Génie Logiciel

Awa DIATTARA
awa.diattara@ugb.edu.sn

Processus de développement logiciel

Plan du cours

1. Qu'est ce qu'un processus de développement logiciel
2. Activités de développement logiciel
3. Schéma général d'un processus de développement logiciel
4. Modèles de développement logiciel
5. Importance de la documentation dans le projet

Qu'est ce qu'un processus de développement logiciel ?

- Définition : ensemble d'activités successives, organisées en de la production d'un logiciel.

- En pratique :
  - Pas de processus idéal
  - Choix du processus en fonction des contraintes (taille des équipes, temps, qualité,…)

  ✓ **Donc adaptation de « processus types » aux besoins réels**

Processus de développement logiciel

- Activités de développement logiciel
  - Analyse des besoins
  - Spécification
  - Conception
  - Programmation
  - Validation et vérification
  - Livraison
  - Maintenance

- Pour chaque activité : utilisation et production de documents.

Activités du développement logiciel

- Analyse des besoins. Comprendre les besoins du client.
  - Objectifs généraux, environnement du futur système, ressources disponibles, contraintes de performance,…
  - Fournie par le client (expert du domaine d'application, futur utilisateur,…).

- Spécification
  - Établir une description claire de ce que doit faire le logiciel (fonctionnalités détaillées, exigences de qualité, interface…)
  - Clarifier le cahier des charges (ambiguïtés, contradictions) en listant les exigences fonctionnelles et non fonctionnelles.

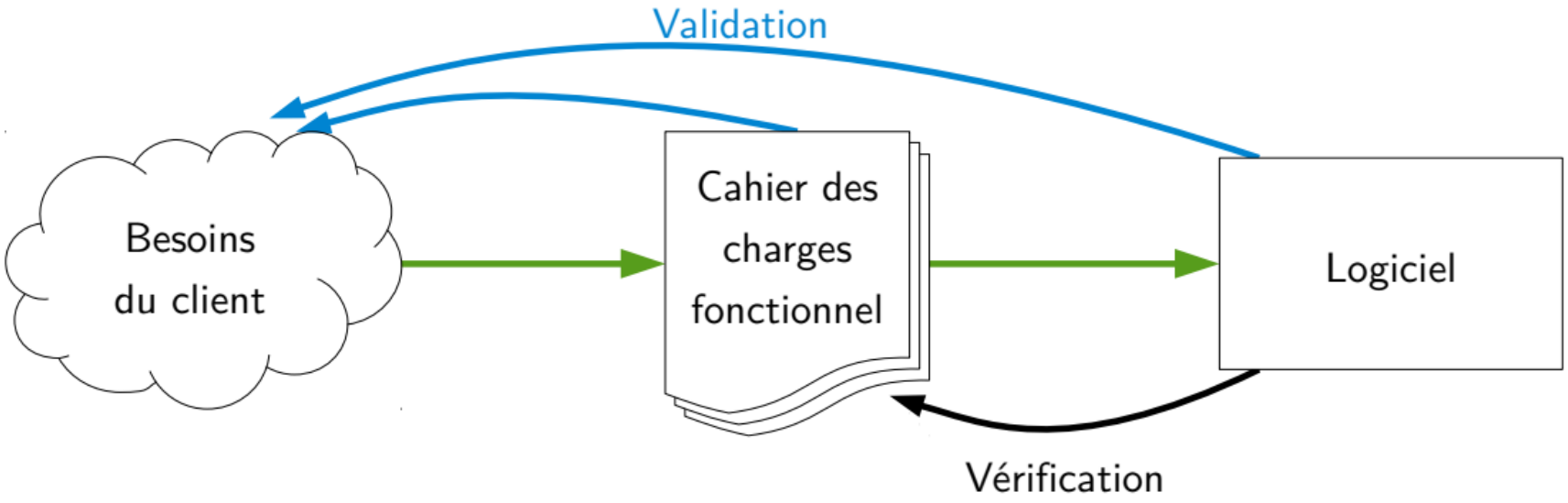
Activités du développement logiciel

- Conception. Élaborer une solution concrète réalisant la spécification
  - Analyse, choix de la modélisation, description de l'architecture du système en composants (avec interface et fonctionnalités)
  - Réalisation des fonctionnalités par les composants (algorithmes, organisation des données)
  - Réalisation des exigences non fonctionnelles (performance, sécurité,…)

- Programmation. Implantation de la solution conçue
  - Choix de l'environnement de développement, du/des langage(s) de programmation, de normes de développement, codage du logiciel dans le langage cible…

Activités du développement logiciel

- Test
  - Vérification et Validation
    - Vérification : assurer que le logiciel satisfait sa spécification.
    - Validation : assurer que les besoins du client sont satisfaits (au niveau de la spécification, du produit fini…)



Activités du développement logiciel

- Déploiement ou livraison.
  - Phase ou étape où le logiciel est installé, mis en production et rendu accessible aux utilisateurs finaux
  - Le logiciel quitte l'environnement de développement pour être utilisé en conditions réelles.

