Agile

Méthodologies

CONSULTING BSTORM

Méthodologies

Qu'est ce qu'une méthodologie ?

Cadre établi afin de

1. **structurer**
2. **planifier**
3. **contrôler**

le processus de développement d'un système d'information.

Les disciplines du développement de software

1. **Analyse des exigences**
   - Comprendre les besoins du client.
2. **Conception** (Design)
   - Définir la solution technique.
3. **Développement**
   - Implémenter la solution.
4. **Validation** (Testing)
   - S'assurer que la solution répond adéquatement aux besoins.
5. **Déploiement**
   - Intégration globale et mise en production.
6. **Maintenance**

Différentes méthodologies

Les différences entre différentes méthodologies de développement résident essentiellement dans :

1. L'importance donnée à chaque activité.
2. La séquence permise entre chaque activité.

Large variété de méthodologie: en cascade, en spirale, incrémentale, agile...

Waterfall

Fordisme

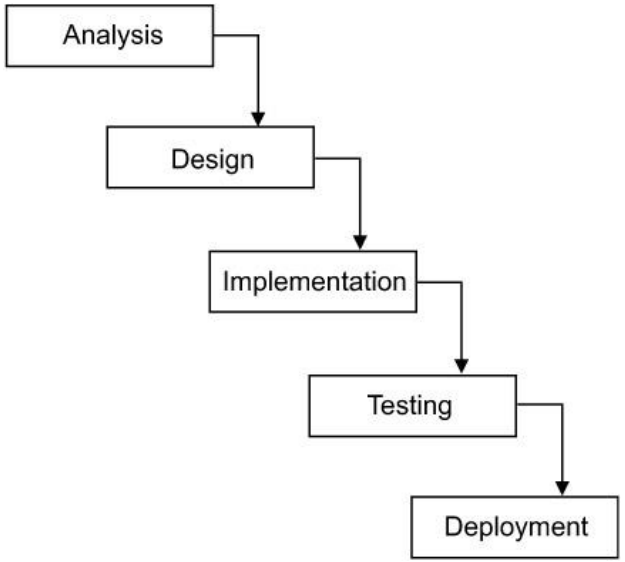
Au début des années 1900, **Henry Ford** a popularisé le fait que pour construire des produits complexes, il faut que les travailleurs se spécialisent. Ford révolutionna le secteur de l'automobile en créant le **Fordisme** où le travail est dit séquentielle.

- Augmentation de la productivité.
- Très efficace pour les produits ou services standardisés.

Méthode en cascade - Waterfall

Depuis les années 70, le secteur de l'IT utilise la méthode dite en cascade qui se caractérise par des **phases séquentielles**, qui se succèdent après la validation des livrables produits lors de la phase précédente.

- Chaque phase dépend des **livrables** de la phase précédente et correspond à une **spécialisation** des tâches.
- Le chef de projet de s'engager sur un **planning détaillé de réalisation**, prévoyant les jalons de début et fin de phases, et les activités à mener.



Réflexion

1. Quels sont les **avantages** et les **inconvénients** d'une telle méthode?
2. Quels sont les **critères favorable** / **défavorable** pour cette méthode ?

Avantages

1. **Approche simple**
   - Facile de créer planning et budget pour des tâches séquentielles.
   - Facile à suivre. Il suffit de suivre le planning et les instructions des phases précédentes.
2. **Economies d'échelle**
   - La spécialisation des équipes et le travail séquentielle permet des économies d'échelle.
3. **Pérennisation de la connaissance du projet**
   - Comme les équipes ne travaillent pas ensemble, il est nécessaire de fournir une documentation aussi importante que le code.

