Génie logiciel : cycle de vie du logiciel

Karën Fort

karen.fort@univ-lorraine.fr / https://members.loria.fr/KFort

Institut des sciences du Digital Management & Cognition
COMPOSANTE DE L'UNIVERSITÉ DE LORRAINE

UNIVERSITÉ DE LORRAINE

Quelques sources d'inspiration

- Cours de Lydie du Bousquet, UGA (avec son accord)
- Méthodes agiles : vidéo (en)
- Pour aller plus loin sur Scrum : vidéo (en)

Wooclap


1 Allez sur wooclap.com
2 Entrez le code d'événement dans le bandeau supérieur
Code d'événement
AROMEI
Activer les réponses par SMS

Wooclap

Exercice

Quels sont les mots que vous inspire le terme "génie logiciel" ?

Définition

Génie logiciel, ingénierie logicielle, *software engineering*

[...] science de génie industriel qui étudie les méthodes de travail et les bonnes pratiques des ingénieurs qui développent des logiciels. Le génie logiciel s'intéresse en particulier aux procédures systématiques qui permettent d'arriver à ce que des logiciels de grande taille **correspondent aux attentes du client**, *soient* **fiables**, *aient un* **coût d'entretien réduit** *et de bonnes* **performances** *tout en respectant les délais et les coûts de construction.*

`https://fr.wikipedia.org/wiki/G%C3%A9nie_logiciel`

Génie logiciel

- propose des méthodes systématiques et des outils
- pour atteindre, dans le développement :
  - la prédictabilité

Inspiré de Lydie du Bousquet

Génie logiciel

- propose des méthodes systématiques et des outils
- pour atteindre, dans le développement :
  - la prédictabilité
  - la précision

Inspiré de Lydie du Bousquet

Génie logiciel

- propose des méthodes systématiques et des outils
- pour atteindre, dans le développement :
  - la prédictabilité
  - la précision
  - la limitation des risques

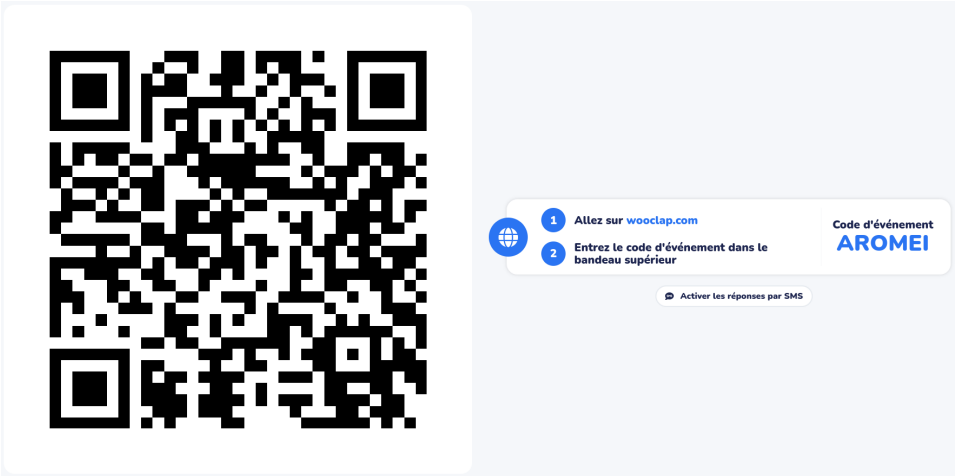
Inspiré de Lydie du Bousquet

Génie logiciel

- propose des méthodes systématiques et des outils
- pour atteindre, dans le développement :
  - la prédictabilité
  - la précision
  - la limitation des risques
  - le professionnalisme

Inspiré de Lydie du Bousquet

Wooclap


1 Allez sur wooclap.com
2 Entrez le code d'événement dans le bandeau supérieur
Code d'événement
AROMEI
Activer les réponses par SMS

Wooclap

Exercice

Quelles sont les activités qui composent un projet logiciel ?

Activités classiques dans le développement logiciel

- analyse des besoins

Inspiré de Lydie du Bousquet

Activités classiques dans le développement logiciel

- analyse des besoins
- spécifications

Inspiré de Lydie du Bousquet

Activités classiques dans le développement logiciel

- analyse des besoins
- spécifications
- architecture logicielle

Inspiré de Lydie du Bousquet

Activités classiques dans le développement logiciel

- analyse des besoins
- spécifications
- architecture logicielle
- conception

Inspiré de Lydie du Bousquet

Activités classiques dans le développement logiciel

- analyse des besoins
- spécifications
- architecture logicielle
- conception
- implémentation (développement)

Inspiré de Lydie du Bousquet

Activités classiques dans le développement logiciel

- analyse des besoins
- spécifications
- architecture logicielle
- conception
- implémentation (développement)
- test

Inspiré de Lydie du Bousquet

Activités classiques dans le développement logiciel

- analyse des besoins
- spécifications
- architecture logicielle
- conception
- implémentation (développement)
- test
- documentation

Inspiré de Lydie du Bousquet

Activités classiques dans le développement logiciel

- analyse des besoins
- spécifications
- architecture logicielle
- conception
- implémentation (développement)
- test
- documentation
- installation, déploiement

Inspiré de Lydie du Bousquet

Activités classiques dans le développement logiciel

- analyse des besoins
- spécifications
- architecture logicielle
- conception
- implémentation (développement)
- test
- documentation
- installation, déploiement
- formation et support

Inspiré de Lydie du Bousquet

Activités classiques dans le développement logiciel

- analyse des besoins
- spécifications
- architecture logicielle
- conception
- implémentation (développement)
- test
- documentation
- installation, déploiement
- formation et support
- maintenance

