

INFOF-307 - GÉNIE LOGICIEL ET GESTION DE PROJETS

- EXTREME PROGRAMMING -
- INTRODUCTION -

Pluquet Frédéric

Assistants: A. Cnudde, A. Laval, Y. Molinghen, A. Reynouard

2021-2022

ULB

RÉFÉRENCE



SOMMAIRE

- **Introduction**
- Gestion de projet
- Gestion de l'équipe
- Gestion de la qualité du code
- Autres considérations
- Conclusions

ETAPES DE DÉVELOPPEMENT

- **Appréhender** le problème
- **Analyser** le problème
 - Comprendre finement le problème
 - Définir et modéliser la solution
- **Réaliser** la solution
- **Tester**
- **Déployer** (mise en production)
- **Maintenir** le logiciel dans le temps

COMPRENDRE FINEMENT LE PROBLÈME

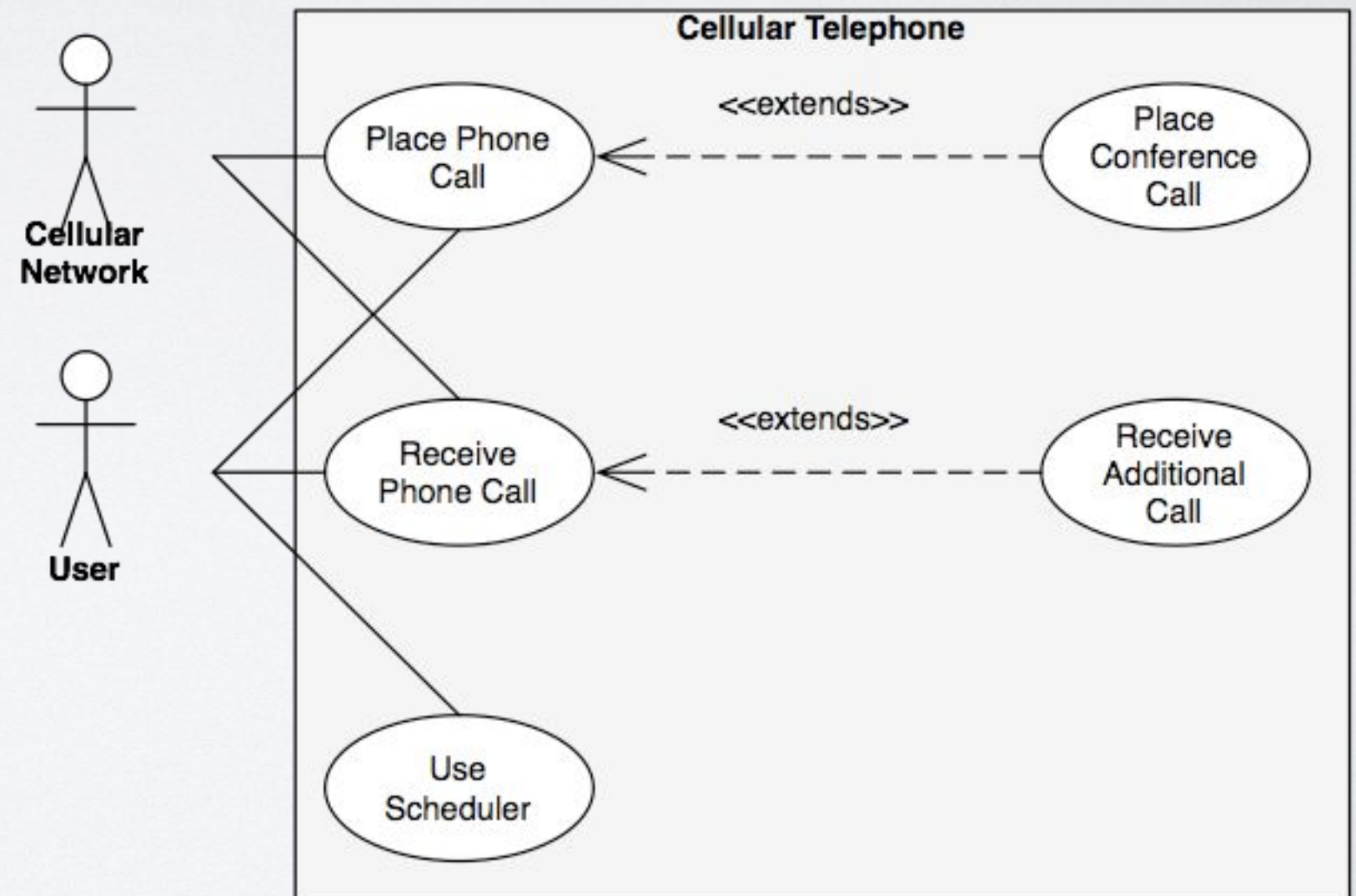
COLLECTE DES BESOINS - RAPPELS

- Requirements
 - Functional requirements
 - Non-functional requirements
 - Domain requirements
- Software Requirements Document (SRD)

