - 1 Navigator (stratège)
- Code review
 - Revue de code par l'équipe

Grandes lignes du code review

[Source image commadot](http://commadot.com)



[Source image evoketechnologies](http://evoketechnologies.com)



III. FRAMEWORK SCRUM



HISTORIQUE

- Développé dans les années 90
- Créateurs
 - [Ken Schwaber](#)
 - [Jeff Sutherland](#)
- SCRUM GUIDE
 - Document officiel qui partage la vision de SCRUM par ses créateurs
 - Les versions
 - Première en 2010
 - Précédente version 2017
 - Version actuelle **2020**

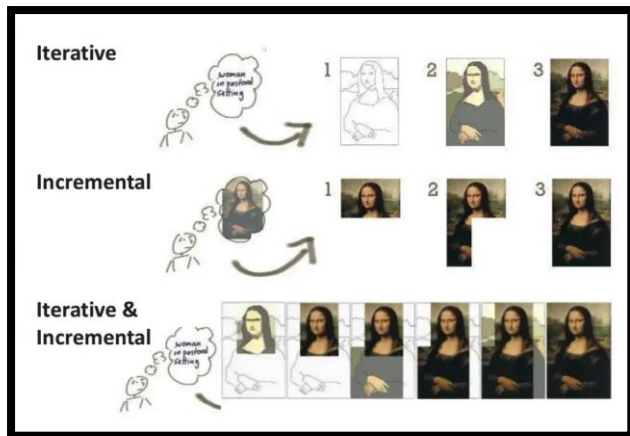
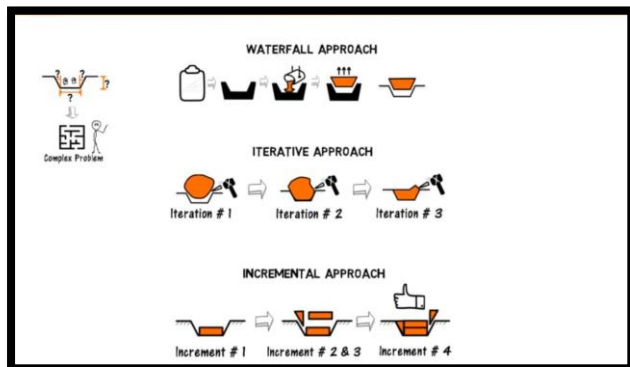
FRAMEWORK

- SCRUM est
 - Un Framework léger
 - Un ensemble des **règles** pour **guider les relations et les interactions** entre les individus participant au projet
 - **Volontairement incomplet** pour laisser la liberté à **l'intelligence collective** (capacité à résoudre des problématiques en travaillant ensemble avec des profils divers)
 - Un Framework Agile
 - Axé sur la **valeur du produit** et sa **maximisation**
 - **Adapté aux solutions innovantes et complexes**
 - Une aide pour les équipes à générer rapidement de la valeur aux produits et services à développer
 - Utilisé pour des projets dans divers domaines (n'est plus exclusivement réservé à l'IT)

THÉORIE : EMPIRISME ET LEAN

- La théorie de SCRUM se base sur l'Empirisme et la pensée LEAN
- Empirisme
 - Courant philosophique qui dit que toutes connaissances viennent de l'expérience
 - Appliqué dans le cadre SCRUM, cela signifie que les « *décisions à prendre doivent s'appuyer sur l'observation de faits* »
- Pensée du LEAN
 - Réduction du gaspillage et concentration sur l'essentiel
 - Meilleure performance
 - Meilleure rentabilité
 - Meilleure qualité

APPROCHE ITÉRATIVE ET INCRÉMENTALE



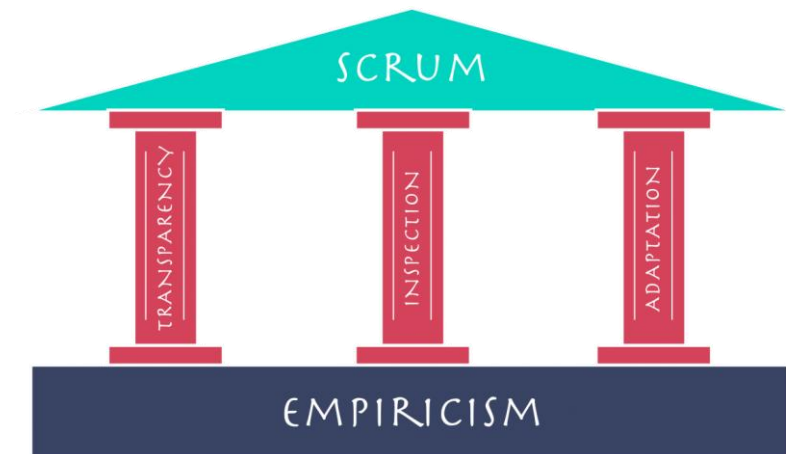
[Source image Vscrum Team](#)