Inspiré de Lydie du Bousquet

Processus de développement

aka méthodologie de développement, cycle de vie du développement logiciel, processus logiciel

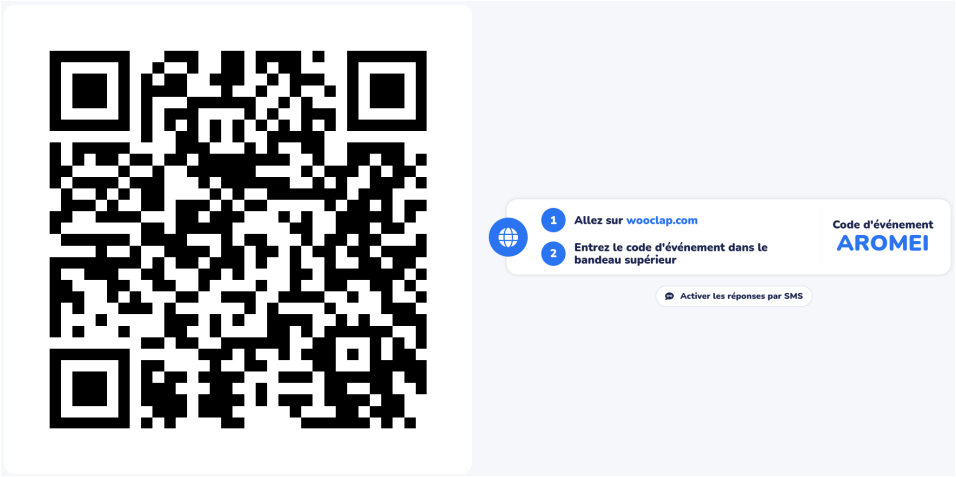
Définition

Méthodologie utilisée pour le développement d'un produit logiciel.
Organisation des tâches ou activités nécessaires au développement.

$\rightarrow$ plusieurs modèles

Inspiré de Lydie du Bousquet

Wooclap


1 Allez sur wooclap.com
2 Entrez le code d'événement dans le bandeau supérieur
Code d'événement
AROMEI
Activer les réponses par SMS

Wooclap

Exercice

Quels modèles de développement logiciel connaissez-vous ?

À connaître. . .

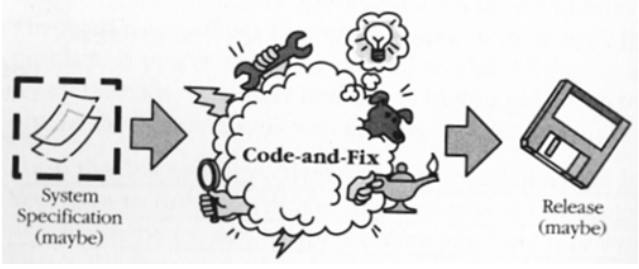
un ingénieur informaticien doit :

- savoir reconnaître un processus de développement
- savoir identifier l'ordre des étapes
- connaître les avantages et les inconvénients de chacun
- choisir le processus le plus adapté à la situation

Inspiré de Lydie du Bousquet

Code and fix

- pas vraiment de conception
- les programmeurs codent directement
- à un moment donné, on commence les tests (souvent trop tard)
- les inévitables bugs doivent être corrigés avant de lancer le produit (on espère)



Inspiré de Lydie du Bousquet

Code and fix : avantages et inconvénients

\+ donne l'impression d'être efficace et rapide

Code and fix : avantages et inconvénients

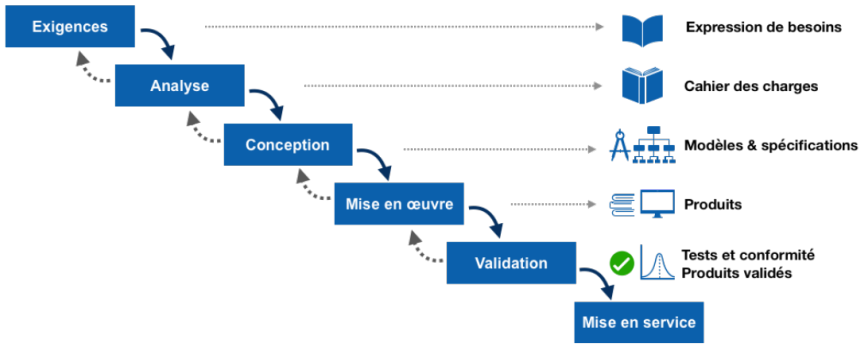
+ donne l'impression d'être efficace et rapide
- très souvent, ni l'un ni l'autre

Code and fix : avantages et inconvénients

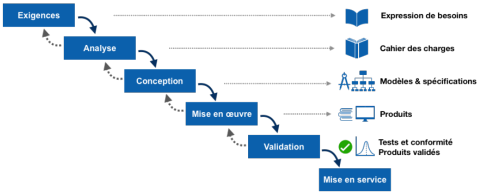
+ donne l'impression d'être efficace et rapide
- très souvent, ni l'un ni l'autre
- convient mal au développement en équipe et aux moyens/gros projets

Développement en cascade

Approche séquentielle dans laquelle le développement est vu comme se déroulant logiquement suivant différentes phases qui se succèdent :



Développement en cascade : l'idée

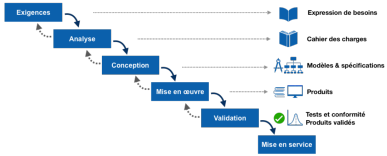


Approche structurée :

