Gestion de la production dans l'industrie du numérique
8IFG147

Introduction

Aymen Sioud
Département Informatique et Mathématique
Courriel : aymen.sioud@uqac.ca

Tables des matières

- Introduction
- Objectifs
- Contenu
- Calendrier

Introduction

- **Vos compétences**: *programming in the small*
  - programmation individuelle sur de petits problèmes
  - Algo., langages de programmation, structure de données
  - (parfois) un peu de méthodologie: analyse descendante

- **En entreprise**: *programming in the large*
  - travail en équipe pour de projets longs et complexes
  - spécifications des exigences de départ peu précises
  - dialogue avec le client-utilisateur: parler métier
  - organisation, planification, gestion de risque
  - Conséquence: démarche ingénierie : **génie logiciel**

Introduction: pourquoi le GL?

- Alors pourquoi le génie logiciel ?
  - Pour rationaliser la *production*, le *déploiement* et la *maintenance* du logiciel.
- Quelle portée de la discipline?
  - On s'intéresse au logiciel en tant que ***produit***.
  - On s'intéresse à son cycle de vie en tant que ***processus*** et aux participants humains
- Qu'est-ce qu'on vise?
  - La ***qualité*** du logiciel: multiples facteurs, difficilement mesurables.
  - La ***qualité*** du processus (***maturité***): contrôle, reproductibilité, améliorations.

Génie logiciel: définition

- Le GL doit fournir des méthodes de conception de systèmes complexes permettant:
  - Une prise en compte du client
  - Une démarche de qualité
  - Une organisation du travail en équipe

- IEEE *Standards Collection: Software Engineering* «The application of a **systematic**, **disciplined**, **quantifiable** approach to the **development**, **operation**, and **maintenance** of software, that is the application of engineering to software»

Génie logiciel: définition

- Le GL consiste à **développer** et à **utiliser** des **méthodes** et des **outils** dans le but de produire un logiciel de **qualité** en respectant les contraintes de budgets en **coût** et en **temps**

- GL: Méthodologie de construction en ***équipe*** d'un logiciel complexe et à multiples versions

- Programmation Vs. Génie logiciel (approximation):
  - **Programmation**: activité personnelle
  - **Génie logiciel**: activité d'équipe, la partie programmation ne représentera qu'entre 10% et 30% du coût total

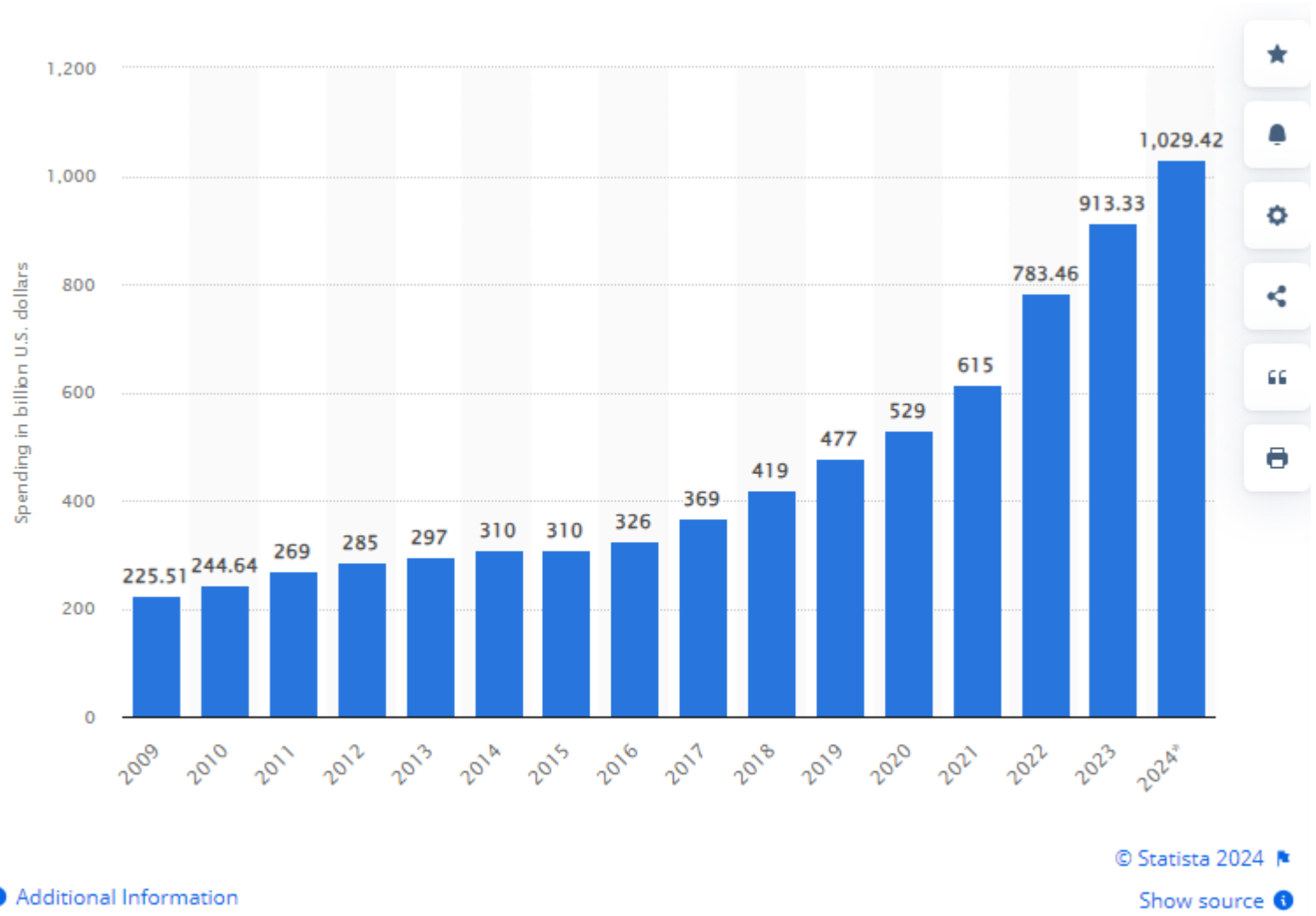
Logiciel: aspects économiques

- *The econimies of ALL developed nations are dependent on software*: [Sommerville, 2008]
- Importance économique du logiciel:
  - Importance croissance de l'informatique dans l'économie
  - Coût de logiciel supérieur à celui de matériel
  - Coût de maintenance supérieur au coût de conception