[Source image bootcamp uxdesign](#)

- « *Approche à la fois itérative et incrémentale pour optimiser la prédictibilité et le contrôle de risque.* »
- Les équipes SCRUM ont toutes les **compétences** et **expertises** pour travailler sur le projet
- Les membres de l'équipe **partagent** leurs compétences et expertises au sein de l'équipe

LES 3 PILLIERS DE SCRUM

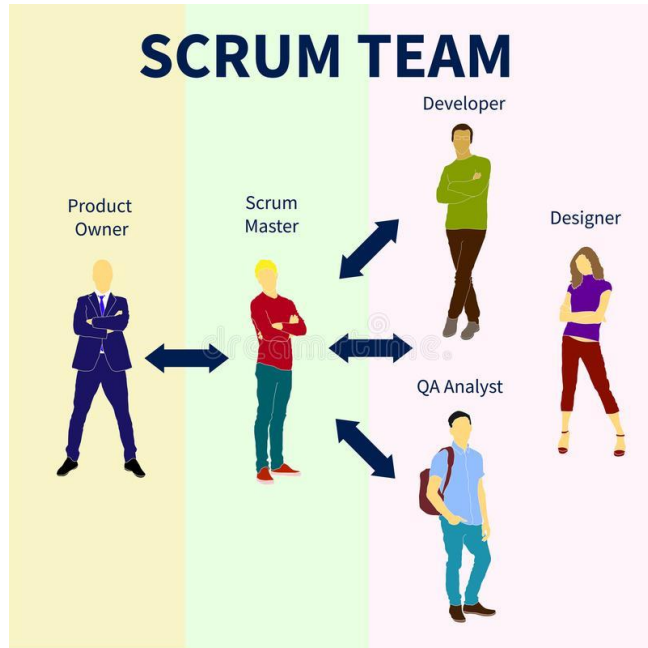
[Source image wikiagile](#)



- **Transparence**
 - Tout ce qui concerne le produit doit être **visible par toutes les personnes impliqués**
 - Peu de transparence augmente le risque et diminue la valeur du produit
 - La transparence permet l'inspection
- **Inspection**
 - Inspection fréquente du travail en cours pour **détecter rapidement les écarts**
 - L'inspection permet l'adaptation
- **Adaptation**
 - **S'ajuster** dès que les écarts deviennent inacceptables pour atteindre l'objectif fixé

L'ÉQUIPE SCRUM

Source image dreamstime



- 1 Scrum Master
 - 1 Product Owner
 - Developers (3 à 9 individus)
-
- Pas de hiérarchie
 - Pas de sous-équipe
 - Pas d'autres rôles que les 3 ci-dessus

COMPÉTENCES ET EXPERTISE DE L'ÉQUIPE

- Les équipes SCRUM sont
 - **Cross-functional** (Multifonctionnel ou transversal)
 - Les membres ont toutes les compétences requises pour produire de la valeur à chaque itération de la réalisation du produit
 - **Self-managing** (Auto-gestion)
 - Les membres décident de qui fait quoi, quand et comment
 - Les équipes sont **soutenues par leur direction** dans leurs prises de décision
 - Tout le monde est responsable de la création de valeur.
- Les équipes petites plus petites ont tendances à mieux s'organiser et à être plus productif (c'est une généralité et non une vérité absolue)
 - Plusieurs équipes SCRUM peuvent travailler sur le même produit
 - Plusieurs teams
 - Avoir le **même Product Goal** (Objectif produit)
 - Avoir le **même Product Backlog** (Carnet de produit)
 - Et le **même Product Owner (PO)**