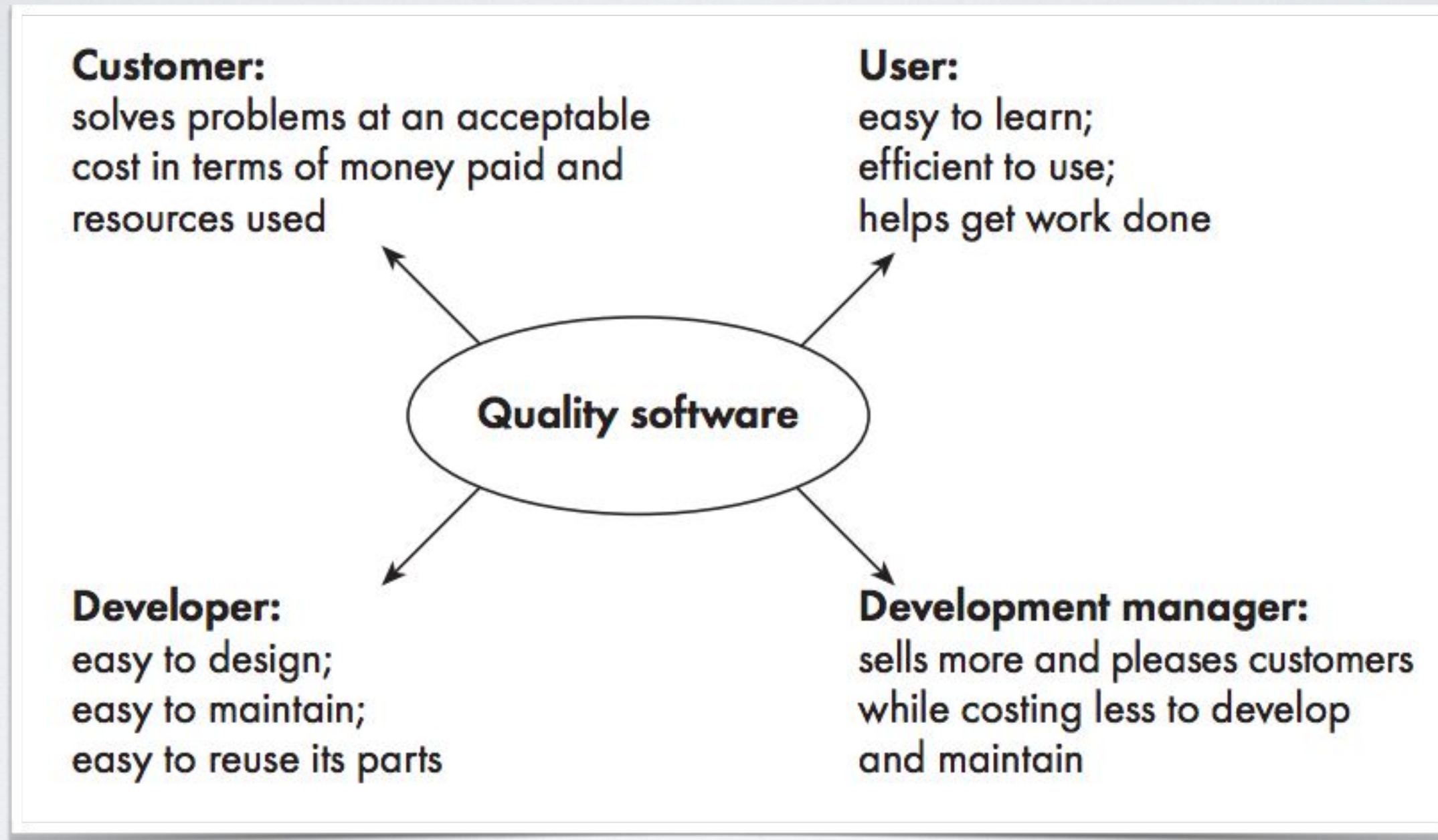
DÉFINIR ET MODÉLISER LA SOLUTION

UNIFIED MODELING LANGUAGE (UML)

- Diagrammes de cas d'utilisation (Use Cases)
- Diagrammes de classes
- Diagrammes de séquence
- Diagramme d'états
- Diagramme de composants
- Diagramme de déploiement



QUALITÉS D'UN LOGICIEL



Object-Oriented Software Engineering: Practical Software Development Using Uml and Java, McGraw Hill Higher Education

QUALITÉS D'UN LOGICIEL

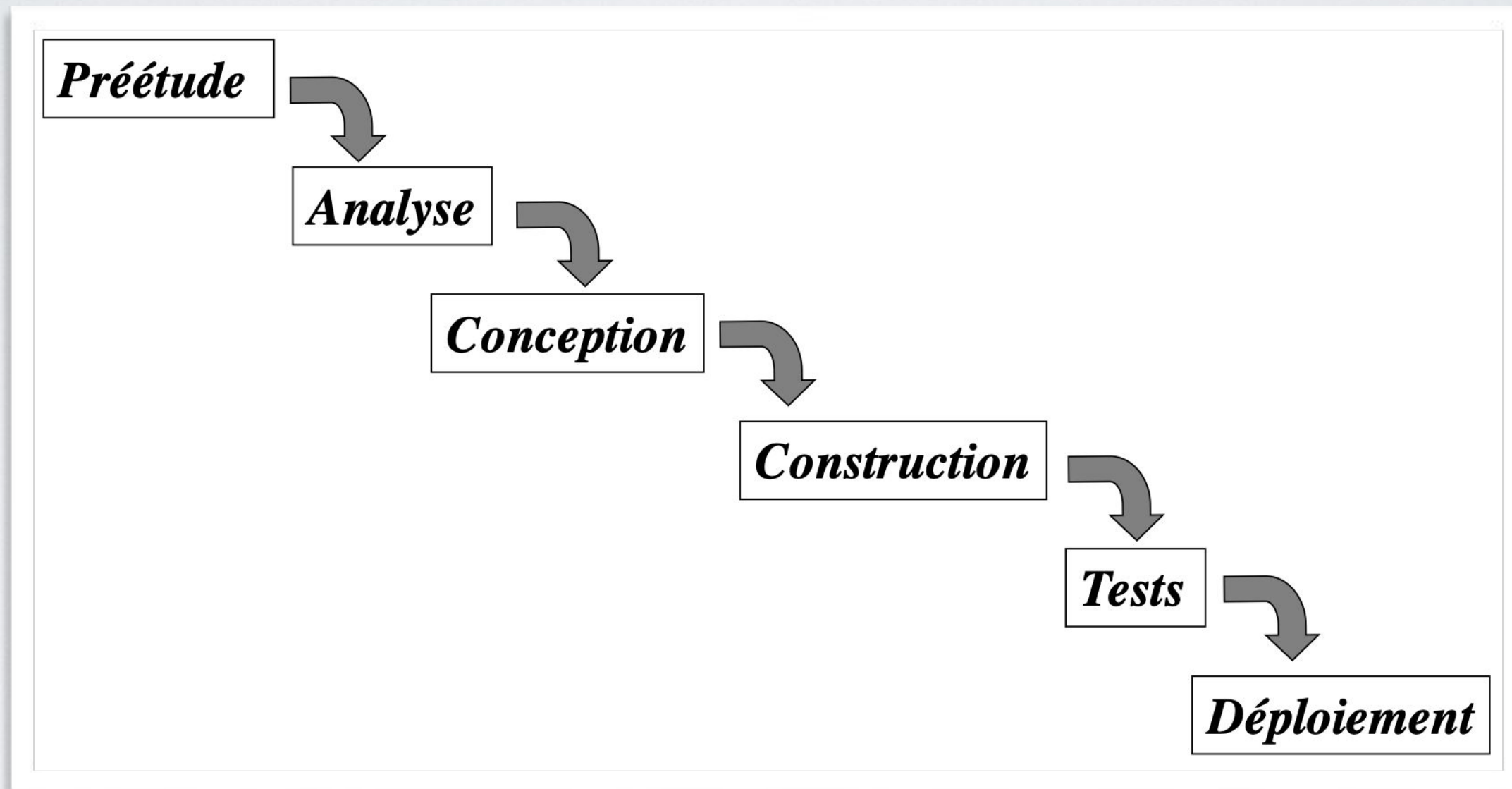
- «Ensemble des caractéristiques d'une entité qui lui confèrent l'aptitude à satisfaire des besoins exprimés ou implicites », ISO 8402:1995 (généralisée par ISO 9000:2005 et ISO 9000:2015)
- **Pertinence**
- **Apport de bénéfices**
- **Fonctionnement correct (pas de bugs, pas de perte de données, sécurité)**
- **Performance**
- **Facilité d'apprentissage**
- **Évolutivité**
- **Réutilisabilité**
- **Portabilité**

MÉTHODOLOGIES

- Procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client.



