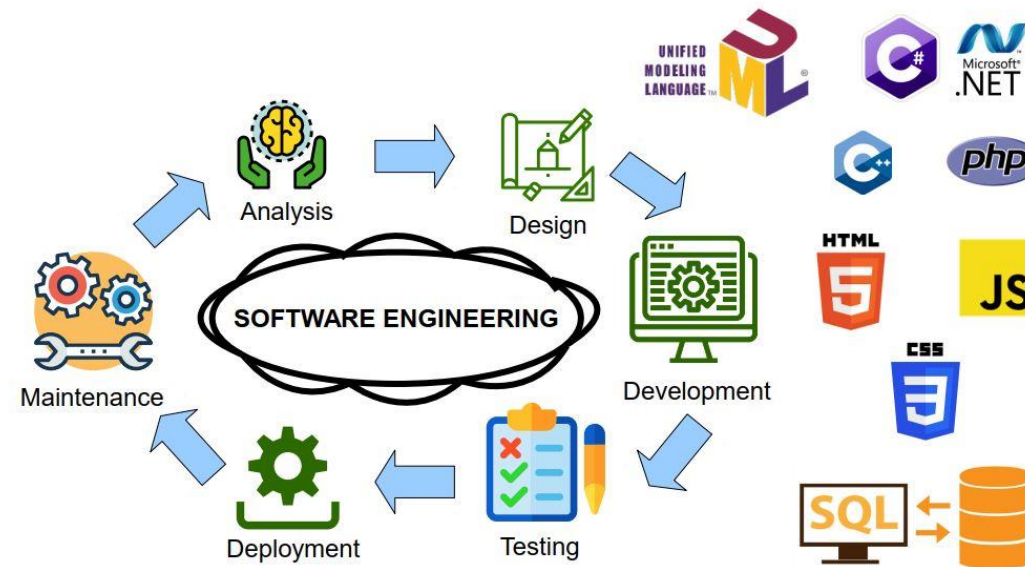


INF3523

Architecture et Génie des Logiciels



Dr El Hadji Bassirou TOURE
Université Assane Seck de Ziguinchor
2022 - 2023

Le Génie Logiciel

Le Génie Logiciel ou l'Ingénierie Logicielle

- Systèmes informatiques
 - 80 % de logiciel & 20 % de matériel
- Principaux fabricants de matériel (bien maîtrisé et fiable)



- Problèmes de l'informatique sont relatifs au logiciel

Spécificités du logiciel

- Un produit immatériel, dont l'existence est indépendante du support physique
- Un objet technique fortement contraint
- Un cycle de production différent
 - La reproduction pose peu de problèmes,
 - Seule la première copie d'un logiciel a un coût

Crise du logiciel ==> Génie Logiciel

- 1968 : Crise du logiciel ☹️
- L'informatique ne répond pas aux attentes
- L'informatique coûte très cher
- Aujourd'hui ☹️

Percentage of Projects That Fail

The statistics for software project delivery show that **only one in three software projects are truly successful**.

According to Standish Group's **Annual CHAOS report**, 66% of technology projects (based on the analysis of 50,000 projects globally) end in partial or total failure.

Just one in three software projects are considered truly successful with large projects most at risk of failure

	SUCCESSFUL	CHALLENGED	FAILED
Large	8%	26%	41%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%

Software Project Failure Rate

- <https://www.atlascode.com/blog/why-software-projects-fail/>

Le Génie Logiciel ou l'Ingénierie Logicielle

- L'expression « Génie Logiciel » (Software Engineering) est né 😊
- Comment faire des logiciels de qualité ? 😞
- Quels sont les critères de qualité pour un logiciel ? 😞
- Qu'attend-on d'un logiciel ? 😞

Qualité du logiciel

- Les qualités du logiciel:
 - Externes: observables par l'utilisateur.
 - Internes: d'intérêt pour le développeur du logiciel
 - Influencent directement les qualités externes.
 - Ex. Le logiciel doit être vérifiable (interne) pour pouvoir affirmer qu'il est fiable (externe).