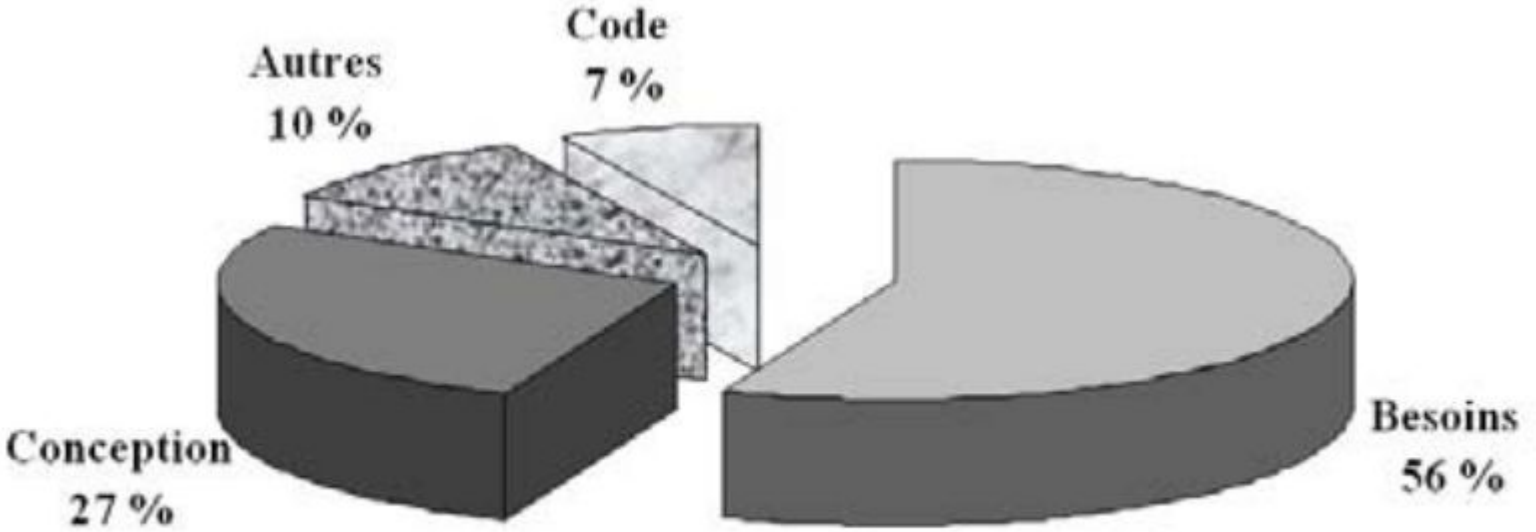
Inconvénients

1. **Rigidité de l'approche prédictive**
   - Pas toujours possible au début de projet d'anticiper tous les éléments d'un projet.
   - Difficulté à respecter le planning et budget.
   - La préoccupation majeure du chef de projet devient alors de coller au plus près au plan car tout retard ou imprévu est perçu comme échec, incompétence.

2. **L'effet tunnel**
   - Peu de possibilité pour un client de donner un feedback. Lorsqu'il peut le faire, il est souvent trop tard pour réaliser une modification. Hors il est souvent très difficile pour un client de décrire correctement son besoin du 1er coup.

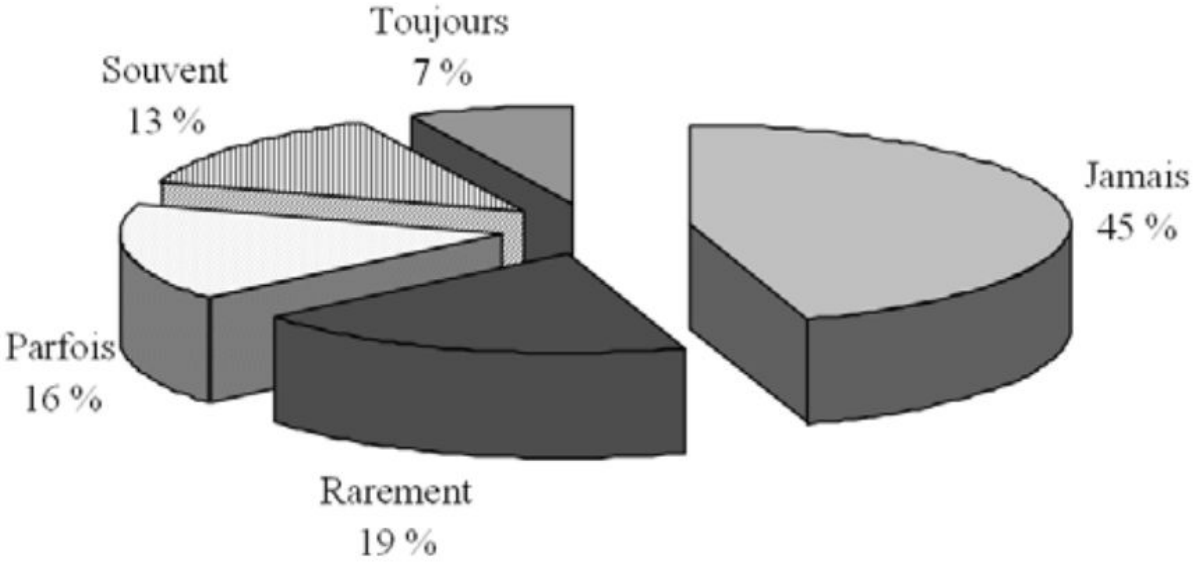
3. **Difficile d'apporter des modifications en cours de projet**
   - Il faut retourner dans les phases antérieures où les équipes sont probablement affecté à un autre projet.

Approche prédictive : origines des défauts logiciels



Source : J Johnson, Keynote speech, XP2002 (Sardaigne, Italie)

L'effet tunnel : % de fonctionnalités utilisées



Source : J Johnson, Keynote speech, XP2002 (Sardaigne, Italie)

Inconvénients

4. **Une mauvaise communication**
   - Peu importe la méthodologie, il y a toujours de problèmes de communication entre toutes les parties prenantes d'un projet. Puisque les équipes ne travaillent pas ensemble, ces difficultés sont exacerbées.
5. **L'implication tardive des développeurs**
   - Les développeurs n'interviennent que très tard dans les projets. Il est impossible pour eux d'apporter des modifications dans la conception.
6. **La levée tardive des facteurs à risques**
   - Les phases d'intégration, de tests et de présentation au client ne se font qu'à la fin du projet.
7. **Une documentation pléthorique**
   - On passe parfois plus de temps à documenter l'application qu'à la développer.

Exigences pour la méthode en cascade

1. Les exigences sont **claires** et correctement **définies**.
2. L'environnement de travail **stables**.
3. La technologie est bien **connue** et **mature**.
4. Il n'y a rien de nouveau ou d'inconnu dans le projet (**prévisibilité**).
5. De nombreux projets semblables ont déjà été exécutés avant.
6. Le projet est court (quelques mois maximum).

Qui utilise ce genre de méthode ?

- Institutionnel
- Banque / Assurance
- Aéronautique
- ….

Chaos Report

Quelles sont les chances de réussites d'un projet ?

Quelles sont les chances et les facteurs de réussites d'un projet ? Pour comprendre le taux de réussite des projets on va se baser sur le **Chaos Report de 2015**.

