

# Méthodes Agiles

Joseph Gillain - CogniTIC



# Objectifs

- Introduire les participants aux méthodologies de développement dites "Agiles" et plus particulièrement à **SCRUM**

AGILE = Méthodologie  
AGILE >< Langage

# Tables des matières

1. Phase Initiale: Introduction à Agile
2. Phase de Sprint:
  - a. Itération 1 - Aperçu de SCRUM
  - b. Itération 2 - Les rôles
  - c. Itération 3 - La vision du produit
  - d. Itération 4 - Le Product Backlog
  - e. Itération 5 - Le Planning
  - f. Itération 6 - Un contrat SCRUM
3. Phase de Clotûre: Case Study

# Phase initiale

*...ou l'on compare les méthodes Agile aux méthodes traditionnelles*

# Méthodes de développement de software

Qu'est ce qu'une méthode de développement?

*Cadre établi afin de structurer, planifier et contrôler le processus de développement d'un système d'information*

# Méthodes de développement de software

Quelles sont les disciplines d'un processus de développement?

# Les disciplines du développement de software

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

# Les activités du développement de software

Les différences entre différentes méthodologies de développement résident essentiellement dans:

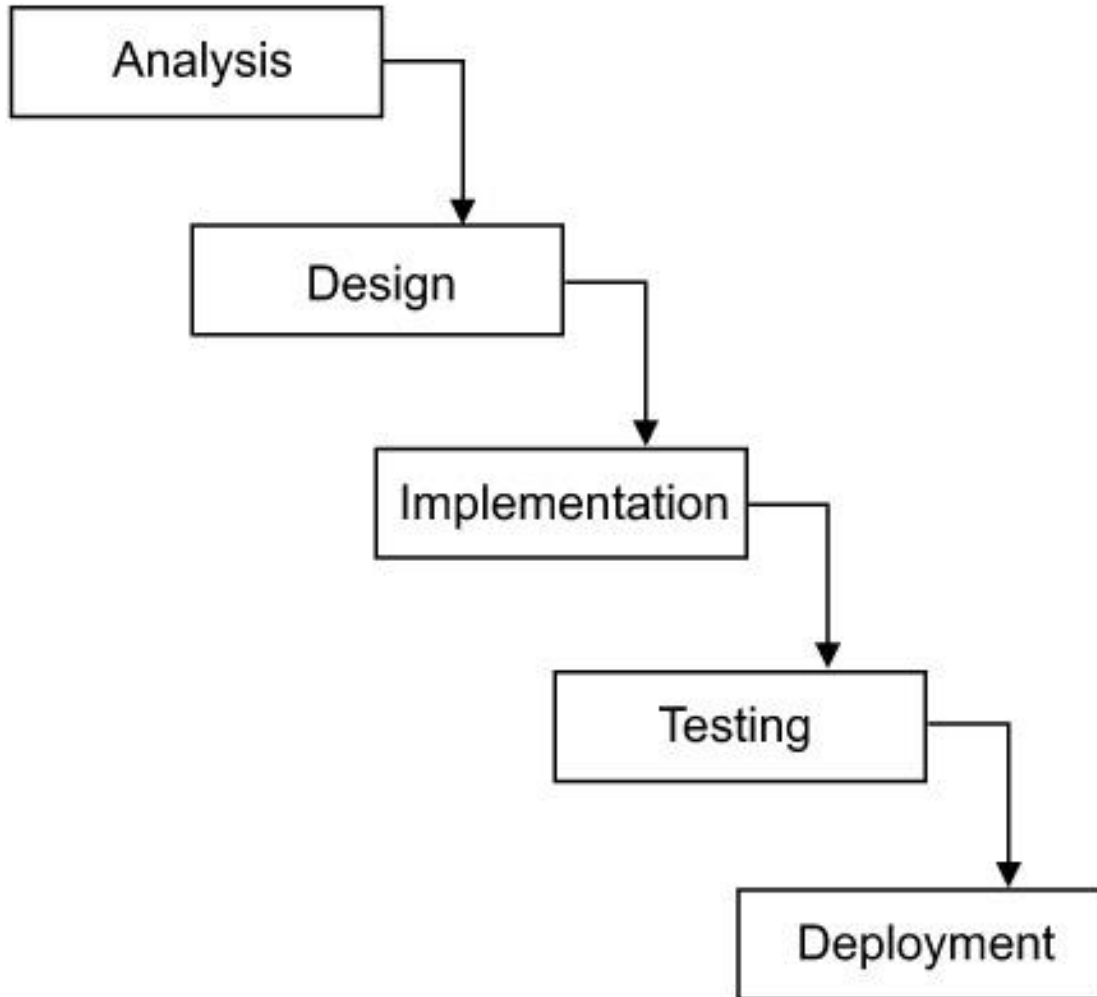
- l'importance donnée à chaque activité
- la séquence permise entre chaque activité