MÉTHODES EN CASCADE (WATERFALL)

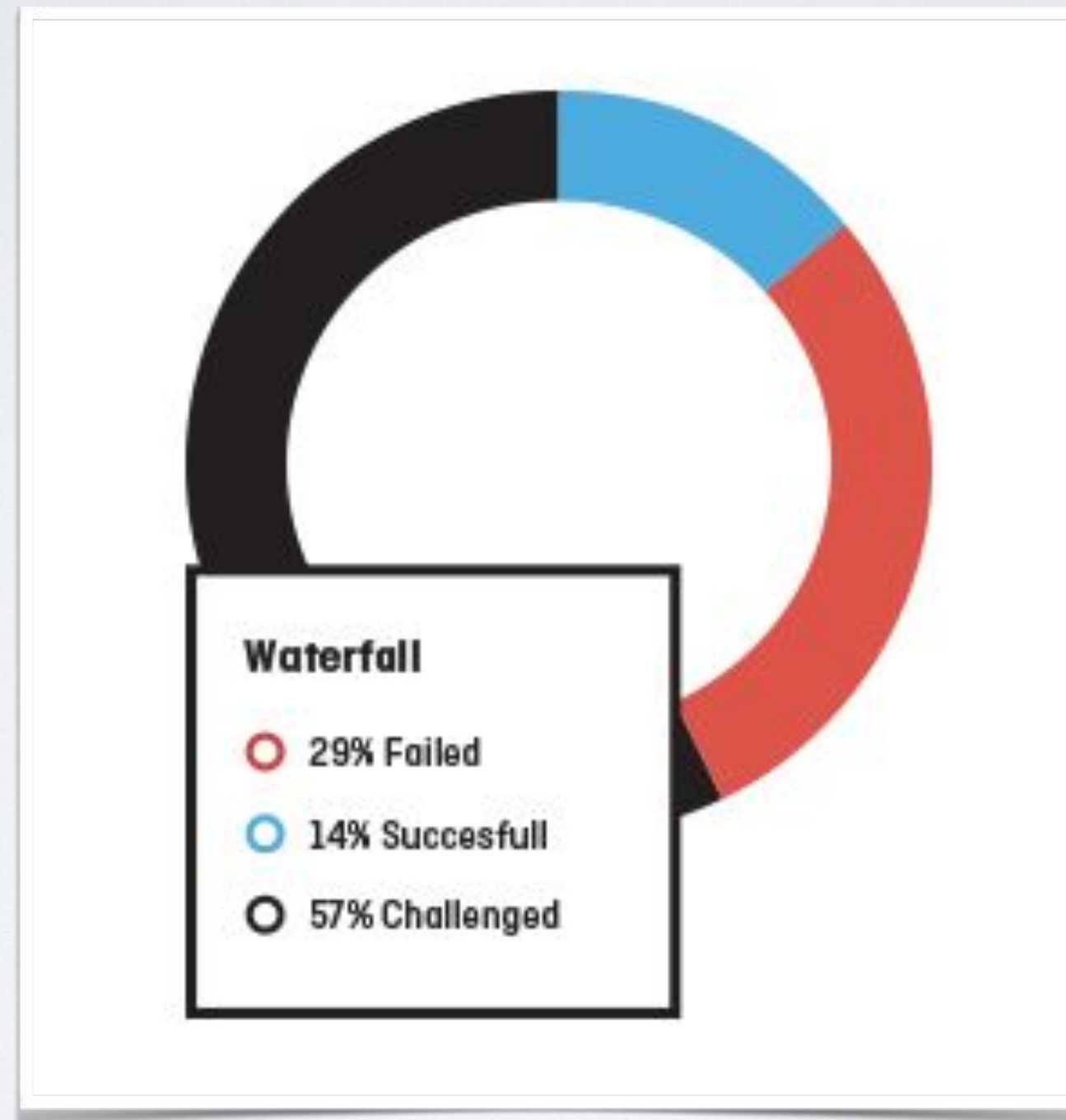


LES LIMITES DU WATERFALL

- Définition du cahier des charges
 - puis, document de spécifications
 - puis, document de conception générale
 - puis, document de conception détaillée
 - puis, plans de tests
- et finalement... on code l'application !

**Basées sur une idée:
définir *très*
précisément ce que
l'on souhaite
construire
avant de le construire**

LES LIMITES DU WATERFALL



LES LIMITES DU WATERFALL

- Pourquoi ?
 - Spécifications ambitieuses et conception poussée mais construction difficile avec remise en question des choix initiaux du projet
 - Difficile de définir *a priori* un logiciel
 - L'analyse est l'une des premières phases, où l'analyste connaît le moins le contexte
 - Le client n'intervient pas dans la construction

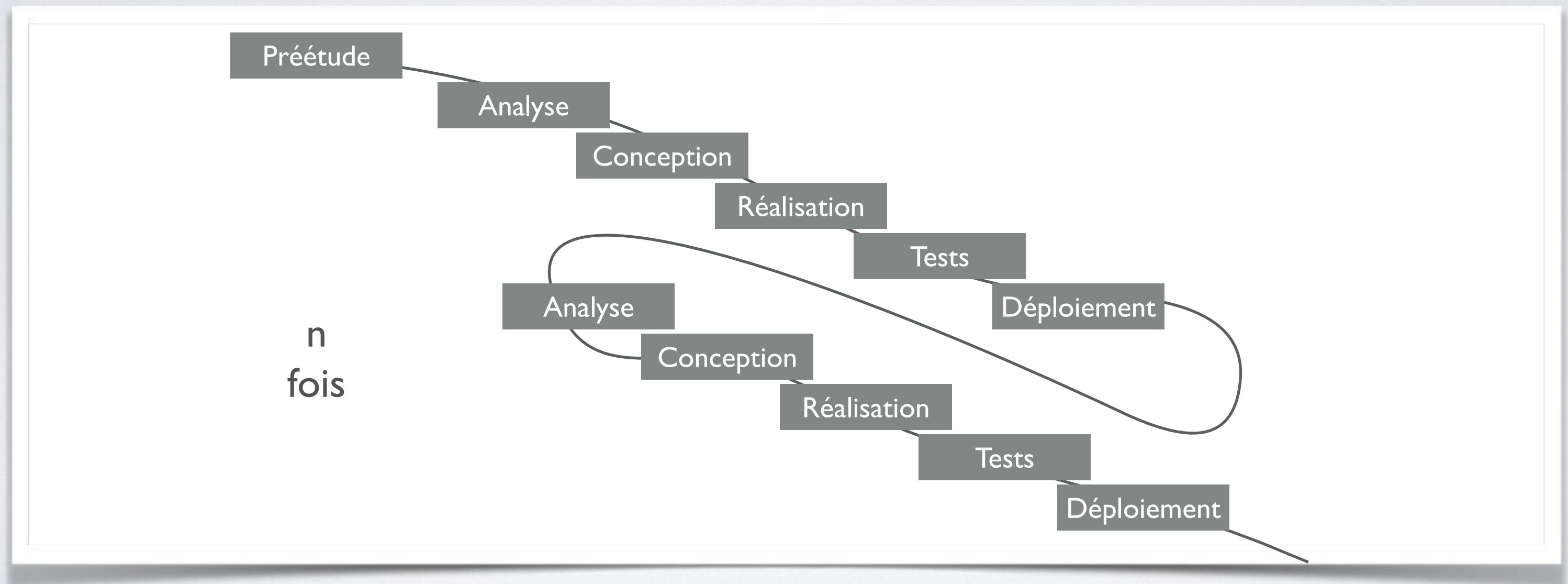
LES LIMITES DU WATERFALL

- Avantages :
 - Découpe en phases => parallélisation possible
 - Les managers ont une vision faussement simple de la gestion du temps
 - Utile si besoin de tout spécifier avant de construire
 - Par exemple: si la construction est onéreuse, si la qualité de chaque étape doit être évaluée finement, ...

