



Management de la Qualité (Qualité Logicielle)

Pr Abdelali ZAKRANI & Pr Said NOUH

Plan

- ❖ Introduction
 - L'importance de la qualité logicielle
 - Désastres historiques des logiciels
- ❖ Modèle de qualité logicielle (ISO/IEC 25010:2011)
 - Modèle de qualité d'utilisation
 - Modèle de qualité du produit
 - Caractéristiques de la qualité d'utilisation
 - Caractéristiques de la qualité du produit
 - L'assurance qualité
- ❖ Principes pour la maintenabilité (SOLID)
 - Responsabilité unique (Single responsabilité)
 - Ouverture/fermeture (Open/Closed)
 - Substitution de Liskov (Liskov substitution)
 - Séparation des interfaces (Interfaces Segregation)
 - Inversion des dépendances (Dependency inversion)
- ❖ Tests:
 - Tests unitaires
 - Tests d'intégration
 - Inspection continue de la qualité du code
 - Tests fonctionnels
 - Test de performance

Chapitre 1:

Introduction, l'importance de la qualité logicielle et citation de quelques désastres historiques dû à des logiciels

Pr Abdelali ZAKRANI & Pr Said NOUH

3

Introduction & définitions

Spécificité des logiciels

- Un produit immatériel, dont l'existence est indépendante du support physique
- Un objet technique fortement contraint
 - Fonctionne ou ne fonctionne pas
 - Structure complexe
 - Relevé des modes de travail du domaine technique
- Un cycle de production différent
 - La reproduction pose peu de problèmes, seule la première copie d'un logiciel a un coût
 - Production à l'unité
 - Semblable au Génie Civil (ponts, routes...)

4

Spécificité des logiciels

Exemple:



Produit palpable	Produit non palpable
Produit manufacturé	Produit Développé
Assemblage facile de composants	Assemblage difficile (réutilisation)
Maintenance Par Remplacement des composants	Maintenance par l'équipe de développement elle-même
Procédé de développement industrialisé	Procédé de développement entièrement basé sur les humains

5

Domaine des ‘sciences de l’ingénieur’ dont la finalité est la *conception*, la *fabrication* et la *maintenance* de systèmes *logiciels complexes, sûrs et de qualité* (‘Software Engineering’ en anglais). Aujourd’hui les économies de tous les pays développés sont dépendantes des systèmes logiciels.

Le génie logiciel (GL) se définit souvent par opposition à la ‘programmation’, c’est à dire la production d’un programme par un individu unique, considérée comme ‘facile’.

Dans le cas du GL il s’agit de la fabrication *collective* d’un *système complexe*, concrétisée par un ensemble de documents de conception, de programmes et de **jeux de tests** avec souvent de *multiples versions*, et considérée comme ‘difficile’.

6

Le triangle QCD (qualité, coût, délais)



7

Qu'est ce que le génie logiciel: Définition

Règle du CQFD : Coût Qualité Fonctionnalités Délai

Le GL se préoccupe des *procédés de fabrication des logiciels* de façon à s'assurer que les 4 critères suivants soient satisfaits.

- Le système qui est fabriqué **répond aux besoins** des utilisateurs (correction fonctionnelle).
- **La qualité correspond au contrat de service initial.**

La qualité du logiciel est une notion multiforme qui recouvre :

- la *validité* : aptitude d'un logiciel à réaliser exactement les tâches définies par sa spécification,
- la *fiabilité* : aptitude d'un logiciel à assurer de manière continue le service attendu,
- la *robustesse* : aptitude d'un logiciel à fonctionner même dans des conditions anormales,
- l'*extensibilité* : facilité d'adaptation d'un logiciel aux changements de spécification,
- la *réutilisabilité* : aptitude d'un logiciel à être réutilisé en tout ou partie,