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



# Méthode en cascade

# Méthode en cascade



# Réflexion

*Quels sont les avantages et les inconvénients d'une telle méthode?*

# Réflexion

*Quels sont les critères  
favorable/défavorable pour cette  
méthode?*

# Avantages

1. Le temps alloué dans les premières phases des projets permet d'atteindre des **économies d'échelle**
2. La **documentation** est aussi importante que le code. Cela permet de pérenniser la connaissance au sein du projet
3. **Approche simple**, disciplinée et facile à comprendre

# Inconvénients

1. La rigidité de l'approche
2. L'effet tunnel ou "black box"
3. Une mauvaise communication
4. La levée tardive des facteurs à risques
5. Une documentation pléthorique

# Exigences pour la méthode en cascade

1. L'environnement et les exigences sont **stables**
2. La technologie est bien **connue** et **mature**
3. Il n'y a rien de nouveau ou d'**inconnu** dans le projet  
(**prévisibilité**)
4. De nombreux projets semblables ont déjà été exécutés avant

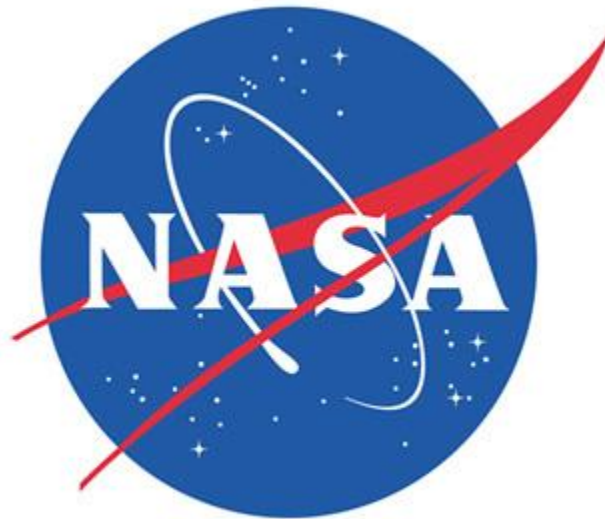
# Méthode en cascade

Qui utilise ce genre de méthode?

