

Gestion de projet

MMI – CM#1 S3

Danielo **JEAN-LOUIS**

But du cours

- Découvrir la méthode Agile
- Découvrir les méthodes de travail issues du manifeste Agile

Apprentissages critiques ciblés

- AC24.02 | Mettre en place ou développer un back office
- AC25.01 | Gérer un projet avec une méthode d'amélioration continue par exemple une méthode agile

SAÉ visée

- SAÉ 3.01 | Intégrer des interfaces utilisateurs au sein d'un système d'information

**Voici ce qu'il se passait aux origines
de la gestion de projet informatique**

Votre entreprise gagne un projet avec un (gros) client. Donc très gros budget.



**L'équipe se met à travailler.
Trois méthodologies possibles :**

"code and fix", cascade et cycle en V

Code and fix

- Absence de gestion de projet
 - Économie d'argent
- Peut fonctionner si on travaille seul
- Peu de visibilité sur le projet

Modèle en cascade

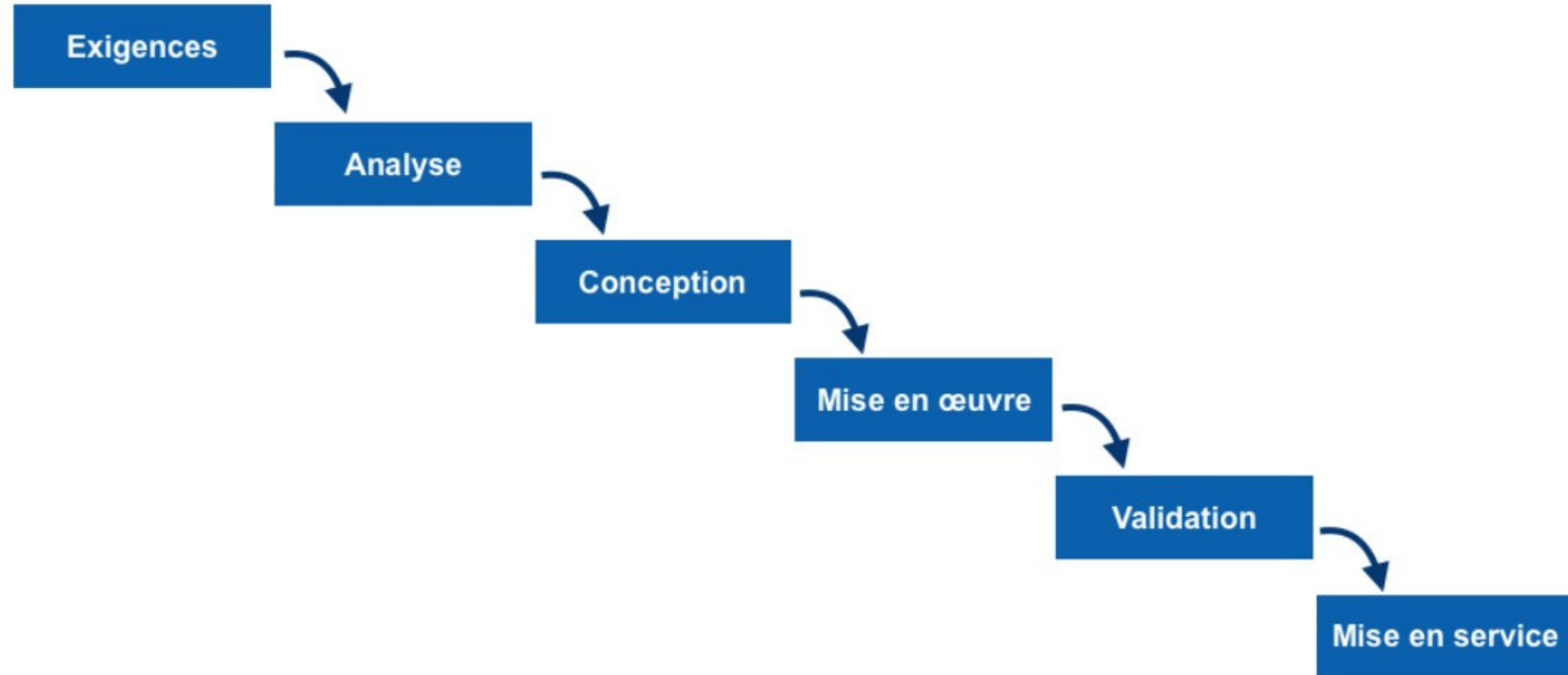


Schéma du modèle en cascade

Source(s) :

