

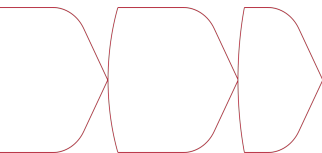
Introduction au Génie Logiciel

Comprendre les bases du développement logiciel

Slimani Mohamed Amine

EHTP

February 5, 2025



Sommaire

Qu'est-ce que le Génie Logiciel ?

Pourquoi le Génie Logiciel est-il important ?

Cycle de vie du développement logiciel (SDLC)

Modèles de développement

Outils et technologies

Bonnes pratiques

Exemple de workflow agile avec Scrum

Exemple de gestion de version avec Git

Défis du Génie Logiciel

Pourquoi c'est important ?

Qu'est-ce que le Génie Logiciel ?

- ▶ **Définition** : Le génie logiciel est l'application d'une approche systématique, disciplinée, et quantifiable au développement, à l'exploitation, et à la maintenance des logiciels.
- ▶ **Objectif** : Produire des logiciels de haute qualité, fiables, et maintenables.
- ▶ **Historique** : Né dans les années 1960 pour répondre à la complexité croissante des projets logiciels.

Pourquoi le Génie Logiciel est-il important ?

- ▶ **Qualité** : Assurer la fiabilité et la performance des logiciels.
- ▶ **Gestion de projet** : Planifier, organiser, et contrôler les ressources.
- ▶ **Maintenance** : Faciliter les mises à jour et les corrections.

Cycle de vie du développement logiciel (SDLC)

- ▶ **Analyse des besoins** : Comprendre les exigences du client.
- ▶ **Conception** : Définir l'architecture et les spécifications techniques.
- ▶ **Implémentation** : Écrire le code.
- ▶ **Tests** : Vérifier que le logiciel fonctionne comme prévu.
- ▶ **Déploiement** : Mettre le logiciel en production.
- ▶ **Maintenance** : Corriger les bugs et ajouter de nouvelles fonctionnalités.

Modèles de développement

- ▶ **Modèle en cascade** : Étapes séquentielles (analyse, conception, implémentation, tests, déploiement).
- ▶ **Modèle itératif** : Développement par cycles itératifs.
- ▶ **Modèle agile** : Approche flexible et collaborative (ex : Scrum, Kanban).

Outils et technologies

- ▶ **Gestion de version** : Git, SVN.
- ▶ **Gestion de projet** : Jira, Trello.
- ▶ **Tests automatisés** : Selenium, JUnit.
- ▶ **Intégration continue** : Jenkins, GitHub Actions.

Bonnes pratiques

- ▶ **Documentation** : Maintenir une documentation claire et à jour.
- ▶ **Revue de code** : Améliorer la qualité du code grâce aux retours des pairs.
- ▶ **Tests unitaires** : Vérifier le bon fonctionnement des composants individuels.

Exemple de workflow agile avec Scrum

- ▶ **Sprint planning** : Planifier les tâches pour le sprint.
- ▶ **Daily stand-up** : Faire le point sur l'avancement.
- ▶ **Sprint review** : Présenter les résultats du sprint.
- ▶ **Sprint retrospective** : Identifier les améliorations pour le prochain sprint.

Exemple de gestion de version avec Git

Commandes Git

```
# Cloner un dépôt
git clone https://github.com/user/repo.git

# Créer une branche
git checkout -b feature-branch

# Faire un commit
git add .
git commit -m "Ajouter une nouvelle fonctionnalité"

# Pousser les modifications
git push origin feature-branch
```

Défis du Génie Logiciel

- ▶ **Gestion des exigences** : Capturer et gérer les besoins changeants des clients.
- ▶ **Complexité** : Gérer la complexité croissante des systèmes logiciels.
- ▶ **Sécurité** : Protéger les logiciels contre les vulnérabilités.

Pourquoi c'est important ?

- ▶ Le génie logiciel est essentiel pour la création de logiciels de qualité.
- ▶ Il englobe des méthodologies, des outils, et des bonnes pratiques pour gérer les projets logiciels.
- ▶ Comprendre le génie logiciel est crucial pour les développeurs et les chefs de projet.

Le génie logiciel est une discipline clé pour la création de logiciels de qualité, fiables, et maintenables. Adoptez les bonnes pratiques et les outils modernes pour réussir vos projets logiciels !