

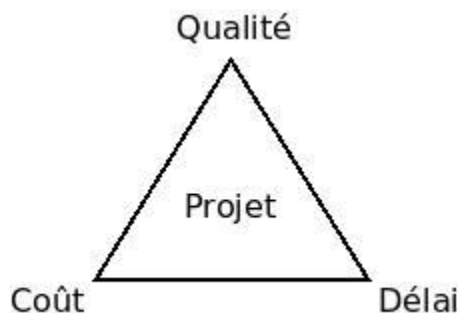
Génie logiciel

Le génie logiciel concerne tous ce qui s'applique à la création de logiciels, que ce soit la conception, le développement et la fabrication de systèmes informatiques, aussi bien d'un point de vue matériels que logiciels.

Le génie logiciel vise à rationaliser et à optimiser le processus de production d'un logiciel. Les enjeux associés sont multiples :

- Adéquation aux besoins du client ;
- Respect des délais de réalisation prévus ;
- Maximisation des performances et de la fiabilité ;
- Facilitation de la maintenance et des évolutions ultérieures.

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



Le génie logiciel couvre l'ensemble du cycle de vie d'un logiciel. Cycle de vie qui désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. L'objectif est de permettre de définir des jalons intermédiaires permettant la validation du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la vérification du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre.

L'IDE :

Un IDE ou environnement de développement intégré est une application logicielle combinant au même endroit tous les outils nécessaires pour un projet de développement logiciel.

Il fournit une interface permettant d'écrire du code, d'organiser des groupes de texte, et d'automatiser les tâches redondantes de programmation.

Plutôt qu'un éditeur de code nu, les IDE combinent les fonctionnalités de multiples processus de programmation en un seul. On retrouve au minimum un éditeur, un compilateur, un débogueur, et des fonctionnalités de complétion de code ou de gestion de code générique. Les plus avancés offrent aussi des capacités de data visualization, de tracing ou de référencement croisé.

Par exemple Microsoft Visual Studio est un IDE.

Les conventions de codage :

Il existe plusieurs conventions de codage, notamment le « coding style » qui désigne un ensemble de règles et de conseils adoptés par les membres d'un projet logiciel pour écrire et mettre en forme du code.

Les conventions de codage visent essentiellement à améliorer la lisibilité du code : elles doivent permettre au programmeur d'identifier « du premier coup d'œil » un maximum de choses dans le code, de se repérer facilement, de savoir où trouver les choses, etc. Une fois adoptés, elles facilitent grandement l'écriture, la maintenance et aident à éviter certaines erreurs.

Tous les langages n'ayant pas forcément de convention de codage spécifiée, il peut être important de se mettre d'accord avec son groupe de projet sur une convention de codage.

Plusieurs critères peuvent rentrer en compte :

- Limiter la taille des modules, écrire des fonctions courtes et simples
- Nombre de caractère limite par ligne (généralement entre 80 et 100 maximum)
- Indentation du code (espace / tabulation)
- Convention de nommage des fonctions, variables, constantes, ... (Mettre des majuscules ou des underscore)
- Et le commentaire du code (ni trop commenté, ni pas assez, commenter uniquement tous ce qui gagne à l'être, et non ce qui clair)

Il existe de nombreux outils permettant d'évaluer la qualité du code, comme par exemple SonarQube, ou encore Visual Assist ou bien DeepScan, ces outils permettent l'analyse de la qualité du code et de la sécurité.

Les débogueurs :

Tout d'abord le débogage correspond à l'action d'éliminer les bogues. Le débogage est un processus de diagnostic, de localisation et d'élimination des erreurs des programmes, ce processus permet aussi la vérification, de tester le programme en cours d'élaboration.

Les débogueurs sont des outils très puissants. Lorsque vous lancez votre programme au sein du débogueur, celui-ci suit le comportement de votre application et si un crash se produit, il s'arrête sur la ligne provoquant l'erreur. À partir de là, votre programme n'est pas réellement arrêté, il est en pause, mais ne pourra pas continuer. Le débogueur vous permet d'analyser son état, de voir les valeurs des variables, d'afficher la liste des fonctions appelées et ainsi de comprendre comment le programme en est arrivé là.

Avec les débogueurs il est également possible d'indiquer à votre débogueur de s'arrêter à une ligne précise de l'application afin d'analyser l'état de celle-ci à cet endroit précis du code, ou même de l'exécuter ligne par ligne pour comprendre comment la machine interprète votre code.

Il existe plusieurs débogueurs, notamment GDB, Code::Blocks, Microsoft Visual Studio ou encore Eclipse.

Le versioning :

Les systèmes de contrôle de version (versioning) sont une catégorie d'outils logiciels qui permettent de gérer les changements apportés à un code source au fil du temps. Grâce au versioning, il est possible de garder une trace de chaque modification du code dans une base de données dédiée. En cas d'erreur, les développeurs peuvent alors revenir en arrière et comparer les versions antérieures du code pour corriger les éventuelles erreurs.

Il existe plusieurs outils de versioning, par exemple GIT (GitLab / GitHub), AWS CodeCommit, Bitbucket ou encore Google Cloud Source Repositories.