- https://fr.wikipedia.org/wiki/Mod%C3%A8le_en_cascade#Critiques

Modèle en cascade

- Enchaînement de phases de travail
- Chaque phase est dépendante de l'autre
 - On ne peut pas commencer la phase N+1 si la phase N n'est pas finie
- Le projet est vérifié dans sa totalité à la fin

Source(s) :

- https://fr.wikipedia.org/wiki/Mod%C3%A8le_en_cascade#Critiques

Modèle en cascade - Problèmes

- Les équipes ne communiquent que durant les phases de validation
- Flexibilité limitée : En cas de changement, il faut faire intervenir toutes étapes concernées
- Le produit n'est visible qu'à la fin

Source(s) :

- https://fr.wikipedia.org/wiki/Mod%C3%A8le_en_cascade#Critiques

Le cycle en V possède des problèmes du même ordre

Cycle en V

- Standard dans les années 80 dans le génie logiciel
 - Provient du monde de l'industrie
- Méthode améliorée de la cascade

Source(s) :

- https://fr.wikipedia.org/wiki/Cycle_en_V

Cycle en V

- Besoins client "immuables"
 - Définition d'un (gros) cahier des charges à partir des exigences client
 - Nécessite une mise à jour régulière du cahier des charges
 - Sert de contrat avec le client

Source(s) :

- https://fr.wikipedia.org/wiki/Cycle_en_V

Cycle en V

- Se déroule de la façon suivante :
 - Le client définit ses besoins
 - L'équipe de production développe
 - Validation client à chaque étape
 - Livraison quand tout est terminé
 - Validation client à la fin

Source(s) :