- une phase doit être terminée avant d'en commencer une nouvelle
- étapes bien identifiées :
  - chaque phase est documentée et validée

Inspiré de Lydie du Bousquet

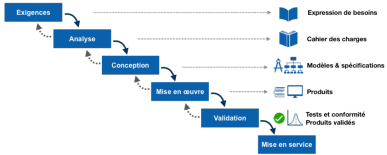
Approche en cascade : avantages et inconvénients



\+ simple

Inspiré de Lydie du Bousquet

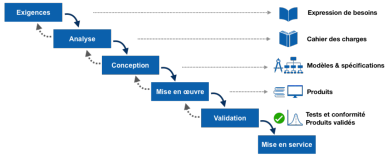
Approche en cascade : avantages et inconvénients



+ simple
+ à chaque phase correspondent des livrables spécifiques

Inspiré de Lydie du Bousquet

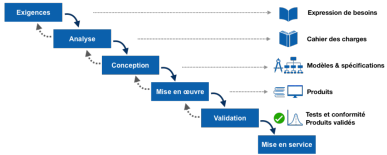
Approche en cascade : avantages et inconvénients



+ simple
+ à chaque phase correspondent des livrables spécifiques
+ vérification à chaque étape pour détecter des erreurs/malentendus

Inspiré de Lydie du Bousquet

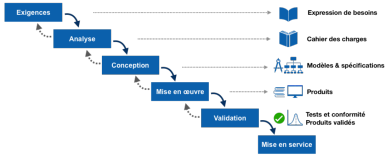
Approche en cascade : avantages et inconvénients



+ simple
+ à chaque phase correspondent des livrables spécifiques
+ vérification à chaque étape pour détecter des erreurs/malentendus
- implique que les besoins ne changent pas

Inspiré de Lydie du Bousquet

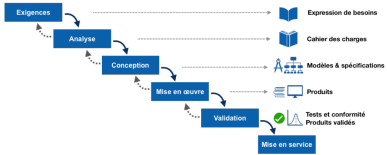
Approche en cascade : avantages et inconvénients



+ simple
+ à chaque phase correspondent des livrables spécifiques
+ vérification à chaque étape pour détecter des erreurs/malentendus
- implique que les besoins ne changent pas
- très difficile de revenir à une étape une fois terminée

Inspiré de Lydie du Bousquet

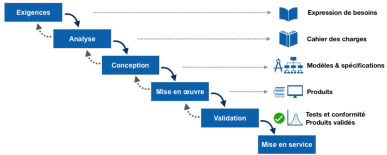
Approche en cascade : avantages et inconvénients



\+ simple
\+ à chaque phase correspondent des livrables spécifiques
\+ vérification à chaque étape pour détecter des erreurs/malentendus
\- implique que les besoins ne changent pas
\- très difficile de revenir à une étape une fois terminée
\- peu de flexibilité, de possibilités d'ajustements

Inspiré de Lydie du Bousquet

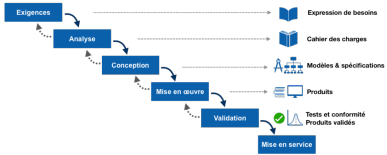
Approche en cascade : avantages et inconvénients



+ simple
+ à chaque phase correspondent des livrables spécifiques
+ vérification à chaque étape pour détecter des erreurs/malentendus
- implique que les besoins ne changent pas
- très difficile de revenir à une étape une fois terminée
- peu de flexibilité, de possibilités d'ajustements
- le produit n'est disponible qu'à la toute fin

Inspiré de Lydie du Bousquet

Approche en cascade : conclusion



Pour les projets :

- de taille moyenne
- avec des spécifications connues et précises

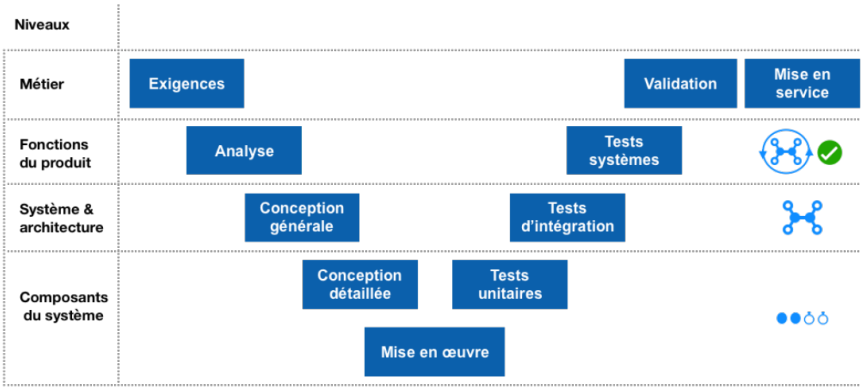
Mais :

- beaucoup de documentation

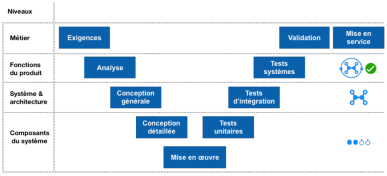
Inspiré de Lydie du Bousquet

Modèle en V

Approche séquentielle (= cascade), mais le test du produit est planifié en parallèle de chaque phase de développement (mais réalisé après) :



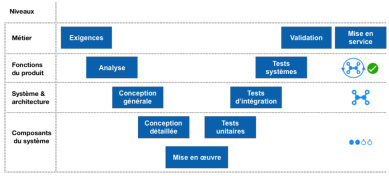
Modèle en V : avantages et inconvénients