LES LIMITES DU WATERFALL

- Désavantages :
 - Compartimentation des équipes
 - Développeurs & analystes
 - sous les chefs d'équipes
 - sous les chefs de projet
- Chaque développeur a son code à faire
 - Le chef d'équipe coordonne les développeurs
- Qualité de code faible (car développement local)
 - Temps de développement plus long

MÉTHODES ITÉRATIVES



MÉTHODES AGILES

- Groupes de pratiques de projets
- Applicables à divers types de projets
- Pierre angulaire : le manifeste agile

MANIFESTE AGILE : VALEURS

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

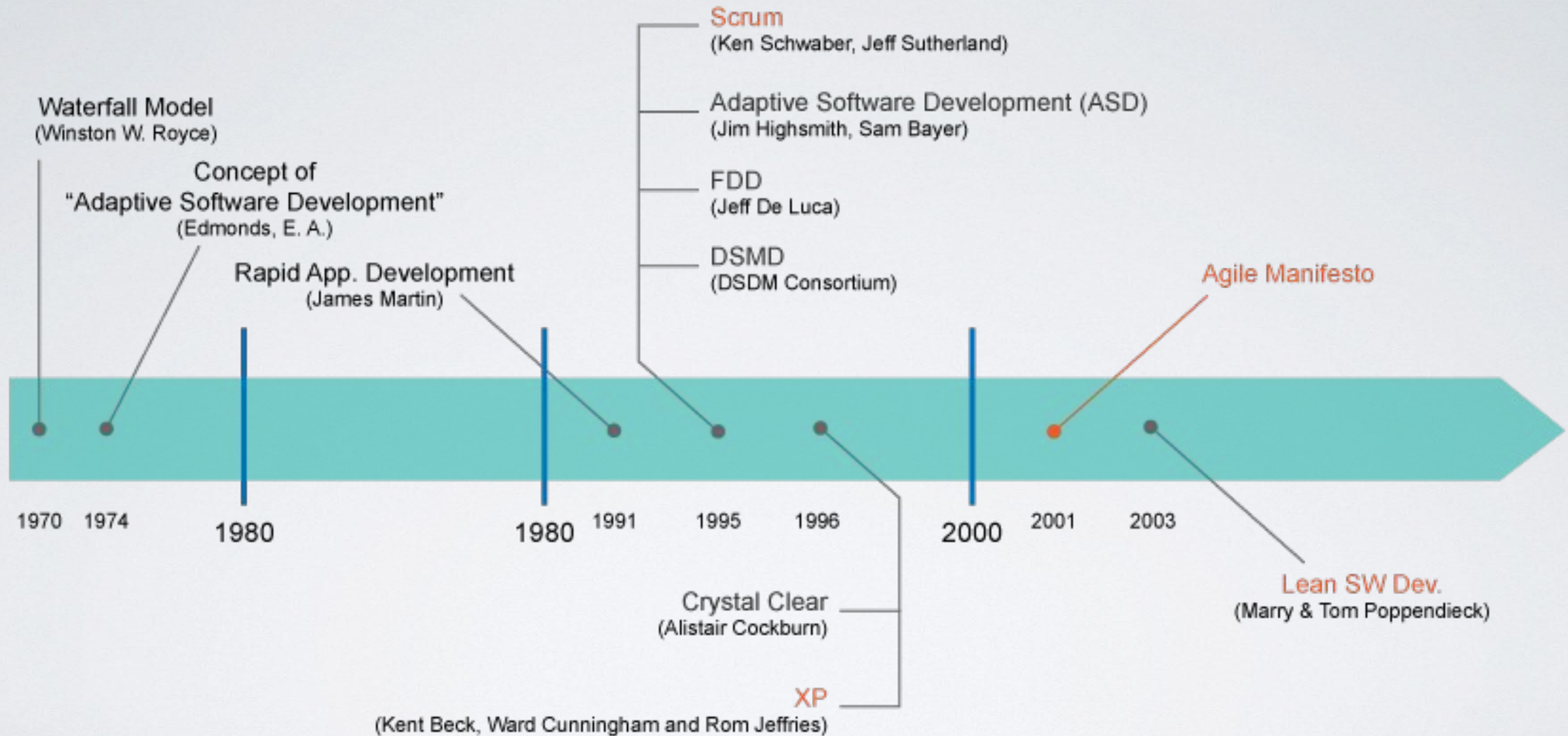
Un logiciel qui fonctionne plus qu'une documentation exhaustive.

La collaboration avec les clients plus que la négociation contractuelle.