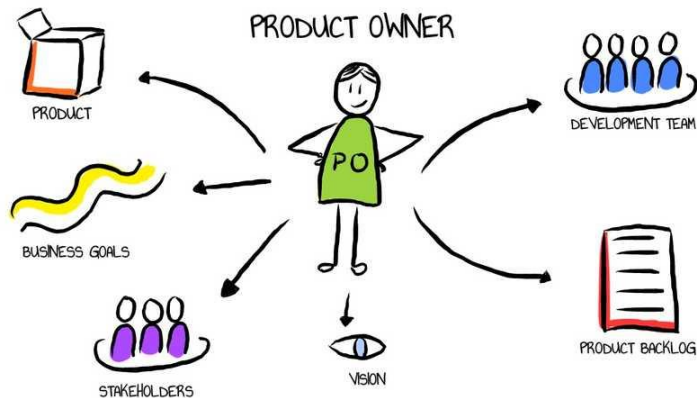
Source image Letsscrumit



- Le rôle *developers* regroupent toutes les personnes responsables de la création d'au moins un incrément de valeur à chaque sprint. (équipe de réalisation)
- Taille maximal 9 individus
- Responsabilités
 - Plan de Sprint (plan d'action)
 - *Sprint Backlog*
 - Adhérer à la *Definition of Done*
 - Adapter leur plan chaque jour pour atteindre le *Sprint Goal* (objectif du sprint)
 - Être professionnels

PRODUCT OWNER

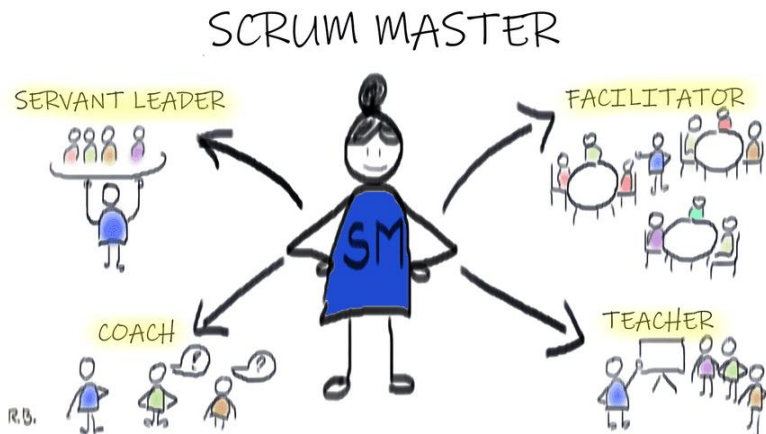
Source image Letsscrumit



R.B.

- Une seule personne
- Représente les autres parties prenantes (client, utilisateurs finaux etc.) au sein de l'équipe
- Responsabilités
 - Maximiser la valeur du produit
 - Gérer efficacement le **Product Backlog (PB)**
 - Développer et communiquer le **Product Goal**
 - Créer, communiquer et prioriser les **Product Backlog Items**
 - Rendre le PB transparent, visible et compréhensible
 - Peut déléguer une partie de son travail aux autres membres de l'équipe cependant il demeure le seul responsable et le dernier mot lui revient.

Source image Letsscrumit



- Responsabilités
 - S'assurer de la **bonne application de SCRUM**
 - Tous les événements ont lieu avec le respect du temps (*timebox*) et que les membres concernées y participent
 - Aide l'équipe et l'organisation à comprendre *SCRUM*
 - Efficacité de l'équipe
 - Aider à l'amélioration des pratiques
 - Au service de l'équipe
 - En tant que **facilitateur**
 - Celui qui **lève les obstacles**
 - **Coach** l'équipe dans l'autogestion et partage des compétences
 - Aide l'équipe à se concentrer sur la maximisation de la valeur