Activités du développement logiciel

- Maintenance
  - Types de maintenance
    - **Correction** : identifier et corriger des erreurs trouvées après la livraison.
    - **Adaptation** : adapter le logiciel aux changements dans l'environnement (format des données, environnement d'exécution,…)
    - **Perfection** : améliorer la performance, ajouter des fonctionnalités, améliorer la maintenabilité du logiciel.

Schéma général d'un processus de développement

- Remarques.
  - Il est très rare d'appliquer un processus comme une unique séquence des activités précédentes.
  - En général, un logiciel est le fruit de plusieurs itérations.
  - Chaque itération contient les activités de spécification, conception, implémentation, validation et évolution.
  - Il existe plusieurs modèles de processus qui organisent de façon différente ces activités.

Modèles de processus de développement

Modèles de processus développement de logiciel

- La modèles de processus de développement logiciel aident les développeurs à sélectionner la stratégie pour développer le produit logiciel.

- Ce sont des « *plans de travail* » qui permettent de planifier le développement.

Modèles de processus développement de logiciel

- Plusieurs modèles :
  - Modèle en cascade
  - Modèle en V
  - Développement par prototypage
  - Développement incrémental
  - Modèle orienté réutilisation
  - Modèle en spirale
  - Méthodes agiles
  - ✓ En pratique : mélange de divers modèles

Modèles de développement

Modèle en cascade

Qu’est ce que le modèle en cascade ?

- Le modèle en cascade ou waterfall en anglais, est un modèle de gestion de projet.
- Il utilise une approche linéaire et séquentielle des différentes phases et activités nécessaires à la livraison du produit.
- Ce modèle peut être utilisé dans divers secteurs : allant du développement logiciel à la gestion de projets d’ingénierie.
- Il a été présenté pour la première fois par Herbert D. Bennington en 1956 lors d'un Symposium sur les méthodes de programmation avancées.
- Mais il doit son succès à l’informaticien Winston Walker Royce qui a fait une analyse critique des modèles linéaires dans son essai *Managing the development of large Software Systems*, publié en 1970.

Qu’est ce que le modèle en cascade ?

- Dans son essai, Royce proposait comme alternative un modèle itératif et incrémental dans lequel chaque phase reposait sur la précédente et en vérifiait les résultats.

