Introduction au Génie Logiciel

Séance 1 : Généralités

L. Laversa
laversa@irif.fr

Université Paris Cité

28 janvier 2025

. Remerciements à E. Bigeon

Plan du cours

1. Généralités (aujourd'hui)
2. Formalisme UML
3. Outils de Génie Logiciel
   - Bonnes pratiques de programmation
   - Automatisation
   - Design pattern
   - Tests

Évaluation

1. **Contrôle continu** (CC)
   a. quizz sur moodle
      - début de TD (9)
      - moyenne des 8 meilleurs
      - 60% de la note de CC
   b. TP noté
      - 2 dernières séances
      - 40% de la note de CC
2. **Examen final** (F)

$$\text{Note} = \max(CC,F) * 0{,}6 + \min(CC,F) * 0{,}4$$

Évaluation

1. **Contrôle continu** (CC)
   a. quizz sur moodle
      - début de TD (9)
      - moyenne des 8 meilleurs
      - 60% de la note de CC
   b. TP noté
      - 2 dernières séances
      - 40% de la note de CC
2. **Examen final** (F)
3. **Bonus** (B)
   - Défense d'architecture
   - Après UML
   - En groupe de 1 à 3
   - Dossier + soutenance

$$\text{Note} = \max(CC,F) * 0{,}6 + \min(CC,F) * 0{,}4$$

$$\text{Note} = \max(CC,F) * 0{,}5 + \max(\min(CC,F), B) * 0{,}3 + \min(\min(CC,F), B) * 0{,}2$$

Définitions

Logiciel

désigne un programme

Définitions

Logiciel

désigne un programme (code compilé ou source, librairies) et les documents nécessaires à son installation, utilisation, développement et maintenance.

Définitions

Logiciel

désigne un programme (code compilé ou source, librairies) et les documents nécessaires à son installation, utilisation, développement et maintenance.

Génie logiciel

désigne l'ensemble des méthodes, des techniques et d'outils concourant à la production d'un logiciel de qualité avec maîtrise des coûts et délais.

Les étapes (1)

Faisabilité Étude préalable qui questionne la faisabilité et la pertinence du projet, ainsi que les contraintes techniques (coût, temps, qualité) et les alternatives possibles.

Les étapes (1)

Faisabilité Étude préalable qui questionne la faisabilité et la pertinence du projet, ainsi que les contraintes techniques (coût, temps, qualité) et les alternatives possibles.

Spécification Description formelle du programme à développer, création d'un cahier des charges et des procédures de validation.

Les étapes (1)

Faisabilité Étude préalable qui questionne la faisabilité et la pertinence du projet, ainsi que les contraintes techniques (coût, temps, qualité) et les alternatives possibles.

Spécification Description formelle du programme à développer, création d'un cahier des charges et des procédures de validation.

Conception Choix techniques (architecture, technologies, librairies) adaptés à la spécification, et décision des tests d'intégration.

Les étapes (2)

Implémentation Développement du code source, tests unitaires et documentation.

Les étapes (2)

Implémentation Développement du code source, tests unitaires et documentation.

Intégration Assemblage du programme, tests d'intégration.

Les étapes (2)

Implémentation Développement du code source, tests unitaires et documentation.

Intégration Assemblage du programme, tests d'intégration.

Validation Tests de validation sur l'exécutable.

Les étapes (2)