- Source
  - https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf

Taux de succès

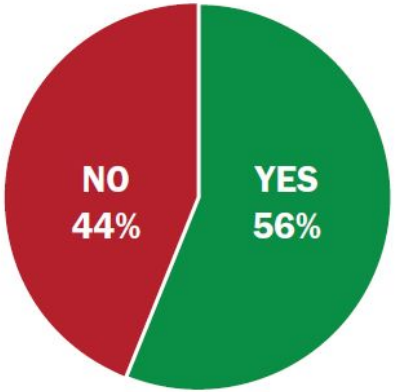
MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

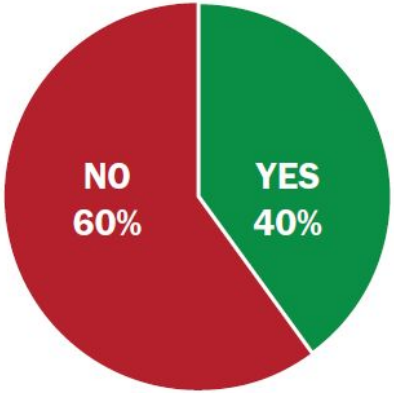
Respect des contraintes

ONBUDGET



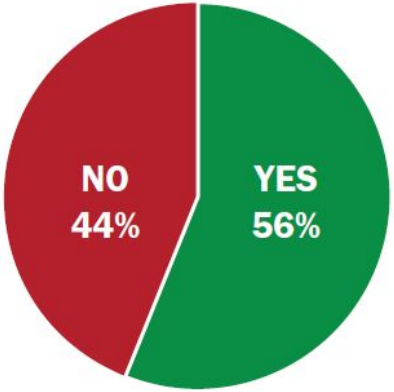
The percentage of projects that were OnBudget from FY2011–2015 within the new CHAOS database.

ONTIME



The percentage of projects that were OnTime from FY2011–2015 within the new CHAOS database.

ONTARGET



The percentage of projects that were OnTarget from FY2011–2015 within the new CHAOS database.

La taille du projet

PROJECT SIZE BY CHAOS RESOLUTION

	SUCCESSFUL	CHALLENGED	FAILED	TOTAL
Grand	6%	51%	43%	100%
Large	11%	59%	30%	100%
Medium	12%	62%	26%	100%
Moderate	24%	64%	12%	100%
Small	61%	32%	7%	100%

The size of software projects by the Modern Resolution definition from FY2011–2015 within the new CHAOS database.

CHAOS RESOLUTION BY PROJECT SIZE

	SUCCESSFUL	CHALLENGED	FAILED
Grand	2%	7%	17%
Large	6%	17%	24%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
TOTAL	**100%**	**100%**	**100%**

The resolution of all software projects by size from FY2011–2015 within the new CHAOS database.

Par industrie

CHAOS RESOLUTION BY INDUSTRY

	SUCCESSFUL	CHALLENGED	FAILED
Banking	30%	55%	15%
Financial	29%	56%	15%
Government	21%	55%	24%
Healthcare	29%	53%	18%
Manufacturing	28%	53%	19%
Retail	35%	49%	16%
Services	29%	52%	19%
Telecom	24%	53%	23%
Other	29%	48%	23%

The resolution of all software projects by industry from FY2011–2015 within the new CHAOS database.

Par région du monde

CHAOS RESOLUTION BY AREA OF THE WORLD

	SUCCESSFUL	CHALLENGED	FAILED
North America	31%	51%	18%
Europe	25%	56%	19%
Asia	22%	58%	20%
Rest of World	24%	55%	21%

The resolution of all software projects from FY2011–2015 by the four major areas of the world.

Par complexité

CHAOS RESOLUTION BY COMPLEXITY

	SUCCESSFUL	CHALLENGED	FAILED
Very Complex	15%	57%	28%
Complex	18%	56%	26%
Average	28%	54%	18%
Easy	35%	49%	16%
Very Easy	38%	47%	15%

The resolution of all software projects by complexity from FY2011–2015 within the new CHAOS database.

Par méthodologie

CHAOS RESOLUTION BY AGILE VERSUS 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 all 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.

Agile

Nouvelles méthodologies

Dans les années 90, plusieurs méthodologies ont été créées pour améliorer la productivité et la qualité du travail fournis par les équipes IT.

- XP programming.
- Feature-driven Development.
- Kanban.
- Lean.
- Scrum.
- Etc.

Réflexion

Comment pourrait-on modifier le processus waterfall afin de l'améliorer ?

Empirisme

Agile est fondé sur l'**Empirisme**.

- **La connaissance provient exclusivement de l'expérience.**
- On prend des décisions en fonction de ce que l'on sait pas sur ce que l'on croit savoir.
- Agile propose une approche itérative et incrémentale pour optimiser la prévisibilité et contrôler les risques.

Les 3 piliers de l'Empirisme

1. **Inspection**
    a. Remise en question constante sur le projet et le process.
2. **Adaption**
    a. Modifier les plans pour faire face à ce que l'on a appris de nouveau durant nos inspections.
3. **Transparence**
    a. Partager ouvertement les informations nécessaires.

Incrémentation et itération

Incrémentation

Le développement va être découpé en plusieurs parties. Chacun des développements vient enrichir l'existant. Un incrément est donc une avancée dans le processus de développement.