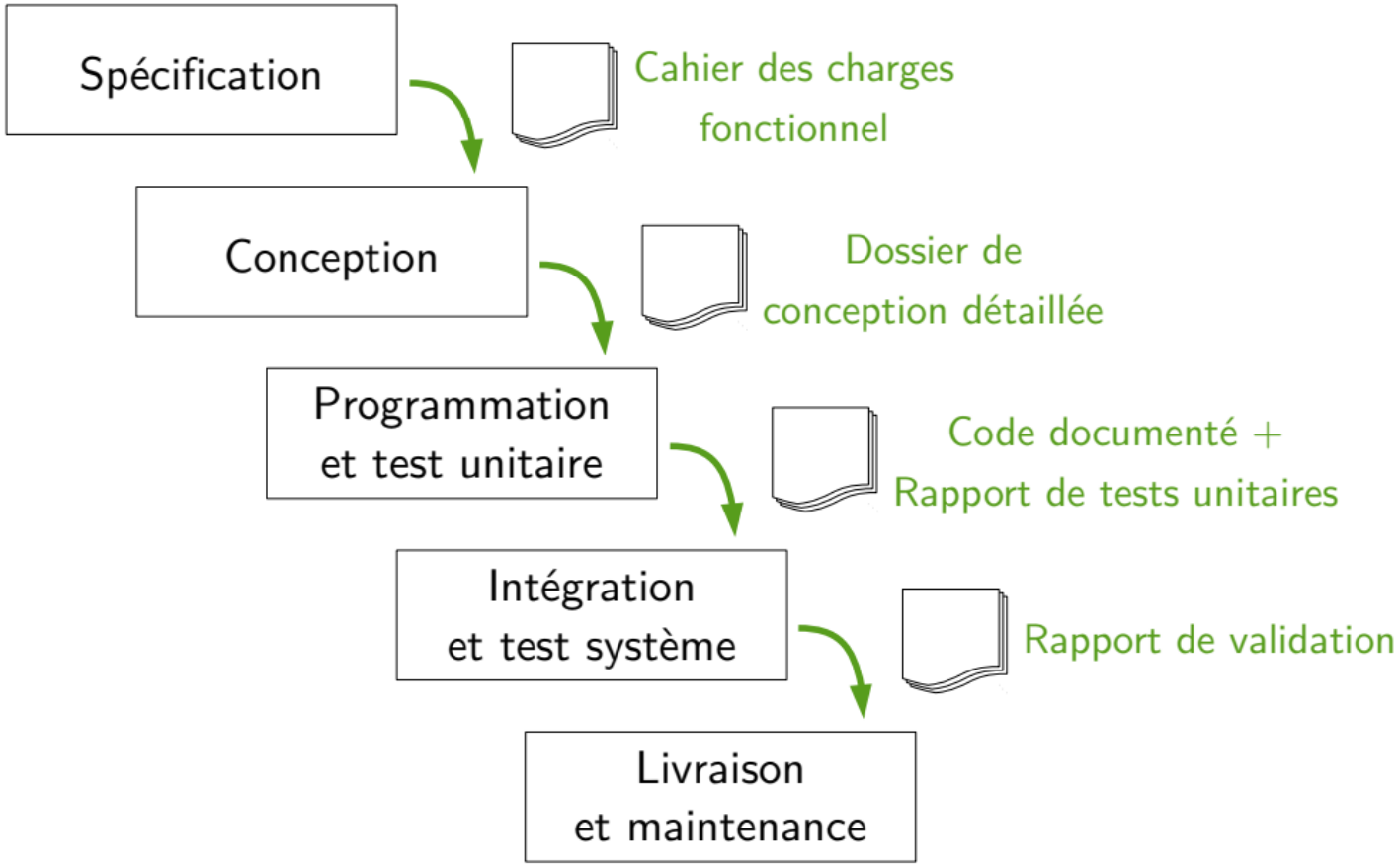
- Il proposait un modèle en 7 phases : Exigence système, Exigences logicielle, Analyse, Conception, Implémentation, Test, Exploitation.

- Le modèle doit sa grande notoriété au standard américain DoD-STD-2167, qui a par la suite, a proposé une forme extrêmement simplifiée du modèle de gestion développée par Royce.

Modèle en cascade

Modèle en cascade (Royce, année 70)

- Chaque étape doit être terminée avant que ne commence la suivante.
- A chaque étape, production d'un document base pour l'étape suivante.



Modèle en cascade

Modèle en cascade : caractéristiques

- Hérite des méthodes classiques d'ingénierie → s'adapte bien à un contexte où le logiciel fait partie d'un système complexe englobant.
- Production de documents entre chaque phase → améliore le suivi du projet.
- Chaque phase doit se terminer pour commencer la suivante.
- Découverte d'une erreur → retour à la phase à l'origine de l'erreur et nouvelle cascade, avec de nouveaux documents.
- Coût de modification d'une erreur important, donc choix en amont cruciaux (typique d'une production industrielle).

Modèle en cascade : avantages

- Plus adapté aux petits projets ou à ceux dont les spécifications sont bien connues et fixes.
- Adaptés également à des projets où les systèmes sont développés sur plusieurs sites (facilite la planification du projet).