Collaboration avec le Product Owner

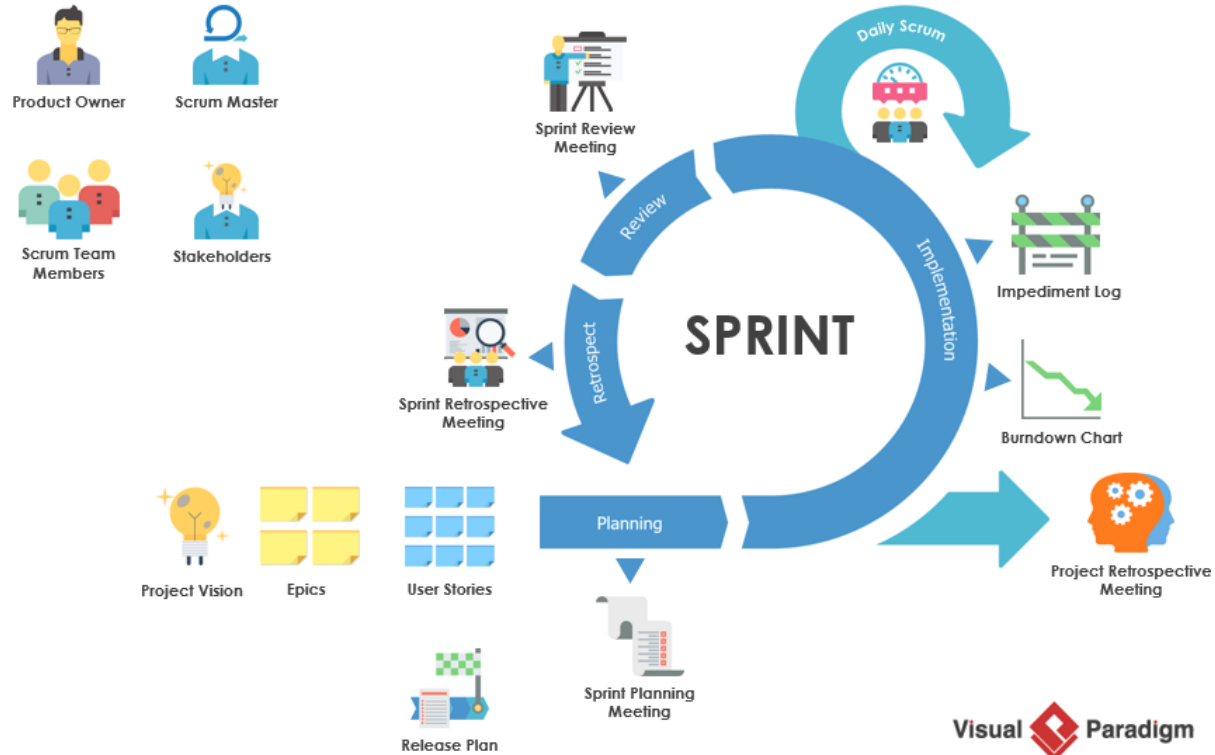
- Aide à trouver des techniques plus efficace pour communiquer le *Product Goal*
- Aide à gérer efficacement le *Product Backlog*
- Aide à comprendre et à décrire clairement les *Product Backlog Items*
- Facilite la collaboration avec les parties prenantes du projet

Collaboration avec l'organisation (entreprise)

- Guide, entraîne et coach l'organisation à adopter *SCRUM*
- Planifier et conseil l'organisation à mettre en place *SCRUM*
- Supprime les barrières entre les parties prenantes et l'équipe *SCRUM*

CYCLE SCRUM [source image visual paradigm](#)

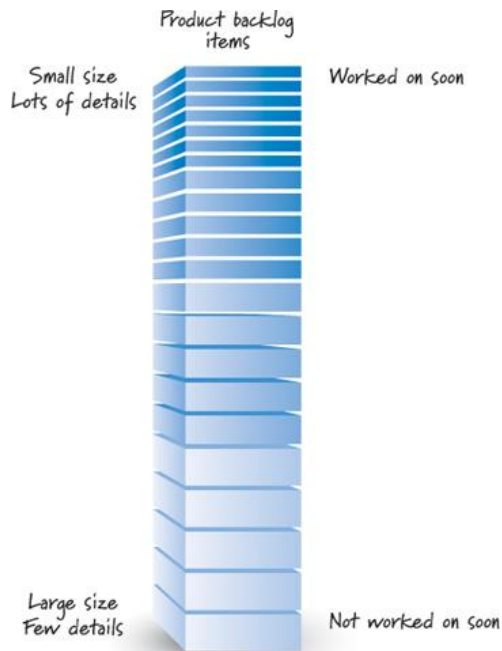
The Agile – Scrum Framework



ARTEFACTS DE SCRUM

- Artefact = travail ou valeur
- Pour
 - Maximiser la transparence des informations
 - Permettre l'inspection
 - Permettre l'adaptation
- 3 artefacts SCRUM
 1. **Product Backlog**
 2. **Sprint Backlog**
 3. **Increment**
- Chaque artefact est associé à un engagement de transparence
 1. Le **Product Backlog** est associé à l'engagement du **Product Goal**
 2. Le **Sprint Backlog** au **Sprint Goal**
 3. L'**Increment** à la **Definition of Done**

Source image Informat



Liste émergente et ordonnée

- Ensemble de tous les éléments, exigences, améliorations, corrections visant à **construire ou améliorer un produit**
- *Product Backlog* contient
 - Besoins fonctionnels
 - Besoins non-fonctionnels (NFRs)
 - Bugs
 - Etc.
- Les éléments constituant le *Product Backlog* peuvent être déclinés en
 - *Epic* (macro fonctionnalité)
 - *User Story* (fonctionnalité)
 - Tâche (plus petite unité de travail)
 - Etc.

