

Synthèse Génie Logiciel

Le génie logiciel applique des principes d'ingénierie au développement logiciel. Il vise la réussite de projets logiciels à travers des méthodes, des bonnes pratiques et des outils. Il cherche à optimiser la production logicielle en équilibrant les coûts, les délais et la qualité, tout en considérant les besoins du client, des performances, de la fiabilité, de la facilité de maintenance et dévolution.

Dimensions du Génie Logiciel :

- Couverture : Tout le cycle de vie d'un logiciel.
- Activités Clés : Analyse des besoins, architecture, conception, production de code, gestion des versions, tests, documentation.
- Travail Collaboratif : Méthodologies et interactions avec l'équipe et les clients.

Architecture logicielle (Software architecture) : C'est la vue de haut niveau du système définissant la structure et les fonctions principales du logiciel. Elle concerne les choix de langages, les technologies, et les interactions entre les différentes sous-parties du logiciel.

Conception Logicielle (Software design) : C'est une phase qui détaille le fonctionnement opérationnel et les interactions des composants logiciels. S'occupe de l'organisation du code, du stockage de données, et des fonctionnalités des sous-parties.

Patrons logiciels (Software patterns) :

- Patrons d'Architecture : Modèles de haut niveau pour organisation et interactions (ex : MVC, micro-services).
- Patrons de conception (design pattern) : Solutions réutilisables pour résoudre les problèmes de conception spécifiques (ex : Singleton, Factory).

Production du Code Source :

- Convention de Nommage : Camel-case, snake-case, etc.
- Langage du Code : Généralement en anglais
- Formatage du Code : Règles pour organisation (indentation, espaces).
- Commentaires : Annotations pour clarifier le code.

Gestion des Versions, c'est la pratique de suivi et de gestion des modifications du code source, pour faciliter la collaboration et le suivi des changements :

- La gestion centralisée (ex : SVN) repose sur un serveur unique de contrôle de versions
- La gestion décentralisée (ex : GIT) permet à chaque développeur de travailler avec une copie complète du dépôt avec une suivi des modifications et une gestion des branches.

Tests : Tests de validation (fonctionnement et conformité aux exigences), tests d'intégration (Compatibilité entre modules ou composants du logiciel), tests unitaires (Fonctionnement de petites parties méthodes ou classes).

Documentation Technique : Pour développeurs, testeurs, ingénieurs. Elle couvre l'architecture, la conception et les choix techniques. Elle utilise l'UML pour la modélisation.

Documentation du Code Source : Pour développeurs. Elle détaille les fonctions, les classes, les méthodes.

Documentation Utilisateur : Pour les clients et les utilisateurs finaux. Elle inclut des guides d'utilisation, une FAQ, des procédures. Elle peut utiliser des captures d'écran ou des schémas.