- https://fr.wikipedia.org/wiki/Cycle_en_V

Cycle en V

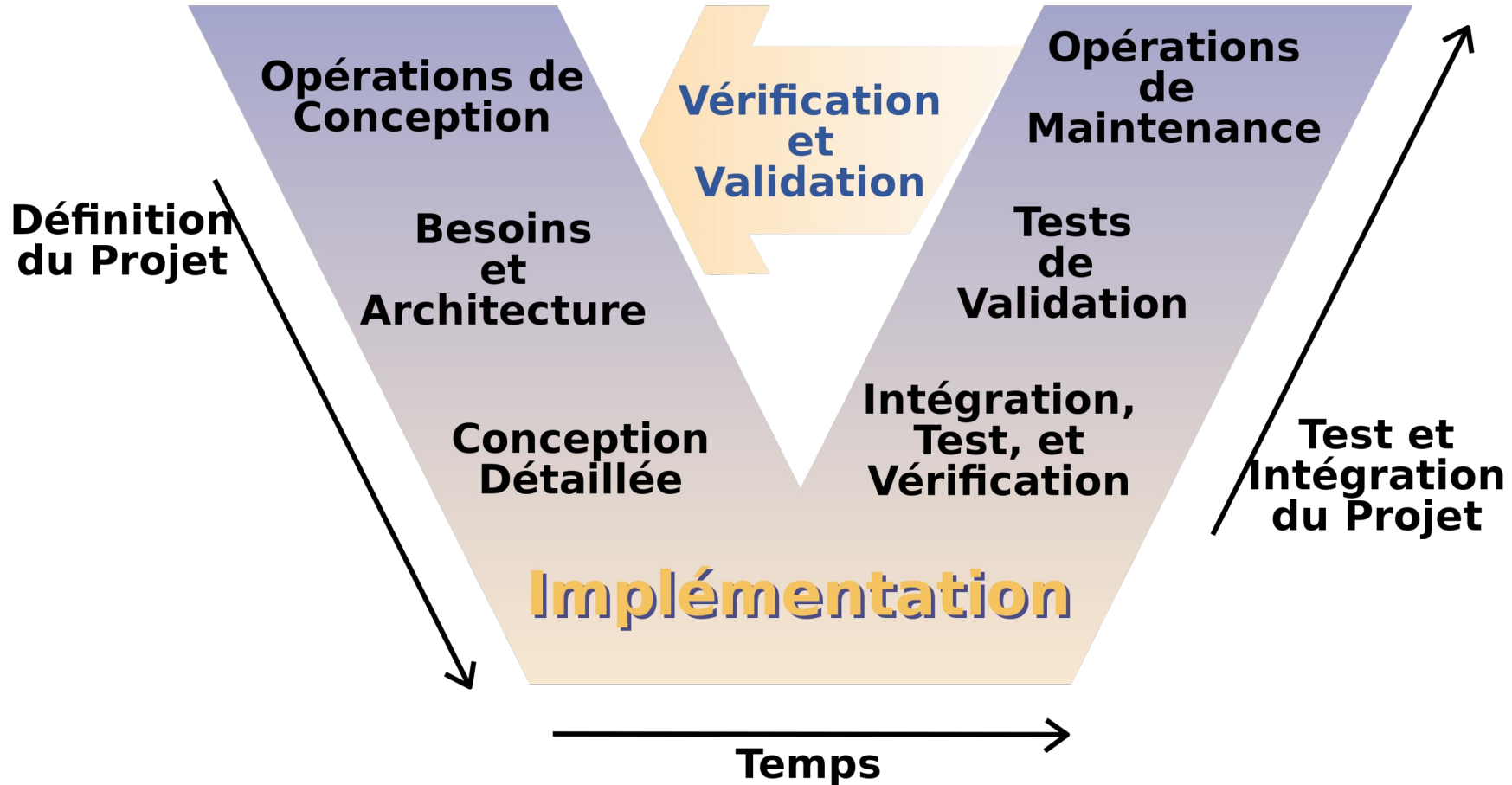


Schéma du cycle en V

En bref, c'est du travail à la chaîne

**Quels problèmes
posent le cycle en V ?**

Problèmes du cycle en V

- Faible implication de l'équipe de développement
 - Elle ne fait qu'exécuter
- Faible adaptabilité du projet
 - Si un besoin nouveau et pertinent apparaît difficile de l'implémenter
- Retards compensés par des charrettes

Cycle en V - Problèmes

- Livraison du projet en fin de cycle
 - Risque très élevé de bugs nombreux
 - Le client découvre tout à la fin (et se plaint beaucoup)
 - Le projet peut être dépassé à l'arrivée
 - Budget augmente
- Documentation trop dense

Source(s) :

- https://fr.wikipedia.org/wiki/Cycle_en_V#Critiques

Cycle en V - Problèmes

- Création d'un projet monolithique :
 - Chaque fonctionnalité est développée l'une à la suite de l'autre
- Planning et projet immuables
 - **Approche prédictive du projet**
- Responsabilité non partagée en cas d'échec du projet
 - “J’ai fait ma part, ce sont eux qui ont échoué”

Source(s) :

- https://fr.wikipedia.org/wiki/Cycle_en_V#Critiques

Cycle en V - Problèmes

- Création de conflits par l'intermédiaire du chef de projet
 - Il impose des tâches
- Manque de relation client
 - Le client intervient à la fin et au début
- ...