Modèle en cascade : critiques

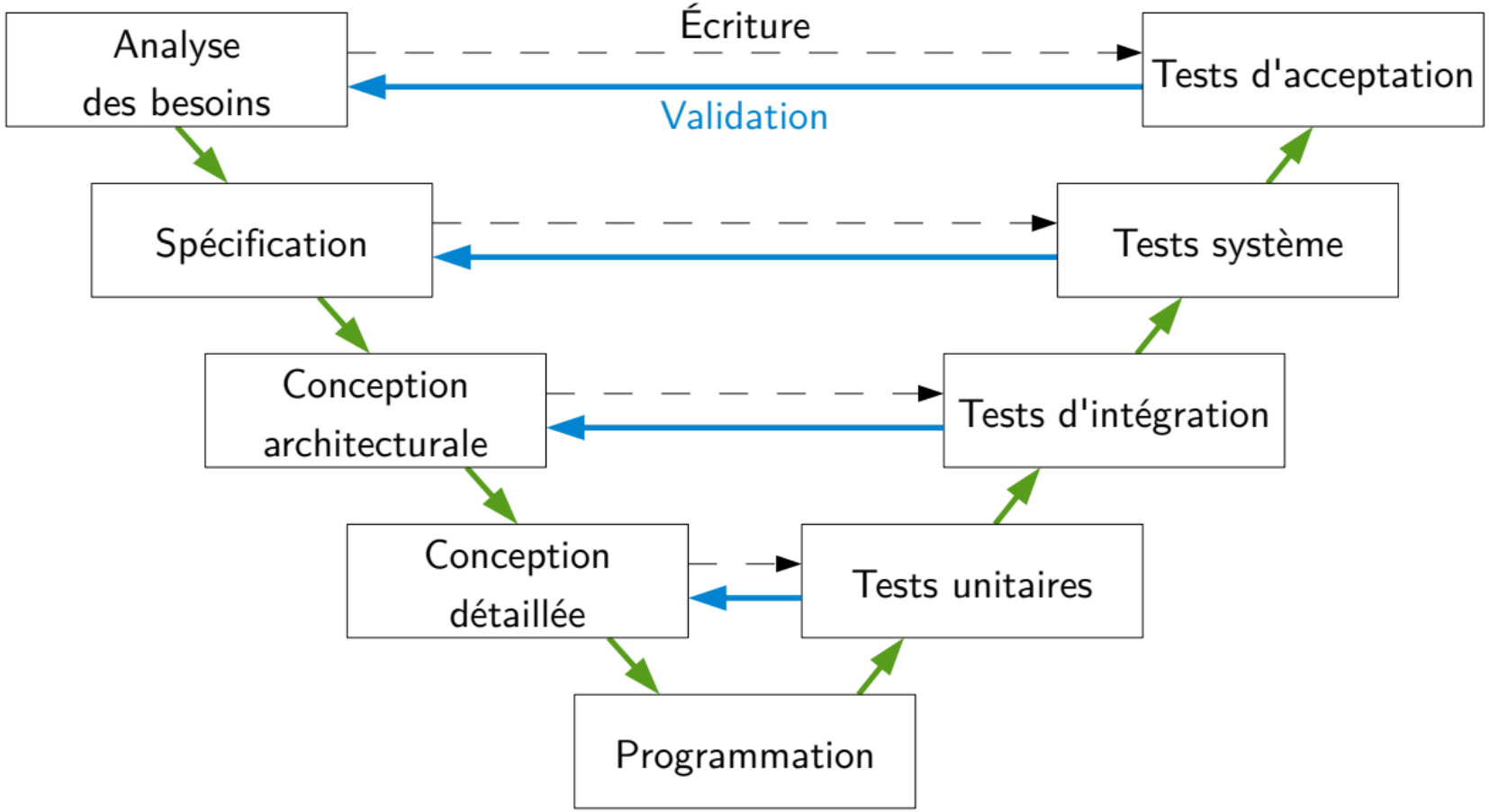
- Pas toujours adaptée à une production logicielle, en particulier difficile de s'adapter aux changements des besoins utilisateurs.
- Inadapté au développement de systèmes dont la spécification est difficile à formuler à priori.
- Les tests sont prévus tardivement.
- Le modèle en cascade rend coûteux le développement itératif puisque la rédaction des documents de validation de chaque phase demande beaucoup de travail.
- Erreurs produites lors de la phase de spécification impliquent de reprendre toutes les phases du processus.

Modèles de développement

Modèle en V

Modèles de processus de développement

❑ Modèle en V



✓ Plus adapté aux projets de taille et complexité moyennes.

Modèles de processus de développement

- Modèle en V :
  - Normalisé, très utilisé en informatique industrielle et télécommunication.
  - C'est un cycle orienté test :
    - A chaque activité correspond une activité de vérification
    - La vérification est prise en compte au moment même de la création
      - Permet d'éviter d'énoncer une propriété qu'il est impossible de vérifier objectivement une fois le logiciel réalisé

Niveaux de test

- **Test unitaire** : test de chaque unité de programme (méthode, classe, composant), indépendamment du reste du système.
- **Test d'intégration** : tests des interactions entre les composants (interfaces et composants compatibles).
- **Test système** : test du système complet par rapport à son cahier des charges.
- **Test d'acception** (recette) : fait par le client, validation par rapport aux besoins initiaux

Modèle en V : faiblesses

- Les choix techniques sont validés très (trop) tard par test.
- Beaucoup de choix techniques sont pris sans disposer d'information concrète (prise de risques).
- Il faut attendre longtemps pour savoir si on a bien construit le logiciel (difficile d'impliquer les utilisateurs car le logiciel n'est disponible qu'à la dernière phase).