8

Qu'est ce que le génie logiciel: Définition

- la *compatibilité* : aptitude des logiciels à pouvoir être combinés les uns aux autres.
 - l'*efficience* : aptitude d'un logiciel à bien utiliser les ressources matérielles telles la mémoire, la puissance de l'U.C., etc.
 - la *portabilité* : facilité à être porté sur de nouveaux environnements matériels et/ou logiciels,
 - la *traçabilité* : capacité à identifier et/ou suivre un élément du cahier des charges lié à un composant d'un logiciel.
 - la *vérifiabilité* : facilité de préparation des procédures de recette et de certification,
 - l'*intégrité* : aptitude d'un logiciel à protéger ses différents composants contre des accès ou des modifications non autorisés,
 - la *facilité d'utilisation, d'entretien*, etc.
- **Les coûts restent dans les limites prévues au départ.**
 - **Les délais restent dans les limites prévues au départ.**

9

Introduction & définitions

Qualité logicielle

- **Déf. intuitive** : La qualité est la conformité avec les besoins, l'adéquation avec l'usage attendu, le degré d'excellence ou , tout simplement, la valeur de quelque chose pour quelqu'un.
- **Déf. ISO (Organisation internationale de normalisation)** : Ensemble des traits et des caractéristiques d'un produit logiciel portant sur son aptitude à satisfaire des besoins exprimés ou implicites
- **Déf. IEEE** : La qualité du logiciel correspond au degré selon lequel un logiciel possède une combinaison d'attributs désirés.
Attributs: efficacité, sécurité, facilité d'utilisation, performance,...
- **Déf. de Crosby** : La qualité du logiciel correspond au degré selon lequel un client perçoit qu'un logiciel répond aux multiples attentes.
==> Nécessité de faire les tests du logiciel pour vérifier la qualité du logiciel
- **Déf. de Pressman** : Conformité aux exigences explicites à la fois fonctionnelles et de performances, aux standards de développements explicitement documentés et aux caractéristiques implicites qui sont attendues de tous logiciels professionnellement développés.
- En génie logiciel, la qualité logicielle est une appréciation globale d'un logiciel, basée sur de nombreux indicateurs.

10

Grande complexité

- Interdépendance des packages, des modules,...
- Une classe utilise une autre,
- ...
- Structure complexe
- ➔ difficile d'évaluer la qualité

**Opportunités limitées
de détection de
("bugs")**

Très difficile de faire exhaustivement tous les tests de tous les cas d'utilisation
➔ difficile d'évaluer la qualité

Invisibilité du produit

Pas de manifestation physique visible
➔ difficile d'évaluer la qualité

Pr Abdelali ZAKRANI & Pr Said NOUH

11**Sujet à une relation
client-fournisseur****Exige des interfaces
avec d'autres systèmes****Exige un travail d'équipe****Exige la coopération et
coordination avec d'autres
équipes de développement****Exige la maintenance
pendant plusieurs années****Exige la poursuite du projet
alors que l'équipe change****12**

- Le logiciel est une composante majeure des systèmes informatiques (environ 80% du coût) utilisés pour :
 - communication (ex. syst. téléphone, syst. email)
 - santé (monitoring)
 - transport (ex. automobile, aéronautique)
 - échanges économiques (ex. commerce)
 - Gestion des entreprises,...
 - ...
- Les défauts du logiciel sont extrêmement coûteux en terme
 - d'argent
 - de réputation des entreprises de développement et leurs clients
 - de perte de vie (logiciels de contrôle de réanimation dans les cliniques,...)

Pr Abdelali ZAKRANI & Pr Said NOUH

13

La première sonde du programme Mariner de la NASA

- Mariner 1 est la première sonde du programme Mariner de la NASA. Lancée pour une mission de survol de la planète Vénus.
- Destruction 294,5 secondes après son lancement.
- Cause directe:
 - Le programme de contrôle ne lisse plus les valeurs de vitesse et réagit brutalement à des variations mineures
- Impact:
 - Coût : 18,5 millions de dollars
- Origine:
 - Erreur dans le code source du système de contrôle



Pr Abdelali ZAKRANI & Pr Said NOUH

14

■ Une autre cause invoquée pour l'incident de Mariner 1

- Le bug suivant, trouvé dans le code Fortran I, ne semble pas avoir eu de conséquence