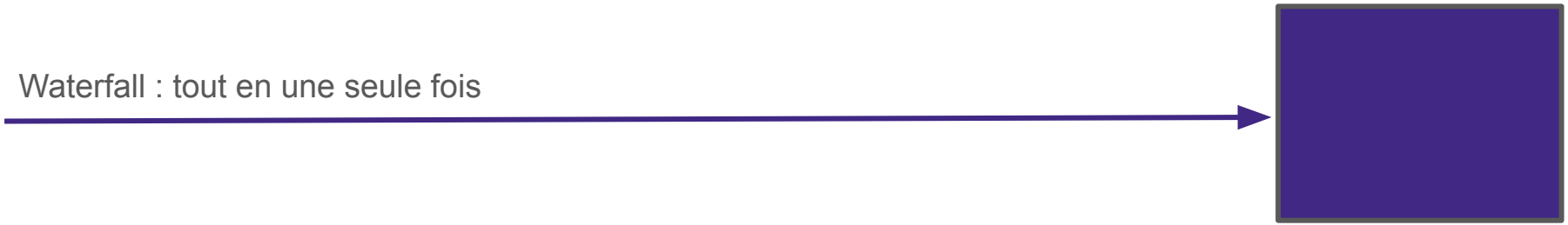
Itération

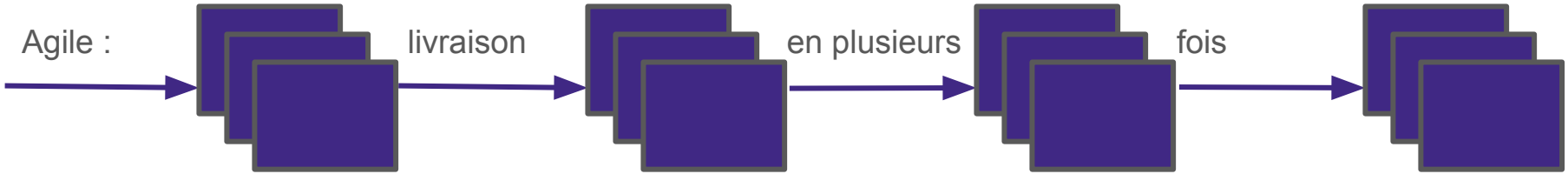
A chaque itération, les parties prenantes ont l'occasion de faire des commentaires sur l'augmentation la plus récente, ainsi que sur le produit dans son ensemble. Ce retour sera incorporé dans la version suivante.