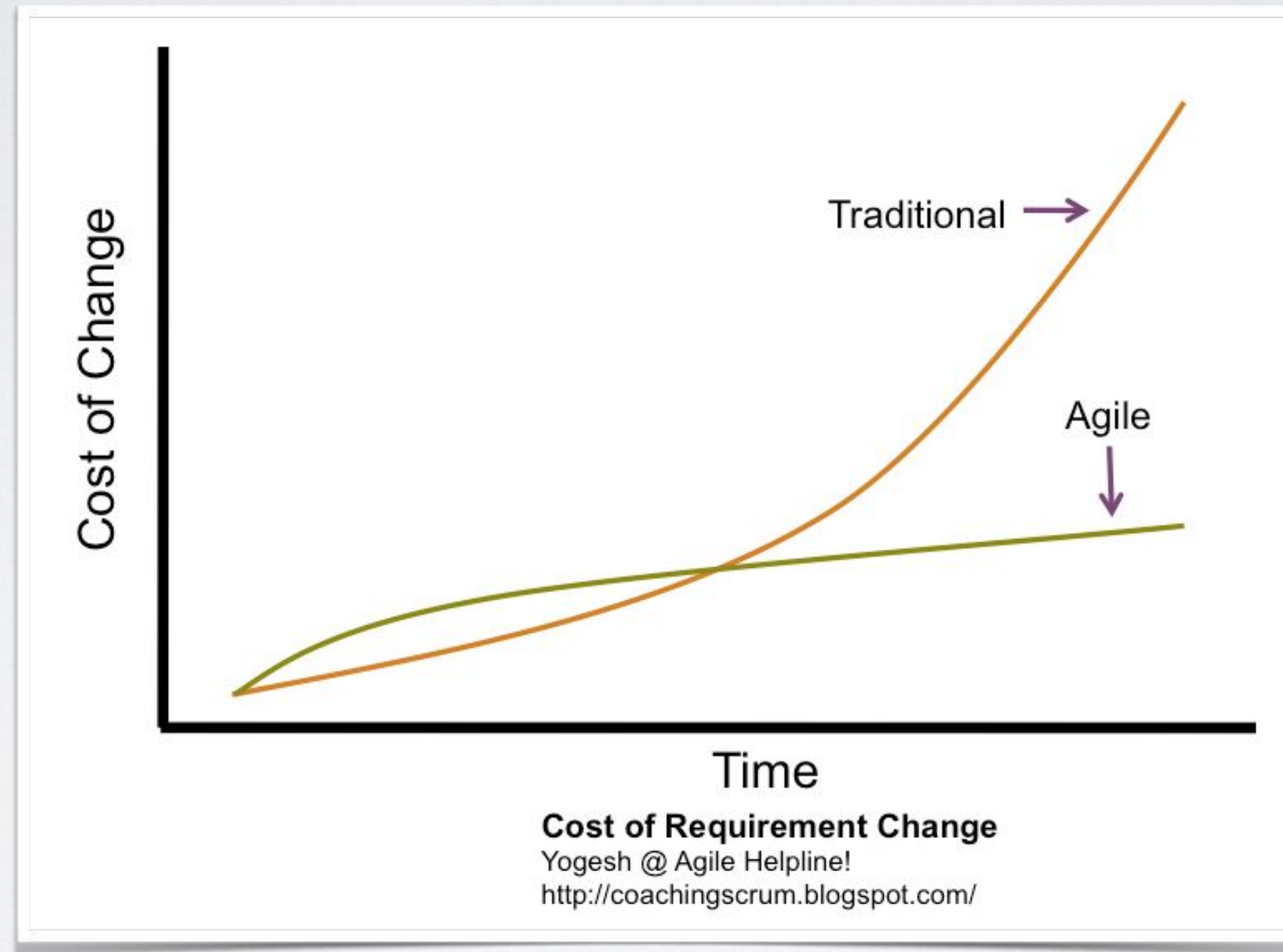
L'adaptation au changement plus que le suivi d'un plan.

Cela signifie que bien qu'il y ait de la valeur dans les éléments situés à droite, la préférence se porte sur les éléments qui se trouvent sur la gauche.