- Une amélioration de la qualité du logiciel est indispensable

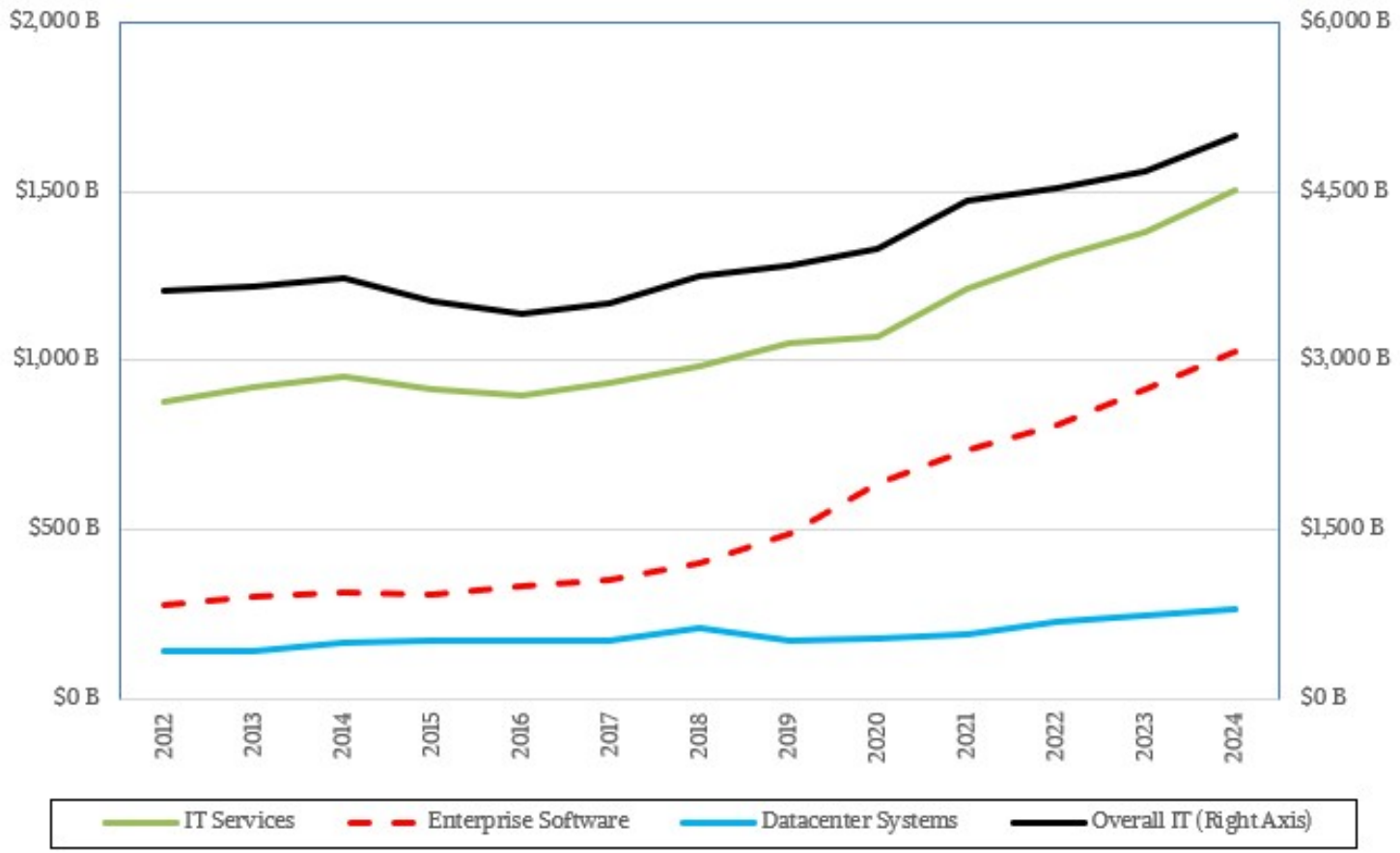
Dépenses annuelles en TI

(https://www.statista.com/statistics/247554/global-enterprise-application-software-revenue/)