DO5I=1.5

C'est une affectation! L'instruction correcte (une boucle pour I variant de 1 à 5) aurait du être:

DO5I = 1,5

Ou plutôt

DO 5 I=1, 5



Pr Abdelali ZAKRANI & Pr Said NOUH

15



- **Rejet du système Socrate à la SNCF - 1990 :**
- **Cause directe:** importantes difficultés de déploiement et d'utilisation
- **Impact:** report de la clientèle vers d'autres moyens de transport.
- **Origine:** Rachat d'un système développé pour une compagnie aérienne

Destruction d'Ariane 5 1996

- **Cause directe:** conversion entier/flottant non autorisée
- **Impact:** Explosion 30s après le décollage et une année de retard pour le programme Ariane 5 et 500 millions \$
- **Origine:**
 - Reprise du code spécifié pour Ariane 4
 - Absence de tests de validation pré-vol
 - Désactivation de la gestion des exceptions
 - Conservation du code inutile

Pr Abdelali ZAKRANI & Pr Said NOUH

16

Désastres historiques des logiciels

Bogue 2000 et PeopleSoft



Bug de l'an 2000 : la pendule indique janvier 1900 au lieu de janvier 2000.



Bogue de l'an 2000

- **Cause directe:** Codage de la date sur 2 caractères
- **Impact:** Très nombreuses mesures préventives et correctives et le coût estimé des centaines de milliards \$

Echec du déploiement de PeopleSoft (éditeurs d'ERP) chez Avis 2004

- **Cause directe:** Multiples retards de déploiement et nombreux surcoûts d'adaptation
- **Impact:** Abandon du projet et le coût est environ 45 millions d'euros
- **Origine:**
 - Complexité du logiciel mal perçu
 - Charge d'intégration et d'adaptation sous-estimée

Pr Abdelali ZAKRANI & Pr Said NOUH

17

Désastres historiques des logiciels

Panne des trains en Europe de l'Est (2022)

- **Panne des trains en Europe de l'Est (2022):** Article Publié le 17 /03/022 : <https://www.lesechos.fr/industrie-services/tourisme-transport/ukraine-un-bug-chez-alstom-perturbe-le-trafic-ferroviaire-en-europe-de-l'est-1394375>

« Trains retardés, voire annulés, les réseaux ferrés de plusieurs pays d'Europe et d'Asie étaient fortement ralenti ce jeudi. La raison : un bug informatique touchant un système de signalisation ferroviaire fourni par Alstom a affecté la circulation de trains en Pologne, en Italie et en Asie, mais le problème est connu et est en voie d'être réglé, a indiqué le constructeur ferroviaire »

• Il s'agit d'un problème de formatage de l'heure touchant un système initialement fourni par Bombardier Transports - entreprise rachetée par Alstom début 2021. Et « il n'y a pas eu de cyberattaque » contrairement à ce qu'ont pu craindre les autorités polonaises, a précisé un porte-parole.

• 800 kilomètres de lignes touchées en Pologne

- En Pologne, la panne a entraîné d'importants retards et des annulations de trains, dont certains devaient transporter des réfugiés ukrainiens
- <https://www.latribune.fr/technos-medias/telecoms/un-bug-logiciel-a-l-origine-de-la-panne-geante-chez-orange-886661.html>
- <https://www.rocketprojet.com/29-bugs-informatiques-catastrophe/>
- <https://horustest.io/blog/les-10-bugs-informatiques-les-plus-couteux-de-l-histoire/>

Pr Abdelali ZAKRANI & Pr Said NOUH

18

❖ **L'incident du Therac-25 (années 1980)** : Le Therac-25 était un dispositif de traitement du cancer par rayonnement. En raison d'une défaillance logicielle, il a administré par erreur des doses de rayonnement mortelles à plusieurs patients. Plusieurs décès et blessures graves ont résulté de ces erreurs.

