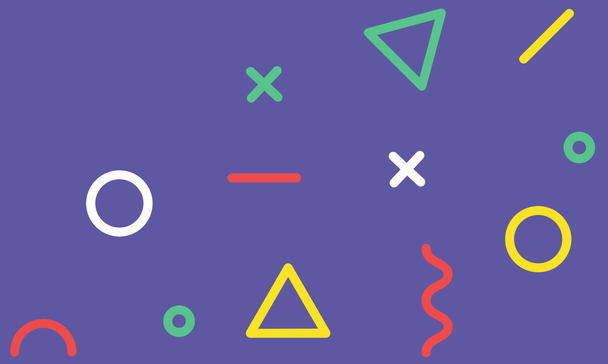


Processus Unifiés et Méthodologies Agile

Hamdi Bouslah
hamdi.bouslah@gmail.com

Exercice

Vous décidez de construire la maison de votre rêve !

Comment vous imaginez votre maison ?

Budget ? Caractéristiques ? Délais ?

Exercice

Vous travaillez dans une agence web en tant que chef de projet, un client vous demande de lui développer une application Web de crowdfunding (https://fr.ulule.com/) en ligne :

Votre client vous demande un Devis (en Dinars) + la date de livraison

Un cahier de charge détaillé est communiqué par le client, les modules à développer sont :

M1 : Création d'un Projet
M2 : Consultation des Projets (Listing + Filtres)
M3 : Contribution à un projet (Paiement en ligne + Signature électronique)
M4 : Commentaires et Avis
M5 : Administration (gestion des utilisateurs , validation des projets)
M6 : Publication des articles
M7 : Authentification

Hypothèse 1 : Technologies Angular / NodeJS , Postresql
Hypothèse 2 : votre équipe est composé de 5 personnes : architecte , 3 développeurs,
1 designer web.

Exercice

	IHM	Complexité	Charge en JH
M1 : Création d'un Projet	1	Moyenne	8
M2 : Consultation des Projets (Listing + Filtres)	1	Simple	6
M3 : Contribution à un projet (Paiement en ligne + Signature électronique)	4	Complexe	48
M4 : Commentaires et Avis	2	Simple	12
M5 : Administration (gestion des utilisateurs , validation des projets, gestion des contributions)	5	Moyenne	40
M6 : Publication des articles	2	Simple	12
M7 : Authentification	2	Moyenne	16
			Total 142

Hypothèse 3 : il faut 6 JH pour développer une interface Simple , 8 JH pour développer une IHM de complexité moyenne et 12 pour développer une IHM Complexe

Exercice

Q1 : Quel Prix vous allez proposer à votre client ?

Q2 : Quelle date de livraison vous allez communiquer à votre Client ?

Q3 : Quand est ce que votre client découvrira l'application pour la première fois ?

Q4 : qu'est ce que vous faites si votre client détecte des bugs ou des anomalies ?

Q5 : Est-ce que l'équipe est complète ? Est-ce qu'il vous manque des profils ?

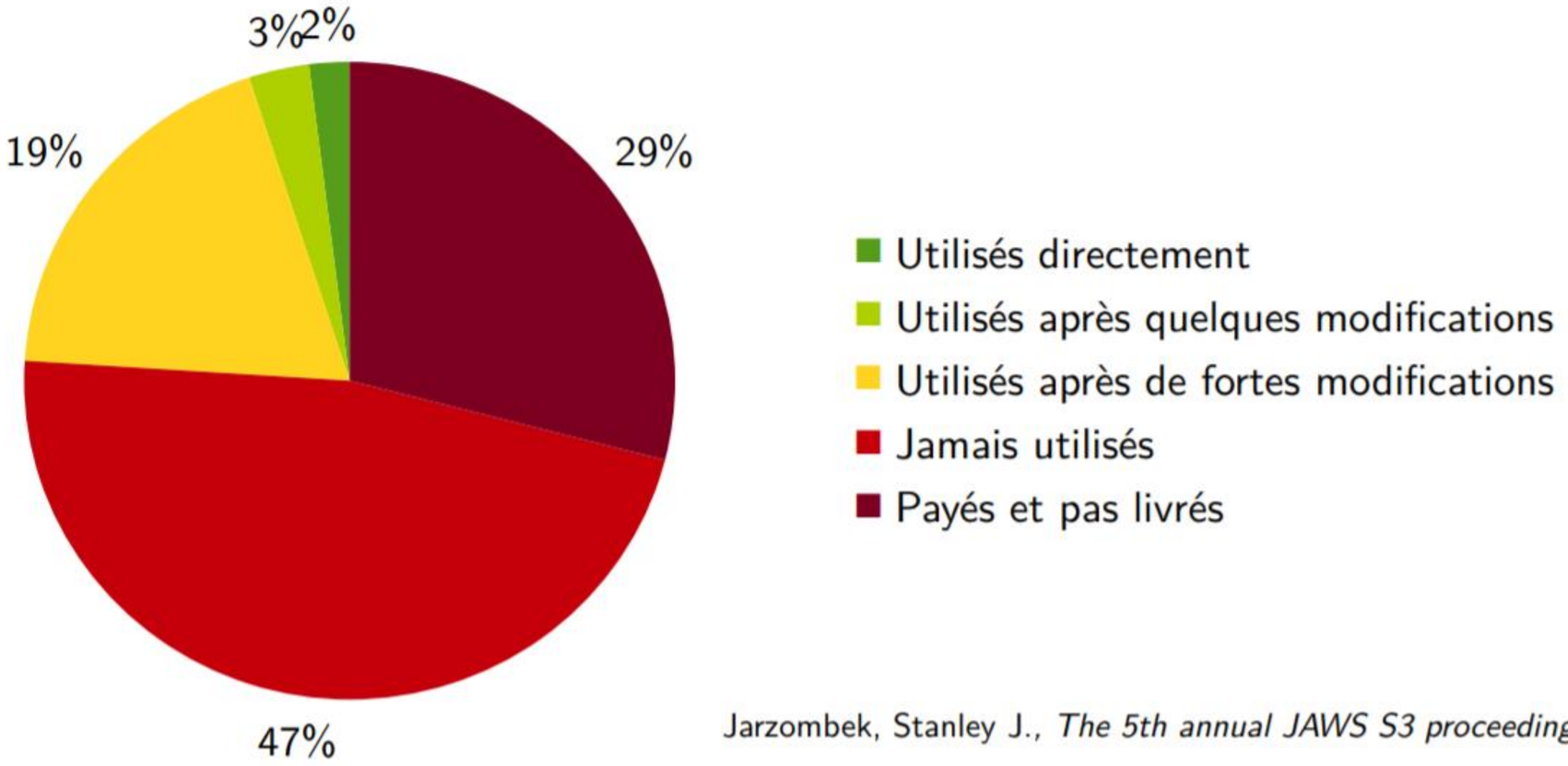
Q6 : Une fois votre solution est en production ? Qui fera le support ?