\+ simple et facile à mettre en oeuvre

Inspiré de Lydie du Bousquet

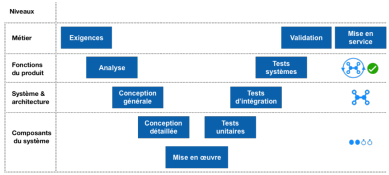
Modèle en V : avantages et inconvénients



+ simple et facile à mettre en oeuvre
+ les tests sont préparés avant ou en parallèle du développement

Inspiré de Lydie du Bousquet

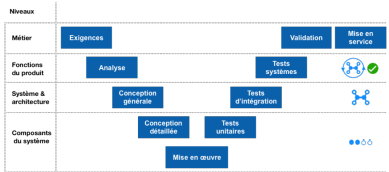
Modèle en V : avantages et inconvénients



+ simple et facile à mettre en oeuvre
+ les tests sont préparés avant ou en parallèle du développement
+ accélère et améliore la validation

Inspiré de Lydie du Bousquet

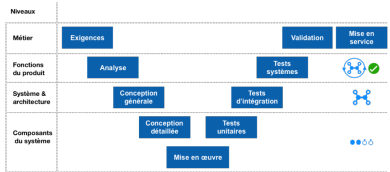
Modèle en V : avantages et inconvénients



+ simple et facile à mettre en oeuvre
+ les tests sont préparés avant ou en parallèle du développement
+ accélère et améliore la validation
- mêmes problèmes qu'avec l'approche en cascade

Inspiré de Lydie du Bousquet

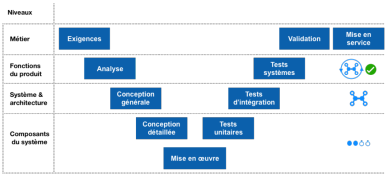
Modèle en V : avantages et inconvénients



+ simple et facile à mettre en oeuvre
+ les tests sont préparés avant ou en parallèle du développement
+ accélère et améliore la validation
- mêmes problèmes qu'avec l'approche en cascade
- si des changements apparaissent à mi-chemin, le plan de test et les exigences doivent être mis à jour

Inspiré de Lydie du Bousquet

Modèle en V : conclusion



Pour les projets :

- où la validation est un point clé, avec une équipe Qualité
- avec des spécifications connues et précises

Inspiré de Lydie du Bousquet

Wooclap

1 Allez sur wooclap.com

2 Entrez le code d'événement dans le bandeau supérieur

Code d'événement
AROMEI

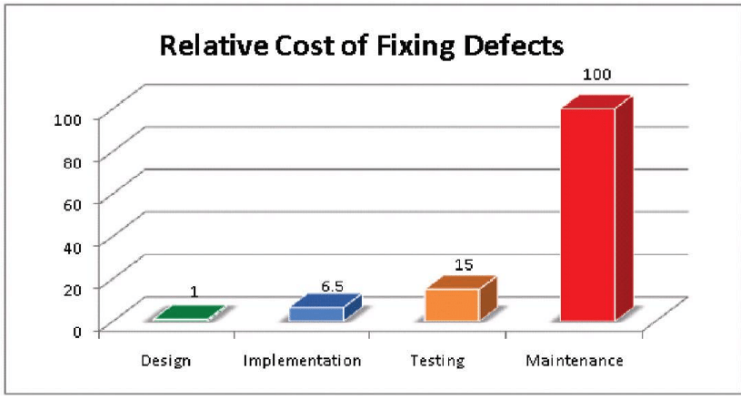
Activer les réponses par SMS

Wooclap

Exercice

Il est plus coûteux de corriger un bug à l'étape de la maintenance par rapport à la conception, mais dans quelle proportion ?

Coût d'une correction de bug à différentes phases du développement



IBM System Science Institute Relative Cost of Fixing Defects [Dawson et al., 2010]

Développement incrémental/itératif

Objectif :

- être capable de fournir un produit final adapté
- réduire les risques en découpant le projet en petites parties et proposer davantage de capacités d'adaptation pendant le développement

Différentes possibilités :

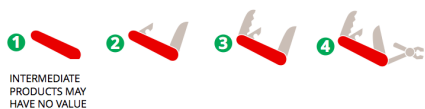
- réaliser une série de mini cascades :
  - toutes les phases de la cascade sont réalisées pour une petite partie
  - avant de réaliser l'incrément suivant
- les exigences générales sont définies avant de procéder au développement des petites cascades correspondant aux incréments individuels du système, ou
- l'analyse des exigences, l'architecture et le coeur du système sont définies, puis on développe successivement les différents incréments

Inspiré de Lydie du Bousquet

Développement incrémental

Principes :

- les exigences sont collectées globalement
- le développement est découpé en parties
- ces parties sont développées à différents moments et intégrées pour générer les releases (versions)
- chaque incrément apporte de la valeur logicielle (par ex. ajout d'un package)



Inspiré de Lydie du Bousquet

Développement incrémental : avantages et inconvénients

\+ première version livrée plus tôt que dans un modèle en cascade

Inspiré de Lydie du Bousquet

Développement incrémental : avantages et inconvénients

+ première version livrée plus tôt que dans un modèle en cascade
+ les parties les plus importantes peuvent être développées en priorité, l'ordre du reste peut être modifié

Inspiré de Lydie du Bousquet

Développement incrémental : avantages et inconvénients

+ première version livrée plus tôt que dans un modèle en cascade
+ les parties les plus importantes peuvent être développées en priorité, l'ordre du reste peut être modifié
+ les utilisateurs peuvent donner leur avis pour ajuster les exigences pour chacune des versions