Qualités attendues

- Qualité : Utilité
 - Adéquation entre
 - Le besoin effectif de l'utilisateur
 - Les fonctions offertes par le logiciel
 - Réponse
 - Bonne analyse des besoins
 - Bonne communication (langage commun, démarche participative) entre acteurs

Qualités attendues

- Qualité : Utilisabilité
 - « Degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité, efficience et satisfaction, dans un contexte d'utilisation spécifié » (ISO 9241-11,1998)
 - Facilité d'apprentissage : comprendre ce que l'on peut faire avec le logiciel, et savoir comment le faire
 - Facilité d'utilisation : importance de l'effort nécessaire pour utiliser le logiciel à des fins données
- Convivialité
 - Qualité d'un logiciel que les utilisateurs trouvent agréable et facile à utiliser.
- Réponse
 - Bonne maîtrise du mode opératoire des utilisateurs
 - Bonne adaptation de l'ergonomie des logiciels aux utilisateurs

Qualités attendues

- Qualité : Fiabilité
 - Correction, justesse, conformité :
 - le logiciel est conforme à ses spécifications, les résultats sont ceux attendus
 - Robustesse, sûreté :
 - le logiciel fonctionne raisonnablement en toutes circonstances,
 - rien de catastrophique ne peut survenir, même en dehors des conditions d'utilisation prévues
- Réponse
 - Utilisation de méthodes formelles, de langages etc.
 - Vérifications, tests
- Tout logiciel correct est-il fiable ?

Qualités attendues

- Qualité : Robustesse
 - Qualité du logiciel qui se comporte de façon convenable MÊME dans les circonstances non prévues par la spécification.
 - Comportement du logiciel possiblement non spécifié en cas de
 - mauvaises entrées
 - pannes matérielles
 - ajout de composants
 - etc.
 - Si la spécification décrit également les exigences concernant le traitement des cas exceptionnels:
 - correction = robustesse ?

Qualités attendues

- Qualité : Interopérabilité
 - Un logiciel doit pouvoir interagir en synergie avec d'autres logiciels
- Réponse
 - SOA (Architectures Orientées Service)
 - Microservices
 - API

Qualités attendues

- Qualité : Performance
 - Les logiciels doivent satisfaire aux contraintes de temps d'exécution
 - Les logiciels doivent aussi supporter la scalabilité (montée à l'échelle)
 - Mesures de performance
 - Complexité algorithmique
 - Mesure – monitoring (collecte de données, identif. bouchons)
 - Analyse - construire un modèle et l'analyser (preuves math.)
 - Simulation - construire un modèle et l'exécuter (stat. et prob.)

Qualités attendues

- Qualité : Portabilité
 - Un même logiciel doit pouvoir fonctionner sur plusieurs machines/OS différents
- Réponse
 - Rendre le logiciel indépendant de son environnement d'exécution (voir interopérabilité)
 - Techniques de virtualisation
 - Faire l'hypothèse d'une configuration minimale (facilités et mémoire)
 - Utiliser technologies permettant de déterminer la config. et de s'adapter à elle.

Qualités attendues

- Qualité : Réutilisabilité
 - Ne pas réinventer la roue, si possible :
 - 80 % du code est du « tout venant » qu'on retrouve à peu près partout
 - 20 % du code est spécifique
 - Réponse
 - tech. conception orientées objet (instanciation, abstraction, généricité)
 - Construire un logiciel à partir de composants prêts à l'emploi
 - « Design Patterns »
 - tech. conception à base de composants (code reuse)

Qualités attendues

- Qualité : Vérifiabilité
 - Peut être vérifié facilement (par analyse formelle ou test).
 - Quelles propriétés peut-on vouloir vérifier? Comment ?
 - correction
 - fiabilité
 - performance
 - etc.

Qualités attendues

- Qualité : Maintenance facile
 - Qualité du logiciel dont l'entretien est facilement réalisable.
 - Réduire la quantité de maintenance corrective (zéro défaut)
 - Rendre moins coûteuses les autres types de maintenance
 - Adaptative, perfective, préventive, etc.
- Réponse
 - Réutilisabilité, modularité
 - Vérifier, tester
 - Structures de données complexes et algorithmes simples
 - Anticiper les changements à venir

Qualités attendues

- Qualité : Maintenance facile
 - Types de maintenance:
 - Corrective(20% des coûts):
 - Enlever les erreurs résiduelles ou introduites pendant la maintenance.
 - Adaptative (20%):
 - Ajuster le logiciel suite à la modification de son environnement.
 - Perfective (+50%):
 - Modifier le logiciel pour améliorer ses qualités (ajout de nlls fcts, performance, etc.)

Qualités du logiciel : Points de vue

- La qualité est perçue différemment selon
 - Le point de vue de celui qui l'observe: développeur, utilisateur, client, etc.
 - Le domaine (contexte): logiciel critique? Editeur de texte?
- Les différents points de vue de la qualité selon Garvin
 - Vue transcendantale: on sait reconnaître la qualité mais on ne peut la définir.
 - Vue "utilisateur": Évalue si le logiciel répond à l'usage qu'on désire en faire.
 - Vue "fabriquant" (manufacturing): Évalue si le logiciel est conforme à sa spécification.
 - Vue "produit": Évalue si le logiciel offre les caractéristiques attendues pour le type de produit.
 - Vue basée sur la valeur: Évalue ce que le client est prêt à payer pour le logiciel.