<https://www.futura-sciences.com/tech/actualites/informatique-science-decalee-premier-bug-informatique-histoire-papillon-nuit-84432/>

❖ **L'effondrement de la bourse de Tokyo (années 1999)** : Une panne logicielle a provoqué une interruption majeure des échanges à la Bourse de Tokyo en 1999. Cela a eu un impact considérable sur les marchés financiers mondiaux.

<https://www.courrierinternational.com/article/couac-une-panne-historique-oblige-la-bourse-de-tokyo-fermer-pour-la-journee>

Pr Abdelali ZAKRANI & Pr Said NOUH

19

❖ **L'incident de Knight Capital (2012)** : Une erreur logicielle a conduit à une série de transactions boursières défectueuses, causant une perte de plus de **400 millions de dollars en moins d'une heure et forçant l'entreprise à faire faillite**.

https://www.lemonde.fr/economie/article/2012/08/07/les-deboires-de-knight-capital-specialiste-du-trading-haute-frequence_1743230_3234.html

<https://www.lefigaro.fr/flash-eco/2012/08/06/97002-20120806FILWWW00169-knight-capital-sauve-de-la-faillite.php>

❖ **Le bug Heartbleed (2014)** : Heartbleed était une faille de sécurité critique dans la bibliothèque OpenSSL, largement utilisée pour la sécurité sur Internet. Cette vulnérabilité a permis à des attaquants de voler des données sensibles, y compris des informations de connexion et des clés de chiffrement.

https://www.lemonde.fr/technologies/article/2014/04/09/une-enorme-faille-de-securite-dans-de-nombreux-sites-internet_4397995_651865.html

<https://www.numerama.com/sciences/29026-heartbleed-ssl-openssl.html>

Pr Abdelali ZAKRANI & Pr Said NOUH

20

Désastres historiques des logiciels

❖ **Le crash du Boeing 737 Max (2018 et 2019)** : Deux accidents mortels impliquant des avions Boeing 737 Max ont eu lieu en 2018 et 2019 (La mort de 346 personnes). Les enquêtes ont révélé que des défaillances logicielles, notamment le système MCAS (Maneuvering Characteristics Augmentation System), avaient contribué aux accidents. **Ces défaillances peuvent entraîner une interdiction de vol mondiale de ces avions.**

- https://fr.wikipedia.org/wiki/Suspension_de_vol_du_Boeing_737_Max

- https://www.lepoint.fr/monde/securite-aerienne-une-annee-2019-plombee-par-le-boeing-737-max-01-01-2020-2355579_24.php#11

❖ **La faille Equifax (2017)** : En 2017, Equifax, l'une des principales agences de crédit aux États-Unis, a révélé une faille de sécurité majeure qui a exposé les données personnelles de millions d'Américains. **Equifax paye 700 millions de dollars**. C'est le prix le plus fort payé par une entreprise pour une affaire de piratage.

https://www.lemonde.fr/pixels/article/2017/10/03/le-piratage-d-equifax-du-a-une-faille-informatique-non-corrigee_5195179_4408996.html

<https://www.lefigaro.fr/secteur/high-tech/piratage-de-donnees-equifax-paye-700-millions-de-dollars-pour-eteindre-les-enquetes-20190722>

Pr Abdelali ZAKRANI & Pr Said NOUH

21

Désastres historiques des logiciels

❖ **La cyberattaque SolarWinds (2020)** : En décembre 2020, il a été révélé qu'une cyberattaque sophistiquée avait compromis le logiciel de gestion réseau de SolarWinds, exposant de nombreuses organisations gouvernementales et privées à des risques de sécurité importants.

<https://www.lebigdata.fr/solarwinds-cyberattaque-historique-usa>

https://fr.wikipedia.org/wiki/Cyberattaque_de_2020_contre_les_%C3%89tats-Unis

❖ **La panne de l'application mobile de vaccination COVID-19 de l'État de New York (2021)** : L'application mobile utilisée pour gérer les rendez-vous de vaccination contre la COVID-19 dans l'État de New York a connu des problèmes techniques, **entravant la prise de rendez-vous et la distribution des vaccins**.