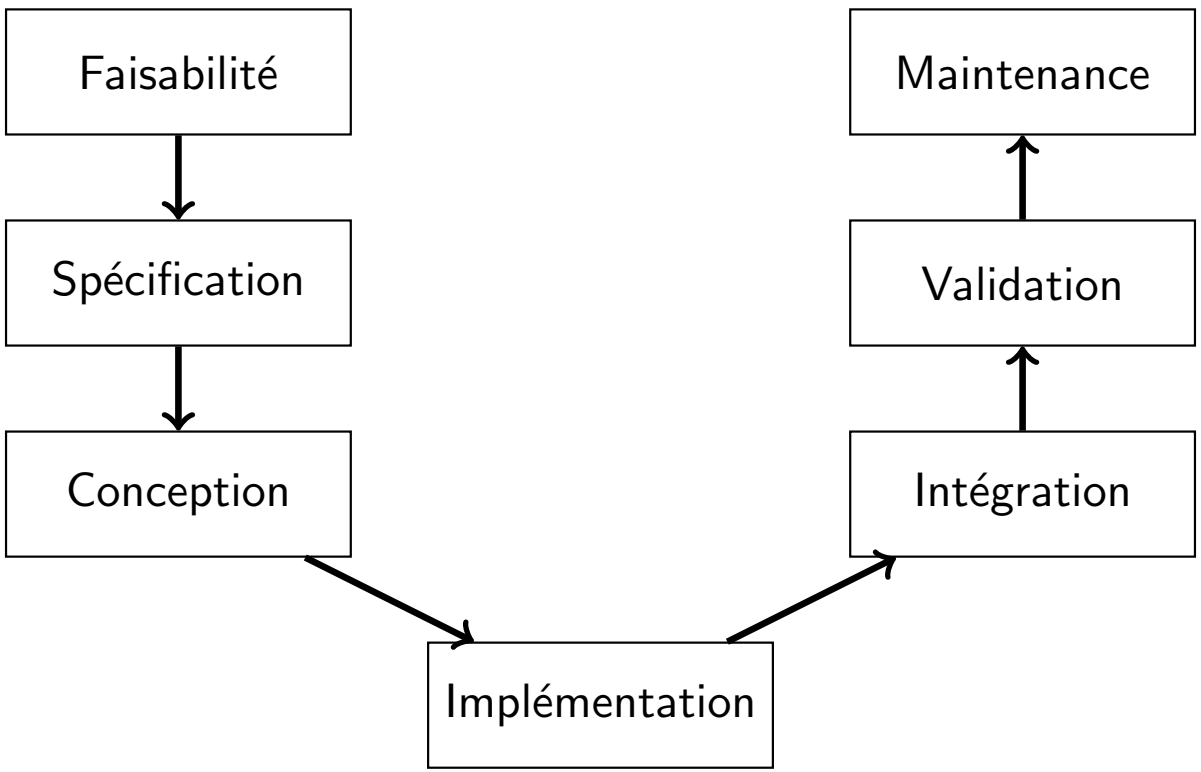
Implémentation Développement du code source, tests unitaires et documentation.

Intégration Assemblage du programme, tests d'intégration.

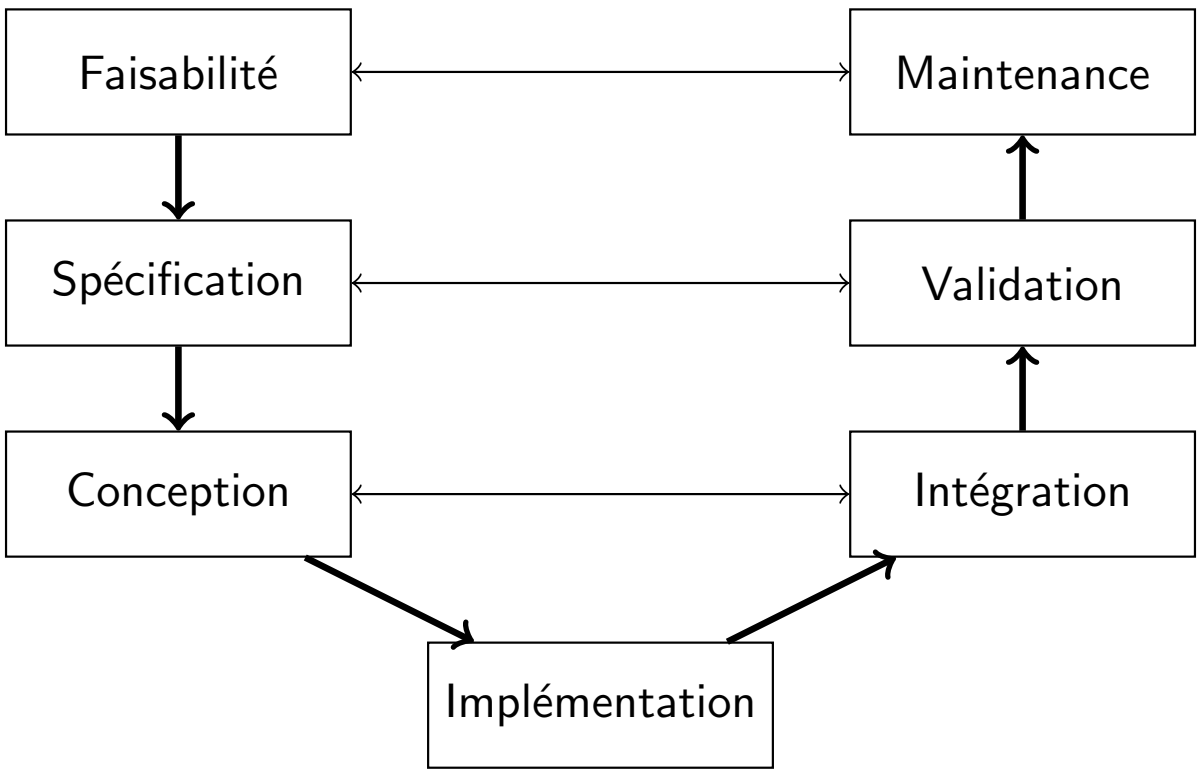
Validation Tests de validation sur l'exécutable.