USER STORY (RÉCIT UTILISATEUR)

Source image Justinmind

User Story

As a <role>
I want <goal>
so that <benefit>

Acceptance criteria:
(Conditions of Satisfaction)

...
...

As an Account Manager
I want a sales report of my account
to be sent to my inbox daily
So that I can monitor the sales
progress of my customer portfolio

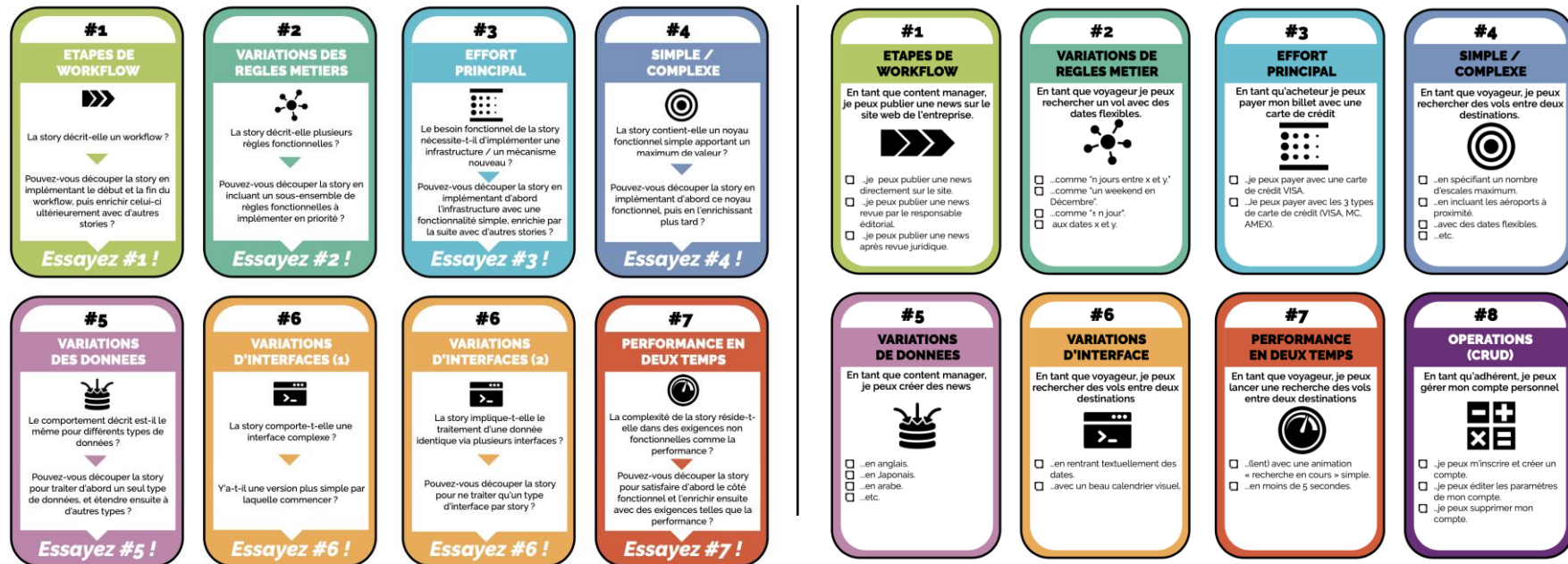
Acceptance criteria:

1. The report is sent daily to my inbox
2. The report contains the following sales details: ...
3. The report is in csv format.