Inspiré de Lydie du Bousquet

Développement incrémental : avantages et inconvénients

+ première version livrée plus tôt que dans un modèle en cascade
+ les parties les plus importantes peuvent être développées en priorité, l'ordre du reste peut être modifié
+ les utilisateurs peuvent donner leur avis pour ajuster les exigences pour chacune des versions
+ tests et gestion des risques plus faciles que dans un modèle en cascade

Inspiré de Lydie du Bousquet

Développement incrémental : avantages et inconvénients

+ première version livrée plus tôt que dans un modèle en cascade
+ les parties les plus importantes peuvent être développées en priorité, l'ordre du reste peut être modifié
+ les utilisateurs peuvent donner leur avis pour ajuster les exigences pour chacune des versions
+ tests et gestion des risques plus faciles que dans un modèle en cascade
- architecture choisie en général au début $\Rightarrow$ besoin d'une définition complète de tout le système avant de le découper en parties

Inspiré de Lydie du Bousquet

Développement incrémental : avantages et inconvénients

+ première version livrée plus tôt que dans un modèle en cascade
+ les parties les plus importantes peuvent être développées en priorité, l'ordre du reste peut être modifié
+ les utilisateurs peuvent donner leur avis pour ajuster les exigences pour chacune des versions
+ tests et gestion des risques plus faciles que dans un modèle en cascade
- architecture choisie en général au début $\Rightarrow$ besoin d'une définition complète de tout le système avant de le découper en parties
- coût total plus élevé que celui d'une cascade

Inspiré de Lydie du Bousquet

Développement itératif

Principes :

- construire une première version
- obtenir des retours clients et raffiner
- continuer jusqu'à ce que le produit soit satisfaisant

Inspiré de Lydie du Bousquet

Développement itératif : avantages et inconvénients

+ limitation des fonctionnalités rarement utilisées, focalisation sur les fonctionnalités utilisées souvent

Inspiré de Lydie du Bousquet

Développement itératif : avantages et inconvénients

+ limitation des fonctionnalités rarement utilisées, focalisation sur les fonctionnalités utilisées souvent
+ produit utilisable presque immédiatement et constamment

Inspiré de Lydie du Bousquet

Développement itératif : avantages et inconvénients

+ limitation des fonctionnalités rarement utilisées, focalisation sur les fonctionnalités utilisées souvent
+ produit utilisable presque immédiatement et constamment
+ généralement utilisé dans les **méthodes agiles** (voir plus loin)

Inspiré de Lydie du Bousquet

Développement itératif : avantages et inconvénients

+ limitation des fonctionnalités rarement utilisées, focalisation sur les fonctionnalités utilisées souvent
+ produit utilisable presque immédiatement et constamment
+ généralement utilisé dans les **méthodes agiles** (voir plus loin)
- pression accrue sur l'utilisateur (qui doit participer davantage)

Inspiré de Lydie du Bousquet

Développement itératif : avantages et inconvénients

+ limitation des fonctionnalités rarement utilisées, focalisation sur les fonctionnalités utilisées souvent
+ produit utilisable presque immédiatement et constamment
+ généralement utilisé dans les **méthodes agiles** (voir plus loin)
- pression accrue sur l'utilisateur (qui doit participer davantage)
- chaque phase d'une itération est rigide, sans superposition

Inspiré de Lydie du Bousquet

Développement itératif : avantages et inconvénients

+ limitation des fonctionnalités rarement utilisées, focalisation sur les fonctionnalités utilisées souvent
+ produit utilisable presque immédiatement et constamment
+ généralement utilisé dans les **méthodes agiles** (voir plus loin)
- pression accrue sur l'utilisateur (qui doit participer davantage)
- chaque phase d'une itération est rigide, sans superposition
- nécessite un niveau d'excellence technique supérieur (que les autres méthodes)

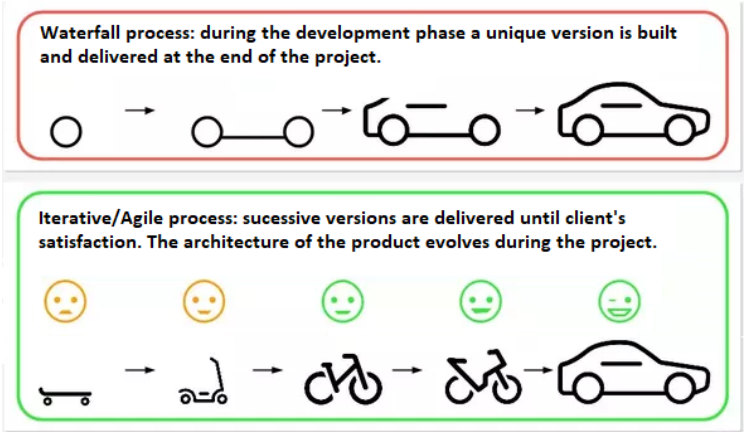
Inspiré de Lydie du Bousquet

Développement itératif : avantages et inconvénients

+ limitation des fonctionnalités rarement utilisées, focalisation sur les fonctionnalités utilisées souvent
+ produit utilisable presque immédiatement et constamment
+ généralement utilisé dans les **méthodes agiles** (voir plus loin)
- pression accrue sur l'utilisateur (qui doit participer davantage)
- chaque phase d'une itération est rigide, sans superposition
- nécessite un niveau d'excellence technique supérieur (que les autres méthodes)
- des problèmes coûteux d'architecture ou de conception peuvent apparaître du fait que toutes les exigences ne sont pas collectées dès le début pour tout le cycle de vie