Exercice

Q7 : Si votre client vous demande des améliorations , comment vous allez procéder ? Qui fera la mise à jour de la solution ? Peut-on automatiser le déploiement ?

Introduction

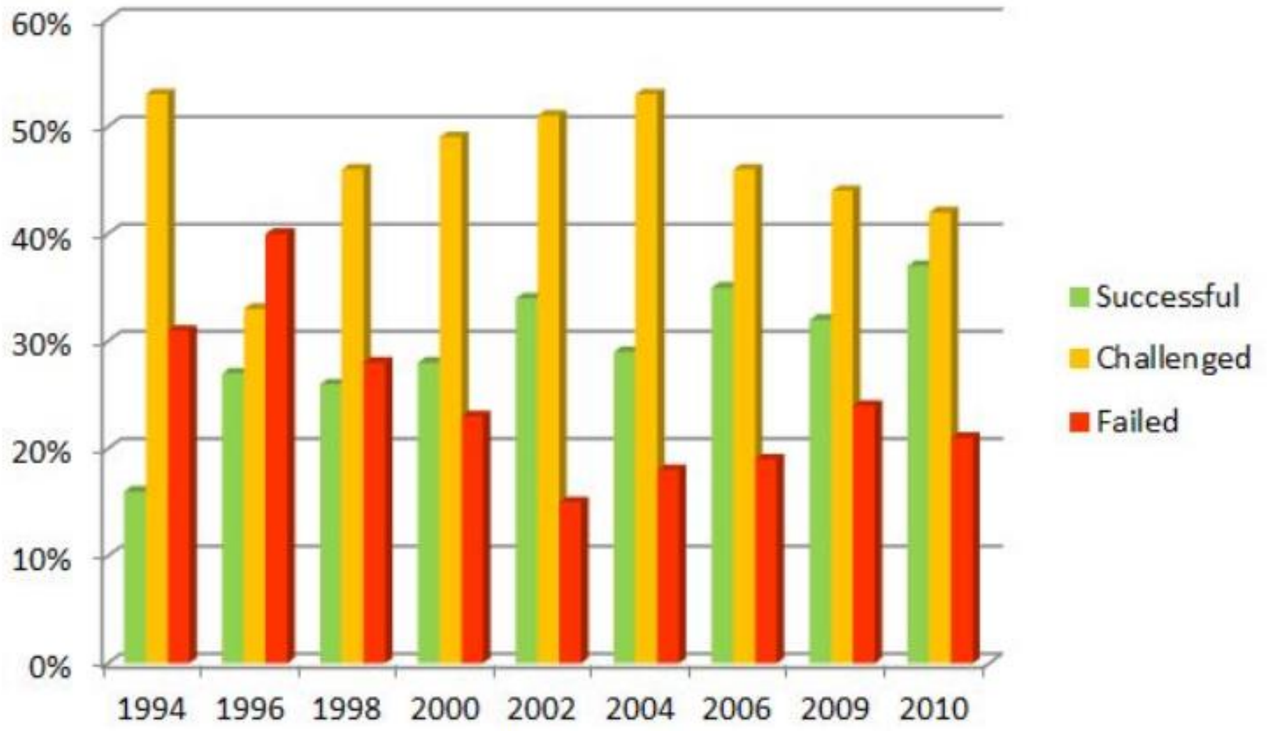
Étude du *Department of Defense* des États-Unis sur les logiciels produits dans le cadre de 9 gros projets militaires



Jarzombek, Stanley J., *The 5th annual JAWS S3 proceedings*, 1999

Introduction

Les projets technologiques continuent d'échouer à un rythme alarmant, et pour un coût sensible ! Il est temps d'apprendre de nos erreurs…



Chaos report - historique
D'après les données du Standish Group

Qu’est ce que le logiciel ?

- IEEE 610.12
  « Des instructions (programmes informatiques), des procédures, et possiblement de la documentation associée et des données appartenant à l’opération d’un système informatique. »

- Le logiciel est omniprésent dans nos sociétés !
  - Le logiciel est de plus en plus *complexe*
  - Le logiciel est de plus en plus *critique*

- L’industrie du logiciel est un des moteurs de la nouvelle économie, L’IT et IA bouleversement notre vie dans les années qui viennent

Spécificités du logiciel

- Un produit **immatériel**, dont l'existence est indépendante du support physique
  - Semblable à une œuvre d'art (roman, partition…)

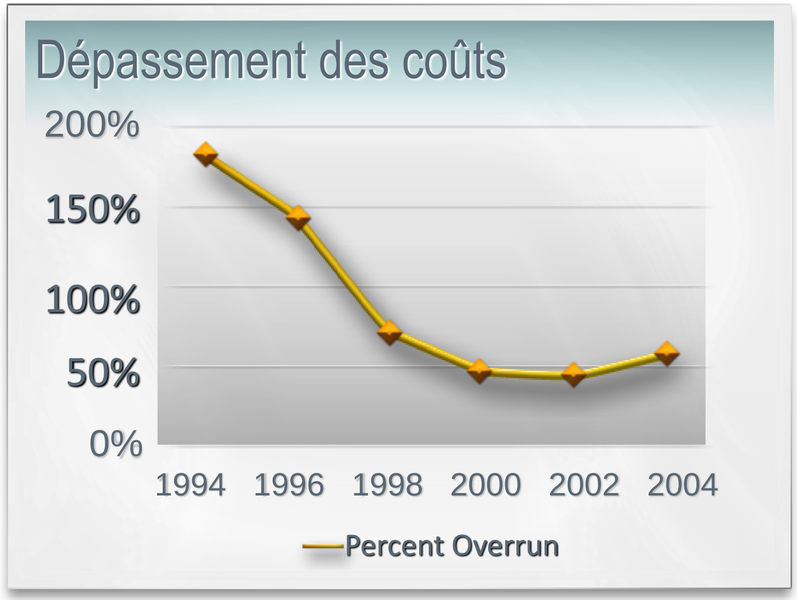
- Un objet technique fortement contraint
  - Fonctionne ou ne fonctionne pas
  - Structure complexe
  - Relève des modes de travail du domaine technique