Qualités et type de logiciels

- Systèmes d'information et de commerce électronique
 - Gestion d'informations: création/destruction, consultation et mise à jour de données.
 - Qualités délicates:
 - Intégrité des données
 - Sécurité
 - Disponibilité des données
 - Performance des transactions
 - Convivialité.

Qualités et type de logiciels

- Systèmes temps réel
 - Doit répondre aux événements dans un laps de temps limité.
 - Temps de réponse est une question de correction et non de performance.
 - Ex.: système de monitoring de la température, système de guidage d'une sonde
- Qualités délicates
 - Correction, fiabilité, robustesse
 - Interopérabilité, convivialité

Qualités et type de logiciels

- Systèmes distribués
 - Ordinateurs indépendants connectés par un réseau de communication.
 - Caractéristiques
 - Distribution: données ou contrôle?
 - Qualités délicates
 - fiabilité, robustesse
 - performance
 - interopérabilité

Qualités et type de logiciels

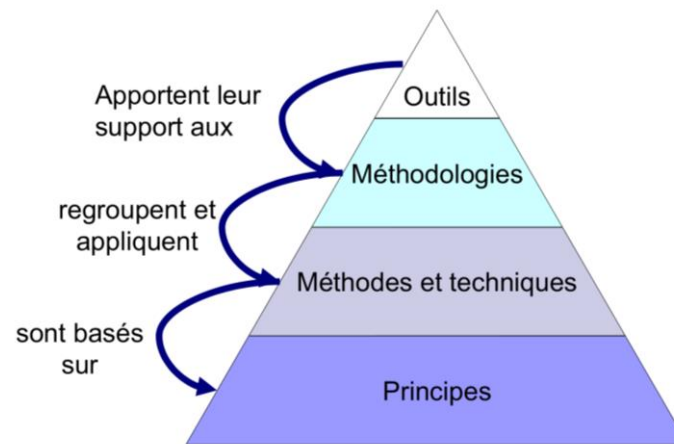
- Systèmes embarqués
 - Système intégré à un dispositif, une machine ou un autre système pour le piloter.
 - Exemples: système de freinage automobile.
 - Caractéristique
 - Interface avec autre système (non humain).
 - Qualités délicates
 - Interopérabilité
 - réutilisation

Qualité du logiciel : Métrique

- Comment mesure-t-on la qualité?
 - Ex. On évalue la fiabilité d'un pont en mesurant la pression qu'il peut supporter.
 - 1. Définir avec précision la qualité à évaluer.
 - 2. Etablir une métrique i.e. formule qui permet d'évaluer une qualité à partir de différents paramètres mesurables.

Définition du Génie Logiciel

- GL : « Discipline de l'informatique qui regroupe un ensemble de connaissances, de procédés et des acquis scientifiques pour la conception, la mise en œuvre, la vérification et la documentation de logiciels dans le but d'en optimiser la production, le support et la qualité »
- GL : « Le génie logiciel c'est un processus de résolution de problème, en utilisant des méthodes, des techniques et des outils informatiques. »



Problèmes du logiciel

- Une erreur peut conduire à l'introduction d'un défaut (faute ou bogue)
- Si un défaut dans le code est exécuté, cela peut (ou ne pas) causer une défaillance
- Les erreurs peuvent survenir pour de nombreuses raisons :
 - contraintes de temps,
 - faillibilité humaine,
 - inexpérience ou manque de compétence,
 - manque de communication,
 - complexité,
 - nouvelles technologies, etc.

Problèmes du logiciel

- Certaines défaillances sont causées par des défauts dans le code
- D'autres peuvent être causées par les conditions environnementales
 - climat, température, etc.
- Les résultats inattendus ne sont pas tous des défaillances.
 - Des faux positifs peuvent se produire en raison d'erreurs dans la façon dont les tests ont été exécutés,
 - ou en raison de défauts dans les données de test, l'environnement de test

Erreurs historiques dans les logiciels

- La sonde Mariner vers Vénus s'est perdue dans l'espace à cause d'une erreur dans un programme FORTRAN.
- En 1981, le premier lancement de la navette spatiale a été retardé de deux jours à cause d'un problème logiciel.
- L'explosion d'Ariane 5, le 4 juin 1996, a coûté un demi milliard de dollars (non assuré !)

Développement professionnel de logiciels

Produits logiciels

- Produits génériques
 - Les systèmes logiciels commercialisés et pouvant être vendus à n'importe quel client souhaitant l'acheter.
 - Exemples : Logiciels PC de traitement de textes, logiciels pour un marché spécifique comme les logiciels de gestion bibliothèques.
- Produits personnalisés (sur mesure)
 - Les systèmes logiciels commandés par un client en particulier pour assurer un besoin spécifique.
 - Exemples : Les systèmes de gestion d'un business métier bien défini.

