

# Synthèse

# Génie Logiciel

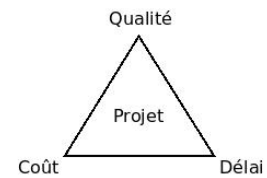


MAHMOUD CHARIF

Développeur Informatique Junior en  
Alternance

Le génie logiciel (software engineering) englobe tous les principes dans le domaine de la création logiciel et regroupe plusieurs principes qui maximise les chances de réussite d'un projet logiciel sur le long terme.

Comme tout projet, la réalisation d'un logiciel est soumise à des exigences contradictoires et difficilement conciliables (triangle coût-délai-qualité).



Ce que je retiens du génie logiciel sont les bonnes méthodes de création logiciel répondant à un problème spécifique tout en prenant compte les enjeux et les contraintes et en se basant sur un modèle de qualité comme ISO/CEI 9126.

Ces solutions prennent souvent la forme de patrons logiciels (software patterns) ou de patrons de conception (design patterns) qui ont été conçus pour répondre à des problèmes spécifiques.

Chaque problème a potentiellement une architecture ou un patron de conception qui lui correspond le mieux. Par exemple, pour des applications nécessitant une grande évolutivité, une architecture basée sur des micro-services pourrait être appropriée. Pour le développement d'interfaces utilisateur en WPF, l'architecture Modèle-View-ViewModel (MVVM) est souvent utilisé.

Pour assurer le succès d'un projet mené en équipe, l'utilisation d'outils de contrôle de version tels que GIT ou SVN permettent de gérer et de conserver l'historique des différentes versions d'un ensemble de fichiers, qui représentent généralement le code source d'un logiciel. Ils favorisent également la collaboration en introduisant le concept de branches, qui permet aux développeurs de travailler sur des fonctionnalités ou des correctifs de manière isolée avant de les intégrer dans la branche principale du projet.

Avec GIT ou SVN, par exemple, il existe plusieurs flux de travail (workflows) que les équipes peuvent adopter.

Il est essentiel de définir la finalité du projet. Cette étape permet d'établir une vision claire des objectifs à atteindre et sert de fondement à toutes les décisions futures concernant le projet.

Il est important de modéliser des diagramme UML pour avoir une vision globale du logiciel en essayant d'appliquer les principes SOLID réfléchir sur le programme avant de commencer à coder dans l'objectif de maximiser la qualité logiciel ainsi que sa maintenabilité.