- Un cycle de production différent
  - La reproduction pose peu de problèmes, seule la première copie d'un logiciel a un coût
  - Production à l'unité

Génie logiciel

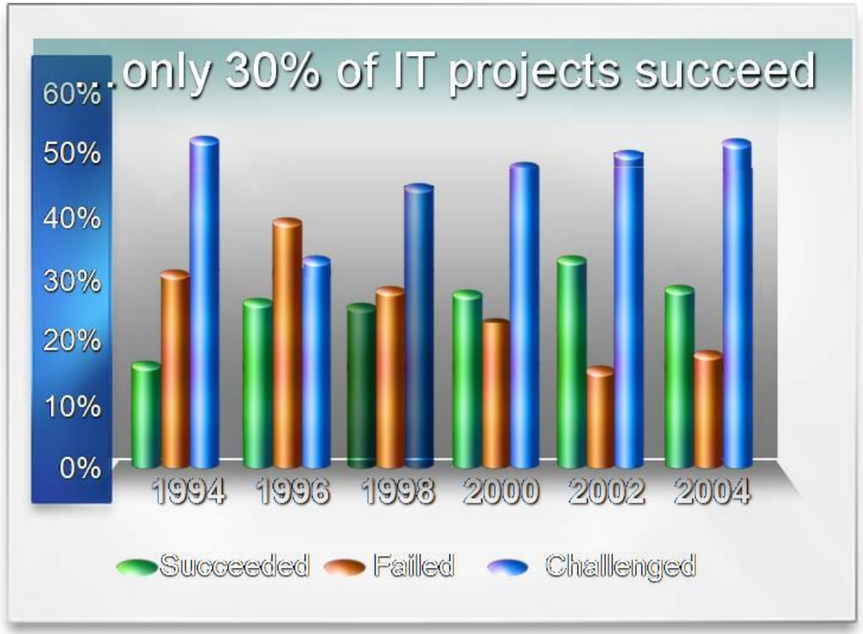
- Nouvelle discipline née en 1968 à Garmish (Allemagne) suite à la fameuse crise du logiciel des années 60 :
    - Durée allongée des projets
    - Complexité croissante des systèmes d'information (taille, performance, distribution, hétérogénéité)
    - Augmentation du volume des données à traiter
- Le génie logiciel veut ramener des solutions aux problématiques suivantes :
    - La fiabilité du logiciel (**Qualité**)
    - Maîtrise des délais de développement prévus (**Productivité**)
    - Rendre le métier de développement un métier économiquement profitable (**Industrialisation**)

Mauvaise nouvelle : crise du logiciel continue !



Source: Standish Group, 2004 Third Quarter Research Report, CHAOS Research Results

30 % seulement des projets de développement réussissent !



Qualité du produit

- Qualité attendue par les utilisateurs :
  - Trop d'acteurs = > Nécessité d'un compromis
  - Acteurs :
    - La MOA,
    - La MOE,
    - Les Utilisateurs.

  - **Les tâches:**
    - Recenser et classer les objectifs, les exigences, les contraintes,
    - Rapprocher les éléments recensés des facteurs qualité standard,
    - Hiérarchiser les facteurs en fonctions des priorités de la charte de projet,
    - En déduire les critères de qualité,
    - Identifier les métriques.

A partir de 1914, la Ford T est entièrement conçue par la chaîne d'assemblage,
temps de montage du châssis 12h30 → 1h33.

Prix 850$ en 1908,

La Ford T en chiffres :
Poids : de 550 kg à 750 kg selon la finition.
Vitesse max : 70 km/h
Prix : 850$ en 1908, 290$ en 1927
(pour info, 850$ à l'époque équivalent à 15 000€ aujourd'hui).
Consommation : 18 l/100 km soit plus de 250 kilomètres d'autonomie.

Les sources d’inspiration des méthodes « agiles »

Les années 1980 et 1990 témoignent de la transformation des systèmes d’organisation largement inspirés du taylorisme et du fordisme. L’expansion des principes de gestion issus du modèle japonais, connu aujourd’hui sous le système de production Toyota (TPS), semble être à l’origine d’un tel changement.

L’autonomation regroupe un ensemble d’outils l’andon qui consiste à arrêter automatiquement la production en cas de problème méthodes (pareto des causes) dont le but est d’inciter l’ensemble des employés d’une entreprise à améliorer la qualité des produits et des services vendus plutôt que les éliminer.

Le principe de juste-à-temps, quant à lui, consiste à organiser son entreprise de telle sorte qu’elle puisse livrer exactement et au bon moment la quantité de biens souhaités par ses clients. La production est tirée par la demande.

Les sources d’inspiration des méthodes « agiles » (2)

Après le succès spectaculaire du Système de Production Toyota traduit sous le terme de *lean management*, de nombreux constructeurs de véhicules ont tenté de reproduire, au sein de leur organisation, l’ensemble des principes et pratiques relevant de cette approche managériale dans le but d’améliorer la performance et la productivité de leur système de production.

La réussite commerciale affichée par les industries *lean* et leur forte médiatisation n’ont fait qu’élargir l’expansion de ce système à différents secteurs, dépassant ainsi son domaine d’application initial, celui de la production industrielle

Agile manufacturing

L'« *agile manufacturing* » a été créé dans les années 1990 par des chercheurs américains, de Lehigh University, chargés d'améliorer la performance des industries américaines. Les défaillances du système de production de masse et les menaces des pays industrialisés d'Asie ont poussé les firmes américaines à changer leur mode de production et à emprunter la voie de l' « agilité ».

L'agile manufacturing a tout d'abord été défini comme une entreprise robuste, adaptative avec une capacité de reconfiguration rapide pour répondre aux opportunités du marché. Une telle entreprise est fondée sur des processus et des structures appropriés ainsi que sur un système coordonné de personnes, d'organisation et de technologies permettant d'acquérir une performance compétitive supérieure à celle des pratiques existantes.