- Les entretiens avec les utilisateurs finaux et le client permettront d'avoir une vision plus précise des besoins
- Découpage le plus petit possible, réalisable en **1 Sprint**
- Un User story permet de découper le besoin en répondant à 4 questions
 - Qui ?
 - Les utilisateurs
 - Formulation : En tant que ...
 - Quoi ?
 - Besoin profond
 - Formulation : Je veux, je souhaite, etc...
 - Pourquoi ?
 - La valeur
 - Formulation : Afin de, dans le but de ...
 - Quels critères d'acceptation (satisfaction)
 - Liste d'éléments pour vérifier que le besoin est satisfait

DIFFÉRENTS TYPES DE DÉCOUPAGE

Source image Myagilepartner.fr



EXERCICE





EXERCICE 3 : CONSTRUCTION DU PRODUCT BACKLOG

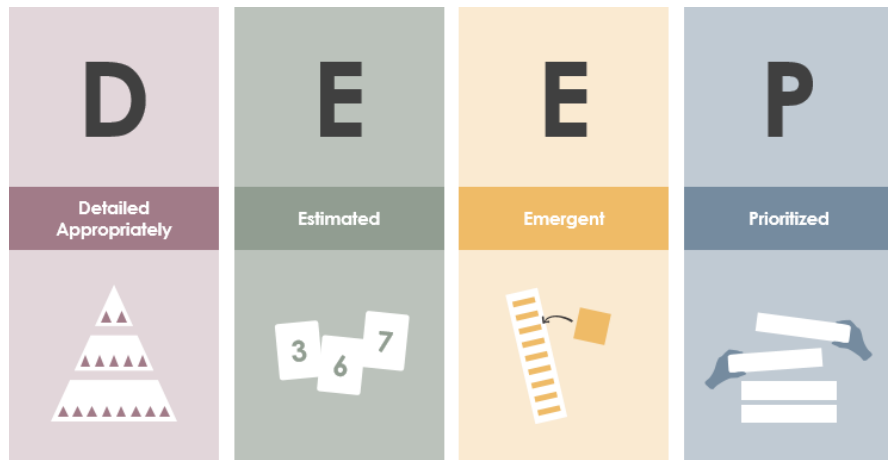
1-exercices/ex3-backlog.md

PRODUCT BACKLOG REFINEMENT

- N'est pas un événement, il s'agit d'un processus d'affinage des Product Backlog Items pour les rendre plus clairs et transparents pour tout le monde
- Lorsqu'un item est suffisamment détaillé, on dit qu'il est « Ready », c'est-à-dire prêt à être sélectionné dans le Sprint Backlog et être réalisé en un Sprint.

ESTIMATION DES PRODUCT BACKLOG ITEMS

Source image Cybermedian



Les éléments du Product Backlog doivent idéalement être

- Suffisamment détaillé et transparent
- Émergent
- Priorisé
 - Classé généralement dans cet ordre
 - Ceux qui ont le plus de valeurs et/ou les plus affinés sont en haut de la pile
 - Ceux qui ont les moins de valeurs et non détaillé en bas de la pile
- Estimé uniquement par les Developers
- Techniques d'estimation
 - Extreme quotation
 - Planning Poker

ESTIMATION AVEC LE PLANNING POKER

Source image Zentao

Agile Estimation Technique

Planning Poker



- Estimer la complexité d'un *US*
 - Consiste à mesurer
 - *La difficulté*
 - *L'incertitude*
 - *Risque*
 - *Périmètre*
 - *Effort*
- **Il ne s'agit pas de déterminer le temps de réalisation de l'*US***
- L'estimation doit être faite uniquement par les Developers
 - PO et SM peuvent estimer uniquement s'ils participent activement à la réalisation d'un élément du Sprint Backlog
- Pour les valeurs, utilisation la suite de *Fibonacci*
 - 1, 2, 3, 5, 8, 13, 21 etc.
- *Storypoint* = carte de complexité attribuée à un élément du Product Backlog

EXERCICE





EXERCICE 4 : PLANNING POKER

`1-exercices/ex4-planning-poker.md`

SPRINT PLANNING

- Événement d'ouverture du Sprint
- L'équipe peut inviter des personnes extérieures pour avoir des conseils et leurs expertises
- Le Sprint Planning doit répondre à 3 questions :
 1. Pourquoi ce Sprint est-il important ?
 1. Le Product Owner expose la valeur qu'elle apporte au produit
 2. L'équipe finalise l'objectif du Sprint avant la fin du meeting (Sprint Goal)
 2. Que peut-on faire durant ce Sprint ?
 1. Les Developers sélectionnent les éléments du Product Backlog sur lesquelles ils ont travailler pour atteindre l'objectif
 2. Ils affinent et s'engagent à les réaliser durant le Sprint. Pour cela, ils s'appuient sur les performances passées, leur projection sur les capacités à venir et la Definition of Done.
 3. Comment le travail choisi sera -t-il réalisé ?
 1. Pour chaque élément du Product Backlog sélectionné, l'équipe des Developers définit un plan d'action qui répond à la Definition of Done
 2. Les Developers sont les seuls à décider de la manière de travail sur un Sprint.
- ***Sprint Planning = Sprint Goal + Product Backlog items sélectionnés + Plan de livraison***

Source image Edinburgh Agile



- Réponds aux questions
 - Pourquoi ?
 - Quoi ?
 - Comment ?
- Est mis à jour tout long du Sprint au fur à mesure que les *Developers* apprennent et implémentent les éléments
- Le périmètre du *Sprint Backlog* peut varier mais elle ne doit pas affecter le *Sprint Goal* ou réduire les exigences de qualité.