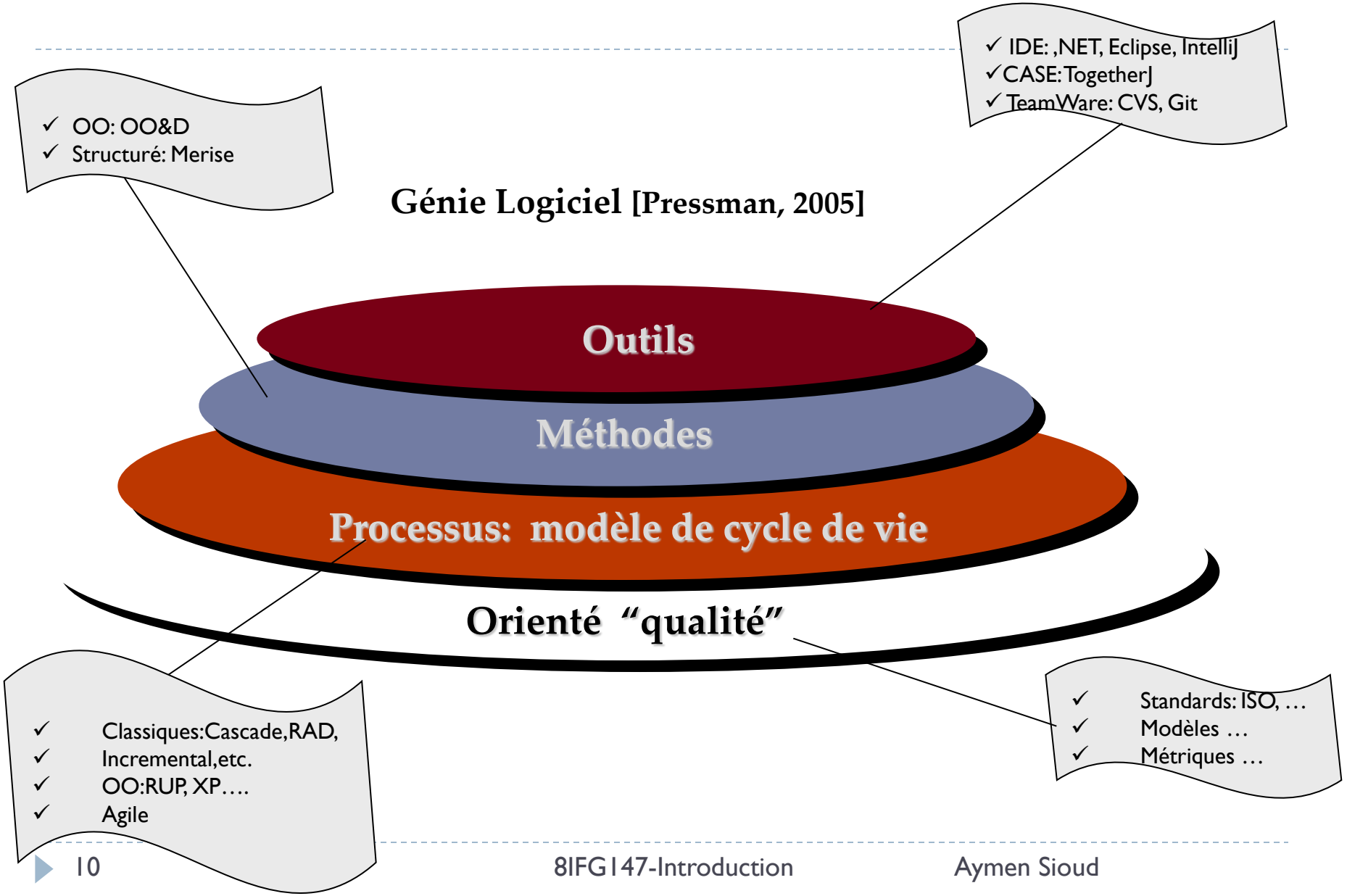


Détails des dépenses TI

(https://www.nextplatform.com/2024/01/18/it-spending-in-2024-cools-thanks-to-change-fatigue/)



Génie logiciel: une technologie par couches



Définition d'un logiciel

- Software is both a **product** (programme, document, …) and a **vehicule** (OS, Network, …) that delivers a product [Pressman, 2005]

- Un logiciel est composé de:
  - programmes
  - structure de données
  - documents

Historique et évolution

- Évolution du matériel et du logiciel:
  - 1950-1964: traitements en lots, logiciel personnalisé
  - 1964-1975: multi-usagers, temps réels, base de données, progiciel
  - 1975-1985: systèmes distribués, systèmes embarqués, matériel à bas coût
  - 1985-2002: ordinateurs personnels, Orienté-objet, systèmes experts
  - 2002-…: ordinateurs multicores, orienté-aspect, orienté composant, Entrepôt de données (datawharehouse), bioinformatique, cloud, mobile, IA …

Le logiciel aujourd'hui c'est

- Système
- Applications
- Logiciel scientifiques / ingénierie
- Logiciel embarqué
- Embedded software
- Ligne de production logiciel (CMMI)
- Application Web/Mobile
- Logiciel IA (robotique, gaming, neural)
- Logiciel en grille (GRID)
- Cloud
- IoT

Ligne de production

- Ensemble de systèmes qui partagent un ensemble commun de caractéristiques et répondant aux besoins d'un marché particulier pour la production d'un logiciel
- Utilisation d'un noyau commun et de composants réutilisables
- Partage un ensemble d'actifs
  - exigences, l'architecture, design patterns, des composants réutilisables, les test cases, et d'autres livrable
- Permet d'élaborer de nombreux produits qui sont conçus en capitalisant sur les points communs entre tous ces produits,

Crise du GL

CHAOS RESOLUTION BY AGILE VS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of software projects from FY2011-2015 within the new CHAOS database, segmented by the Agile process and waterfall method. The total number of software projects is over 10,000

Problème du génie logiciel

1. Accroissement de la demande et des coûts de développement
2. Taille et complexité des logiciels
3. Taille croissante des équipes
4. Spécifications peu précises
5. Évolution rapide des applications

Accroissement de la demande et des coûts de développement

Worldwide IT Spending Forecast (Millions of U.S. Dollars)

	2023 Spending	2023 Growth (%)	2024 Spending	2024 Growth (%)
Data Center Systems	243,063	7.1	261,332	7.5
Devices	699,791	-8.7	732,287	4.6
Software	913,334	12.4	1,029,421	12.7
IT Services	1,381,832	5.8	1,501,365	8.7
Communications Services	1,440,827	1.5	1,473,314	2.3
Overall IT	**4,678,847**	**3.3**	**4,997,718**	**6.8**

Source: Gartner (January 2024)

Taille et complexité des logiciels

- Le logiciel offre de plus en plus de fonctionnalités (système d'information, data warehouse)
- Logiciel souvent non unique, entités à interfacer
- Technologie en mutation (OS et langages en évolution)
- Complexité architecturale (machines distantes, hétérogènes, client-serveur, Intranet, …)

- **Solution** : décomposer le processus de développement, découper en sous-systèmes, se rapprocher d 'un découpage naturel proche de la réalité.

Taille croissante des équipes

- Gestion des compétences variées
- Coordination des travaux et circulation de l'information
- Gestion en parallèle du travail sur une même tâche
- Utilisation d'un langage non ambigu et compréhensible par tous les acteurs du problème
- Problème relié aux délais de plus en plus courts.

- **Solution** : Unification du vocabulaire et méthode d'organisation du travail

Spécifications peu précises

- Spécifications précises, cohérentes et complètes
- Représentent une vue fonctionnelle du système à réaliser : rôle important du client et donc proche du domaine d'application vs. Analyse et conception
- Nécessité d'un formalisme simple et concis compris par tous les partis.

- **Solution** : Modèles pour éclaircir, récapituler et montrer les points clés des spécifications.

Évolution rapide des applications

- Évolution fonctionnelle et technique
- Modification des besoins du client
- Modification de l'activité du client
- Modification de l'environnement technique.

