

## Le génie logiciel

Le génie logiciel (software engineering) représente l'application de principes d'ingénierie au domaine de la création de logiciels. Il consiste à identifier et à utiliser des méthodes, des pratiques et des outils permettant de maximiser les chances de réussite d'un projet logiciel.

Ce dernier possède différents enjeux, atteint à travers différents processus d'optimisation :

Le respect des besoins du client, des délais, la maximisation des performances et la facilité de la maintenance et scalable.

La réalisation de ces logiciels est soumise a diverses exigences, le cout, la qualité et le délai.

Le génie logiciel couvre l'ensemble du cycle de vie d'un logiciel. Il étudie toutes les activités qui mènent d'un besoin à la livraison du logiciel, y compris dans ses versions successives, jusqu'à sa fin de vie.

Les dimensions du génie logiciel sont donc multiples :

analyse des besoins du client, définition de l'architecture du logiciel, choix de conception, règles et méthodes de production du code source, gestion des versions, test du logiciel, documentation, organisation de l'équipe et interactions avec le client.

La très grande majorité des projets logiciels sont menés par des équipes de plusieurs développeurs. Il est de plus en plus fréquent que ces développeurs travaillent à distance ou en mobilité.

Le travail en équipe sur un projet logiciel nécessite de pouvoir :

partager le code source entre membres de l'équipe, gérer les droits d'accès au code, intégrer les modifications réalisées par chaque développeur, signaler des problèmes ou proposer des améliorations qui peuvent ensuite être discutés collectivement.

Pour répondre à ces besoins, des plates-formes de publication et de partage de code en ligne sont apparues. On les appelle parfois des **forges logicielles**. La plus importante à l'heure actuelle est la plate-forme GitHub.

## Bonnes pratiques :

La langue utilisée dans la production du code doit bien entendu être unique sur tout le projet.

Les paramètres de formatage les plus courants sont :

- *taille des tabulations (2 ou 4 espaces) ;*
- remplacement automatique des tabulations par des espaces ;

- *passage ou non à la ligne après chaque accolade ouvrante ou fermante ;*
- ajout ou non d'une espace avant une liste de paramètres ;

L'ajout de commentaires permet de faciliter la lecture et la compréhension d'une portion de code source. L'ensemble des commentaires constitue une forme efficace de documentation d'un projet logiciel.

Il n'y a pas de règle absolue ni de consensus, en matière de taux de commentaires dans le code source. Certaines méthodologies de développement agile (*eXtreme Programming*) vont jusqu'à affirmer qu'un code bien écrit se suffit à lui-même et ne nécessite aucun ajout de commentaires.