Pr Abdelali ZAKRANI & Pr Said NOUH

22

- ❖ **La panne de Facebook (2021)** : En octobre 2021, Facebook a subi une panne mondiale qui a affecté l'accès à ses services, y compris Facebook, Instagram et WhatsApp. **Cette panne a duré plusieurs heures et a eu un impact sur des milliards d'utilisateurs à travers le monde.**
- ❖ **La panne de Zoom (2020)** : En août 2020, l'application de visioconférence Zoom a connu une panne mondiale qui a affecté de nombreuses réunions en ligne, alors que de nombreuses personnes travaillaient à distance en raison de la pandémie de COVID-19.
- ❖ **La panne de Google (2020)** : En août 2020, de nombreux services de Google, y compris Gmail, Google Drive et YouTube, ont été affectés par une panne mondiale, entraînant des perturbations pour les utilisateurs et les entreprises.

Pr Abdelali ZAKRANI & Pr Said NOUH

23

Chapitre 2:

Modèle de qualité logicielle (ISO/IEC 25010:2011)

Pr Abdelali ZAKRANI & Pr Said NOUH

25

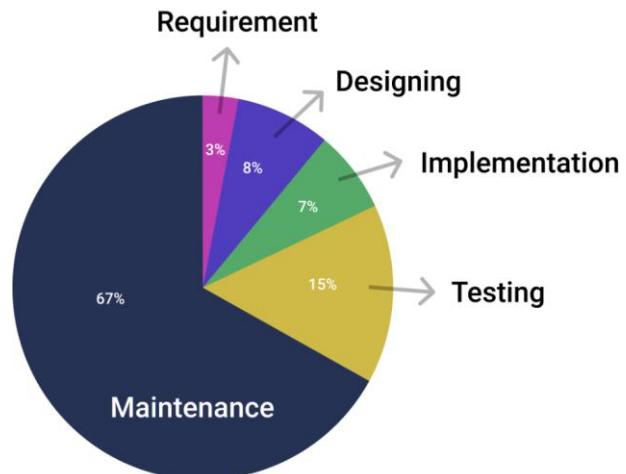
- Le coût du développement du logiciel dépasse souvent celui du matériel.
- Durée de vie d'un logiciel de 10 à 15 ans
- Le coût de la maintenance évolutive et corrective constitue la part prépondérante du coût total.
- Plus de la moitié des erreurs découvertes en phases de tests proviennent d'erreurs introduites dans les premières étapes
- La récupération d'une erreur est d'autant plus coûteuse que sa découverte est tardive.

Pr Abdelali ZAKRANI & Pr Said NOUH

26

Cycle de vie du développement logiciel (SDLC: Software Development Life Cycle)

Coût de la maintenance: Les rapports suggèrent que le coût de maintenance est élevé.


<https://www.openxcell.com/blog/software-maintenance/>

Pr Abdelali ZAKRANI & Pr Said NOUH

27

Modèle de Qualité Logicielle

- La norme ISO/IEC 25010:2011, fait partie de la série de normes internationales SQuaRE(Software product Quality Requirements and Evaluation), est une norme internationale qui traite de la qualité des produits logiciels. <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>
- Elle a été développée par l'Organisation internationale de normalisation (ISO) et la Commission électrotechnique internationale (IEC).
- ISO/IEC 25010 est conçue pour aider les organisations à définir, évaluer et gérer la qualité des logiciels tout au long de leur cycle de vie, de la conception à la maintenance.
- ISO/IEC 25010:2011 définit un modèle de qualité pour les produits logiciels, qui est basé sur huit caractéristiques de qualité principales, chacune d'entre elles étant subdivisée en sous-caractéristiques.