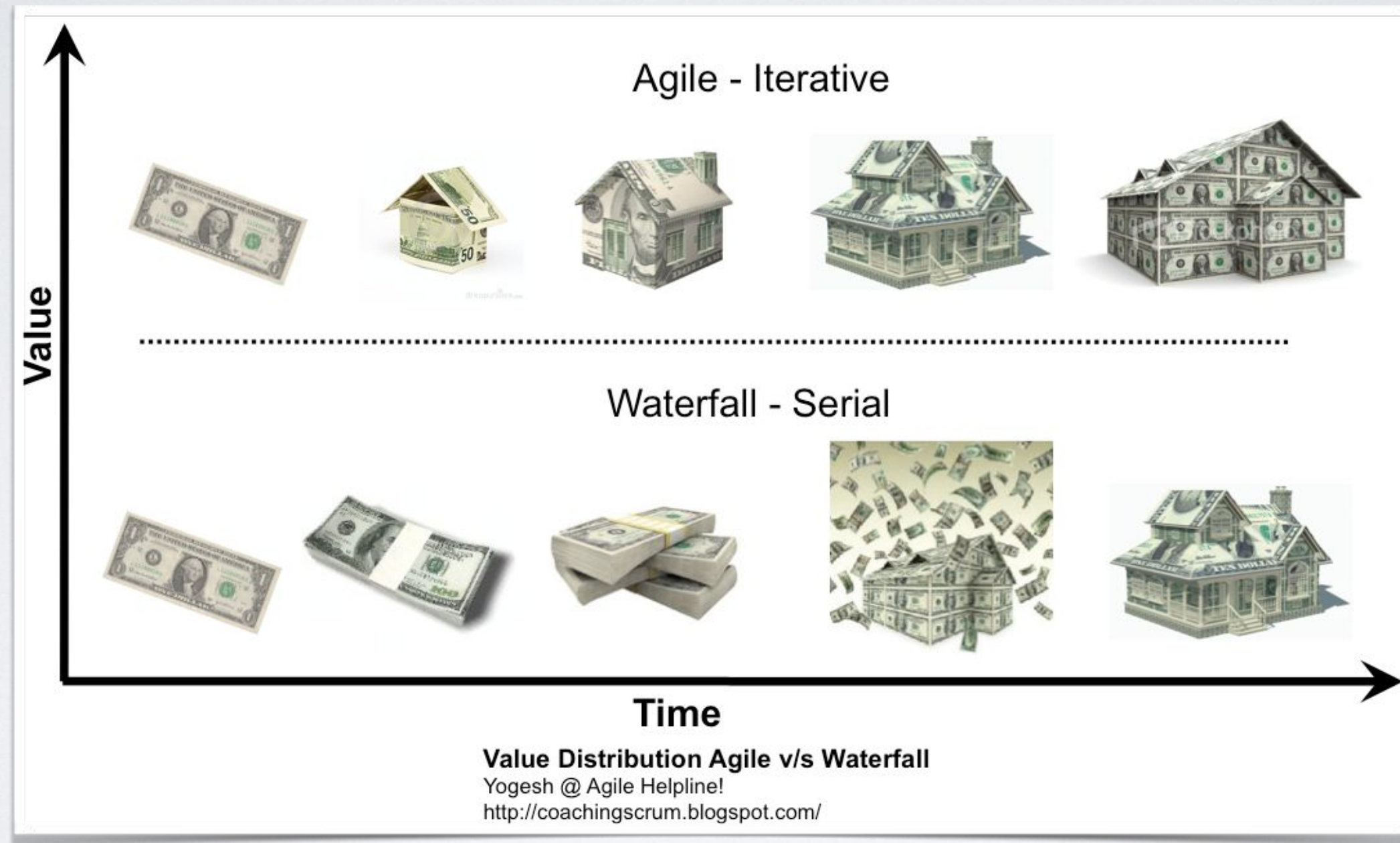
History of Agile



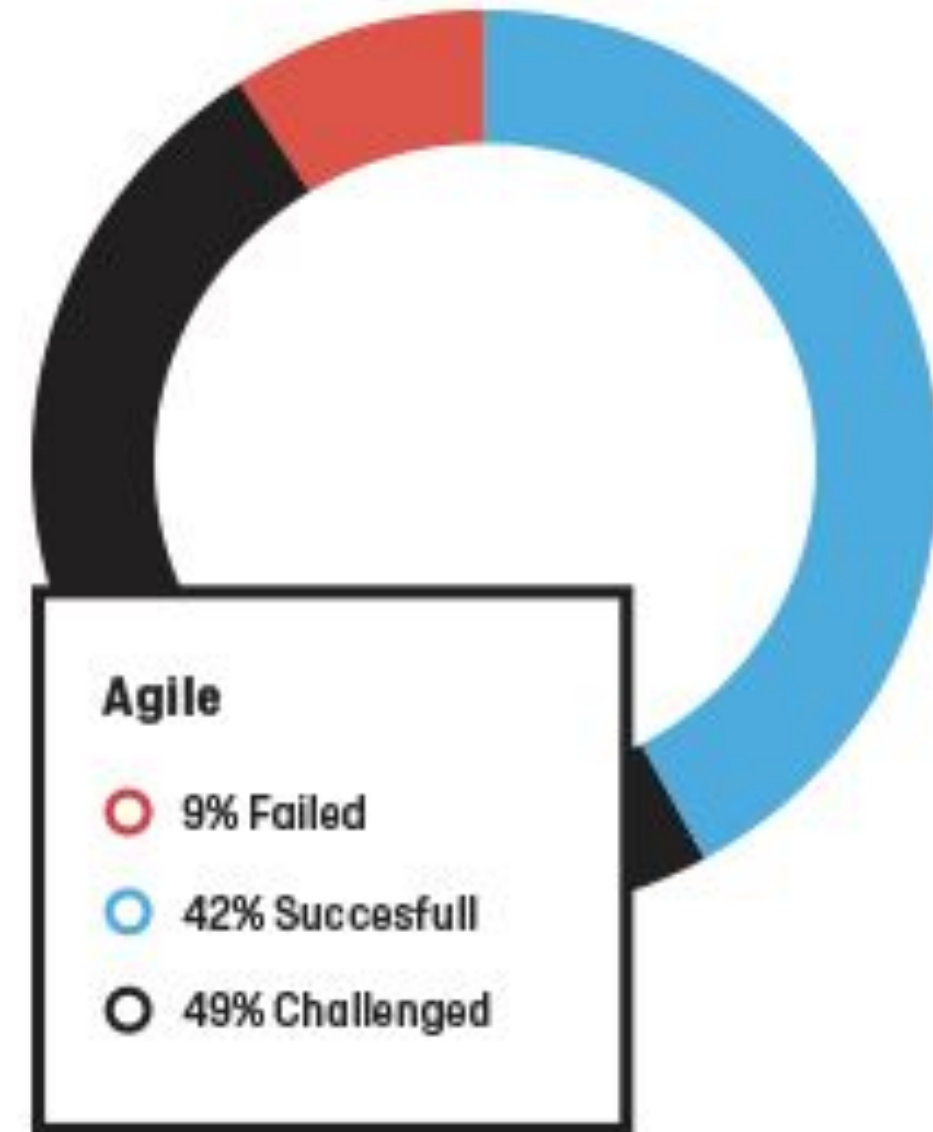
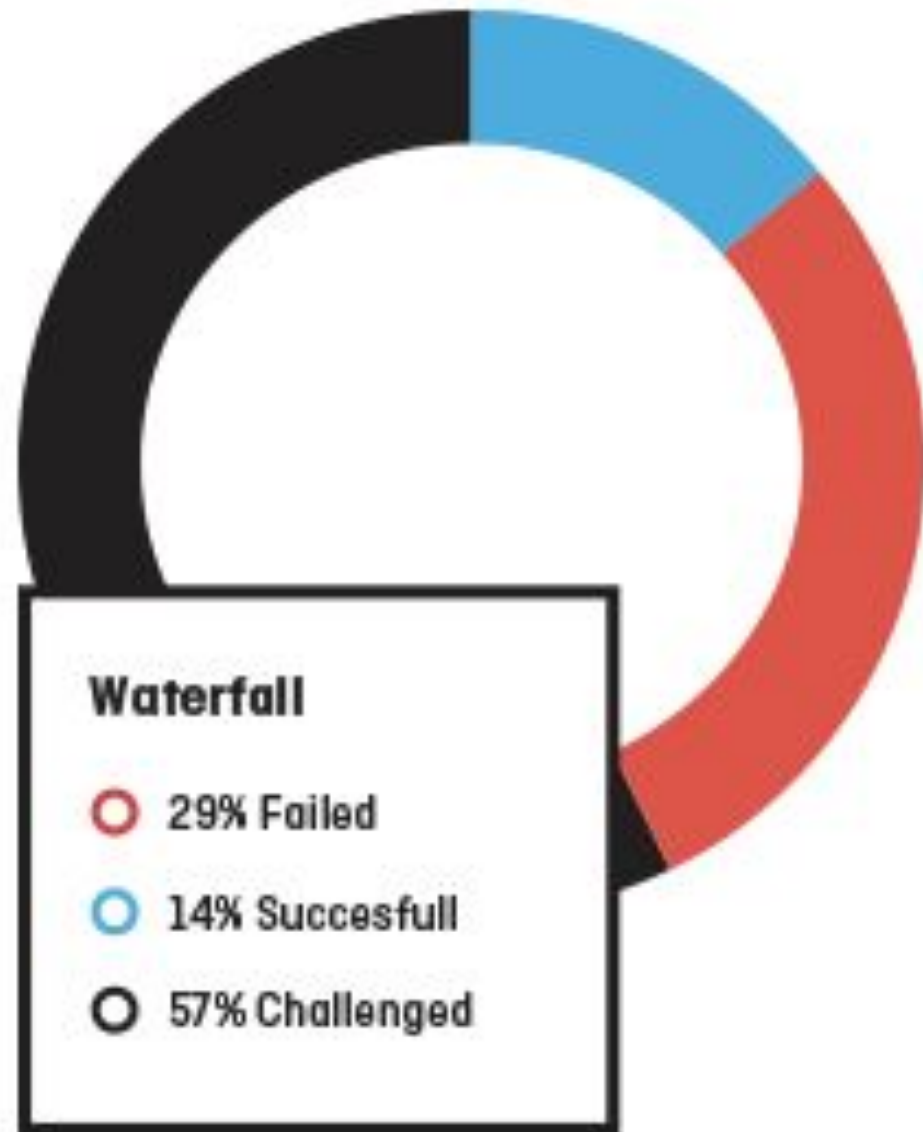
CHANGEMENT DE RÉFÉRENTIEL