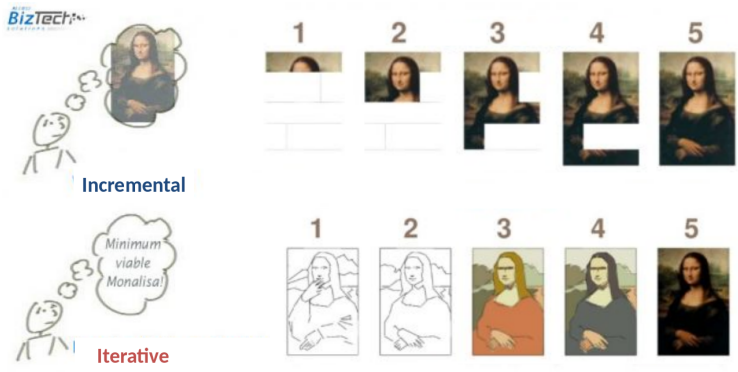
Inspiré de Lydie du Bousquet

Développement en cascade *vs* itératif



Inspiré de Lydie du Bousquet

Développement incrémental *vs* itératif



Inspiré de Lydie du Bousquet

Développement incrémental *vs* itératif

incrémenter signifie ajouter dessus *vs* itérer signifie refaire

Développement incrémental :

- nécessite des exigences bien définies
- entre chaque incrément, on peut modifier une partie des exigences
- plusieurs incréments peuvent être réalisés en parallèle

Développement itératif :

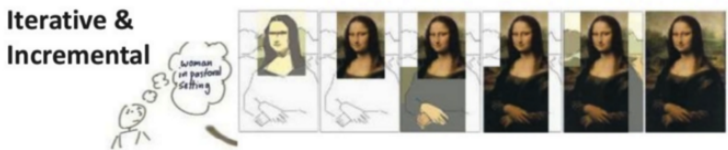
- on commence avec un ensemble d'exigences pour produire une première version
- à chaque itération, on collecte de nouvelles exigences et on fait une nouvelle version

Inspiré de Lydie du Bousquet

Développement agile : depuis début 2000

Principalement itératif :

- cycles rapides (1 à 3 semaines)
- petites *releases*



Exemples de méthodes agiles :

- XP (Extreme Programming)
- Scrum

Inspiré de Lydie du Bousquet

Développement agile : 12 principes (pas vraiment une méthode)

1. satisfaction client grâce à des livraisons rapides et continues de logiciels utiles
2. acceptation des changements dans les exigences, même tard dans le processus de développement
3. versions fonctionnelles livrées fréquemment (semaines plutôt que mois)
4. coopération quotidienne entre les commerciaux et les développeurs
5. projets bâtis autour d'individus motivés, à qui l'on fait confiance
6. conversations en face à face comme forme de communication (localisation unique)
7. progrès mesuré en termes de logiciel fonctionnel ou pas
8. développement continue à un rythme constant
9. attention continue à l'excellence technique et à la conception
10. la simplicité est essentielle (maximiser le travail qui n'est pas à faire)
11. équipes auto-organisées
12. adaptation constante aux changements

Inspiré de Lydie du Bousquet

Méthodes agiles : avantages

- \+ satisfaction client grâce à des livraisons rapides et continues de logiciels utiles
- \+ les gens et les interactions sont privilégiés plutôt que le processus et les outils
- \+ les clients, les développeurs et les testeurs interagissent constamment
- \+ livraisons fréquentes de logiciels fonctionnels (semaines plutôt que mois)
- \+ privilégie les conversations en face à face comme forme de communication
- \+ attention continue à l'excellence technique et à la conception
- +/- coopération quotidienne entre les développeurs et les commerciaux
- +/- adaptation constante aux changements
- +/- les changements dans les exigences sont acceptés, même tard dans le développement

Inspiré de Lydie du Bousquet

Méthodes agiles : inconvénients

- difficulté d'évaluer l'effort nécessaire au développement au début du cycle
- manque d'insistance sur la conception et la documentation
- le projet peut facilement dérailler

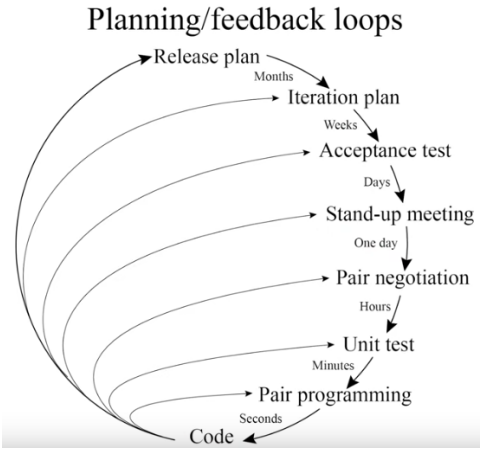
$\Rightarrow$ à utiliser quand les besoins clients n'arrêtent pas de changer (par ex. dans un domaine très dynamique)

Inspiré de Lydie du Bousquet

Exemple de méthode agile : XP (Extreme Programming)

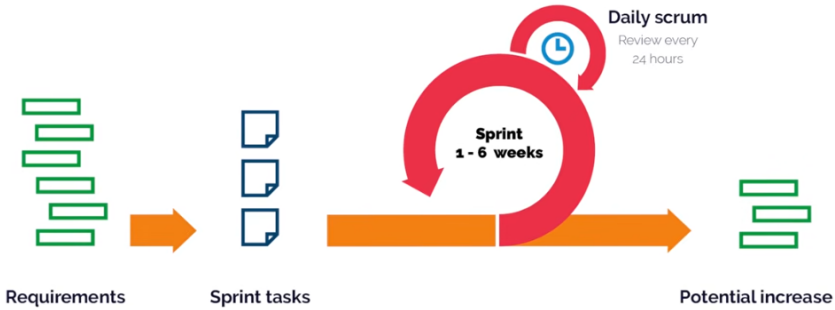
XP est centrée sur les processus d'ingénierie :