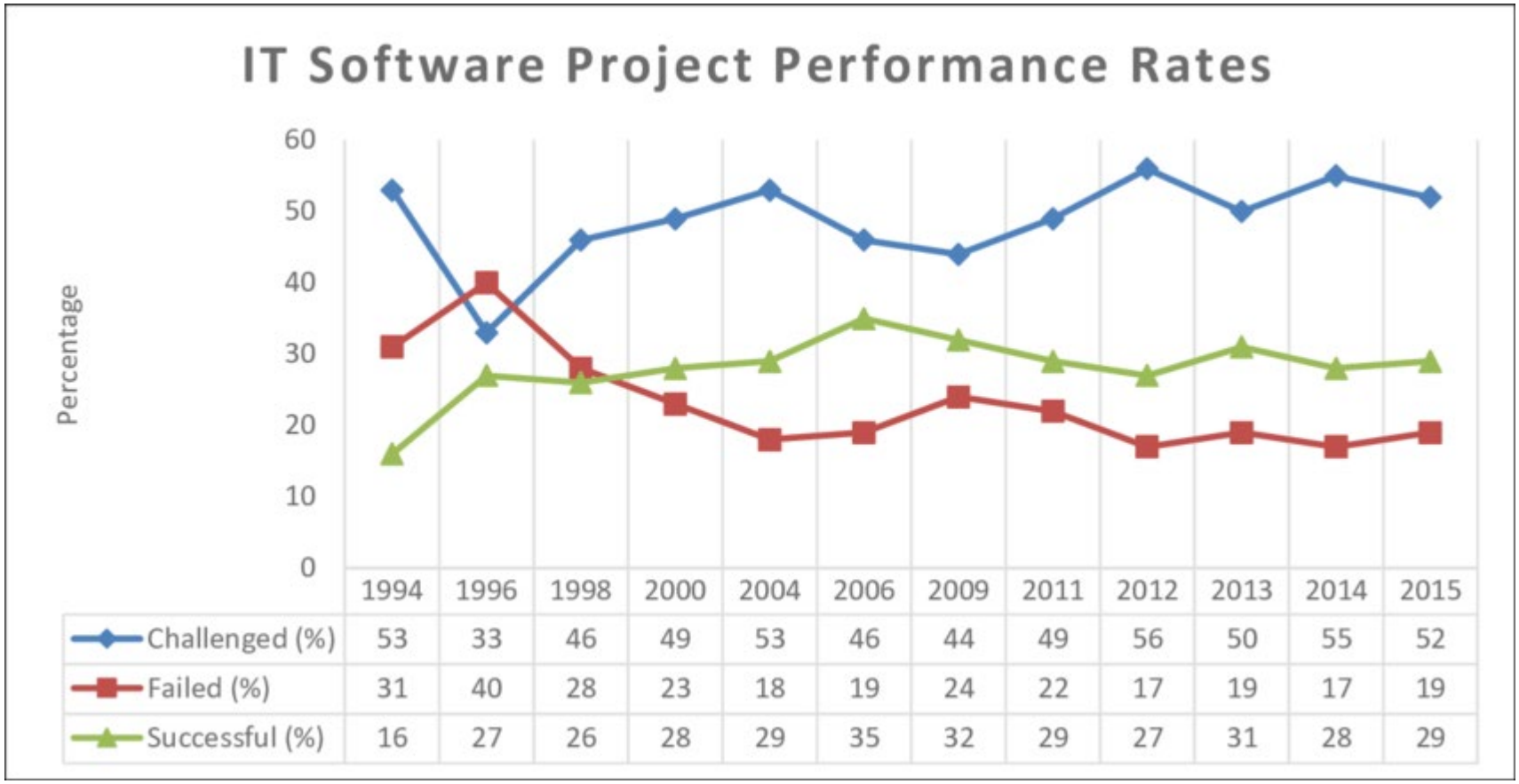
- **Solution** : cycle de vie itératif et incrémental, comme le cycle de vie objet ou des méthodes agiles.

Crise du logiciel

- Les cause de l'échec sont:
  - Mauvaise division et répartition du travail dans une équipe de personnes
  - Mauvaise communication entre les personnes ou entre les équipes-transfert des connaissances
  - Taille du logiciel par rapport à l'effort estimé
  - Relation entre client et concepteur est d'abord un problème de langage (transformation de l'informel vers le formel, mauvaise compréhension des besoins)
  - Logiciels livrés mais jamais utilisés avec succès parce qu'ils ne respectent pas la qualité requise
  - …

Évidence de la crise du logiciel

(Chaos Report statistics for IT software projects from 1994 to 2015 (adapted from Curtis (2012), C Marnewick (2012) and Hastie et al. (2015)))



	1994	1996	1998	2000	2004	2006	2009	2011	2012	2013	2014	2015
Challenged (%)	53	33	46	49	53	46	44	49	56	50	55	52
Failed (%)	31	40	28	23	18	19	24	22	17	19	17	19
Successful (%)	16	27	26	28	29	35	32	29	27	31	28	29

Crise du logiciel: loi de l'évolution de logiciel

- **Loi du changement continuel**: un logiciel doit être continuellement adapté/changé, sinon il devient moins satisfaisant à l'usage!

- **Loi de complexité croissante**: lorsqu'un logiciel change, sa structure tend à devenir plus complexe. Des ressources additionnelles doivent être consacrées à maintenir et à préserver sa structure
- **Loi de croissance constante**: un logiciel doit doter constamment de nouvelles fonctionnalités afin de maintenir la satisfaction des utilisateurs tout au long de sa vie
- **Loi de la qualité déclinante** : la qualité d'un logiciel tend à diminuer face aux changements!

Facteurs de réussite selon Standish

Critère de réussite	Importance
Implication des utilisateurs	19
Soutien de la hiérarchie	16
Formalisation claire des besoins	15
Assurer un découpage correct	11
Attentes réalistes	10
Découpage du projet en petites étapes	9
Compétence de l'équipe projet	8
Appropriation du projet	6
Claire vision de la raison d'être et des objectifs du projet	3
Capacité de travail de l'équipe projet	3
Total	**100**

Scwalbe, p432

Domaines d'applications

- Les principaux domaines d'application
  - Logiciels système
  - Logiciels scientifiques
  - Logiciels embarqués
  - Logiciels d'intelligence artificielle
  - Logiciels d'application Web
  - Jeux vidéo
  - …

Solution à cette crise:
Une méthodologie de développement

1. **Améliorer la productivité :**
   - faciliter le recensement des «vrais» besoins
   - Réutilisation des composants
   - Prototypage rapide

2. **Réduire la complexité**
   - automatisation de la documentation et de la programmation
   - développement incrémental
3. **Améliorer la qualité**
   - Test de logiciel
   - **Gestion de projet**