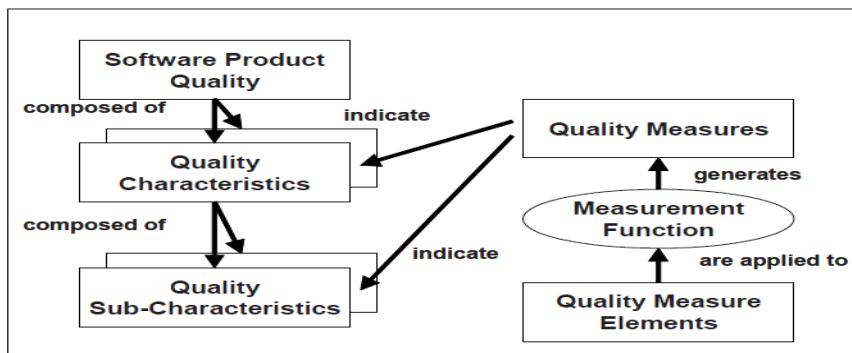
Pr Abdelali ZAKRANI & Pr Said NOUH

28

Modèle de Qualité Logicielle

ISO/IEC (Commission Electrotechnique Internationale) **25010:2011** propose le modèle de référence suivant pour mesurer la qualité des produits logiciels.

Des critères(caractéristiques, sous- caractéristiques) et méthodes de mesure de ces indicateurs de qualité.



<https://www.iso.org/fr/standard/35733.html>

Pr Abdelali ZAKRANI & Pr Said NOUH

29

ISO/IEC 25010:2011

La norme ISO/IEC 25010 :2011 distingue trois (3) catégories de qualité :

- La qualité interne qualifie la qualité du logiciel à partir des mesures statiques du code.
- La qualité externe repose sur les mesures effectuées lors de simulation d'exécution du logiciel, lors des phases de tests par exemple.
- La qualité d'usage définit les objectifs fonctionnels spécifiés par les parties prenantes dans un contexte d'utilisation spécifié.

<https://archipel.uqam.ca/15072/>

Pr Abdelali ZAKRANI & Pr Said NOUH

30

ISO/IEC 25010:2011

- 1) Un modèle de qualité d'utilisation composé de **cinq caractéristiques** (dont certaines sont subdivisées en sous-caractéristiques) qui se rapportent au résultat de l'interaction lorsqu'un produit est utilisé dans un contexte d'utilisation particulier.
- 2) Un modèle de qualité de produit composé de **huit caractéristiques** (qui sont subdivisées en sous-caractéristiques) qui se rapportent aux propriétés statiques du logiciel et aux propriétés dynamiques du système informatique.

Pr Abdelali ZAKRANI & Pr Said NOUH

31

Il existe trois modèles de qualité dans la série **SQuaRE** (Software product Quality Requirements and Evaluation):

- 1) Le modèle de qualité d'utilisation (perçue par le client utilisateur du logiciel)
- 2) Le modèle de qualité du produit dans la présente Norme internationale (ISO/IEC 25010)
- 3) Le modèle de qualité des données dans ISO / CEI 25012.

Les modèles de qualité ensemble servent de cadre pour garantir que toutes les caractéristiques de qualité sont prises en compte.

Ces modèles fournissent un ensemble de caractéristiques de qualité pertinentes pour un large éventail de parties prenantes, telles que: les développeurs de logiciels, intégrateurs de systèmes, acquéreurs, propriétaires, mainteneurs, entrepreneurs, professionnels de l'assurance qualité et du contrôle et les utilisateurs.

Pr Abdelali ZAKRANI & Pr Said NOUH

32

Qualité d'utilisation

Efficacité
(il réponds au besoin du client)

Efficience
(temps d'exécution réduit, économie de mémoire „„)

Satisfaction

Absence de risque

Couverture du contexte
(périmètre et champs de tous les utilisateurs possibles est couvert)

Efficacité

Efficience

Utilité

Confiance

Plaisir

Confort

Atténuation des risques économiques

Atténuation des risques pour la santé et la sécurité

Atténuation des risques environnementaux
(contrôle d'émission de CO2 des usines,...)

complétude du contexte

Flexibilité

Pr Abdelali ZAKRANI & Pr Said NOUH

33

Modèle de Qualité Logicielle

Modèle de qualité du produit

- Complétude fonctionnelle
- Correctivité fonctionnelle
- Pertinence fonctionnelle (une application faite pour une compagnie aérienne → Trains ?)