Source(s) :

- https://fr.wikipedia.org/wiki/Cycle_en_V#Critiques

Méthodes classiques – Problèmes communs

- Approche prédictive de la gestion de projet
 - Laisse peu de place aux imprévus et / ou changements
- Faible implication du client
 - Complexe d'avoir des retours de sa part hors phases de validation

Méthodes classiques – Problèmes communs

- Rédaction d'un cahier des charges
 - Peut être désuet au fil du temps
 - Nécessite de définir tous les besoins au début du projet
- Le résultat n'est visible qu'à la fin
- Emphase sur le processus plutôt que le produit

**Vous avez de fortes chances que votre
projet échoue avec ces méthodes**

Dans leur coin 17 développeurs vont créer des nouvelles méthodes de travail et vont graver dans le marbre leurs principes dans un document appelé le “manifeste Agile”

Le manifeste Agile

- Rédigé en 2001
- Définit la base des méthodes Agile
- Pensé initialement pour l'ingénierie logicielle
 - Peut s'appliquer à d'autres domaines (sous réserve)

Source(s) :

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

Quatre valeurs
Douze principes

Quatre valeurs

Les individus et leurs interactions
plus que les processus et les outils

Des logiciels opérationnels
plus qu'une documentation exhaustive

La collaboration avec les clients
plus que la négociation contractuelle

L'adaptation au changement
plus que le suivi d'un plan

Source(s) :

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

Principes

- Itératives courtes
- Équipes auto-organisées
- Entraide entre le client et les développeurs
- Apporter une valeur ajoutée régulièrement
- ...

Source(s) :

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

Le manifeste Agile

- Inspiré par des méthodes existantes telles que SCRUM ou Kanban
- Base de nombreuses méthodes Agile

Source(s) :

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

Bénéfices pour le client

- Obtention de valeur pour le produit
 - Rapporte plus d'argent
- Amélioration de la qualité
 - Moins de dette technique
- Respect des dates de livraison
- Livraison du produit qu'il attendait

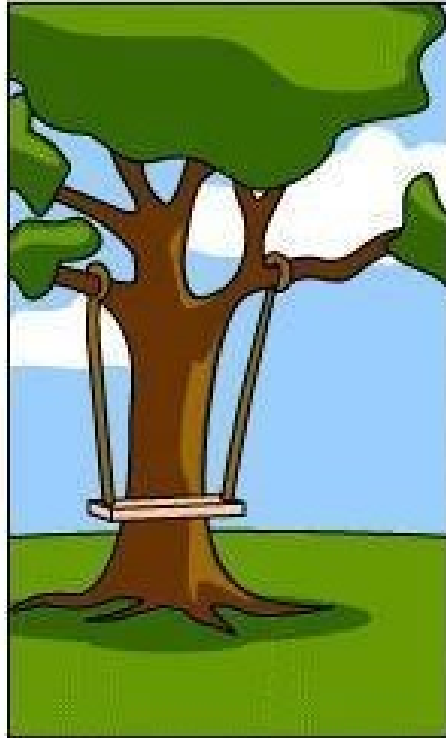
Source(s) :