Gestion de projets

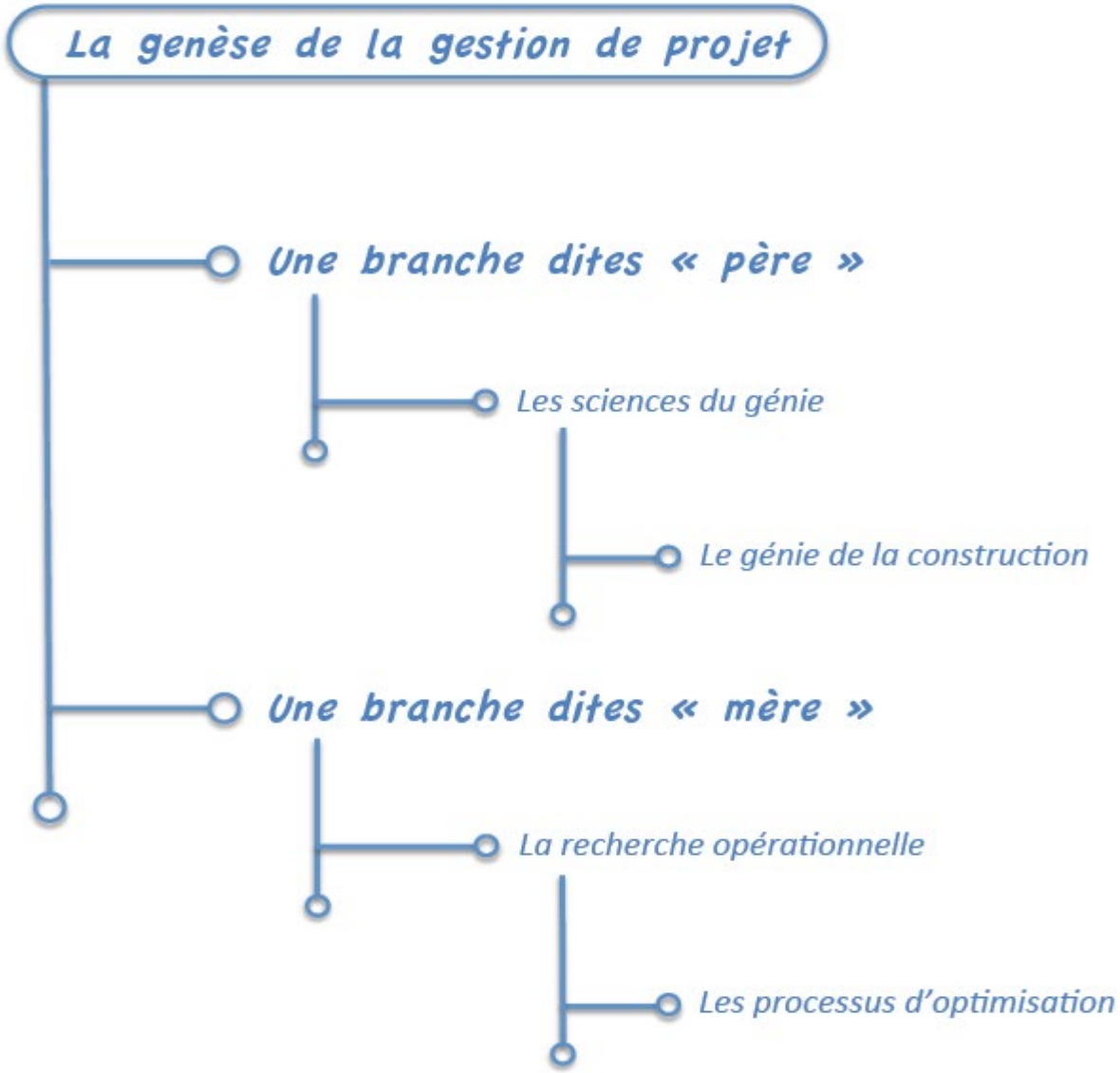
La genèse de la gestion de projet

- Ingénierie
  40 - 50

- Optimisation
  50 - 60

- La gestion de projet

- Ingénierie
- Sciences
- Manufacturier
- Services
- Informatique


La genèse de la gestion de projet
Une branche dites « père »
Les sciences du génie
Le génie de la construction
Une branche dites « mère »
La recherche opérationnelle
Les processus d'optimisation

Tout part après la deuxième guerre mondiale…

- Après la deuxième guerre mondiale
  - Projets aéronautiques
  - Projets de travaux publics
  - Projets d'armement.
- Introduction de la conduite de projet
  - Recherche opérationnelle, gestion en général
- Pratique / Sociétés de service
  - Conseils globaux ou ponctuels