PRODUCTION DE VALEUR



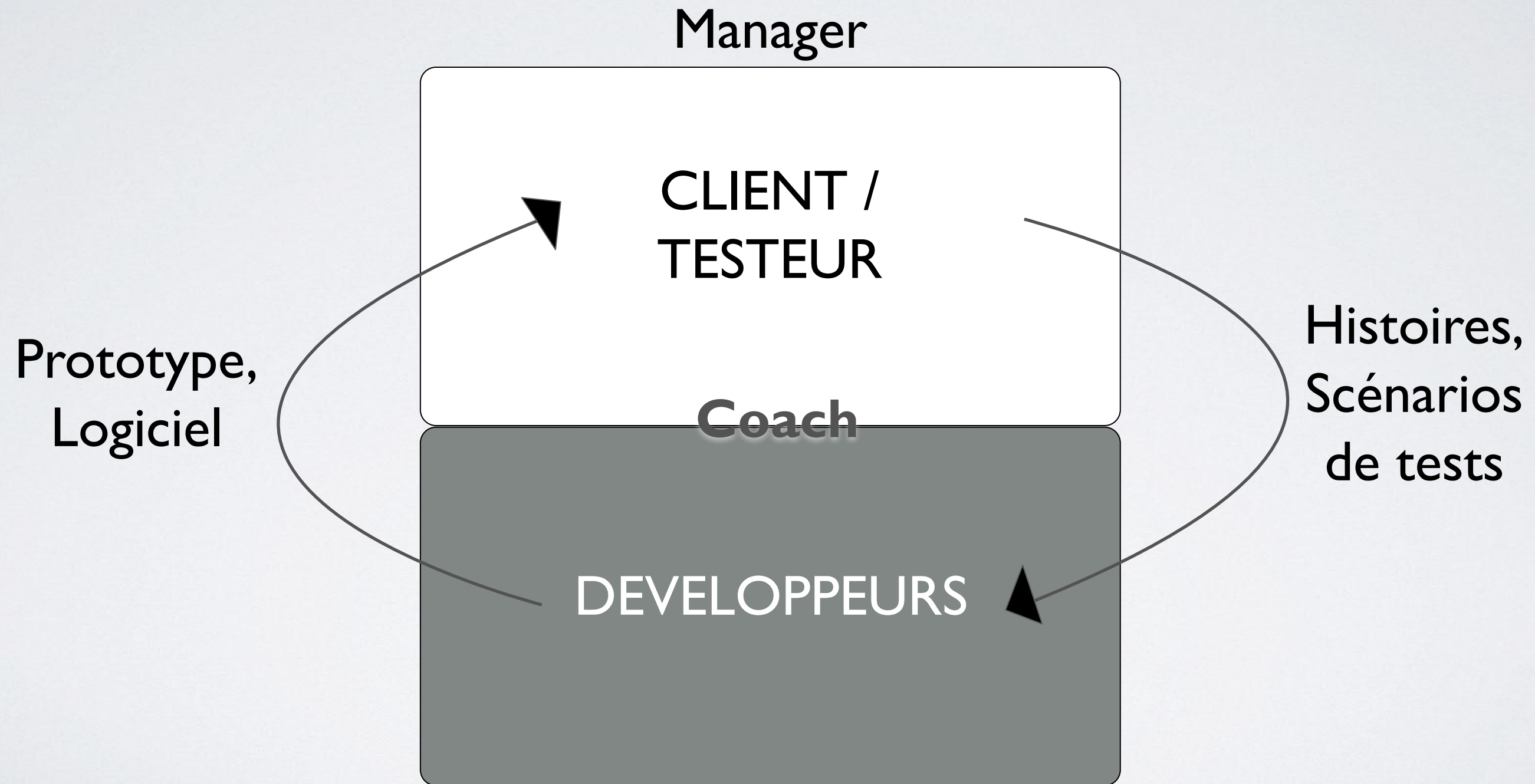
RÉSULTATS OBSERVÉS



XP ?

- e**X**treme **P**rogramming
- Agilité
 - au niveau du code
 - au niveau de l'équipe de développement
 - au niveau de la gestion de projet
- valeurs humanistes, vision positive du développement

RÔLES DANS UNE ÉQUIPE XP



PRATIQUES D'XP

- Pratiques de **gestion de projet**
 - Livraisons fréquentes
 - Planification itérative
 - Client sur site
 - Rythme durable

PRATIQUES D'XP

- Pratiques de **collaboration**
 - Programmation en binôme (pair programming)
 - Responsabilité collective du code
 - Règles de codage
 - Métaphore
 - Intégration continue

PRATIQUES D'XP

- Pratiques de **programmation**
 - Conception simple
 - Remaniement (refactoring)
 - Développement piloté par les tests unitaires
 - Tests de recette