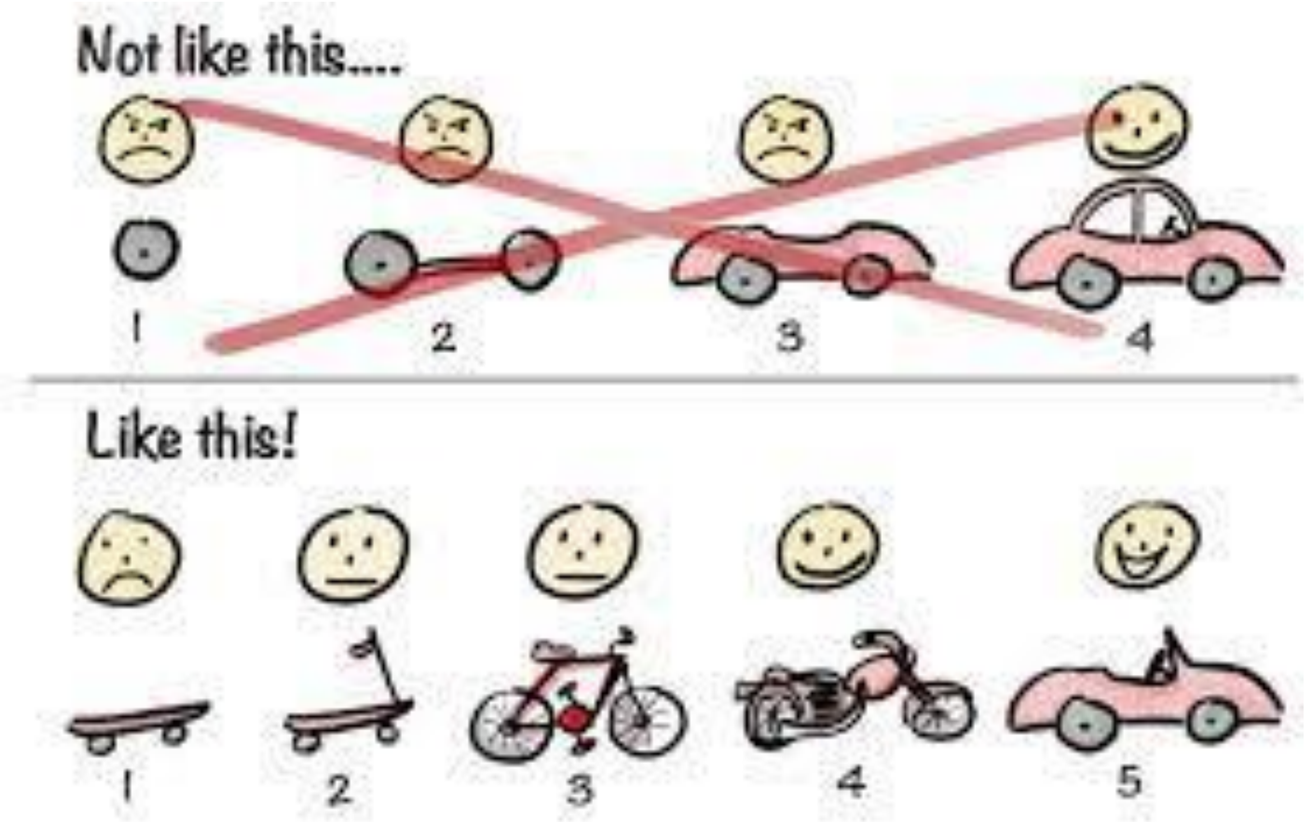
Agile manufacturing Vs Méthode Agile

Concepts	Agile manufacturing	Méthode « agiles »
Réponse rapide aux changements	Capacité de l'entreprise à transformer rapidement les informations collectées en décisions actionnables (Gunasekaran, Yusuf, 2002).	Capacité de s'adapter aux changements à travers de courts cycles de développement.
Cycles de développement rapides	Développement rapide basé sur de courts cycles *concept-to-cash* (Goldman, Nagel & Preiss, 2005 dans Kettunen, 2009).	Cycles courts ne dépassant pas les quelques semaines. Les itérations sont fixées dans le temps.
Reconfiguration	Capacité d'une entreprise à reconfigurer sa structure organisationnelle et ses processus et à réorienter sa production ainsi que ses ressources afin de mieux répondre aux changements (Gunasekaran, Yusuf, 2002).	Reconfiguration au niveau du produit et du projet (restructuration des codes, modification du *design*).
Pro-activité	Saisir les nouvelles opportunités ou provoquer des « ruptures » par le biais de l'innovation (Yusuf, Sarhadi & Gunasekaran, 1999)	Adaptation aux besoins émergents des clients et réduction des activités d'anticipation.
Qualité	La qualité renvoie à la valeur ajoutée perçue par le client. Il n'existe pas de compromis précis sur la qualité (Liker, 2004 dans Kettunen, 2009).	La qualité est considérée comme une norme. Recours à des pratiques d'ingénierie pour améliorer la qualité (tests unitaires, intégration continue).

Agile manufacturing Vs Méthode Agile

Stratégie des processus industriels	Les capacités industrielles sont déterminées en fonction des objectifs stratégiques. Les industries agiles sont du type make-to-order [27] (Narasimhan, Swink & Kim, 2006)	Les méthodes de développement agiles sont de nature *engineering to order* [28] : les fonctionnalités sont redéfinies et stabilisées durant la conception du produit.
Leadership	Le management scientifique est remplacé par le principe d'*empowerment*. Les personnes disposent d'une autonomie, accordée par leur hiérarchie, pour organiser leur travail. (Gunasekaran, Yusuf, 2002).	Renforcement et auto-organisation des équipes.
Entreprise guidée par les connaissances	Les connaissances collectives constituent un avantage compétitif pour l'organisation (Yusuf, Sarhadi & Gunasekaran, 1999).	La circulation libre de l'information et les connaissances tacites sont valorisées.
Prises de décisions	L'autorité est distribuée, les prises de risque et le partage de connaissances sont encouragés. (Shafer, 1997 dans Kettunen, 2009).	La responsabilité est partagée entre les membres d'une équipe de développement agile. la collaboration et la communication fréquente sont encouragées.

Démarche incrémentale



« The basic idea behind iterative enhancement is to develop a software system incrementally, allowing the developer to take advantage of what was being learned during the development of earlier, incremental, deliverable versions of the system. Learning comes from both the development and use of the system, where possible » (Basili & Turner, 1975).

Les premières publications sur les méthodes « agiles » apparaissent en 1991 avec le développement de la méthode RAD[34] par James Martin.