Définition

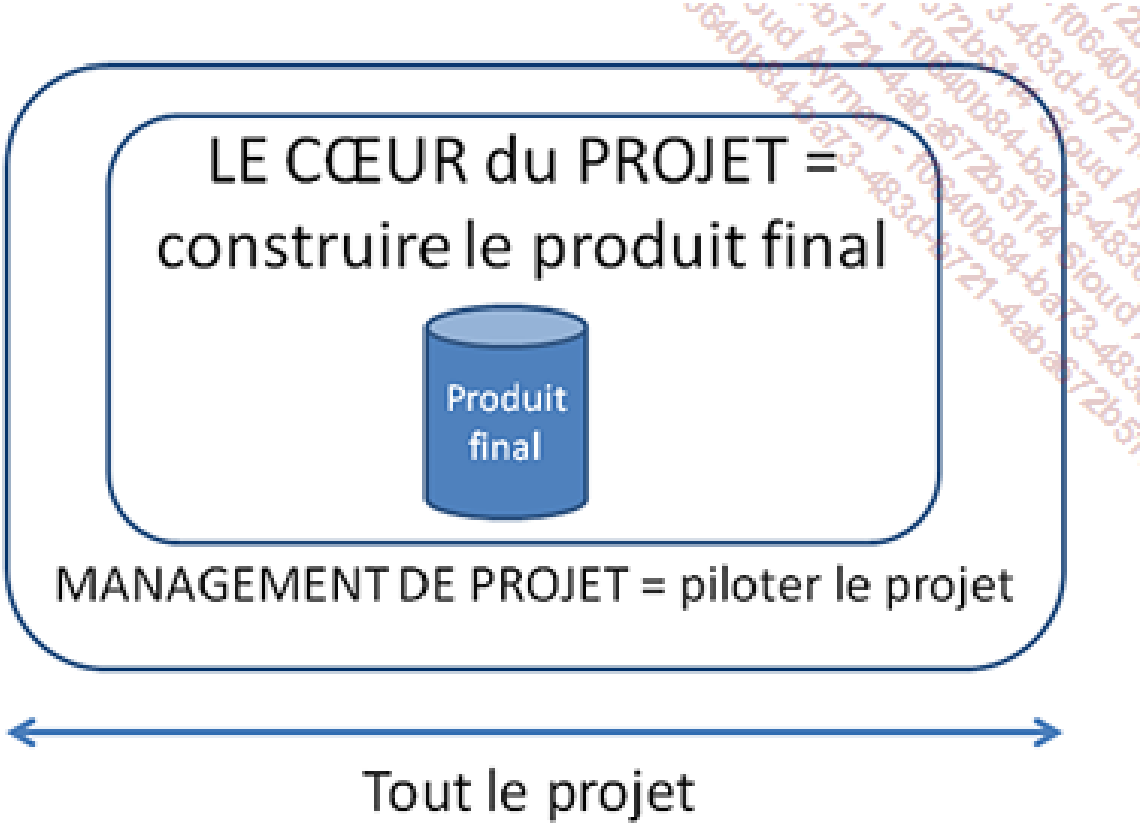
- Grand Dictionnaire Terminologique

- Une *réalisation* ***unique, limitée*** *dans le temps et comportant un* ***ensemble de tâches*** *cohérentes, utilisant des* ***ressources*** *humaines, matérielles et financières en vue d'atteindre les* ***objectifs*** *prévus au mandat, tout en respectant des* ***contraintes*** *particulières.*

Définitions normalisées

- ISO 10006
  - un projet est un « processus unique, qui consiste en un ensemble d'activités coordonnées et maîtrisées comportant des dates de début et de fin, entrepris dans le but d'atteindre un objectif conforme à des exigences spécifiques telles que les contraintes de délais, de coûts et de ressources ».

Projet et gestion de projet



Cycle de vie – Approche standard

- Ce cycle de vie standard se compose des phases suivantes :
  - étude de faisabilité (analyse)
  - définition des solutions (objectif à atteindre)
  - conception détaillée (contrats de réalisation)
  - Réalisation (exécution des contrats)
- Plusieurs inconvénients
  - Découpage inadéquat aux systèmes d'informations
  - Impossibilité de maîtriser coûts et temps
  - Problème de faisabilité (que pouvons nous faire, ce n'est pas une réponse par oui ou par non)

Spécificités des projets informatiques

- Trois types de projets informatiques
  - Développement
    - Encadrement d'un processus de GL
    - problématique de gestion de portée et, corollairement, d'estimation de l'effort donc des coûts et de la durée donc de gestion de projets
  - Entretien
    - Encadrement d'un processus de GL
    - problématique de gestion des modifications donc de gestion des configurations
  - Exploitation
    - Encadrement d'un service
    - problématique de continuité et qualité de service

Modèle de développement / Procédé

- Il faut construire le découpage temporel en fonction des caractéristiques de l'entreprise et du projet.
- Introduction des *modèles de développement (process models) ou modèles de cycle* de vie.
- Les principaux modèles sont :
  - le modèle du *code-and-fix*
  - le modèle de la transformation automatique
  - le modèle de la cascade
  - le modèle en V
  - le modèle en W
  - le modèle de développement évolutif
  - le modèle de la spirale
  - AGILEs
  - Scrum
  - ....

Rappels

- Les solutions suggérées pour amortir les problèmes :
  - décomposer le processus de développement, découper en sous-systèmes, se rapprocher d 'un découpage naturel proche de la réalité.
  - Unification du vocabulaire et méthode d'organisation du travail
  - Modèles pour éclaircir, récapituler et montrer les points clés des spécifications.
  - Cycle de vie itératif et incrémental, comme le cycle de vie objet et des méthodes agiles.

Agilité

L'agilité

- L'agilité est devenue la réponse à ces problèmes.
- Mais qu'est-ce que l'agilité exactement ?
- Et comment peut-elle transformer non seulement la manière dont nous travaillons, mais aussi la manière dont nous pensons et interagissons avec le monde autour de nous ?

Le manifeste agile

- Insister sur l'importance de l'adaptabilité, de la collaboration et de la réponse rapide au changement.
- Mettre l'humain au centre de chaque décision et reconnaît que chaque projet est un voyage d'apprentissage.
- Ne se résume pas à suivre un ensemble de pratiques ou d'utiliser certains outils.
- C'est avant tout une transformation culturelle.
- Exige une remise en question profonde des méthodes de travail, des valeurs et des principes qui guident nos actions.

Qu'est-ce le développement agile

(https://www.agilealliance.org/agile101/)

- Agile software development is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the **Manifesto for Agile Software Development** and the **12 Principles** behind it. When you approach software development in a particular manner, it's generally good to live by these values and principles and use them to help figure out the right things to do given your particular context.