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

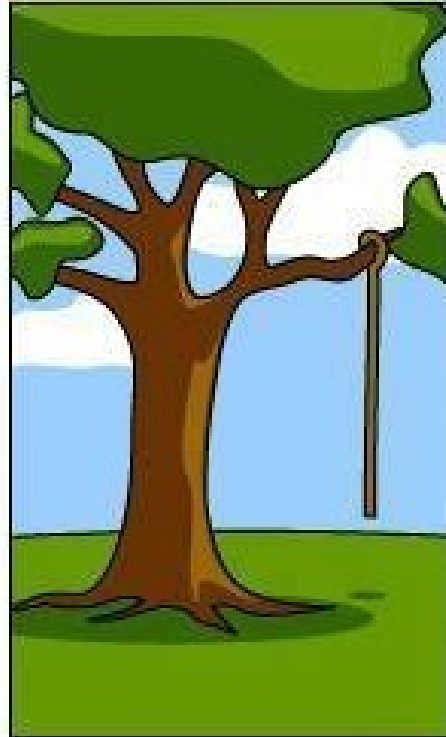
Bénéfices pour le client



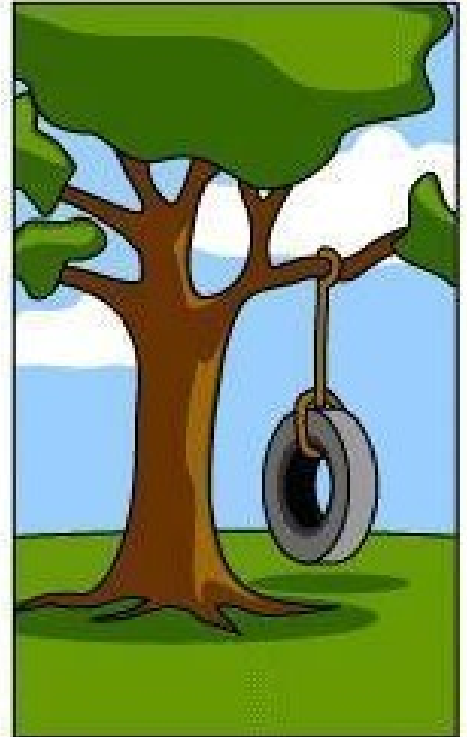
Comment le client a décrit son projet



Comment le chef de projet l'a compris.



Ce qui été livré



Ce que le client voulait en réalité.

Avec les méthodes Agile, on veut éviter ce genre de cas

Point Professionnel – *On est Agile*

- *Buzzword* du milieu professionnel
- Beaucoup d'entreprises se revendiquent et l'appliquent très mal
- Possibilité pour vous de vous valoriser en apportant la (vraie) méthode, voir même en l'enseignant auprès de vos collègues

Pourquoi être Agile ?

- Pallier aux manquements du cycle en V (et autres méthodes du même genre)
- Reposer sur un cycle de développement itératif, incrémental et adaptatif
 - Échecs moins problématiques
- Impliquer le client et les équipes (et l'utilisateur) → Intelligence collective

Source(s) :

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

"L'intelligence collective ou de groupe se manifeste par le fait qu'une équipe d'agents coopérants peut résoudre des problèmes plus efficacement que lorsque ces agents travaillent isolément"

Source(s) :

- https://fr.wikipedia.org/wiki/Intelligence_collective

Pourquoi être Agile ?

- Livraison plus régulière du produit
 - Limite les échecs
 - Maximise les profits
- Mieux prévoir, anticiper les besoins
- Mieux s'adapter aux changements du client / marché

Source(s) :

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

Pourquoi être Agile ?

- Augmenter la satisfaction client et des utilisateurs finaux
- Plus d'implication de l'équipe de production
 - Meilleures idées, fonctionnalités, motivation
- Définir un cap, une **vision** pour le projet

Source(s) :

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

Vision produit

- Intention du produit, son rôle, à quoi il sert, qu'est-ce qui est attendu
 - **Ce n'est pas la solution technique**
- Sans vision, risque élevé :
 - De délivrer un produit peu pertinent
 - Prendre du retard
 - Créer des réunions inutiles

Pourquoi être Agile ?

- Pour produire un produit de qualité
 - Moins cher sur le long terme à gérer
- **Mettre à la poubelle votre GANTT, votre Road Map, bref tout calendrier de planification**
 - Bref, limiter le plus possible la prédiction

Source(s) :

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

Pourquoi être Agile ?

- Délivrer régulièrement une nouvelle version du projet à la fin d'un **sprint**
 - Approche incrémentale : morceau par morceau le produit est livré

Source(s) :

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

Sprint

- Courte itération, répétable et fixe
- Implique tous les développeurs
- Doit apporter de la valeur au produit à la fin de la période

Sprint

- Existe dans presque toutes les méthodes Agile
- Appelé également “itération”
- Composante de l’approche empirique de la gestion de projet
 - S’oppose à l’approche prédictive

Approche prédictive

- Gestion de projet de "l'ancien temps"
- Ne permet pas de s'adapter correctement aux imprévus :
 - Absence d'un membre de l'équipe
 - Retards de livraison
 - ...