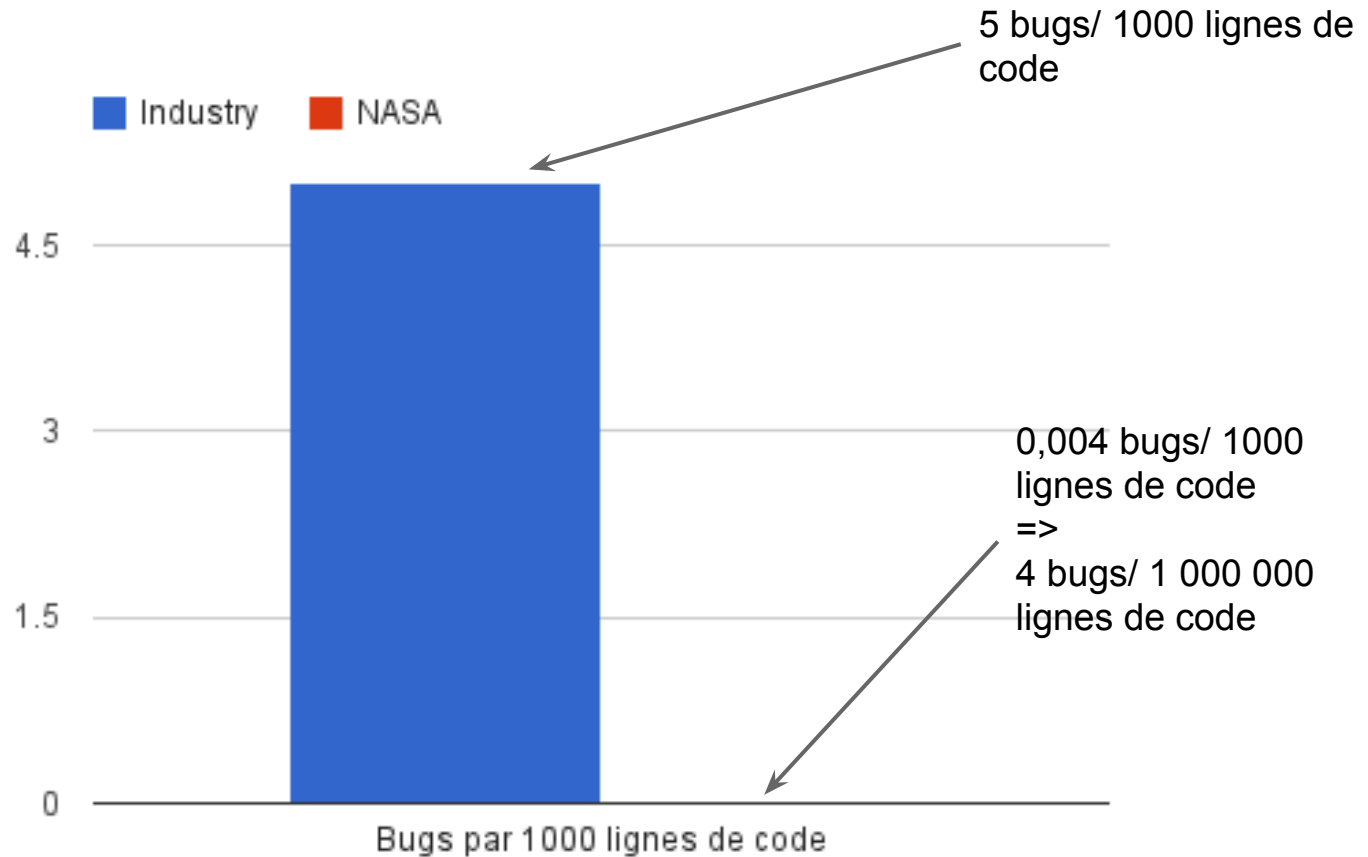


# Méthode en cascade

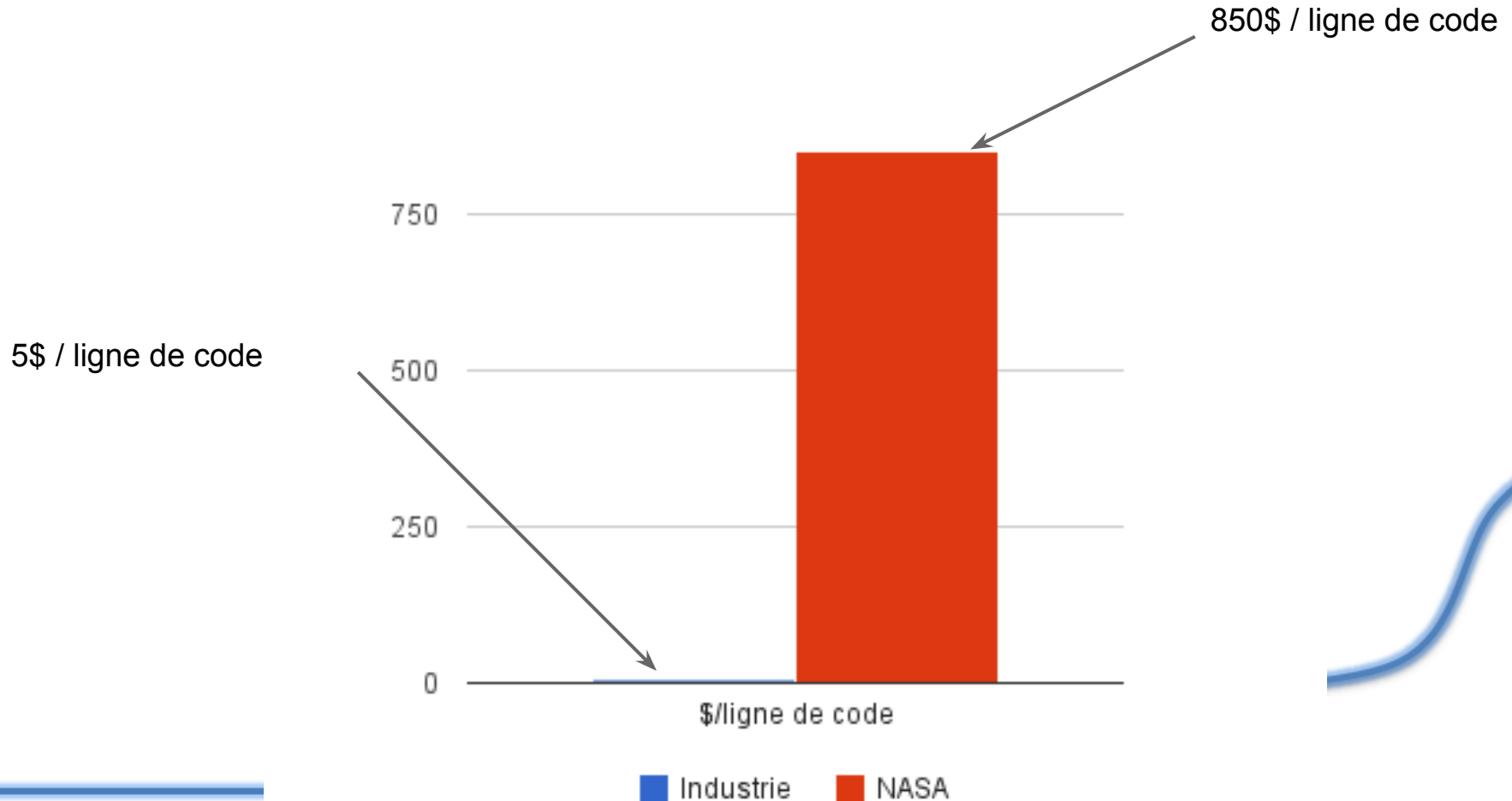
Qui utilise ce genre de méthode?