INCREMENT ET DEFINITION OF DONE

Increment

- Bout de produit opérationnel (qui est utilisable) qui tend vers le Product Goal.
- Chaque nouvel Increment s'ajoute à la somme de tous les increments précédents
- Un sprint donne lieu à la création d'un ou plusieurs Increments
- A la fin du Sprint, lorsqu'un Increment n'est pas opérationnel, il ne doit pas être présenté lors du Sprint Review, il est réestimé et replacé dans le Product Backlog.
- Un Increment peut être délivré dès qu'il satisfait à la Definition of Done.
- On ne doit pas attendre le Sprint Review pour le délivrer un increment terminé

Definition of Done (DoD)

- Apporter de la transparence sur la définition de travail fini ou terminé
- Généralement des standards de l'organisation (SO) que la Scrum Team doit suivre
 - S'il n'y a pas de SO, la Scrum Team doit créer sa propre DoD
 - S'il y a plusieurs équipes sur le même produit, la DoD doit être commune
- Un increment doit satisfaire au DoD pour être présenté au Sprint Review et être potentiellement livrable en production
- Il n'y a pas l'obligation de livrer les Increments en production à chaque fin de Sprint
- La stratégie de livraison est déterminée par l'équipe SCRUM.

ÉVOLUTION DU DOD

Initial DoD	Mature DoD	Stringent DoD
<p>All acceptance criteria met</p> <p>Unit test coverage >85%</p> <p>Functional test passed</p> <p>No known defects</p> <p>Peer code review</p> <p>Peer code review passed</p> <p>Documentation completed</p>	<p>All business functionality and acceptance criteria met</p> <p>No Dependence unattended</p> <p>Unit test coverage >85%</p> <p>No errors on coding standards</p> <p>Technical Debts <5 days (subjective)</p> <p>Maintainability index for the module/Code >90</p> <p>Functional test automation >75%</p> <p>Functional test passed</p> <p>No known defects</p> <p>Peer code review passed</p> <p>Peer code review passed</p> <p>Documentation completed</p>	<p>All business functionality and acceptance criteria met</p> <p>No Dependence unattended</p> <p>No build failures</p> <p>Integration testing passed</p> <p>Unit test coverage >85%</p> <p>No errors in coding standards</p> <p>Technical Debts < 5 days (subjective)</p> <p>Maintainability index for the modules/Code>90</p> <p>Functional test automation 75%</p> <p>Functional test passed</p> <p>Regression test passed</p> <p>PEN test passed</p> <p>Load test passed</p> <p>Performance test passed</p> <p>No known defects</p> <p>Peer code review passed</p> <p>Documentation completed</p> <p>Compliance Documents updated</p> <p>Regulatory requirements complete</p> <p>UAT approved</p>