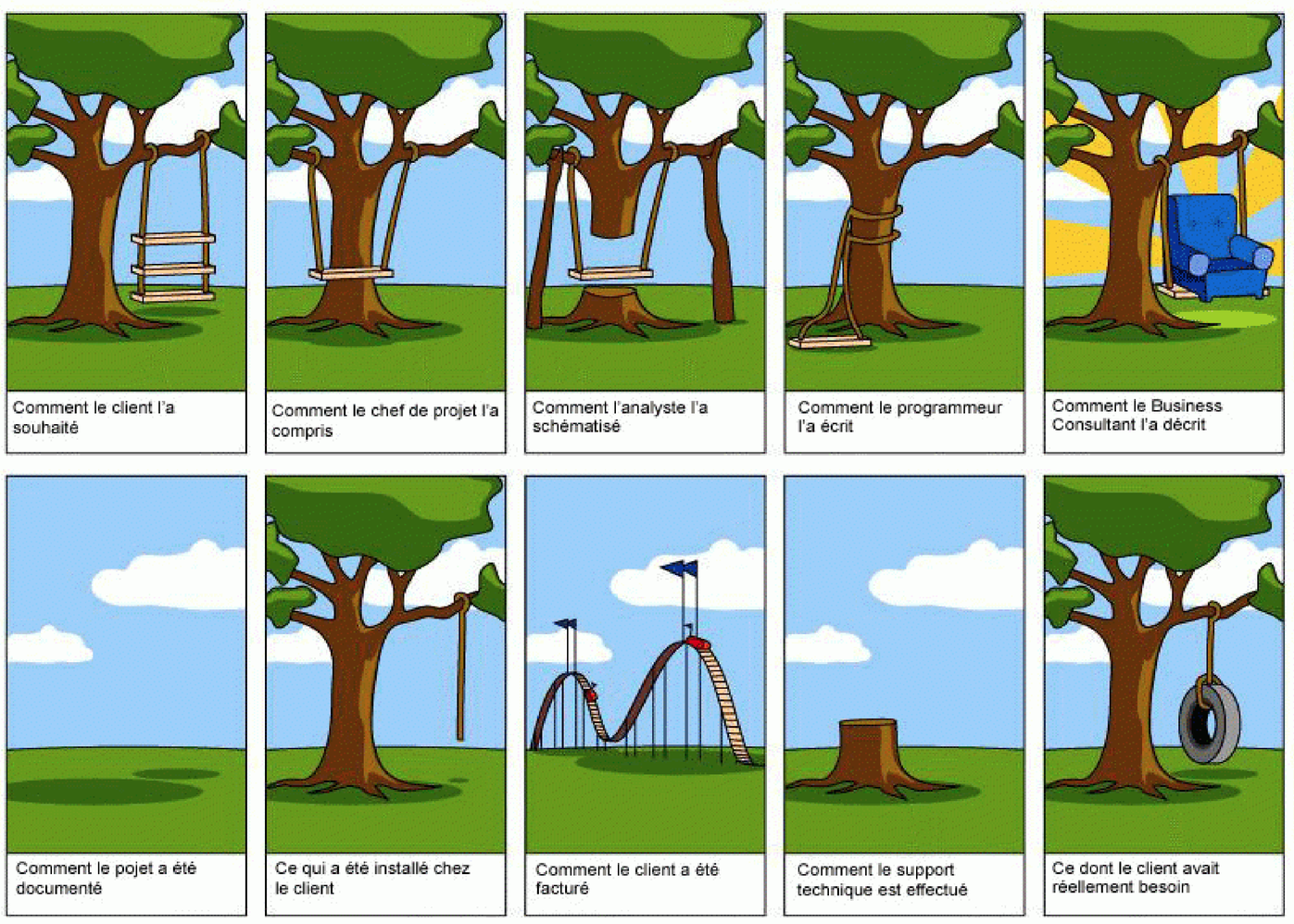
le développement conduit par les tests

Si l'ensemble des méthodes « agiles » s'appuient sur des principes et des valeurs communes, elles se démarquent, toutefois, par rapport aux pratiques et instrumentations de gestion qu'elles mobilisent.

- extreme programming
- Scrum
- lean développement

Scrum c’est une Philosophie

Une autre approche de gestion de projet



Méthodologies et Processus de Fabrication

Sans méthodologie

Bénéfices :

- Marche si vous travaillez seul
- Pas de surcoûts dûs à la gestion de projet

Inconvénients :

- Rapidement compliqué
- Pas de visibilité

Méthodologies et Processus de Fabrication

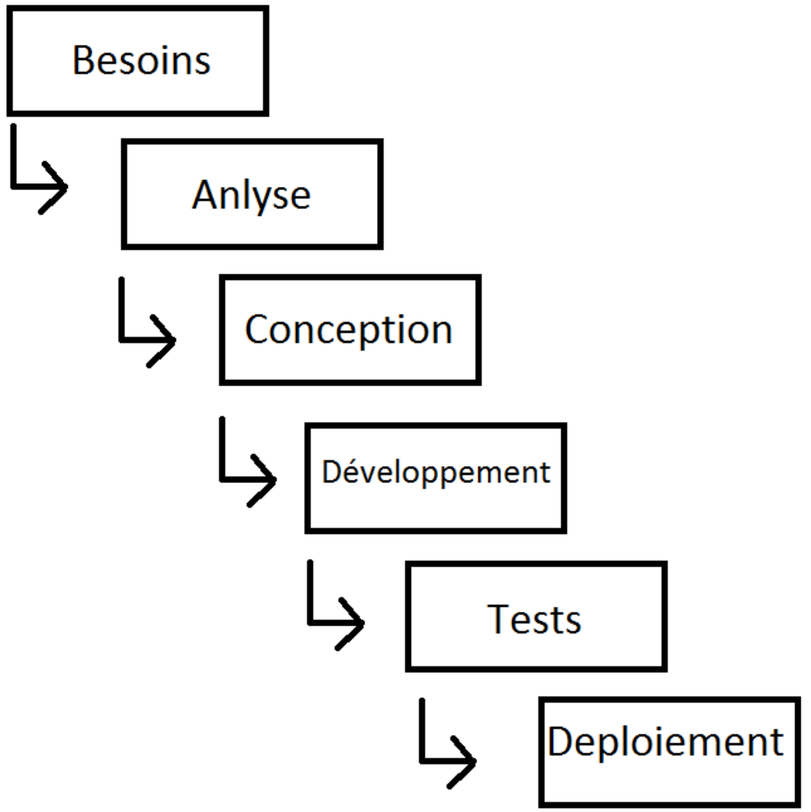
Cascade

Bénéfices :

- tâches établies au début du projet
- segmentation claire des responsabilités
- passation de tâches aisées

Inconvénients :