Enfants et parents de la méthode Agile

- eXtreme Programming (XP)
- sAFE : Scaled Agile Framework
- **SCRUM**
- Less : Large-scale scrum
- **Kanban**
- ...
- Votre méthode

Source(s) :

- <http://agilemanifesto.org/iso/fr/manifesto.html>
- <https://www.nutcache.com/fr/blog/les-methodes-agiles/>

Votre méthode Agile peut être la modification d'une méthode existante ou encore la combinaison de plusieurs méthodes... mais restons simples. Nous découvrirons au fil des cours des méthodes dites traditionnelles.

Source(s) :

- <http://agilemanifesto.org/iso/fr/manifesto.html>
- <https://www.nutcache.com/fr/blog/les-methodes-agiles/>
- <http://www.agilealliance.com/>

Points communs aux méthodes Agile

- Itérations courtes
 - L'équipe livre régulièrement une mise à jour du produit
- Communication permanente
 - On évite de changer régulièrement de membres
 - On fait part de ses problèmes aux autres

Source(s) :

- <http://agilemanifesto.org/iso/fr/manifesto.html>
- <https://www.nutcache.com/fr/blog/les-methodes-agiles/>
- <http://www.agilealliance.com/>

Points communs aux méthodes Agile

- Adaptabilité
 - A chaque problème, l'équipe essaye de trouver une solution et prend du recul sur l'historique
- Compréhension
 - L'équipe sait ce qu'elle doit faire. Tout le monde est d'accord sur les tâches (dans la mesure du possible) et le produit

Source(s) :

- <http://agilemanifesto.org/iso/fr/manifesto.html>
- <https://www.nutcache.com/fr/blog/les-methodes-agiles/>
- <http://www.agilealliance.com/>

Points communs aux méthodes Agile

- Documentation
 - Limitée au minimum

Source(s) :

- <http://agilemanifesto.org/iso/fr/manifesto.html>
- <https://www.nutcache.com/fr/blog/les-methodes-agiles/>
- <http://www.agilealliance.com/>

**On parlera de culture ou approche
Agile lorsqu'une équipe respecte la
méthode.**

Point Professionnel – *On est Agile mais...*

- On veut garder des méthodes de l'ancien temps
 - Road maps
 - Retro-planning (un gantt en somme)
 - ...
- On va vérifier tout le temps ce qu'il se passe
 - Micromanagement

Et parfois, ça tourne mal, très mal. Par exemple :
Voir détails de l'échec d'Agile à la MACIF

Ces comportements sont des *red flags* concernant la pérennité de la méthode Agile sur le projet

A partir du moment où vous respectez les 4 valeurs et 12 principes définies dans le manifest Agile. Il est possible de créer votre méthode pour convenir à **vos besoins**.

Source(s) :

- <http://agilemanifesto.org/iso/fr/manifesto.html>
- <https://www.nutcache.com/fr/blog/les-methodes-agiles/>
- <http://www.agilealliance.com/>

En bref – La méthode Agile

- 4 valeurs et 12 principes
- Méthodologie s'opposant aux méthodes traditionnelles de gestion de projet
 - Approche empirique
- Augmente le taux de réussite d'un projet

Source(s) :