Processus Incrémental

Processus Incrémental

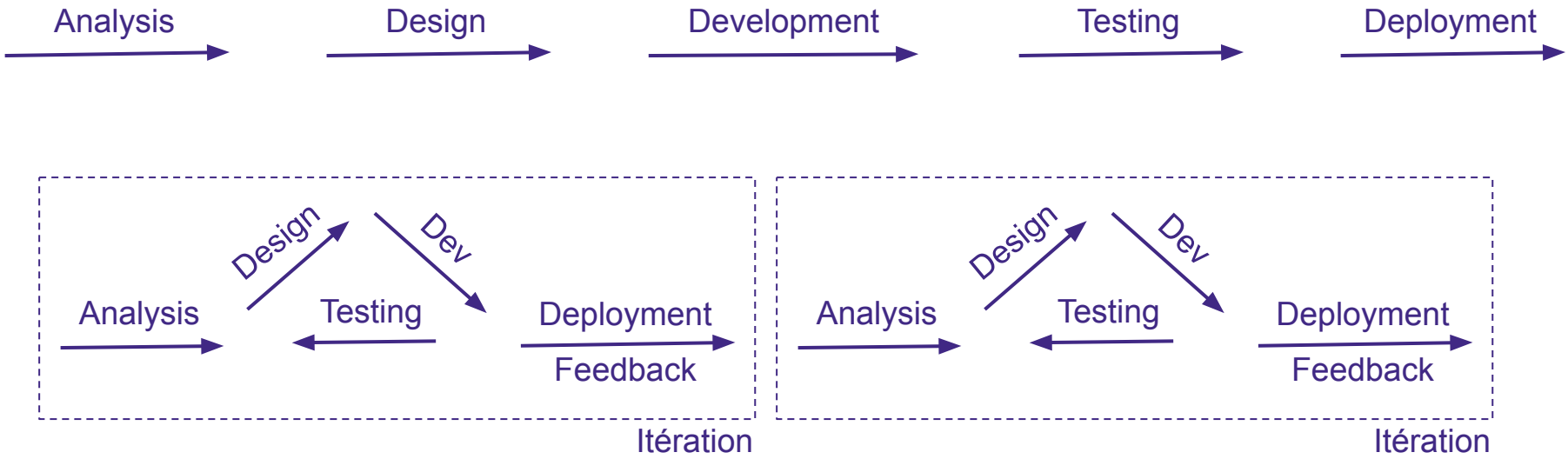
Waterfall : tout en une seule fois





Processus Itératif

Processus Itératif



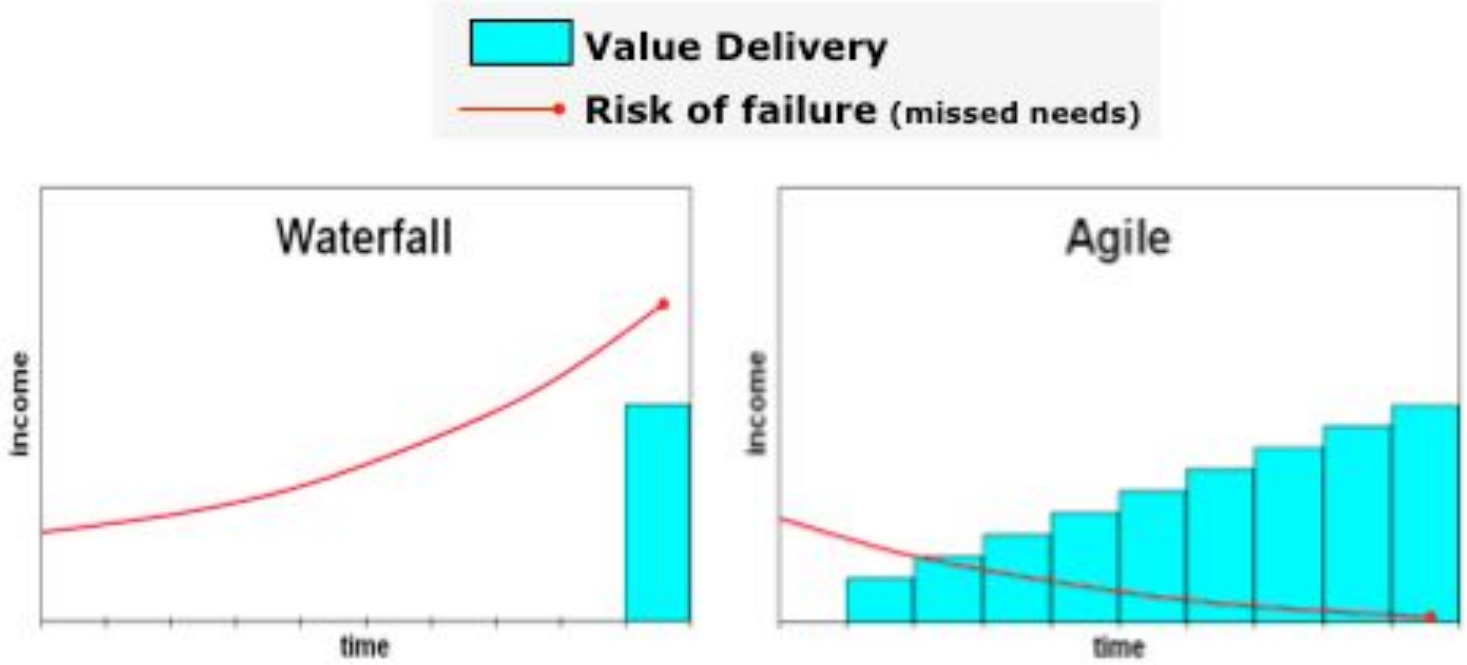
Avantages de releases fréquentes

1. **La communication est de meilleure qualité**
   - L'utilisateur a la possibilité de clarifier ses exigences au fur et à mesure.
   - Malentendus, incompréhensions, incohérences sont mis en évidence très tôt dans le projet. Il est donc encore possible de les corriger.
2. **Plus de transparence**
   - Permet de montrer l'avancement du projet au fur et mesure.
   - Le client reçoit des « preuves » tangibles de l'avancement du projet.
3. **Risque réduit**
   - Créer un produit prêt à être expédié à tout moment.
   - Les tests sont effectués à chaque itération, les anomalies détectées sont corrigées au fur et à mesure.

Avantages de releases fréquentes

4. **Meilleure valeur commerciale**
   - Produit des revenus dès le début.
   - Récupérer les coûts plus tôt.
   - Diminuer l'investissement global.
5. **Les coûts sont contrôlés**
   - Les coûts sont limités au périmètre de l'itération.
   - On ne perd que les efforts de cette itération et non la totalité du projet.
   - On peut arrêter à n'importe quelle moment, puisqu'on a une version finie à la fin de chaque itération.
6. **Possibilité d'exploiter méthodiquement les leçons tirées d'une itération**
7. **L'équipe prend confiance**

Avantages de releases fréquentes



Le Manifeste Agile

Le Manifeste Agile

En février 2001, un groupe de 17 développeurs s'est réuni à Snowbird (Utah) pour discuter de la possibilité de créer une meilleure méthode de développement.

Cette réunion a débouché sur un document intitulé **The Agile Manifesto** (Le Manifeste Agile), qui décrit 4 valeurs fondamentales et 12 principes pour améliorer la manière dont on construit des softwares.

- https://agilemanifesto.org/

Mythe sur l’origine

Une idée fausse commune est que le Manifeste Agile a été écrit, puis les méthodologies Agiles spécifiques en ont toutes été dérivées. En fait, c'était le contraire.

Au moment de la rédaction du manifeste, plusieurs méthodologies Agiles différentes avaient déjà commencé à prendre forme et à gagner du terrain auprès des équipes de développement de logiciels.

Le Manifeste était une tentative pour unir les partisans de ces méthodologies afin que les idées puissent être partagées et qu’ils puissent avancer ensemble.

4 Valeurs Fondamentales

1. Priorité aux personnes et aux interactions par rapport aux procédures et aux outils
    - Travail en groupe, communication avec les clients plutôt que de travailler uniquement à partir d'un outil de gestion de projet.
2. Priorité aux applications opérationnelles par rapport à une documentation pléthorique
    - Qui lit des centaines de pages de documentation ?
    - Souvent obsolète avant la fin du projet.
    - L'objectif est d'avoir un software qui fonctionne, pas une documentation.
    - Se limiter à la documentations succinctes à jour, documentation permanente du code.

4 Valeurs Fondamentales

3. Priorité à la collaboration avec le client par rapport à la négociation de contrat
   - Etablir une relation de confiance avec le client.
   - Feedback régulier du client, solution répondant réellement aux attentes.
4. Priorité de l'acceptation du changement par rapport à la planification
   - S'adapter au fur et à mesure de l'avancement du développement.
   - Planning flexible, modifications possibles après 1ère version du système.

12 Principes

1. Prioriser la satisfaction du client
   - La plus grande priorité est de satisfaire le client en lui livrant très tôt et régulièrement des versions fonctionnelles de l'application source de valeur.
   - Le client peut décider de la mise en production de l'application.
2. Accepter les changements
   - Accueillir les demandes de changement à bras ouverts, même tard dans le processus de développement. Les méthodologies agiles exploitent les changements pour apporter au client un avantage concurrentiel.
   - Produire des systèmes flexibles.

12 Principes

3. Livrer en permanence des versions opérationnelles de l'application
   - Livrer le plus souvent possible des versions opérationnelles de l'application, avec une fréquence comprise entre deux semaines et deux mois, avec une préférence pour l'échelle de temps la plus courte.
   - Objectif : livrer une application qui satisfasse aux besoins du client.
4. Coopérer quotidiennement
   - Clients et développeurs doivent coopérer quotidiennement tout au long du projet.
5. Construire des projets autour d'individus motivés
   - Leur donner l'environnement et le support dont ils ont besoin et leur faire confiance pour remplir leur mission.

