

COMMENT L'IA VA TRANSFORMER LE CYCLE DE DÉVELOPPEMENT LOGICIEL^(IA)

[expleo]

1 Planification & analyse des exigences

Les grands modèles de langage (LLM) peuvent assister les business analysts ou product owners dans la relecture des cahiers des charges afin d'identifier les ambiguïtés et les incohérences. En parallèle, l'analytique prédictive peut identifier les risques potentiels dans la définition du périmètre, à partir des données de projets passés.

Réduit le risque de reprises coûteuses, améliore l'alignement avec les objectifs business et renforce la conformité dès le départ.

2 Conception et architecture

L'IA peut aider les architectes et ingénieurs logiciels en automatisant l'analyse de code legacy pour cartographier les dépendances, suggérer une approche de refactorisation afin de passer de systèmes monolithiques à une architecture microservices, et simuler l'architecture pour évaluer la scalabilité, les performances et les compromis de sécurité avant le codage.

Réduit les coûts et les délais liés aux systèmes IT legacy et diminue le risque de reprises coûteuses après la phase de déploiement.

3 Codage et implémentation

Les développeurs passent du codage manuel au vibe coding (co-crédation), où des outils comme GitHub Copilot et Cursor génèrent des composants logiciels entiers (de l'architecture et la logique aux interfaces et à la documentation) à partir de prompts en langage naturel. Cependant, l'agentic coding (autonomie de l'IA) devrait bientôt devenir la norme.

Augmente la productivité des équipes de développement tout en accélérant le time-to-market.

5 Déploiement et production

L'IA peut aider les ingénieurs DevOps à automatiser la génération de templates Infrastructure-as-Code (IaC) pour des environnements multi-cloud ou hybrides, et à assurer une surveillance continue avec détection d'anomalies.

Réduit les coûts et délais de déploiement, améliore la fiabilité du système.

4 Tests et assurance qualité

L'IA peut aider les analystes qualité à générer des cas de test fonctionnels et assister les ingénieurs qualité dans l'automatisation des tests d'API (Application Programming Interface), d'UI (User Interface) et de performance grâce à des scripts auto-apprenants qui s'adaptent aux évolutions de l'application.

Étend la couverture des tests à moindre coût, améliorer la fiabilité des mises en production et renforcer la confiance des parties prenantes.