# Statistiques NASA-Soft. Eng.



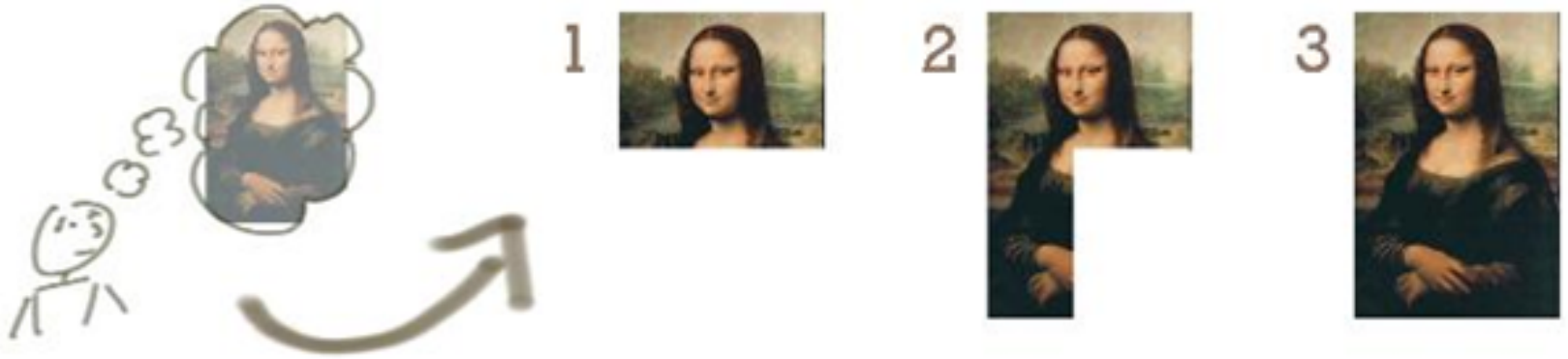
# Statistiques NASA-Soft. Eng.