Adéquation fonctionnelle

1

- Comportement temporel
- Utilisation des ressources
- Mémoire utilisé

Efficacité des performances

2

- Coexistence avec autres applications
- Interopérabilité et communication avec autres applications

Compatibilité

3

- Apprentissage facile pour utilisation
- Opérabilité
- Protection contre les erreurs de l'utilisateur
- Esthétique de l'interface utilisateur
- Accessibilité

Utilisabilité

4

- Disponibilité
- Tolérance aux pannes
- Récupération (Docs MS word par exemple)

Fiabilité

5

- Confidentialité
- Intégrité
- Non-répudiation (assume sa Responsabilité)

Sécurité

6

- Modularité
- Réutilisabilité
- Analyssabilité
- Modificabilité
- Testabilité

Maintenabilité

7

Voir détails dans les diapositives suivantes

- Adaptabilité
- Instalabilité
- Remplaçabilité

Portabilité

8

Pr Abdelali ZAKRANI & Pr Said NOUH

34

Modèle de Qualité Logicielle

Modèle de qualité du produit

Les huit caractéristiques de la qualité du produit logiciel:

- Fonctionnalité** : Cette caractéristique concerne la capacité du logiciel à fournir les fonctionnalités requises, y compris sa pertinence, son exactitude, sa complétude et sa conformité aux spécifications fonctionnelles.
- Fiabilité** : Il s'agit de la capacité du logiciel à maintenir ses performances au fil du temps, en évitant les défaillances et les erreurs.
- Utilisabilité** : Cette caractéristique concerne l'ergonomie et la convivialité du logiciel, y compris la facilité d'apprentissage, l'efficacité d'utilisation et la satisfaction de l'utilisateur.
- Efficacité opérationnelle** : Il s'agit de la capacité du logiciel à fonctionner de manière efficiente, en utilisant de manière optimale les ressources disponibles.
- Compatibilité** : Cette caractéristique se réfère à la capacité du logiciel à fonctionner avec d'autres systèmes ou composants sans conflits.
- Sécurité** : Il s'agit de la capacité du logiciel à protéger les données et les fonctionnalités contre les accès non autorisés, les dommages ou les perturbations.
- Maintenabilité** : Cette caractéristique concerne la facilité avec laquelle le logiciel peut être modifié, mis à jour ou réparé.
- Portabilité** : Il s'agit de la capacité du logiciel à être utilisé dans différents environnements ou plates-formes.

Pr Abdelali ZAKRANI & Pr Said NOUH

35

Maintenabilité: degré d'efficacité et d'efficience avec lequel un produit ou un système peut être modifié par les mainteneurs.

Les sous-caractéristiques de la maintenabilité:

- **Modularité**
- **Réutilisabilité**
- **Facilité d'analyse**
- **Facilité de modification**
- **Testabilité**

Pr Abdelali ZAKRANI & Pr Said NOUH

36

Les sous-caractéristiques :

- **Modularité :** degré auquel un système ou un programme informatique est composé de composants discrets de telle sorte qu'une modification apportée à un composant a un impact minimal sur les autres composants
- **Réutilisabilité :** degré auquel un composant peut être utilisé dans plus d'un système ou dans la construction d'autres composants du logiciel.

Pr Abdelali ZAKRANI & Pr Said NOUH

37

Les sous-caractéristiques :

- **Facilité d'analyse (Analysabilité)** : le degré d'efficacité et d'efficiency avec lequel il est possible d'évaluer l'impact sur un produit ou un système d'une modification prévue d'une ou de plusieurs de ses parties, ou de diagnostiquer un produit en ce qui concerne les défauts ou les causes de défaillances, ou d'identifier les parties à modifier. Si vous avez un code source d'une application que vous n'avez pas touché depuis 2 ans, est-ce que vous pouvez l'analyser facilement pour faire des modifications dedans ou d'y ajouter de nouvelles fonctionnalités et être sûr qu'il n'y a pas des risques associés à ces changements ? Vous n'avez pas touché un module qui influencera sur un autre ?
- **Facilité de modification (Modificabilité)**: degré auquel un produit ou un système peut être modifié de manière efficace et efficiente sans introduire de défauts ou dégradation de la qualité du produit existant

Pr Abdelali ZAKRANI & Pr Said NOUH

38

Les sous-caractéristiques :

- **Testabilité** : degré d'efficacité et d'efficiency avec lequel des critères de test peuvent être établis pour un système, un produit ou des composants et des tests peuvent être effectués pour déterminer si ces critères ont été satisfaits.