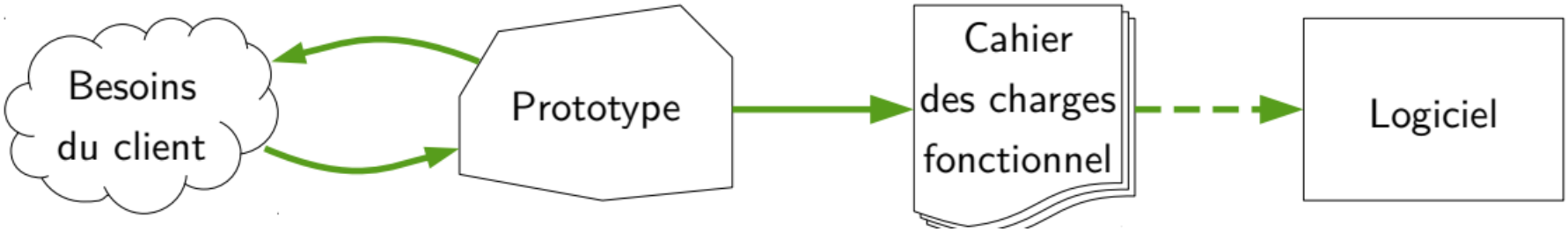
Modèles de développement

Modèle par prototypage

Modèles de processus de développement

❑ Développement par prototypage (développement évolutif)

- Développement rapide d'un prototype avec le client pour valider ses besoins.
- Écriture de la spécification à partir du prototype, puis processus de développement linéaire.



➢ Avantages : validation concrète des besoins, moins de risques d'erreur de spécification.

Modèles de processus de développement

Modèle de développement par prototypage : caractéristiques

- Augmente les chances de répondre aux besoins de l'utilisateur.
- Ne dispense pas d'écrire la spécification du système car il faut s'assurer que l'implémentation est correcte.
- Particulièrement adapté aux projets de taille moyenne (inférieur à 100000 lignes de code) comme par exemple des solutions intégrées pour les petites entreprises.

Modèle de développement par prototypage : critiques

- Difficile de gérer un projet utilisant ce modèle car la visibilité de l'avancement du développement est peu claire.
  - Nécessite un très bon chef de projet capable de se faire une idée de l'état d'avancement du système.
- Plus difficile de structurer correctement le logiciel (définir de bonnes abstractions, modulariser efficacement) car les prototypes sont par définition des produits « bricolés ».
- Coût en terme de tests et de validation du produit final peut être très important.
  - Des approches mixtes intégrant le module par prototypage pour produire un premier prototype validé et un module en cascade pour reconstruire correctement un produit final constituent en général un bon compromis.