12 Principes

6. Favoriser le dialogue direct
   - La méthode la plus efficace pour communiquer des informations à une équipe et à l'intérieur de celle-ci reste la conversation en face à face.
7. Mesurer l'avancement du projet en fonction de l'opérationnalité du produit
   - Le fonctionnement de l'application est le premier indicateur d'avancement du projet.
8. Adopter un rythme constant et soutenable
   - Adapter le rythme pour préserver la qualité du travail sur la durée du projet.
   - Développeurs et utilisateurs devraient pouvoir maintenir un rythme constant indéfiniment.

12 Principes

9. Contrôler continuellement l'excellence technique et à la conception
    - Maintenir le code source propre, clair et robuste.
10. Privilégier la simplicité en évitant le travail inutile
    - La simplicité, art de maximiser la quantité de travail à ne pas faire, est essentielle.
    - Répondre le + simplement aux besoins actuels pour que celui ci soit adaptable.
11. Auto-organiser et responsabiliser les équipes
    - Partage des responsabilités par volontariat.

12 Principes

12. Améliorer régulièrement l'efficacité de l'équipe en ajustant son comportement
    - A intervalles de temps réguliers, l'ensemble de l'équipe s'interroge sur la manière de devenir encore plus efficace, puis ajuste son comportement en conséquence.
    - Environnement en perpétuelle évolution.

Une philosophie

Observations

Agile est une philosophie de travail.

1. Agile n'est pas normatif
   - Décrit un résultat spécifique, mais pas comment l'équipe doit parvenir à ce résultat.
   - Pourquoi ? Chaque situation est différente.
2. Agile ne prétend pas être la seule solution
   - Priorité à… par rapport à…
   - Agile n’est pas la seulement manière de faire, mais souvent la meilleur.

Quelle méthodologie choisir ?

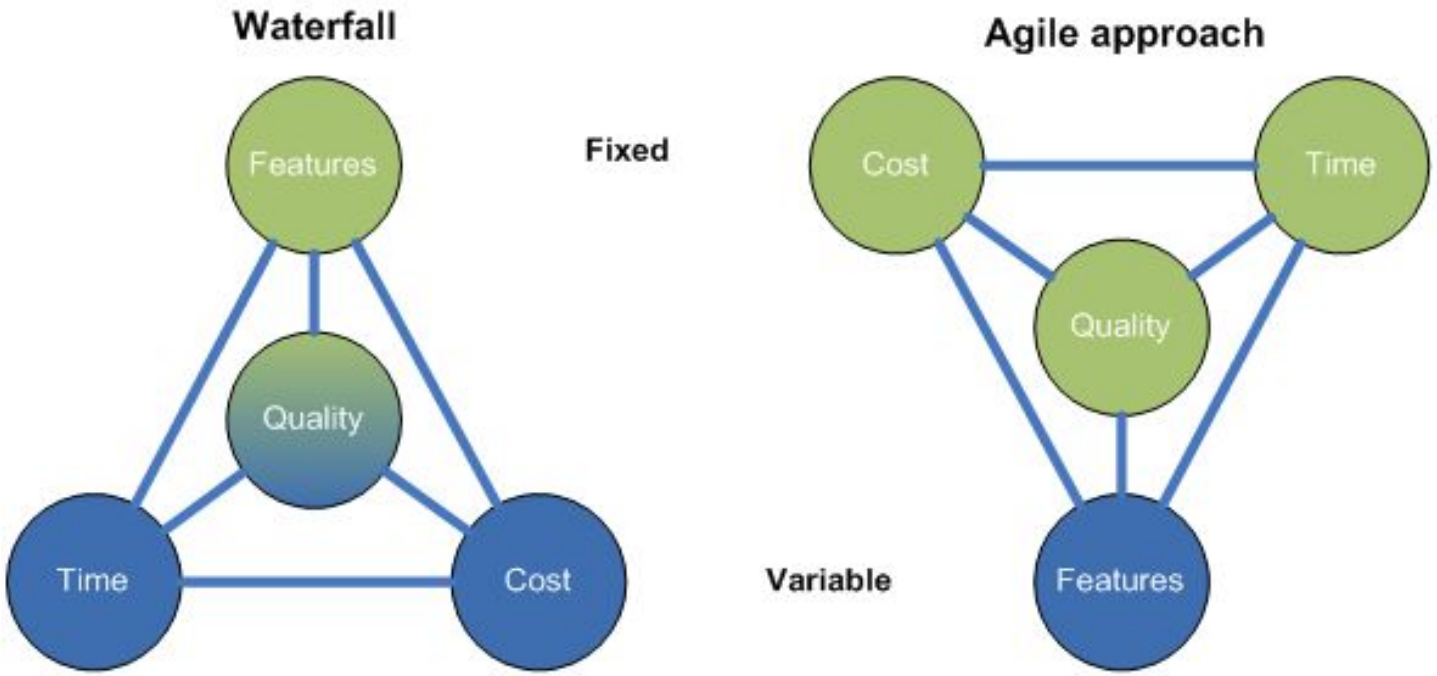
Comparaison

Cycle de vie	Waterfall	Agile
Planification	Prédictive	Adaptative
Documentation	Produite en quantité	Réduite au strict nécessaire
Équipe	Ressources spécialisées	Responsabilisation, initiative et communication
Qualité	Contrôle à la fin du cycle	Contrôle qualité précoce et permanent
Changement	Opposition au changement	Intégré dans le processus

Comparaison

Cycle de vie	Waterfall	Agile
Suivi de l'avancement	Mesure de la conformité aux plans initiaux	Travail restant à faire
Gestion des risques	Processus distinct	Intégré dans le processus
Mesure du succès	Respect des engagements initiaux	Satisfaction client

Comparaison



Quelle méthodologie choisir ?