Exemple: Lors de la création de votre code source, avez-vous fait attention pour séparer les traitements en plusieurs fonctions le maximum possible, pour faciliter les tests unitaires de chaque fonction ?

Pr Abdelali ZAKRANI & Pr Said NOUH

39

L'assurance qualité logicielle (AQL) est un ensemble d'activités planifiées et systématiques de toutes les actions nécessaires pour fournir une assurance suffisante de la qualité logicielle d'un nouveau logiciel ou d'une nouvelle version d'un logiciel est conforme aux exigences et aux attentes établies. Les pratiques d'AQL varient selon le modèle d'affaire et l'industrie où le logiciel est utilisé.

Comment garantir la qualité d'un logiciel ?

L'étape préalable à l'assurance de la qualité d'un logiciel est de respecter les règles de l'art du développement : un code uniforme respectant des standards de codage, une gestion décentralisée du code source, en passant par l'automatisation de toute la chaîne de construction d'un logiciel.

L'assurance qualité implique des approches proactives qui ont pour but d'empêcher la fabrication de produits non conformes. **Le contrôle qualité**, quant à lui, implique des approches réactives destinées à identifier de potentielles non-conformités par le biais d'inspections et de mécanismes de tests.

Pr Abdelali ZAKRANI & Pr Said NOUH

40

L'assurance qualité logicielle (AQ) et le contrôle qualité logiciel (CQ) sont deux concepts essentiels dans le domaine du développement de logiciels, mais ils se différencient par leurs objectifs, leurs approches et leur chronologie.

Assurance Qualité Logicielle (AQ) :

Objectif : L'assurance qualité logicielle vise à prévenir les défauts et à s'assurer que les processus de développement de logiciels sont conformes aux normes, aux meilleures pratiques et aux exigences définies. L'objectif principal est d'améliorer la qualité du produit en intervenant sur les processus.

Approche : L'assurance qualité logicielle adopte une approche proactive. Elle implique l'évaluation continue des processus, des normes, des méthodologies et des compétences des personnes impliquées dans le développement logiciel.

Moment : L'assurance qualité logicielle intervient tout au long du cycle de vie du projet, depuis la phase de planification jusqu'à la livraison finale du produit. Elle est intégrée dans l'ensemble du processus de développement.

Activités : Les activités typiques de l'assurance qualité logicielle incluent la définition de normes de qualité, l'audit de processus, la formation du personnel, la gestion des risques, et la vérification de la conformité aux procédures.

41

Contrôle Qualité Logiciel (CQ) :

Objectif : Le contrôle qualité logiciel se concentre sur la détection des défauts et des problèmes spécifiques dans le logiciel en cours de développement. Son objectif principal est de s'assurer que le produit final est exempt de défauts.

Approche : Le contrôle qualité logiciel adopte une approche réactive. Il s'agit de tester et d'inspecter le logiciel pour identifier les problèmes et les anomalies.

Moment : Le contrôle qualité logiciel intervient généralement à la fin du processus de développement, avant la livraison du produit. Il se concentre sur l'évaluation du produit final. L'approche TDD(test driven développement) adopte le commencement par la préparation des tests qui aident à diriger le développement.

Activités : Les activités typiques de contrôle qualité logiciel comprennent les tests de qualité (tests unitaires, tests d'intégration, tests de système, etc.), les revues de code, l'analyse statique, et l'inspection du logiciel pour identifier les défauts.

L'assurance qualité logicielle se concentre sur l'amélioration continue des processus, tandis que le contrôle qualité logiciel se concentre sur la détection des défauts dans le produit final. Ces deux approches sont complémentaires.

42