# Réflexion

Comment pourrait-on  
améliorer ce processus?

# Processus Incrémental



# Processus Itératif



# **Rapid Application Development**

# Origines

Double constat:

- Manque de concertation entre informaticiens et utilisateurs  
=> *inadéquation du produit aux besoins*
- Durée des méthodes classiques inadaptées à la vitesse d'évolution des technologies



# Principes

- Fondement de base: **Communication**
- Répartition des rôles très structurée:
  - *Maîtrise d'ouvrage*: représente l'utilisateur et détermine les fonctionnalités à développer
  - *Maîtrise d'oeuvre*: apporte les solutions techniques aux problèmes posés par la maîtrise d'ouvrage
  - *Groupe d'animation* et *Rapporteur*: organise la communication du projet

# Maîtrise d'ouvrage

- Responsabilités:
  - Définition des objectifs et des exigences
  - Valider les solutions proposées et élaborées
  - Préparer et piloter le changement induit
- Acteurs:
  - **Maître d'ouvrage**  
Fixe les objectifs
  - **Coordinateur de Projet Utilisateurs ou Maître d'Ouvrage délégué**  
Assure le suivi des objectifs
  - **Responsable de la cohérence et de la qualité fonctionnelle**  
Contrôle la cohérence des décisions  
dans les domaines fonctionnels

# Maîtrise d'oeuvre

- Responsabilités:
  - Proposer et réaliser la solution
  - Livrer des "fonctionnalités"
  - Respecter les directives de qualité
- Acteurs:
  - Maître d'œuvre
  - Pilote de Projet Informatique
  - Responsable par domaine

# Interactions RAD

## Groupe d'animation et de Rapport RAD

### Animateur

- Provoque et conduit les réunions
- N'émet pas d'avis personnel
- Recadre les discussions
- Synthétise et formalise

### Rapporteur

- Produit une documentation automatisée
- Modélise le système en direct lors des réunions de conception
- Rédige les comptes-rendus

### Maîtrise d'ouvrage

- Définit les objectifs et exigences du système
- Valide les solutions proposées et élaborées
- Prépare et pilote le changement

### Maîtrise d'œuvre

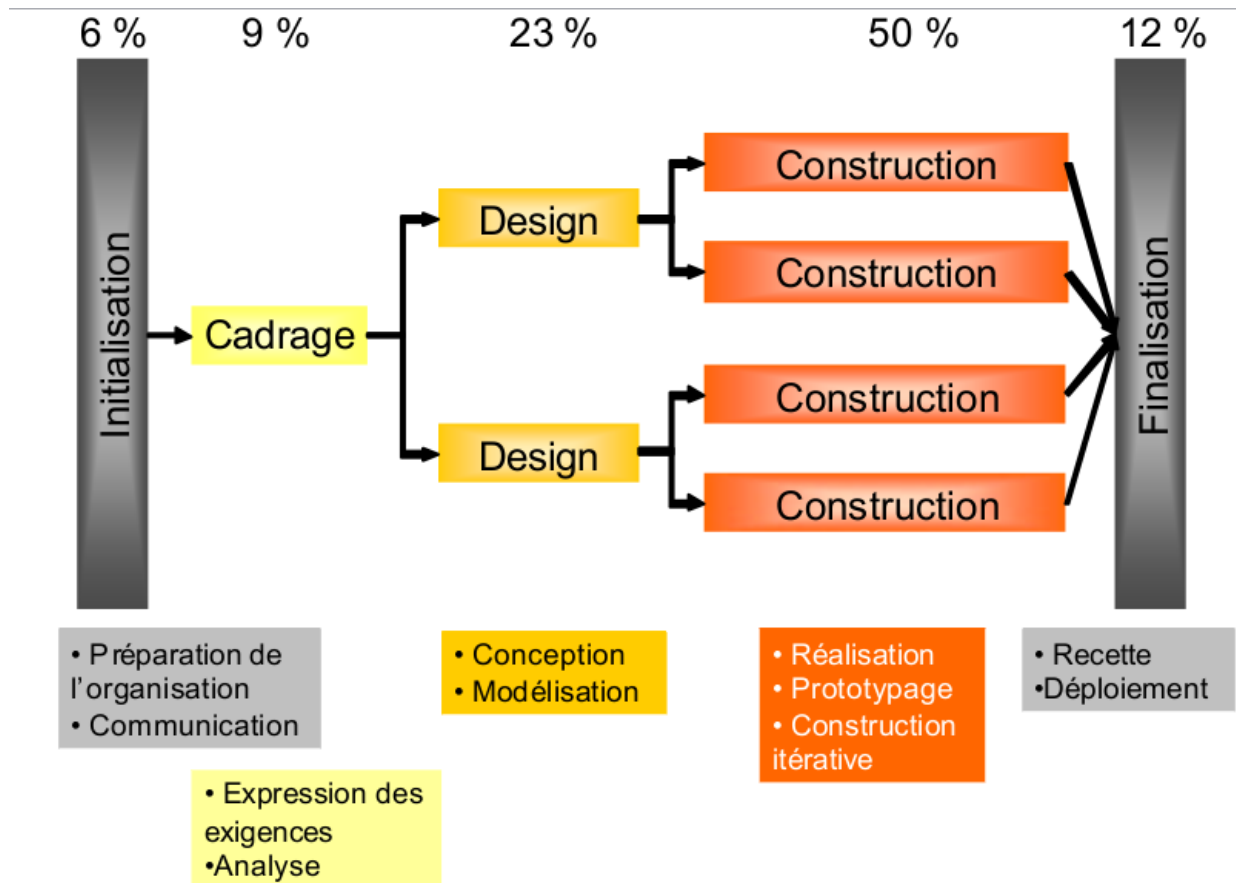
- Propose et réalise la solution
- Livre des « fonctionnalités »
- respecte les directives du Plan d'Assurance Qualité