Agile n'est pas la solution à tous les problèmes, ni la méthode la plus adaptés pour toutes les équipes. Convient pour les projets :

1. Complexes.
2. Avec des deadlines courtes.
3. Avec un risque élevé (inconnue / innovation).

Exercice

Exercice

Déterminez quelle serait la meilleure méthode entre waterfall et agile.

Justifiez votre réponse.

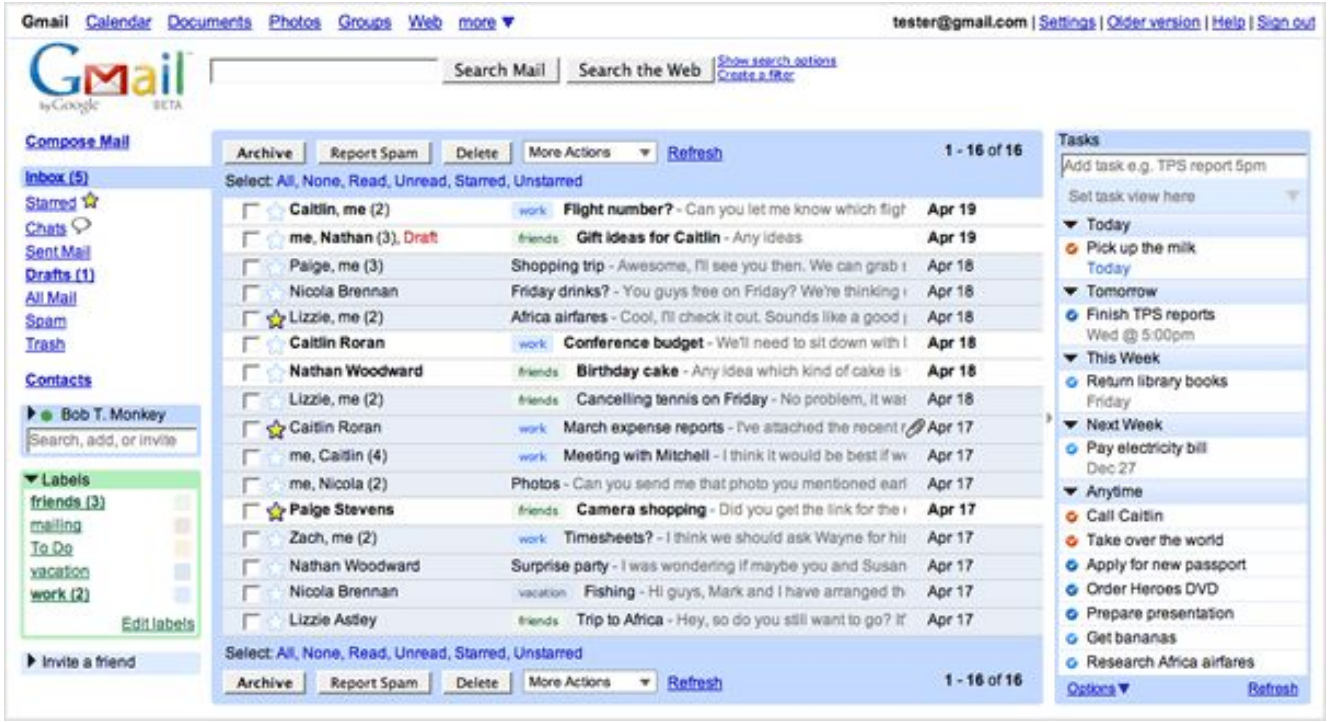
Système de navigation A380

Système de navigation A380

Waterfall

- Airbus dispose de la conception complète.
  - Ils savent ce qu'il veulent.
- Taille équipe.
- Difficulté de mettre en place un système itératif.

Gmail


Gmail Calendar Documents Photos Groups Web more ▼
tester@gmail.com | Settings | Older version | Help | Sign out
Gmail by Google BETA
Search Mail
Search the Web
Show search options
Create a filter
Compose Mail
Inbox (5)
Starred
Chats
Sent Mail
Drafts (1)
All Mail
Spam
Trash
Contacts
Bob T. Monkey
Search, add, or invite
Labels
friends (3)
mailing
To Do
vacation
work (2)
Edit labels
Invite a friend
Archive
Report Spam
Delete
More Actions
Refresh
1 - 16 of 16
Select: All, None, Read, Unread, Starred, Unstarred
Caitlin, me (2) work Flight number? - Can you let me know which flig Apr 19
me, Nathan (3), Draft friends Gift ideas for Caitlin - Any ideas Apr 19
Paige, me (3) Shopping trip - Awesome, I'll see you then. We can grab Apr 18
Nicola Brennan Friday drinks? - You guys free on Friday? We're thinking Apr 18
Lizzie, me (2) Africa airfares - Cool, I'll check it out. Sounds like a good Apr 18
Caitlin Roran work Conference budget - We'll need to sit down with Apr 18
Nathan Woodward friends Birthday cake - Any idea which kind of cake is Apr 18
Lizzie, me (2) friends Cancelling tennis on Friday - No problem, it wa Apr 18
Caitlin Roran work March expense reports - I've attached the recent Apr 17
me, Caitlin (4) work Meeting with Mitchell - I think it would be best if w Apr 17
me, Nicola (2) Photos - Can you send me that photo you mentioned earl Apr 17
Paige Stevens friends Camera shopping - Did you get the link for the Apr 17
Zach, me (2) work Timesheets? - I think we should ask Wayne for hi Apr 17
Nathan Woodward Surprise party - I was wondering if maybe you and Susan Apr 17
Nicola Brennan vacation Fishing - Hi guys, Mark and I have arranged th Apr 17
Lizzie Astley friends Trip to Africa - Hey, so do you still want to go? It Apr 17
Tasks
Add task e.g. TPS report 5pm
Set task view here
Today
Pick up the milk
Today
Tomorrow
Finish TPS reports
Wed @ 5:00pm
This Week
Return library books
Friday
Next Week
Pay electricity bill
Dec 27
Anytime
Call Caitlin
Take over the world
Apply for new passport
Order Heroes DVD
Prepare presentation
Get bananas
Research Africa airfares
Options
Refresh