Spécification de produits logiciels

- Produits génériques
 - La spécification du logiciel est faite par le constructeur du logiciel. C'est ce dernier qui décide de quand et comment appliquer des modifications au logiciel.
- Produits personnalisés (sur mesure)
 - La spécification du logiciel est faite par le client qui décide également des modifications à apporter au logiciel (avec le concours du constructeur).

Génie logiciel

- Le génie logiciel est une discipline d'ingénierie qui s'intéresse à tous les aspects de la production du logiciel allant de la spécification à la maintenance du logiciel.
 - Discipline d'ingénierie
 - Utilisation de théories et méthodes appropriées pour la résolution de problèmes en tenant en compte des contraintes organisationnelles et financières.
 - Tous les aspects de la production du logiciel
 - Pas seulement les procédés techniques de développement utilisés mais également la gestion de projet et le développement d'outils, de méthodes, ..., pour faciliter la production de logiciels.

Processus logiciel

- Une approche de développement utilisée en génie logiciel est souvent appelée un processus logiciel.
- Un processus logiciel est une séquence d'activités qui débouchent sur la production de produits logiciels.
- Quatre principales activités sont communes à tous les processus logiciels.
 - Spécification logicielle
 - Développement logiciel
 - Validation logicielle
 - Evolution logicielle

Activités de processus logiciel

- Spécification logicielle, où les clients et les développeurs définissent le logiciel qui doit être produit ainsi que les contraintes de sa mise en opération.
- Développement logiciel, où le logiciel est conçu et programmé.
- Validation logicielle, où le logiciel est vérifié pour assurer qu'il satisfait les exigences du client.
- Evolution logicielle, où le logiciel est modifié pour s'adapter aux changements des besoins et des exigences du marché.

Principes fondamentaux du génie logiciel

- Quelques principes fondamentaux s'appliquent à tous les types de systèmes, peu importe les techniques de développement employées :
 - « La qualité du processus de fabrication est garante de la qualité du produit »
 - Pour obtenir un logiciel de qualité, il faut en maîtriser le processus d'élaboration
 - La vie d'un logiciel est composée de différentes étapes
 - La succession de ces étapes forme le cycle de vie du logiciel
 - Il faut contrôler la succession de ces différentes étapes

Ethique du génie logiciel

Ethique du génie logiciel

- Le génie logiciel implique de plus grandes responsabilités qu'une simple utilisation de compétences techniques.
- Les ingénieurs du logiciel doivent se comporter de manière honnête et (éthiquement) responsable s'ils veulent se faire respecter comme des professionnels.
- Un comportement éthique est plus que le respect d'un ensemble de lois mais implique également le suivi d'un ensemble de principes moraux.

Enjeux d'une responsabilité professionnelle

- Confidentialité
 - Les ingénieurs doivent normalement respecter la confidentialité de leurs employeurs ou clients, peu importe si un accord formel de confidentialité a été signé ou pas.
- Compétence
 - Les ingénieurs ne doivent pas usurper leur niveau de compétence. Ils ne doivent pas accepter un travail qui est en dehors de leurs compétences.

Enjeux d'une responsabilité professionnelle

- Droits de propriété intellectuelle
 - Les ingénieurs doivent être conscients des lois locales gouvernant l'utilisation de propriétés intellectuelles telles que les brevets, copyrights, ... Ils doivent s'assurer que les droits de propriété intellectuelle sont respectés.
- Mésusage des ordinateurs
 - Les ingénieurs ne doivent pas utiliser leurs compétences techniques pour mésuser les ordinateurs d'autres personnes. Le mésusage d'ordinateurs peut aller du trivial (jouer sur des machines de son employeur) à du très sérieux (propager du virus)

Code d'éthique de l'IEEE/ACM

- Les organisations professionnelles des USA ont coopéré pour produire un code de pratiques éthiques.
- Les membres de ces organisations signent le code en adhérant.
- Le code contient huit principes et est établi par les professionnels du génie logiciel (praticiens, enseignants, chercheurs, étudiants, etc.).

Code d'éthique de l'IEEE/ACM

Software Engineering Code of Ethics and Professional Practice

ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices

PREAMBLE

The short version of the code summarizes aspirations at a high level of the abstraction; the clauses that are included in the full version give examples and details of how these aspirations change the way we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the aspirations and the details form a cohesive code.

Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

Code d'éthique de l'IEEE/ACM

1. PUBLIC - Software engineers shall act consistently with the public interest.
2. CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
3. PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
4. JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.
5. MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. COLLEAGUES - Software engineers shall be fair to and supportive of their colleagues.
8. SELF - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.