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

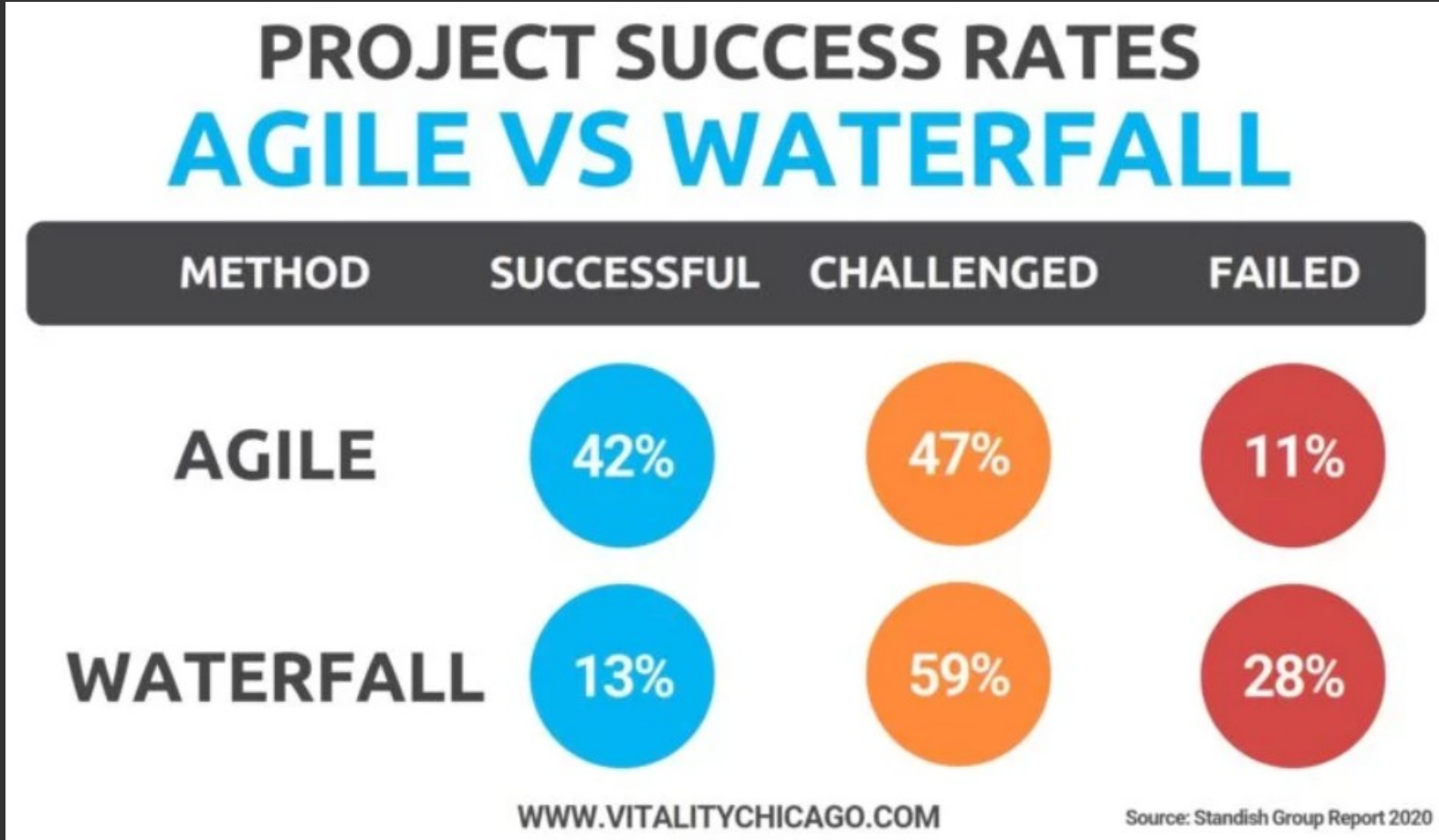
En bref – La méthode Agile

- Améliore le développement de projets (informatique)
 - **La méthode ne fonctionne pas pour tout et avec tout le monde**
- Vise à supprimer la gestion de projet prédictive et ses reliques :
 - Cahier des charges
 - Diagramme de GANTT
 - ...

Source(s) :

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

En bref – La méthode Agile



Source(s) :

- <https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/>

**Par l'absence de cahier des charges,
un projet Agile n'est pas l'anarchie
pour autant.**

Pour aller plus loin

- Choisir l'agilité - Du développement logiciel à la gouvernance
 - Auteurs : Trudel Sylvie et Boisvert Mathieu

Le livre est disponible gratuitement via la bibliothèque en ligne de l'université

Source(s) :

- <http://univ.scholarvox.com.bibdocs.u-cergy.fr/book/88805326>

Questions ?