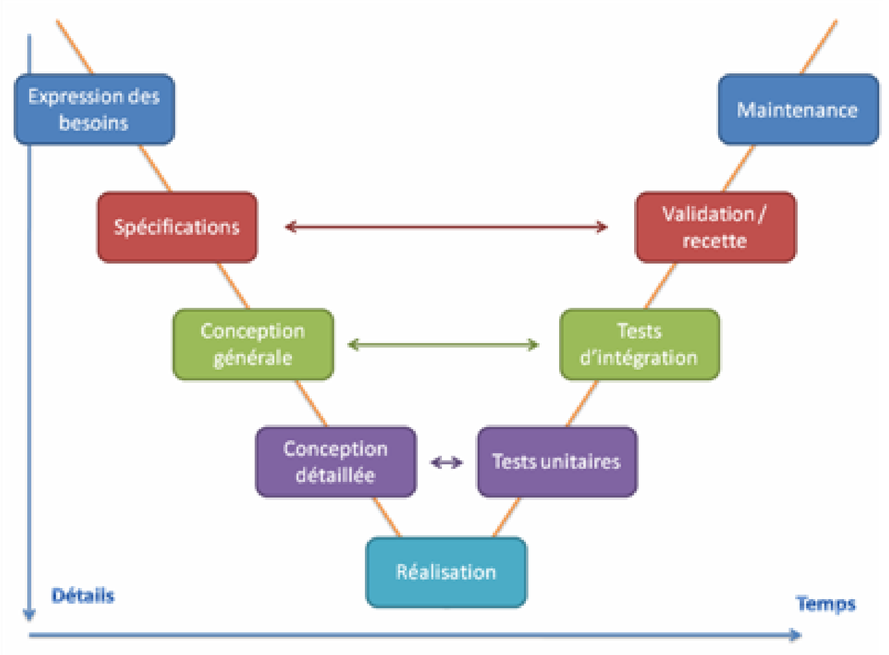
- planification trop rigide alors que les tâches sont change
- la responsabilité du succès du projet n'est pas partagée
- *si le projet échoue : "c'est pas de ma faute, j'ai bien fait r*
- si une tâche n'est pas bien effectuée tout l'édifice est éb
- trop de choses sont faites qui ne sont pas directement li



Méthodologies et Processus de Fabrication

Méthodes Classiques (cycle en V)

- Approche séquentielle
- Faible tolérance à l'erreur
- Mauvaise estimation es charges
- Manque de rigueur dans le pilotage et le suivi du projet.
- Sanction différée



Méthodologies et Processus de Fabrication

Rational unified Process

- itératif et incrémental
- Dirigé par les Cas d'Utilisations
- Centré sur l'architecture
- Focus sur le risque

Historique

- Années 90
  - réaction contre les grosses méthodes
  - prise en compte de facteurs liés au développement logiciel
- Fin années 90: développement de méthodes
  - d'abord des pratiques liées à des consultants, puis des livres
  - XP, Scrum, FDD, Crystal…
- 2001
  - les principaux méthodologues s'accordent sur le « Agile manifesto »
- Depuis
  - projets Agile mixent des éléments des principales méthodes

Quelques Idées reçues !

- Sur un projet agile, il n'y a pas de spécifications, de plan, de processus, d'outils et même pas de contrat".

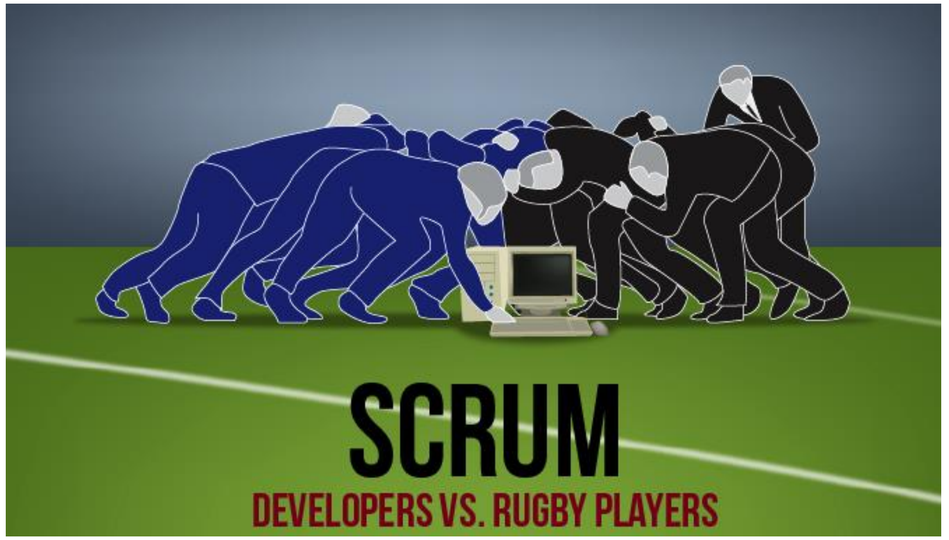
- Un projet Agile ne fonctionne que sur des projets de développement web, qu'avec une dizaine de personnes maximum, qu'avec des super développeurs" ou encore "sur un projet agile, le client change d'avis tout le temps et on tourne en rond à faire et défaire des choses".

- Je fais un support de production, les méthodologies ne sont pas adaptées à mes besoins.

Avantages des méthodologies Agiles

Scrum: Pourquoi ?