Une mentalité avant tout

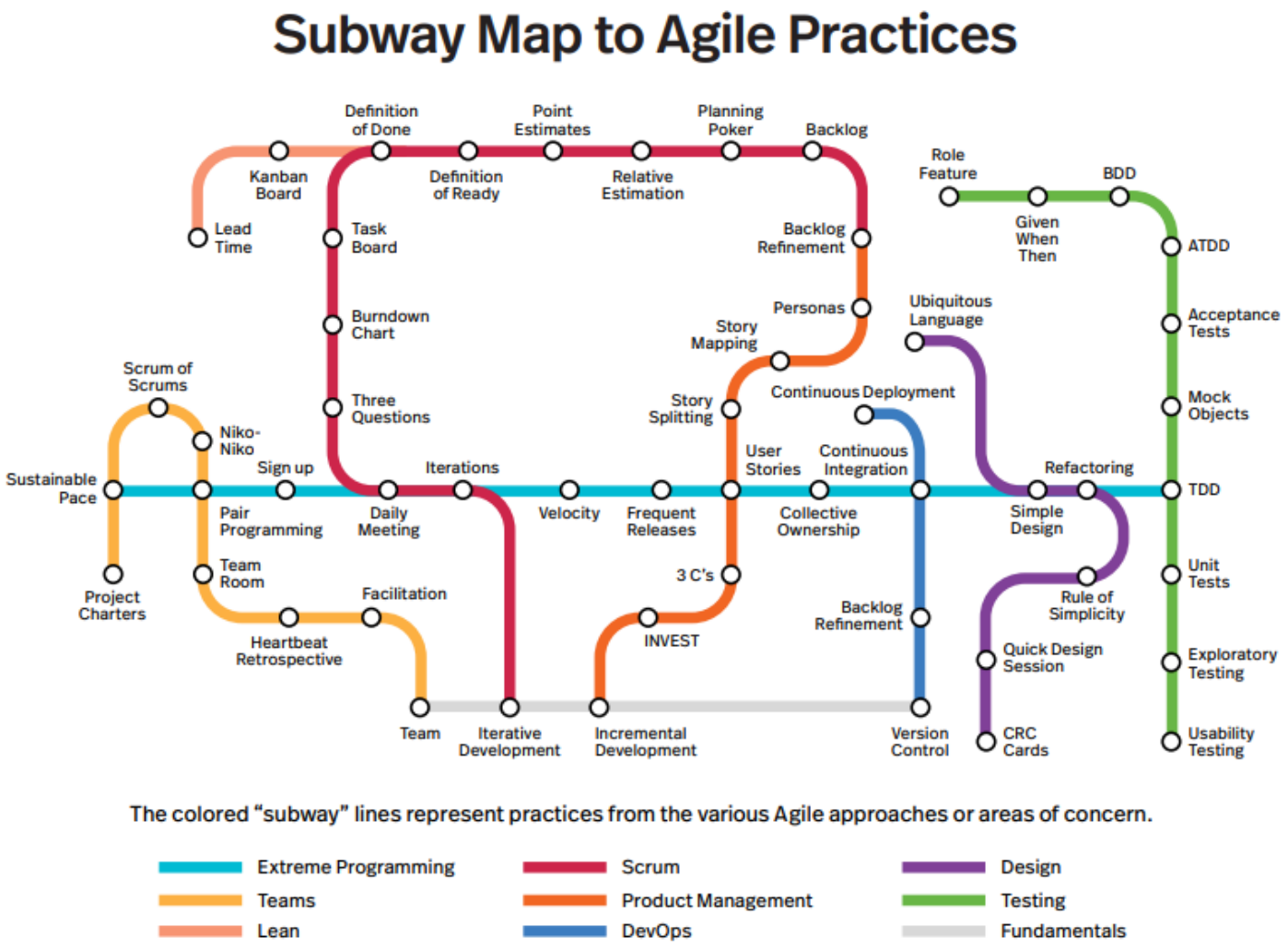
- En 2001
- Groupe de développeurs discutant des défis
- Manifeste agile
  - Un ensemble de valeurs et de principes qui défient les méthodologies traditionnelles de développement et offrent une nouvelle perspective sur la manière dont le travail devrait être accompli.

- Le Manifeste agile, dans sa simplicité, met en avant quatre valeurs clés :

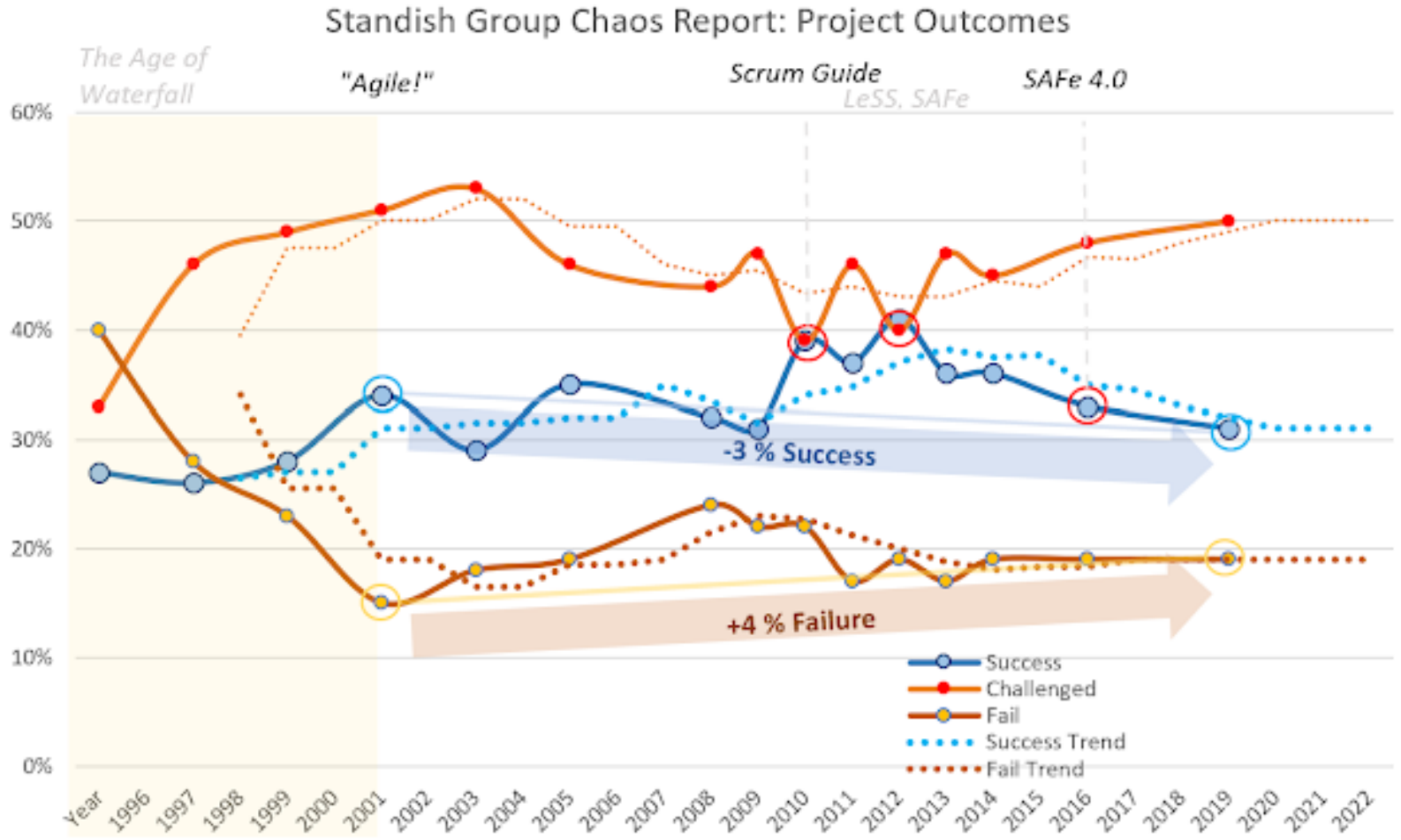
  - les individus et leurs interactions plus que les processus et les outils ;
  - un logiciel fonctionnel plus qu'une documentation exhaustive ;
  - la collaboration avec le client plus que la négociation contractuelle ;
  - l'adaptation au changement plus que le suivi d'un plan.