EXERCICE



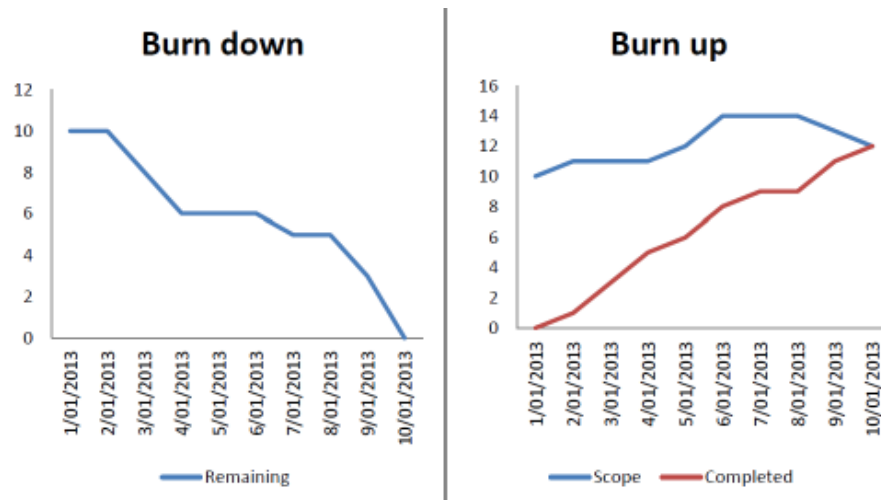


EXERCICE 5 : SPRINT PLANNING

1-exercices/ex5-sprint-planning.md

EVALUER LA PROGRESSION D'UN SPRINT

Source image Clariostechology



Les diagrammes

- Burn Down
 - Indique le travail restant à produire
- Burn Up
 - Indique le travail réalisé
- La prise de décision repose toujours sur l'empirisme, les diagrammes sont justes un moyen d'afficher la progression de l'équipe

SPRINT

- Les sprints plus courts limitent les risques liés aux coûts et à l'effort car le feedback est plus rapide et donc l'adaptation est faite au plus tôt.
- Attention le Sprint 0 n'existe pas dans la théorie de SCRUM, c'est une mauvaise pratique qu'on retrouve dans les entreprises qui consistent à ne pas produire de la valeur au premier sprint mais mettre en place l'architecture, la base de données, l'organisation de l'équipe etc. c'est contre l'esprit de SCRUM qui est de produire de la valeur à chaque Sprint.
- Le Sprint peut être annulé uniquement par le Product Owner si l'objectif du Sprint dévient obsolète.

DAILY SCRUM

- Objectifs
 - Inspecter les travaux en cours vers l'objectif de Sprint et adapter le Sprint Backlog si nécessaire
 - Améliorer la communication entre les Developers
 - Identifier les obstacles et les lever
 - Éliminer la nécessité d'avoir recours à d'autres réunions (réduire ne va pas dire supprimer entièrement cette possibilité)
- Se tient à la même heure et au même endroit pour réduire la complexité
- Participation
 - Obligatoires pour les Developers
 - Le Scrum Master et Product Owner peuvent participer uniquement dans le cas où ils travaillent activement sur un élément du Sprint Backlog. Ils participent avec le rôle de Developers

SPRINT REVIEW

- Session de travail à part entière (ne se limite pas à une présentation du travail réalisé)
- Objectifs
 - Inspecter le travail réalisé durant le Sprint
 - Présenter ce travail aux parties prenantes du projet et la progression vers l'objectif du produit
 - Récolter du feedback pour adapter la direction à prendre pour les prochains Sprints

SPRINT RETROSPECTIVE

- Objectifs
 - Réflexion sur les pistes pour améliorer la qualité et l'efficacité de l'équipe pour les prochains Sprints
 - Inspecter le déroulement du Sprint en cours au niveau
 - Des individus
 - Interactions
 - Processus
 - Outils
 - Definition Of Done (DoD)
 - Identifier les éléments qui ont fait dévier l'équipe de son plan initial et leurs origines
 - Identifier les obstacles, problèmes et comment ils ont été résolus ou non
- Les éléments d'amélioration les plus impactant peuvent être ajoutés au Product BackLog
- L'équipe peut décider durant cet événement de changer le DoD
- Événement qui conclut le Sprint

TIME-BOXING DE TOUS LES ÉVÉNEMENTS SCRUM

- Sprint
 - 1 mois au maximum (entre 1 et 4 semaines)
 - Sprint Planning
 - Pour un Sprint d'un mois, c'est 8 heures maximum
 - Pour un Sprint de moins d'un mois, c'est moins de 8 heures
 - Daily Scrum
 - 15 minutes maximum
- Sprint Review
 - Pour un Sprint d'un mois, c'est 4 heures maximum
 - Pour un Sprint de moins d'un mois, c'est moins de 4 heures
 - Sprint Retrospective
 - Pour un Sprint d'un mois, c'est 3 heures maximum
 - Pour un Sprint de moins d'un mois, c'est moins de 3 heures

EXERCICE





EXERCICES 6 ET 7 : REVIEW ET RETROSPECTIVE

[1-exercices/ex6-sprint-review.md](#)

[1-exercices/ex7-sprint-retrospective.md](#)

ANNEXE



- [Manifeste Agile](#)
- [Principes agiles](#)
- [SCRUM GUIDE 2020 English Version](#)
- [GUIDE SCRUM 2020 en Français](#)

Merci de votre attention et participation
Glodie Tshimini
contact@tshimini.fr



m2iinformation.fr