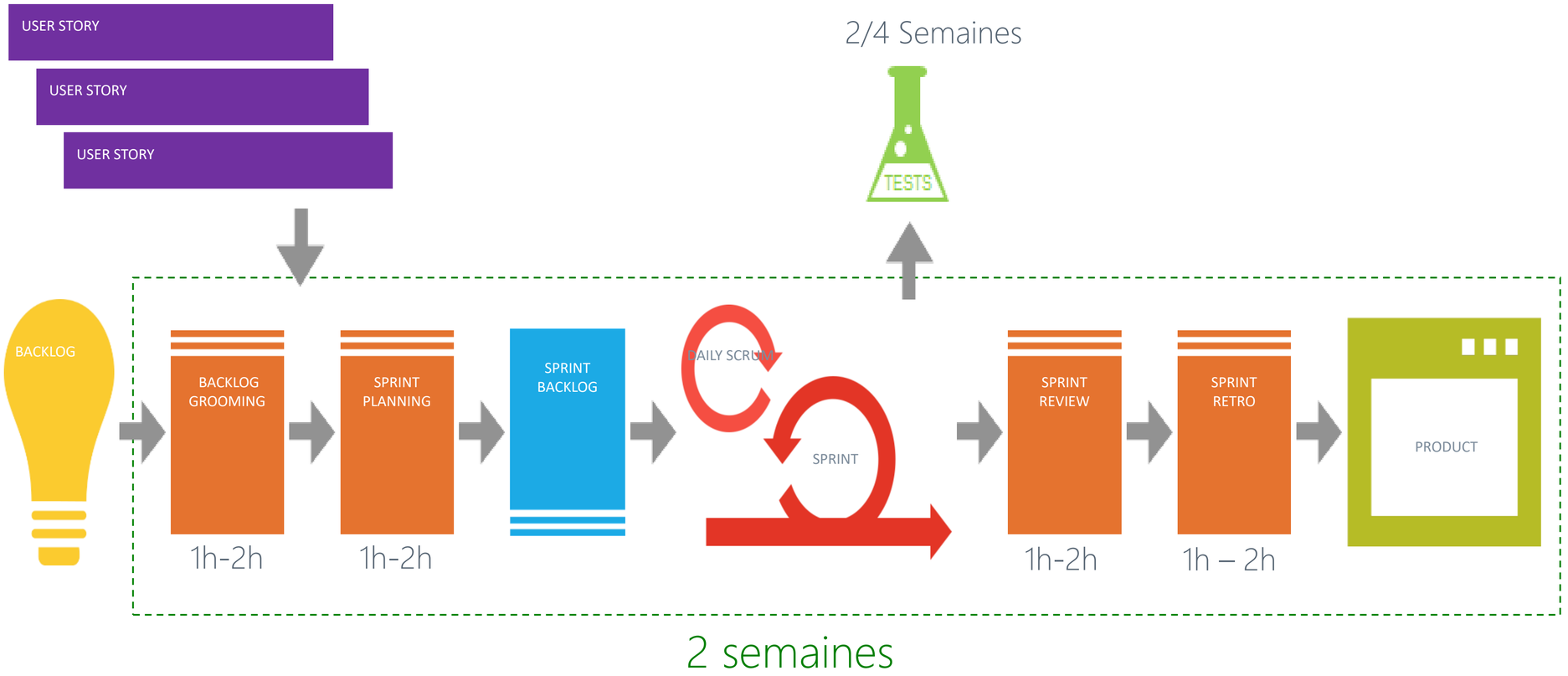
- Le développement Agile n’est pas une question de cycle, c’est une philosophie, une culture.
- Basé sur la réactivité et l’implication du client.
- Responsabilité de l’équipe
- Transparence.
- Inspection (Indicateurs).
- Adaptation (améliorer, réorienter)



Cycle de Production

La vie d'un projet Scrum est rythmée par des Itérations

Cycle de vie d’un projet Scrum



Réunions Scrum (1)

Réunion de planification de sprint

- Fréquence : Début Sprint
- Acteurs : scrum master + équipe + PO
- Partir d’un Product Backlog est complet et ordonnancé
- le PO revoit avec l’équipe la vision du produit et le plan de livraison
- Vérifier les estimations et confirmer qu’elles sont exactes
- Inventaire des tâches
- Découpage en des sous-tâches

Réunions Scrum (2)

Stand-up meeting (10-15 minutes)

- Fréquence : Tous les matins
- Acteurs Scrum master + équipe
- se synchroniser, remonter les obstacles rencontrés, s’entraider, vérifier l’avancement du sprint.
- l’esprit d’équipe.
- Chaque personne répond à 3 questions :
  - Qu'ai-je fait hier pour atteindre l'objectif Sprint ?
  - Que vais-je faire aujourd'hui pour aider l'équipe de développement à atteindre l'objectif Sprint ?
  - Est ce que je vois des obstacles ?

Réunions Scrum (3)

Revue de Sprint

- Fréquence : A la fin de chaque sprint.
- Acteurs : scrum master + équipe + PO + Utilisateurs + Clients
- Présenter les nouvelles fonctionnalités développées au cours du sprint
- Le PO accepte ou refuse les fonctionnalités présentées.
- Calcul de la vélocité

Rétrospective de sprint

- Fréquence : A la fin de chaque sprint.
- Acteurs : Scrum Master + équipe
- Identifier et pondérer les éléments positifs (éléments à cultiver ou source de motivation)
- Identifier les éléments à améliorer
- un plan d’action d’amélioration

Acteurs (1)

Qui fait quoi ?

- **Product owner** (PO) : représentant des clients et des utilisateurs dans le cadre du projet. Son rôle consiste à
- https://www.youtube.com/watch?v=3eljozEWpf8

  - Alimenter le backlog produit.
  - Gérer les priorités
  - Vérifier la conformité de l'application par rapport aux exigences
  - Business Value
  - Partager les connaissances et bien expliquer les USs

- **Scrum Master :** coach et animateur de l'équipe technique
- https://www.youtube.com/watch?v=oheekef7oJk

  - S'assurer de l'application du cadre méthodologique Scrum
  - Aider l'équipe à franchir les différents obstacles
  - S'assurer de la bonne compréhension entre le PO et l'équipe
  - Animer le Stand up meeting.

Acteurs (2)

Qui fait quoi ?

- **Equipe Scrum** : transformer les besoins exprimés par le Product Owner en fonctionnalités utilisables.
  - Développeurs
  - Testeurs
  - Architecte Technique
  - Référents applicatif métiers
  - Systèmes : ISR

Les artéfacts

- Épique (Epic) : Une Epic est une User Story qui est trop volumineuse pour être complétée en un seul Sprint. Dans ces circonstances, on doit la subdiviser en multiples Stories.

- (Product Backlog) : Un Product Backlog est la liste de toutes les User Stories que notre produit idéal pourrait contenir et ce en ordre de priorité. Ces Stories ne sont pas encore subdivisées en tâches et leur description est sommaire sauf pour les plus prioritaires qui feront partie du prochain Sprint .

- Sprint Backlog : Le Sprint Backlog consiste en la liste des User Stories qui composent le Sprint

Définir ses stories (1)

Quelques règles générales

- Tout est une story! (technique, documentation, …) = Avant tout, on cherche une répartition optimale de la charge de travail!
- Indépendant de toute technologie
- Critères I (Independant) N (Negotiable) V (Valuable) E (Estimable) S (Small) T (Testable)
- Obligation de critères d'acceptations = contrat entre le PO et l'équipe = Périmètre fonctionnel = Plan de tests
- Toutes les stories sont soumises à la DoD (« Definition of Done ») et la DoR (« Definition of Ready »)
- Une story est différente d'un cas d'utilisation

Définir ses stories (2)

Quelques règles générales

Quel Format?