Méthodes agiles, différentes pratiques…

(https://www.agilealliance.org/agile101/subway-map-to-agile-practices/)



L’agilité a-t-elle tout réglée ?



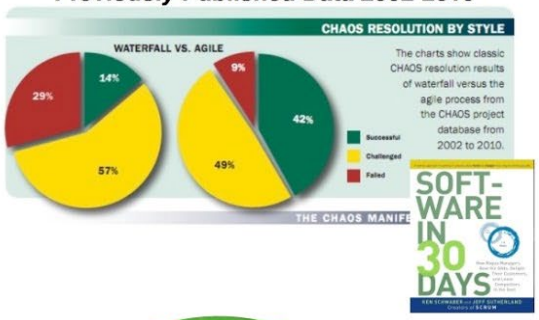
Agilité vous dites …

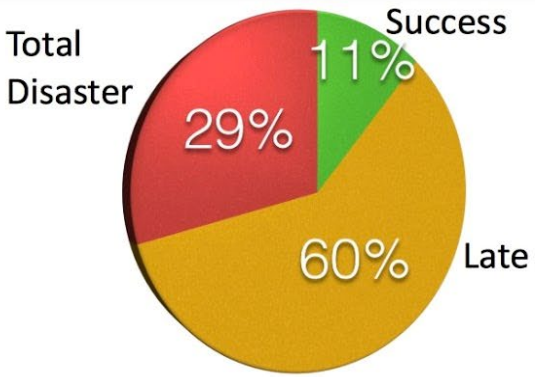
61% of "Agile" Projects are Agile in Name Only

Chaos Report 2015, Standish Group International, Inc.

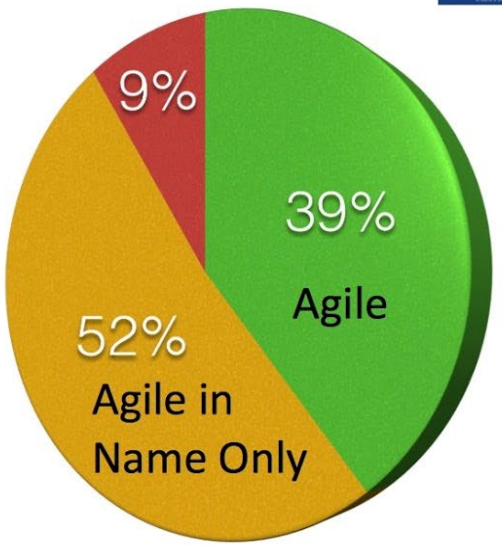
SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

Previously Published Data 2002-2010



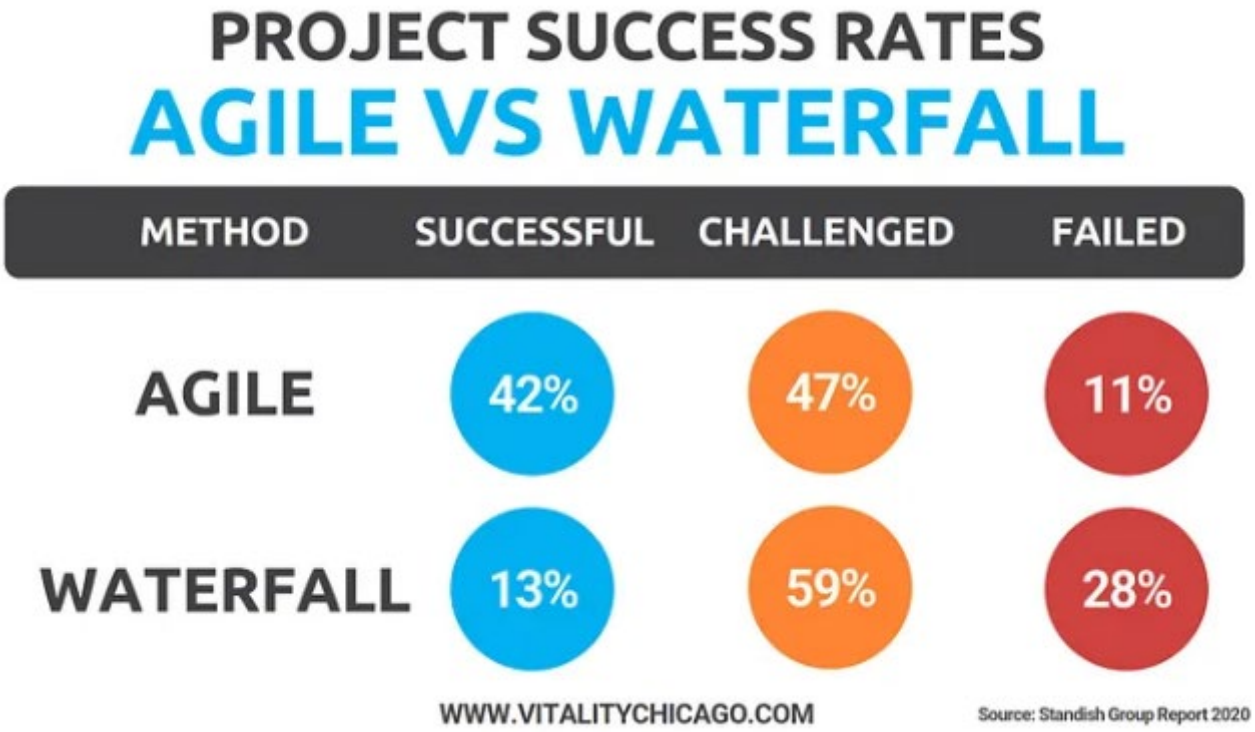


Waterfall 2011-2015 Data



Agile 2011-2015 Data

Encore des échecs



Si c'était aussi facile ☺

- Grands défis dans la gestion de projets, en particulier dans le domaine du développement logiciel
  - L'incertitude intrinsèque

- Naviguer dans l'inconnu
- Anticiper et agir

Objectifs

- Comprendre l'importance d'une bonne gestion de projets
- Développer la compréhension de la méthodologie Agile et du cadre de référence Scrum.
- Connaître les rôles, artéfacts et rituels Scrum.
- Comprendre la planification, le découpage et l'estimation des coûts de projet avec la méthode Agile.
- S'initier à la gestion d'équipes.
- Parfaire ses habiletés de communication et de résolution de conflits.
- Se familiariser avec la gestion de risques.

Plan de cours