- développement orienté par les tests
- test unitaire de tout le code
- programmation en pairs
- intégration continue
- simplicité et clarté
- pas de développement non indispensable
- structure de management horizontale



source

Exemple de méthode agile : Scrum

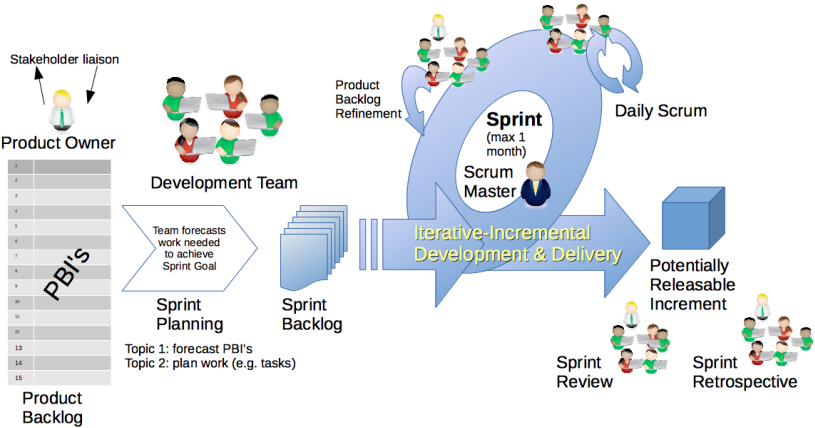


source

Scrum master :

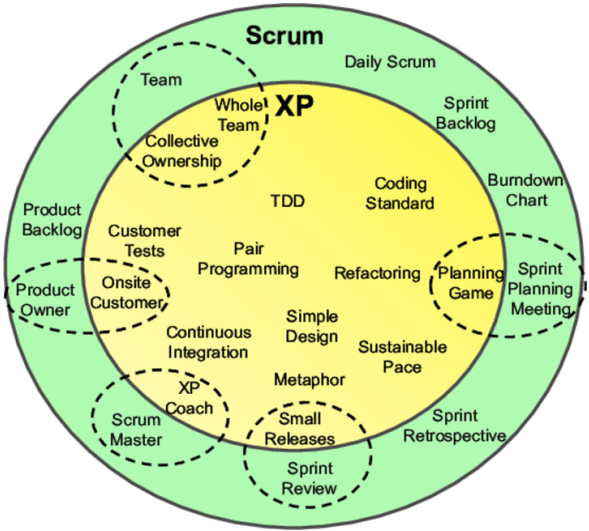
- facilite les progrès de l'équipe
- s'assure que les principes Scrum sont suivis

Exemple de méthode agile : Scrum



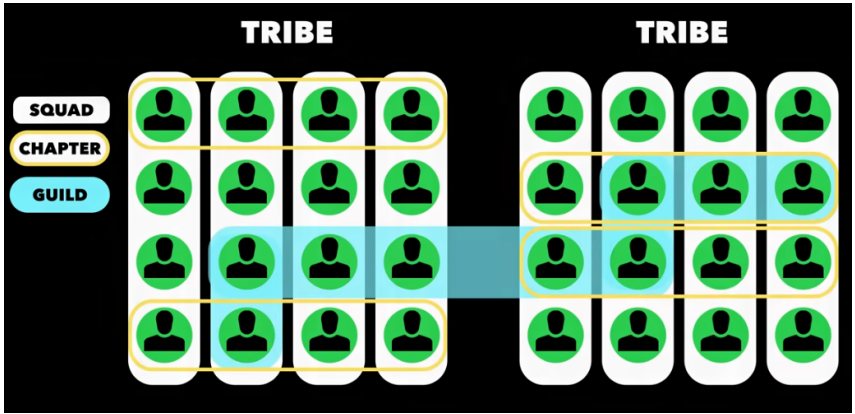
Inspiré de Lydie du Bousquet

Scrum *vs* XP



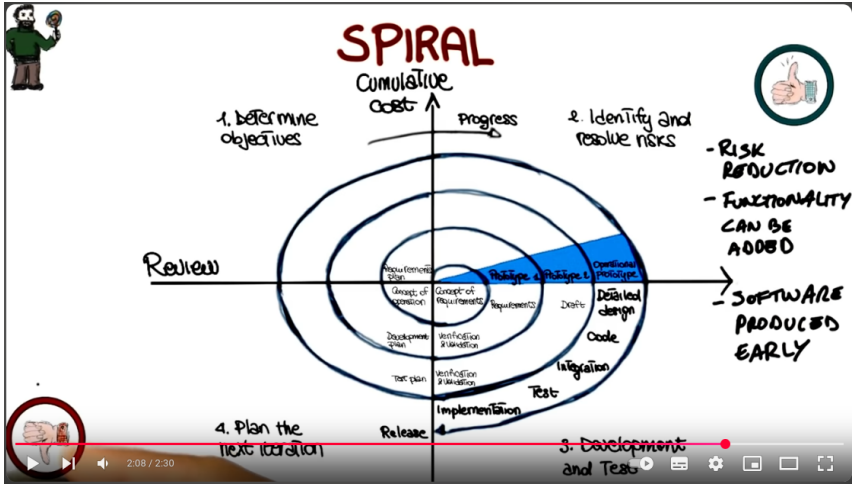
Inspiré de Lydie du Bousquet

Exemple de méthode spécifique : Spotify



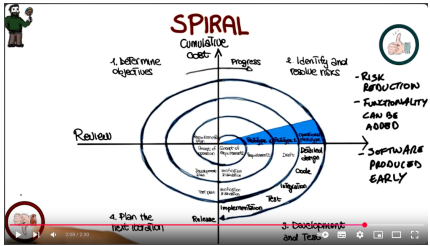
source

Méta modèle en spirale : centré sur l'analyse du risque



source

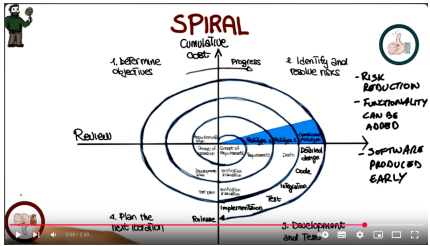
Méta modèle en spirale : avantages et inconvénients