Maintenance Correction, adaptation et amélioration du logiciel.

Cycle de vie d'un logiciel



Cycle de vie d'un logiciel



Qualité d'un logiciel (ISO 25010 [1])

Validité Remplir ses fonctions, suivre son cahier des charges et ses spécifications

- /!\ attribut de qualité particulier, requis absolu pour une application, insensé de comparer les autres critères si la fonctionnalité du logiciel n'est pas remplie.

Beauté de l'interface Degré de satisfaction et de plaisir qu'a un utilisateur à manipuler l'interface du logiciel

Compatibilité Interagir ou cohabiter avec d'autres.

- ex : nom de package unique en java

1. https ://iso25000.com/index.php/en/iso-25000-standards/iso-25010

Qualité d'un logiciel (ISO 25010)

Performance Remplir ses fonctions dans un temps donné, en utilisant un espace mémoire donné, gérer une taille de tâche donnée.

- ex : le caractère entré dans un champ de texte s'affiche en moins de 10 ms ; le logiciel peut tourner sur une carte arduino ; le serveur peut traiter 50 millions de requêtes simultanément.

Fiabilité Résister aux erreurs et conditions anormales (robustesse).

Qualité d'un logiciel (ISO 25010)

Utilisabilité Facilité d'usage de l'interface.

Sécurité Préserver la vie privé ou le secret professionnel, ainsi qu'à identifier correctement les utilisateurs.

Maintenabilité Être maintenu dans la durée, évoluer avec les besoins.

Portabilité Être facilement installé dans de nouveaux environnements.

Scénario de qualité

Scénario de qualité

Version textuelle d'une situation, correspondant à l'un des critères de qualité, dans lequel peut se retrouver le logiciel et des conséquences d'une telle situation.

Scénario de qualité

Scénario de qualité

Version textuelle d'une situation, correspondant à l'un des critères de qualité, dans lequel peut se retrouver le logiciel et des conséquences d'une telle situation.

Exemple

Lorsque l'utilisateur sauvegarde son document, le contenu du fichier correspondant dans le système est remplacé par celui actuellement dans l'éditeur, en moins de une seconde.

Documentation

Spécification Cahier des charges, conception, architecture

Documentation

Spécification Cahier des charges, conception, architecture

Technique Documentation de code source, description d'algorithme, documentation d'API, scripts de configuration et d'installation

Documentation

Spécification Cahier des charges, conception, architecture

Technique Documentation de code source, description d'algorithme, documentation d'API, scripts de configuration et d'installation

Utilisation Manuel d'utilisation contenant les informations d'utilisation pour les utilisateurs finaux.

Documentation

Spécification Cahier des charges, conception, architecture

Technique Documentation de code source, description d'algorithme, documentation d'API, scripts de configuration et d'installation

Utilisation Manuel d'utilisation contenant les informations d'utilisation pour les utilisateurs finaux.

- Chaque documentation a son public