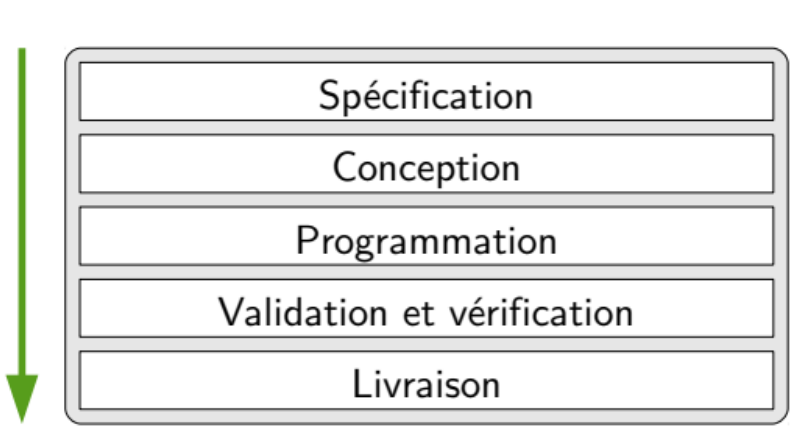
Modèles de développement

Modèle incrémental

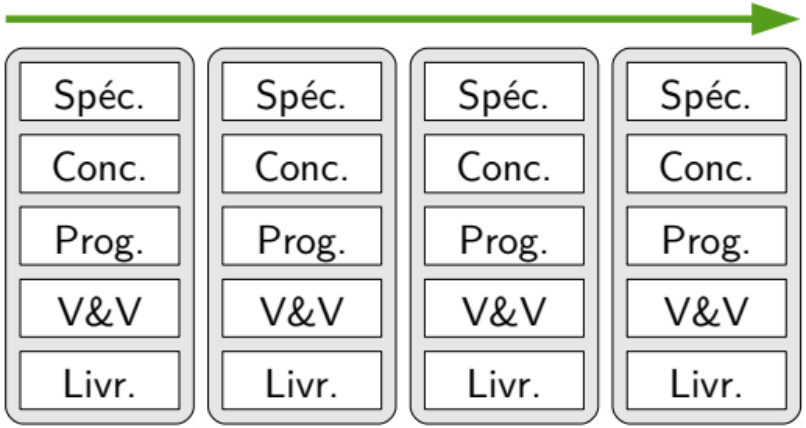
Modèles de processus de développement

Développement incrémental ou développement à livraison incrémentale

- Approche à mi-chemin entre les modèles en cascade et modèle par prototypage
  - Hiérarchiser les besoins du client
  - Concevoir et livrer au client un produit implantant un sous-ensemble de fonctionnalités par ordre de priorité (livraison incrémentale du produit).



Développement en cascade



Développement incrémental

Modèles de processus de développement

- Développement incrémental (développement à livraison incrémentale)

  - Avantage : minimiser le risque d'inadéquation aux besoins (traiter les parties les plus critiques du système en premier)

  - Difficulté : intégration des fonctionnalités secondaires non pensées en amont.

Modèles de processus de développement

Méthodes agiles

- Implication constante du client
- Adaptation au changement Cycles de développement rapides pour réagir aux changements
- Livraison fréquente de version opérationnelles
- …

➢ Avantage : développement rapide en adéquation avec les besoins.

➢ Minimisation des étapes de spécification, documentation → tests, maintenance, …

➢ Complexité de l'adoption de la culture agile

✓ Plus pratique pour de petites équipes de développement car la communication est cruciale.

Home work

- Préparer pour la prochaine séance, un exposé sur le modèle en spirale.

Importance de la documentation dans le projet

Documentation dans le projet

- Objectifs : traçabilité du projet

- Pour le client : avoir une vision claire de l'état d'avancement du projet.

- Pour l'équipe :
  - Regrouper et structurer les décisions prises
  - Avoir des références pour les décisions futures
  - Garantir la cohérence entre les modèles et le produit

- Constitue une base commune de référence
  - Pas de perte d'informations (personne quittant le projet, etc.)
  - Intégration rapide (nouveau arrivant dans le projet, etc.)

Documentation dans le projet

Documents de spécification et conception

- Rédaction : le plus souvent en langage naturel (français).

- Problèmes
  - Ambiguïtés : plusieurs sens pour un même mot selon les personnes ou contextes.
  - Contradictions, oublis, redondances difficiles à détecter
  - Difficultés à trouver une information
  - Mélange entre les niveaux d'abstraction (spécification vs. conception)

Documents de spécification et conception

- Alternatives au langage naturel
  - Langages informels :
    - Langage naturel structuré : modèle de document et règles de rédaction précis et documentés.
    - Pseudo-code : description algorithmique de l'exécution d'une tâche, donnant une vision opérationnelle du système
  - Langages semi-formels :

    Notation graphique : diagrammes structurés accompagnés de texte donnant une vue statique ou dynamique du système
  - Langages formels
    - Formalisme mathématique : propriétés logiques ou modèle du comportement du système dans un langage mathématique.

❑ Documents de spécification et conception

- Alternatives au langage naturel
  - Langages informels ou semi-formels :
    - ➢ Avantages : intuitifs, fondés sur l'expérience, facile à apprendre et à utiliser, répandus.
    - ➢ Inconvénients : ambigus, pas d'analyse systématique
  - Langages formels :
    - ➢ Avantages : précis, analysables automatiquement, utilisables pour automatiser la vérification et le test du logiciel
    - ➢ Inconvénients : apprentissage et maîtrise difficiles

➢ En pratique : utilisation de langages formels principalement pour des logiciels critiques ou restreinte aux parties critiques du système.