QUATRE VALEURS D'XP

Communica-
tion

Simplicité

Feedback

Courage

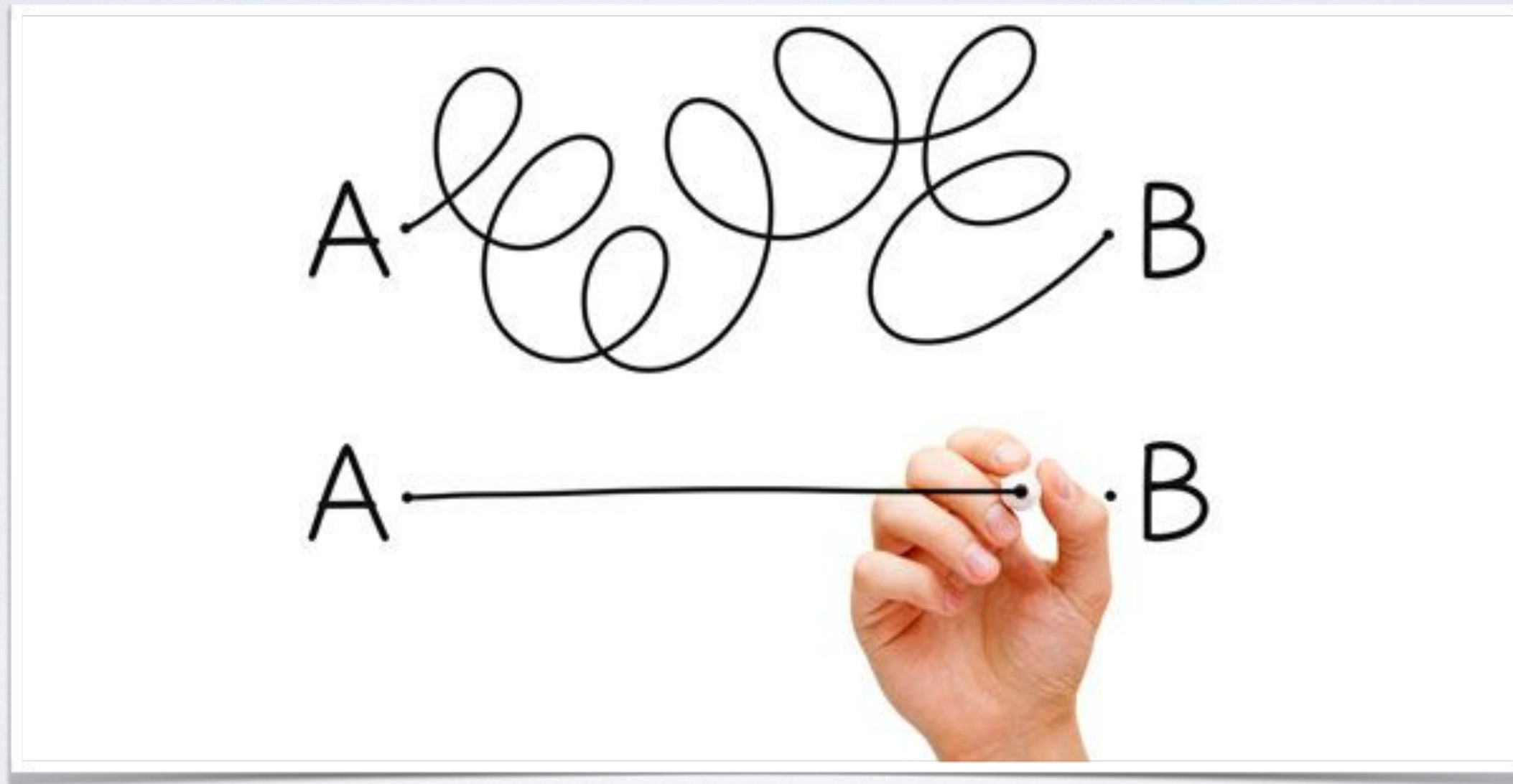
QUATRE VALEURS D'XP: LA COMMUNICATION



QUATRE VALEURS D'XP: LA COMMUNICATION

- Développement = effort *collectif*
 - meilleure communication => meilleure coordination
- XP: accent sur le contact humain
 - la parole est plus rapide
 - les idées sont plus personnelles, résolvant des problèmes liés à une personne
- faiblesse: structuration de l'information et traçabilité

QUATRE VALEURS D'XP: LA SIMPLICITÉ



QUATRE VALEURS D'XP: LA SIMPLICITÉ

- Rester simple
 - « La chose la plus simple qui puisse marcher »
 - « Tu n'en auras pas besoin »
 - « Une fois et une seule fois »
- Au niveau du code
- Avec le client

QUATRE VALEURS D'XP: LE FEEDBACK



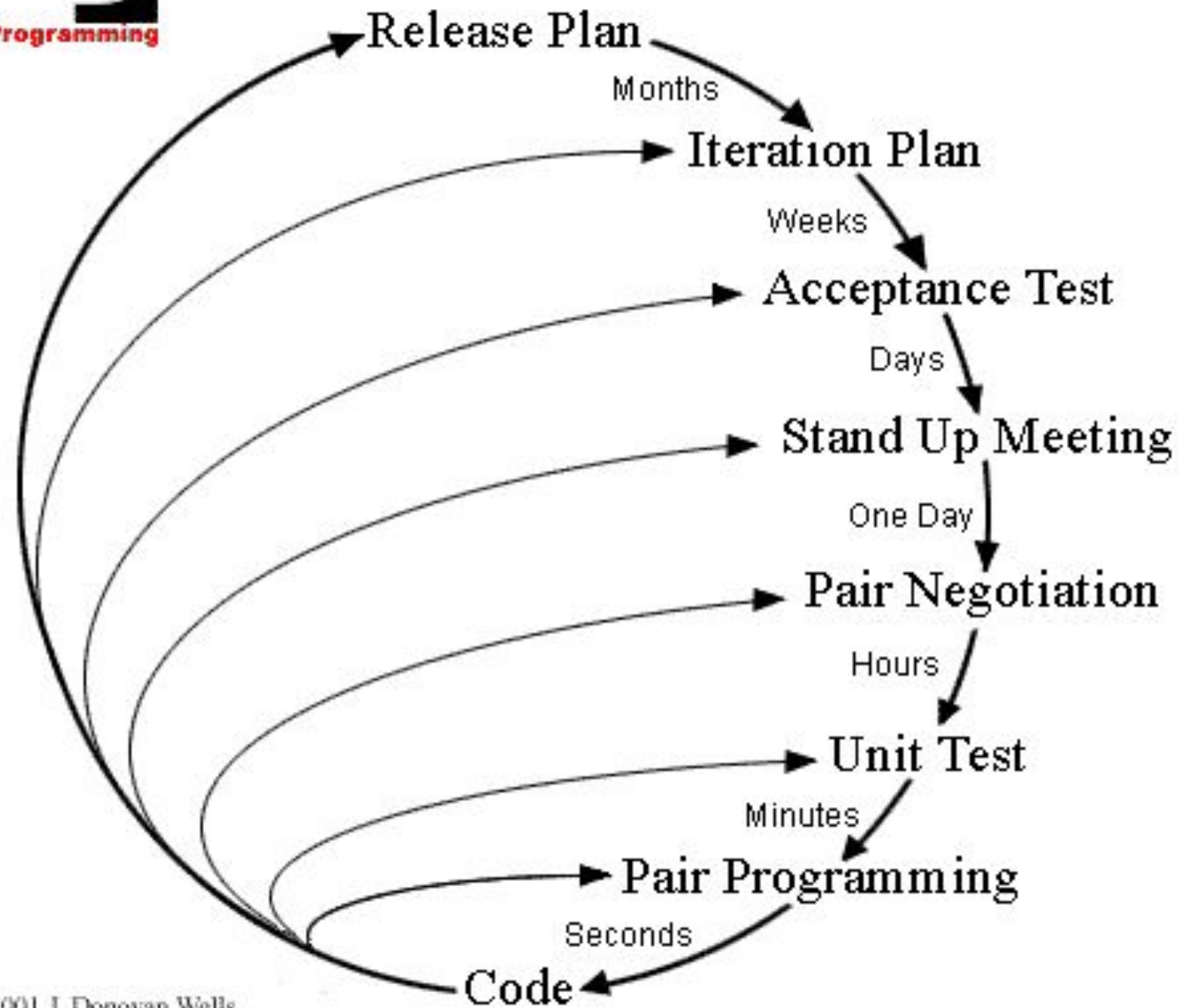
QUATRE VALEURS D'XP: LE FEEDBACK

- Savoir à tout moment où en est le projet
 - client et développeur
- Rectification des erreurs de conception si besoin est
- Réduction des risques
- Utiliser une itération pour planifier la suivante
- Feedback des changements grâce aux tests unitaires
- Feedback du binôme



Planning/Feedback Loops

Zoom Out



Copyright 2001 J. Donovan Wells

QUATRE VALEURS D'XP: LE COURAGE



QUATRE VALEURS D'XP: LE COURAGE

- XP => il faut du cran !
 - se lancer dans un projet sans savoir où on va exactement
 - se borner à réaliser des choses simples
 - se focaliser uniquement sur les besoins du moment
 - pouvoir jeter une partie de code devenue inutile

QUATRE VALEURS D'XP: LE COURAGE

- XP => il faut du cran ! (suite)
 - réécrire du code devenu trop complexe
 - communiquer tout le temps
 - chercher le feedback
 - dévoiler ses faiblesses à son binôme

QUATRE VALEURS D'XP

Communica-
tion

Simplicité

Feedback

Courage

DES QUESTIONS ?