# Les 5 phases d'un RAD

- Initialisation
  - *Préparation de l'organisation et communication*
- Cadrage
  - *Analyse et expression des exigences*
- Design
  - *Conception et modélisation*
- Construction
  - *Réalisation, prototypage*
- Finalisation
  - *Recette et déploiement*

# Cycle de vie d'un projet RAD



Cycle de vie d'un projet RAD

# Initialisation

Préparation de l'organisation et communication

- définit le périmètre général du projet
- établit la structure du travail par thèmes
- recense les acteurs pertinents
- amorce la dynamique du projet

5% du projet

# Cadrage

Analyse et expression des exigences

- Spécification de exigences par les utilisateurs lors d'entretiens de groupe
- 2 à 5 jours de sessions par commission

10% du projet



# Design

## Conception et modélisation

- Validation des modèles organisationnels par les utilisateurs: flux, traitements, données
- Validation d'un premier prototype par les utilisateurs
- 4 à 8 jours de sessions sont prévus par commission

25% du projet

# Construction

Réalisation, prototypage

- Construction au cours de plusieurs sessions itératives de l'application module par module
- Validation des prototypes par l'utilisateur

50% du projet

# Finalisation

Livraison globale et le transfert du système en exploitation et maintenance.

10% du projet

# Unified Process

# Le Processus Unifié

- Processus de développement pour les systèmes orientés objets
- Importance de la **modélisation** et de l'utilisation d'**outils** de support (particulièrement UML)

# Les 4 phases du UP

Chaque projet UP comprend les 4 phases suivantes:

- Inception
- Elaboration
- Construction
- Transition

# Inception

- Vision approximative de la finalité du projet
- Etude d'opportunité
- Définition de périmètre
- Estimations globales

# Elaboration

- Ebauche plus élaborée
- Implémentation de l'architecture du noyau
- Résolutions des risques élevés
- Identification de la plupart des besoins
- Estimations plus réalistes