Gmail

Agile - Google adopte les valeurs Agile

- Priorité au personnes
  - Googleplex, Free friday, …
- Priorité application fonctionnelle
  - Difficile à juger sans information supp.
- Priorité à la satisfaction du client / satisfaction cahier des charges
  - Récolte feedback utilisateurs, abandonne produit/fonctionnalité pas utilisé.
- Priorité changement
  - Fréquente release et roadmap de développement qui s'adapte par rapport à l'évolution du web.

Système de gestion intégrée du stock

Système de gestion intégrée du stock

Pas de réponse tranchée !

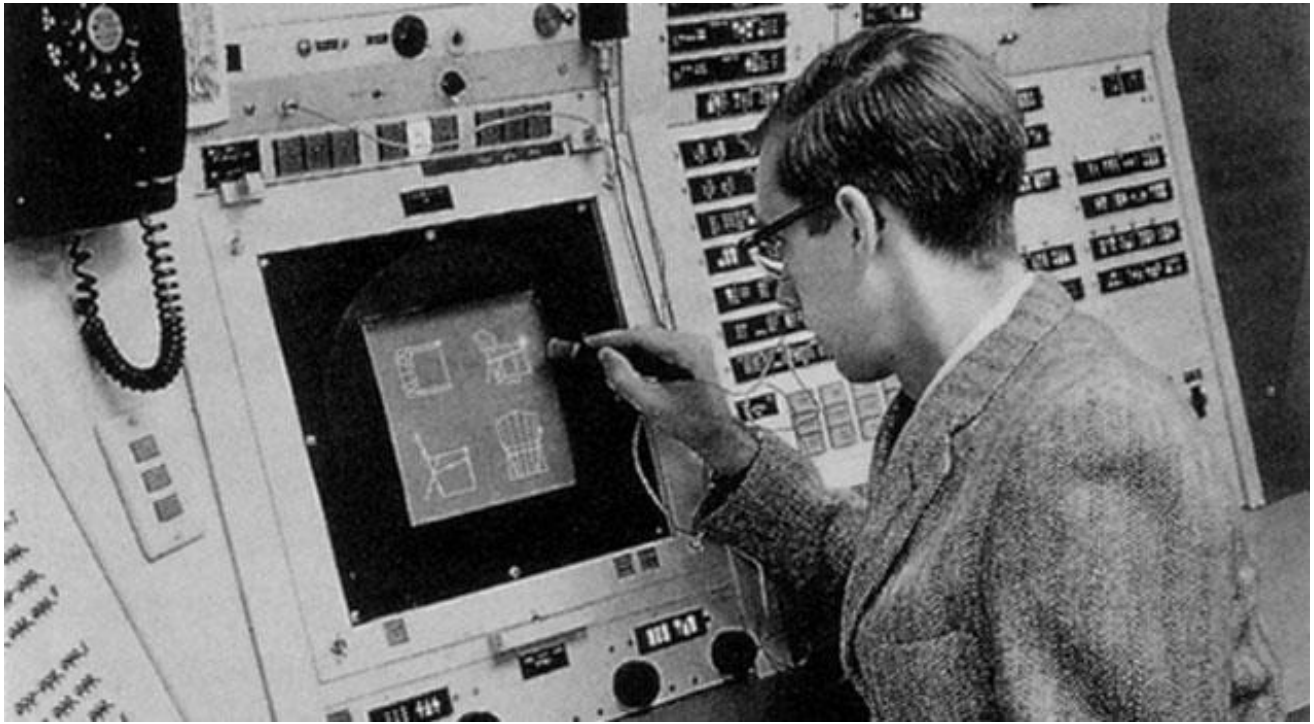
PME - Agile

- Doivent s'adapter constamment à leurs clients et fournisseurs.
- Petite équipe (agile max 10 personnes).

Carrefour - Waterfall

- Ils savent ce qu'ils veulent.
- Stable : opérationnelle depuis des dizaines d'années.
- Grande équipe.

ARPANET (ancêtre d'Internet)



ARPANET (ancêtre d'Internet)

Historiquement - Waterfall

A refaire - Agile

- Car c'est de la recherche, on ne sait pas où l'on va...

Facebook


www.facebook.com/zuck?sk=wall
facebook
Search
Mark Zuckerberg
Works at Facebook Studied Computer Science at Harvard University Lives in Palo Alto, California Knows English, Mandarin Chinese From Dobbs Ferry, New York Born on May 14, 1984
Wall
Info
Share Profile
Report/Block This Person
Wall
RECENT ACTIVITY
"I like dangerous thoughts." on Samuel W. Lessin's status.
Mark Zuckerberg
Steve, you've done so much good for the world already. I hope you get better soon.
January 17 at 11:43am via iPhone
150 people like this.

Facebook

Agile

- Constante évolution.

Scanner médicale

Scanner médicale

- **Waterfall**
- **Agile** (si expérimental)