En points et non en heures ou jours

Ordre de complexité relative entre les stories

Comment?

Planning poker entre les membres de l'équipe de développement

Quand?

Au démarrage de projet pour l'estimation initiale globale

À chaque sprint planning

Qu'estime t-on?

Compliqué VS **Complexe**

Durée de réalisation

La prise en compte des contraintes

Quelques Règles

Le pointage 0 n'existe pas

L'échelle de pointage est la même pour toutes les équipes

Pointage trop haut = subdivision systématique

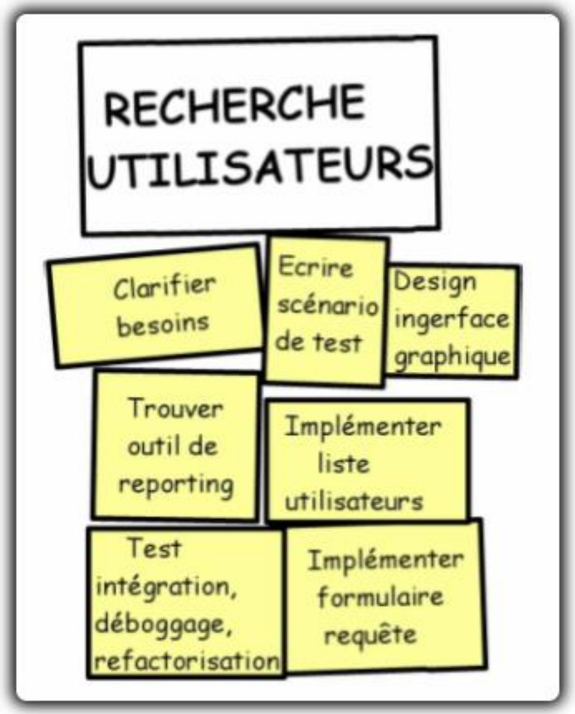
L'estimation d'une story ne peut dépasser la vélocité de l'équipe…

Une story ne peut être implémentée dans deux sprints différents.

Définir ses stories (3)

- Différence entre US et tâche ?

- La distinction est assez simple. Les stories sont des choses livrables qui intéressent le PO/ les clients . Les tâches sont des choses non livrables, ou des choses auxquelles le PO ne s'intéresse pas.



L’estimation: passer de l’abstrait à la réalité

- Les variables fondamentales de calcul des coûts pour un projet agile
  - La vélocité
  - La durée d’un sprint
  - Le taux horaire des différents intervenants
- Formule (très) théorique

Nombre de sprints = Nombre de points / Vélocité estimée par sprint
Effort de développement (heures ou jours) = (Nombre de sprints * Durée d'un sprint)
Coût de développement = Taux horaire * Durée de développement * Nombre d’intervenants

- Toujours un sprint supplémentaire pour la correction et l’ajustement du dernier sprint de développement.

Pratique (1)

- Le projet consiste à créer une ville et construire des bâtiments .
- **Estimation** : la suite de Fibonacci (1, 2,3,5,8, 13 20)

- **1ère Etape** : Choisir une US de référence
- **2ème Etape :** Estimation des story points

- **Bâtiments à construire :**
- Pont
- Steg
- Hôpital
- Crèche
- Ecole Primaire
- Municipalité
- Mosquée
- 2 immeubles
- 7 maisons

Faire un Projet Agile un succès (1) !

- Nécessite une volonté à tous les niveaux
- Nécessite une discipline d'abstraction des possibilités de l'outil en lui-même pour se concentrer sur les besoins réels.
- Nécessite de savoir faire des compromis
- Nécessite un setup et des processus de développement (très) bien rodés
- Impose un modèle de contractualisation adapté
- La plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.
- Accueillir positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.

Faire un Projet Agile un succès (2) !

- Livrer fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.
- Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.
- Réaliser les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.
- Plus de communication et dialogue
- Un logiciel opérationnel / les livrables sont la principale mesure d'avancement.

Faire un Projet Agile un succès (3) !

- Les processus agiles encouragent un rythme de développement soutenable. Ensemble, les PMs, les développeurs, les testeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.

- Une attention continue à l'excellence technique et à une bonne conception renforce l'agilité.

- La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.

- Les meilleures architectures, spécifications et conceptions émergent d'équipes autoorganisées.

- À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

Quiz

Question 1
What kind of software development projects can be executed by Scrum Project Management Framework?

- **Choice-1:** Complete software packages
- **Choice-2:** Customer projects
- **Choice-3:** Sub-systems, components or parts of bigger systems
- **Choice-4:** All kinds of software development projects
- **Choice-5:** None of the given answers
- **Correct Answer:** Choice-4 / All kinds of software development projects

Question 2
Where are the customer requirements stored?

- **Choice-1:** In the Product Backlog
- **Choice-2:** In the Sprint Backlog
- **Choice-3:** In a database
- **Choice-4:** In a Scrum Product Requirement Specification
- **Choice-5:** Nowhere. The Scrum Product Owner knows them
- **Correct Answer:** Choice-1 / In the Product Backlog

Question 3
Which concept is NOT defined in the Scrum Framework?

- **Choice-1:** Scrum Master
- **Choice-2:** Project Manager
- **Choice-3:** Scrum Product Owner
- **Choice-4:** Daily Scrum
- **Choice-5:** Scrum Product Burndown
- **Correct Answer:** Choice-2 / Project Manager

Question 4
In software engineering what are the disadvantages of the classical waterfall model?

- A) End-Product has to be fully anticipated beforehand.
  B) Some requirements are implemented as defined in the beginning of the project, and yet they are not really needed by the customer.
  C) Each phase is strictly separated.

- **Choice-1:** A
- **Choice-2:** B
- **Choice-3:** C
- **Choice-4:** A, B
- **Choice-5:** A, B, C
- **Correct Answer:** Choice-5 / A, B, C

Question 5
What are the advantages of the Scrum Framework?

- **Choice-1:** Fine-grained requirements are only defined when they are really needed.
- **Choice-2:** All activities to design, build and test a certain functionality are kept together in one phase.
- **Choice-3:** Changes are expected and welcomed by Scrum team.
- **Choice-4:** All of the given answers
- **Choice-5:** None of the given answers

Correct Answer: Choice-4 / All of the given answers