\+ réduction des risques

Inspiré de Lydie du Bousquet

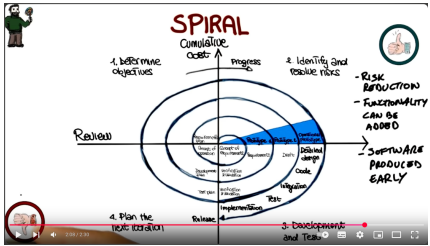
Méta modèle en spirale : avantages et inconvénients



+ réduction des risques
+ fort contrôle de la documentation et de l'acceptation

Inspiré de Lydie du Bousquet

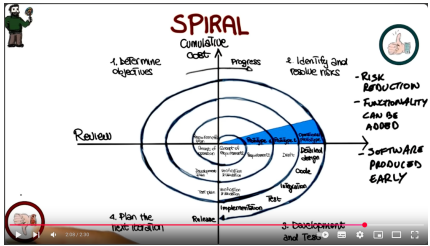
Méta modèle en spirale : avantages et inconvénients



+ réduction des risques
+ fort contrôle de la documentation et de l'acceptation
+ fonctionnalités additionnelles peuvent être ajoutées par la suite (autre cycle)

Inspiré de Lydie du Bousquet

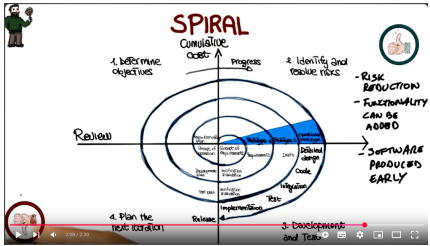
Méta modèle en spirale : avantages et inconvénients



+ réduction des risques
+ fort contrôle de la documentation et de l'acceptation
+ fonctionnalités additionnelles peuvent être ajoutées par la suite (autre cycle)
- modèle coûteux

Inspiré de Lydie du Bousquet

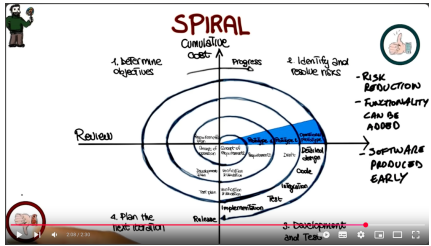
Méta modèle en spirale : avantages et inconvénients



+ réduction des risques
+ fort contrôle de la documentation et de l'acceptation
+ fonctionnalités additionnelles peuvent être ajoutées par la suite (autre cycle)
- modèle coûteux
- niveau d'expertise élevé (analyse de risques)

Inspiré de Lydie du Bousquet

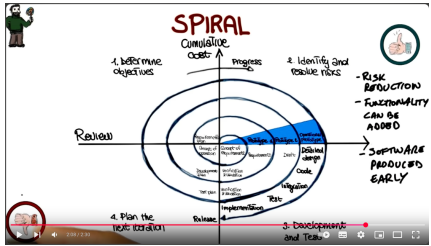
Méta modèle en spirale : avantages et inconvénients



+ réduction des risques
+ fort contrôle de la documentation et de l'acceptation
+ fonctionnalités additionnelles peuvent être ajoutées par la suite (autre cycle)
- modèle coûteux
- niveau d'expertise élevé (analyse de risques)
- succès du projet dépend largement de l'analyse de risques

Inspiré de Lydie du Bousquet

Méta modèle en spirale : avantages et inconvénients



+ réduction des risques
+ fort contrôle de la documentation et de l'acceptation
+ fonctionnalités additionnelles peuvent être ajoutées par la suite (autre cycle)
- modèle coûteux
- niveau d'expertise élevé (analyse de risques)
- succès du projet dépend largement de l'analyse de risques
- pas adapté aux petits projets

Inspiré de Lydie du Bousquet

Méta modèle en spirale : utilisation

- lorsque les coûts et l'évaluation des risques sont importants
- pour les projets à haut/moyen risque
- lorsque les utilisateurs ne sont pas clairs sur leurs besoins
- lorsque les exigences sont complexes
- lorsque des changements significatifs sont à prévoir (recherche et exploration)

Inspiré de Lydie du Bousquet

Récapitulatif

Facteurs	**cascade**	**en V**	**itératif et incrémental**	**agile**
besoins clients pas clairs	mauvais	mauvais	bon	excellent
système complexe	bon	bon	bon	mauvais
système fiable	bon	bon	bon	bon
documentations	excellent	excellent	ça dépend	mauvais
ré-utilisabilité composants	bon	bon	ça dépend	mauvais

Inspiré de Lydie du Bousquet

- les différents modèles
- leurs avantages et inconvénients


Dawson, M., Burrell, D., Rahim, E., and Brewster, S. (2010).
Integrating software assurance into the software development life cycle (sdlc).
Journal of Information Systems Technology and Planning, 3 :49–53.