# Construction

- Implémentation des éléments à moindre risque
- Préparation du déploiement

# Transition

- Béta test
- Déploiement final

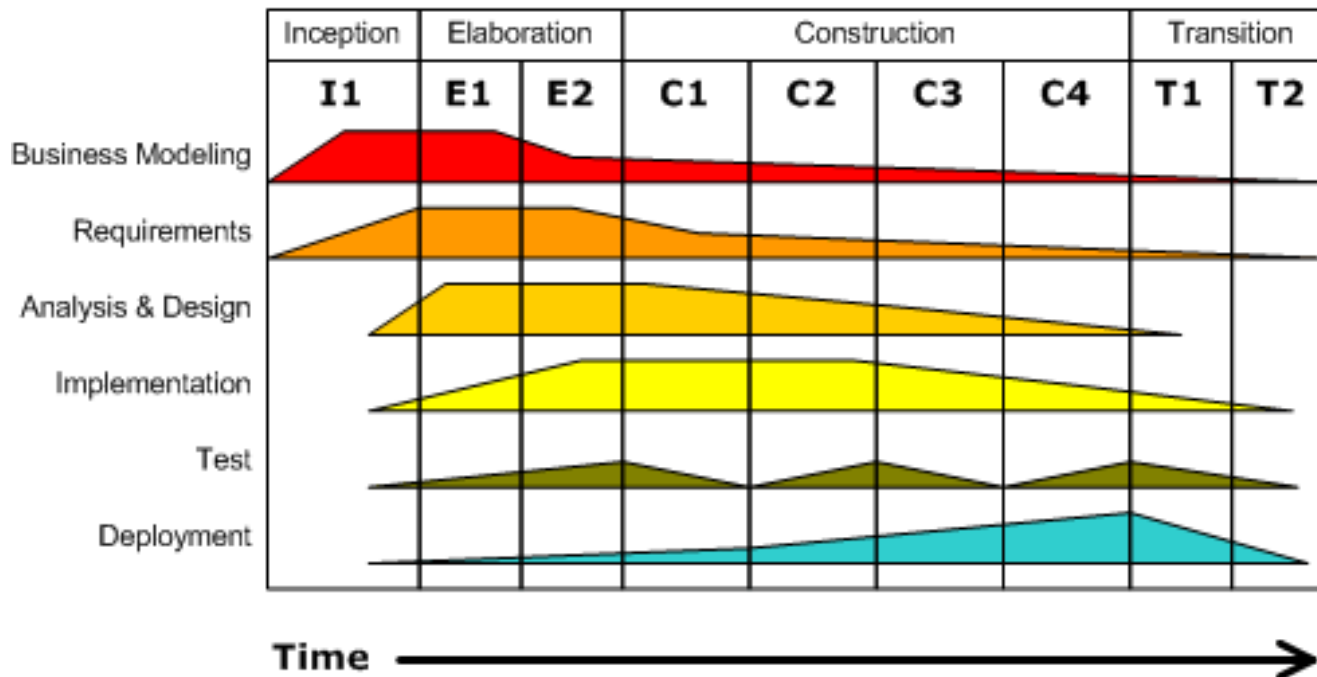
# Les disciplines du UP

Les disciplines sont les ensembles d'activités dans un domaine donné

- Modélisation métier
- Exigences
- Conception
- Implémentation
- Validation
- Déploiement

### Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



# Réflexion

*Quels sont les avantages de l'UP*

# Avantages du développement itératif

- La communication est de meilleure qualité
- La visibilité est meilleure
- La qualité est évaluée en continu
- Les risques sont détectés très tôt
- L'équipe prend confiance
- Les coûts sont contrôlés
- Possibilité d'exploiter méthodiquement les leçons tirées d'une itération

# Exercices

Serait-ce faux de considérer que:

- Inception = Spécification
- Elaboration = Conception
- Construction = Implémentation

# Méthode Agile



# **Le manifeste Agile**

Valeurs et Principes

# Valeurs

- Priorité aux **personnes** et aux **interactions** par rapport aux *procédures* et aux *outils*
  - Travail en groupe, communication
- Priorité aux **applications opératinnelles** par rapport à une *documentation pléthorique*
  - Documentations succinctes à jour, documentation permanente du code

# Valeurs

- Priorité de la **collaboration avec le client** par rapport à la *négociation de contrat*
  - Feedback régulier du client, solution répondant réellement aux attentes
  - Grande maturité du client, relation de confiance
- Priorité de **l'acceptation du changement** par rapport à la *planification*
  - • Planning flexible, modifications possibles après 1ère version du système

# Principes

1. 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. 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*

# Principes

3. 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. 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

# Principes

6. 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. Le **fonctionnement** de l'application est le premier indicateur d'avancement du projet
8. Les méthodes agiles recommandent que le projet avance à un **rythme soutenable** : développeurs et utilisateurs devraient pouvoir maintenir un rythme constant indéfiniment
  - *Adapter le rythme pour préserver la qualité du travail sur la durée du projet*

# Principes

9. Porter une attention continue à l'**excellence technique** et à la conception améliore l'agilité
  - *Maintenir le code source propre, clair et robuste*
10. 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. Les meilleures architectures, spécifications et conceptions sont le fruit d'équipes qui **s'auto-organisent**
  - *Partage des responsabilités par volontariat*

# Principes

12. 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*



# Comparaison

Cycle de vie	Phases séquentielles	Agiles
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
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

# Exercice

## Méthode Agile ?

Comparer les méthodes en cascade, UP et RAD aux valeurs Agiles

Priorités Agiles	Cascade	RAD	UP
Personnes/Communication			
Applications opérationnelles			
Satisfaction du client			
Intégration du changement			

# Exercices

Quelle serait la meilleure méthode de développement pour les projets suivants:

# Systeme de navigation A380



# 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](#)

<input type="checkbox"/>	☆ Caitlin, me (2)	<a href="#">work</a>	<b>Flight number?</b> - Can you let me know which flight	Apr 19
<input type="checkbox"/>	☆ me, Nathan (3), <b>Draft</b>	<a href="#">friends</a>	<b>Gift ideas for Caitlin</b> - Any ideas	Apr 19
<input type="checkbox"/>	☆ Paige, me (3)		Shopping trip - Awesome, I'll see you then. We can grab	Apr 18
<input type="checkbox"/>	☆ Nicola Brennan		Friday drinks? - You guys free on Friday? We're thinking	Apr 18
<input type="checkbox"/>	☆ Lizzie, me (2)		Africa airfares - Cool, I'll check it out. Sounds like a good	Apr 18
<input type="checkbox"/>	☆ Caitlin Roran	<a href="#">work</a>	<b>Conference budget</b> - We'll need to sit down with	Apr 18
<input type="checkbox"/>	☆ Nathan Woodward	<a href="#">friends</a>	<b>Birthday cake</b> - Any idea which kind of cake is	Apr 18
<input type="checkbox"/>	☆ Lizzie, me (2)	<a href="#">friends</a>	<b>Cancelling tennis on Friday</b> - No problem, it was	Apr 18
<input type="checkbox"/>	☆ Caitlin Roran	<a href="#">work</a>	<b>March expense reports</b> - I've attached the recent	Apr 17
<input type="checkbox"/>	☆ me, Caitlin (4)	<a href="#">work</a>	<b>Meeting with Mitchell</b> - I think it would be best if we	Apr 17
<input type="checkbox"/>	☆ me, Nicola (2)		<b>Photos</b> - Can you send me that photo you mentioned earl	Apr 17
<input type="checkbox"/>	☆ Paige Stevens	<a href="#">friends</a>	<b>Camera shopping</b> - Did you get the link for the	Apr 17
<input type="checkbox"/>	☆ Zach, me (2)	<a href="#">work</a>	<b>Timesheets?</b> - I think we should ask Wayne for his	Apr 17
<input type="checkbox"/>	☆ Nathan Woodward		<b>Surprise party</b> - I was wondering if maybe you and Susan	Apr 17
<input type="checkbox"/>	☆ Nicola Brennan	<a href="#">vacation</a>	<b>Fishing</b> - Hi guys, Mark and I have arranged th	Apr 17
<input type="checkbox"/>	☆ Lizzie Astley	<a href="#">friends</a>	<b>Trip to Africa</b> - Hey, so do you still want to go? It	Apr 17

Select: [All](#), [None](#), [Read](#), [Unread](#), [Starred](#), [Unstarred](#)

[Archive](#) [Report Spam](#) [Delete](#) [More Actions](#) ▼ [Refresh](#) 1 - 16 of 16

**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](#)

# Systeme de gestion intégrée du stock





# ARPANET



# Facebook



www.facebook.com/zuck?sk=wall

Search 



 Wall

 Info

[Share Profile](#)  
[Report/Block This Person](#)

## 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

#### 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.



# Scanner médical



**Fin première partie**