



Département Technologies de l'Informatique
Master Pro DSIR

Qualité Performance Logicielle

Présenter par : WALID DHOUIB
dhouib.walid@gmail.com
Auditeur tiers partie QSE et SMSI

1

Plan du cours



Leçon 1. Génie logiciel et modèle de cycle de vie



Leçon 2. Fondamentaux de la qualité logiciel



Leçon 3. Normalisation et exigences de la qualité du logiciel



Leçon 4. Tests logiciels



2

Walid DHOUIB – ISET SFAX

2

3

Première leçon

Nous aborderons ces aspects :

- Génie logiciel
- Cycles de vie



3

Systèmes d'Information

- ▶ C'est un ensemble organisé d'éléments qui permet de regrouper, de classifier, de traiter et de diffuser de l'information sur un phénomène donné
- ▶ L'utilisation de moyens informatiques, électroniques et la télécommunication permettent d'automatiser et de dématerieliser les opérations telles que les procédures d'entreprise.
- ▶ Un ensemble organisé de ressources (matériel, logiciel, personnel, données, procédures...) permettant d'acquérir, de traiter, de stocker des informations (sous forme de données, textes, images, sons...) dans et entre des organisations.

4

Walid DHOUIB – ISET SFAX

4

Exemple de SI

- ▶ Un réseau de guichet automatique de banque
- ▶ Une application de gestion de stocks d'un supermarché
- ▶ Un site web de vente en ligne
- ▶ Une bibliothèque numérique
- ▶ ...



5

Walid DHOUIB – ISET SFAX

5

L'informatisation

- ▶ L'informatique est au cœur de toutes les grandes entreprises.
- ▶ Le système d'information d'une entreprise est composé
 - ▶ de matériels : 20%
 - ▶ La fabrication du matériel est assurée par quelques fabricants seulement
 - ▶ Ce matériel est relativement fiable et le marché est standardisé.
 - ▶ de logiciels. 80%
- ▶ Les problèmes liés à l'informatique sont essentiellement des problèmes de logiciel.

6

Walid DHOUIB – ISET SFAX

6

Les logiciels

- ▶ Un logiciel est un ensemble de programmes, qui permet à un ordinateur ou à un système informatique d'assurer une tâche ou une fonction en particulier
- ▶ Les logiciels peuvent être développés par
 - ▶ Une personne seule,
 - ▶ Une petite équipe,
 - ▶ Un ensemble d'équipes coordonnées.
- ▶ Lors du développement de grands logiciels
 - ▶ importants problèmes de conception et de coordination.
 - ▶ phase absolument décisive
 - ▶ Le coût du développement d'un logiciel est extrêmement important.
 - ▶ Maintenance coûte également très cher
 - ▶ conditionne sa réussite et sa pérennité.

7

Walid DHOUIB – ISET SFAX

7

La nature du logiciel

- ▶ Le logiciel est facile à reproduire
 - ▶ Tout le coût se trouve dans son développement
- ▶ Pour d'autres produits, la fabrication est souvent le processus le plus coûteux
- ▶ Le logiciel est intangible
 - ▶ Il est difficile d'estimer l'effort de développement
- ▶ Le processus de développement est difficile à automatiser
 - ▶ L'industrie du logiciel exige beaucoup de main d'œuvre

8

Walid DHOUIB – ISET SFAX

8

La nature du logiciel

- ▶ Même des informaticiens peu qualifiés peuvent arriver à bricoler quelque chose qui semble fonctionner
 - ▶ La qualité d'un logiciel n'est pas apparente
- ▶ Un logiciel semble facile à modifier
 - ▶ La tentation est forte d'effectuer des changements rapides sans vraiment en mesurer la portée
- ▶ Un logiciel ne s'use pas
 - ▶ Il se détériore à mesure que des changements sont effectués (software aging)
 - ▶ en raison de l'introduction d'erreurs
 - ▶ ou par une complexification indue

9

Walid DHOUIB – ISET SFAX

9

La nature du logiciel

- ▶ Raisons pour lesquelles le logiciel vieillit
 - ▶ maintenance (e.g., bug fixes)
 - ▶ érosion architecturale
 - ▶ inflexibilité dès le début
 - ▶ documentation insuffisante ou inconsistante
 - ▶ duplication de code
 - ▶ manque de modularité
 - ▶ complexité croissante

10

Walid DHOUIB – ISET SFAX

10

La nature du logiciel

► Observations

- Beaucoup de logiciels sont mal conçus et se détériorent rapidement
- La demande pour du logiciel est toujours croissante
 - Le logiciel se trouve en perpétuel "état de crise"
 - L'ingénierie du logiciel est une nécessité
 - processus systématique au lieu de bricolage

11

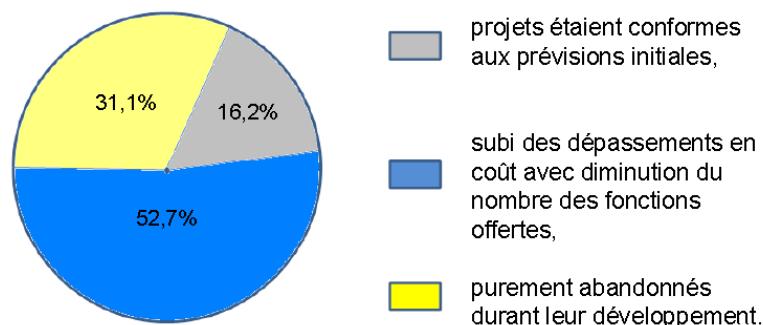
Walid DHOUIB – ISET SFAX

11

Etat de crise

- Une étude du Standish Group (1995) reposant sur un échantillon représentatif

365 entreprises, totalisant 8380 applications



12

Walid DHOUIB – ISET SFAX

12

Etat de crise

- ▶ Pour les grandes entreprises
 - ▶ le taux de succès est de 9% seulement,
 - ▶ 37% des projets sont arrêtés en cours de réalisation,
 - ▶ 50% aboutissent hors délai et hors budget.
- ▶ L'examen des causes de succès et d'échec est instructif
 - ▶ la plupart des échecs proviennent non de l'informatique, mais de la maîtrise d'ouvrage (i.e. le client).

13

Walid DHOUIB – ISET SFAX

13

Crise de l'industrie du logiciel

- ▶ à la fin des années 1970.
- ▶ était principalement due à :
 - ▶ l'augmentation des coûts ;
 - ▶ les difficultés de maintenance et d'évolution ;
 - ▶ la non fiabilité ;
 - ▶ le non respect des spécifications ;
 - ▶ le non respect des délais.
- ▶ L'importance d'une approche méthodologique

14

Walid DHOUIB – ISET SFAX

14

La crise du logiciel

► Etude du gouvernement américain en 1979

| Cas | Coût | % |
|-----------------------------|--------|-----|
| Payés mais jamais livrés | \$3.2M | 47% |
| Livrés mais jamais utilisés | \$2.0M | 30% |
| Abandonnés ou refaits | \$1.3M | 20% |
| Utilisés après modification | \$0.2M | 3% |
| Utilisés tel quel | \$0.1M | 2% |



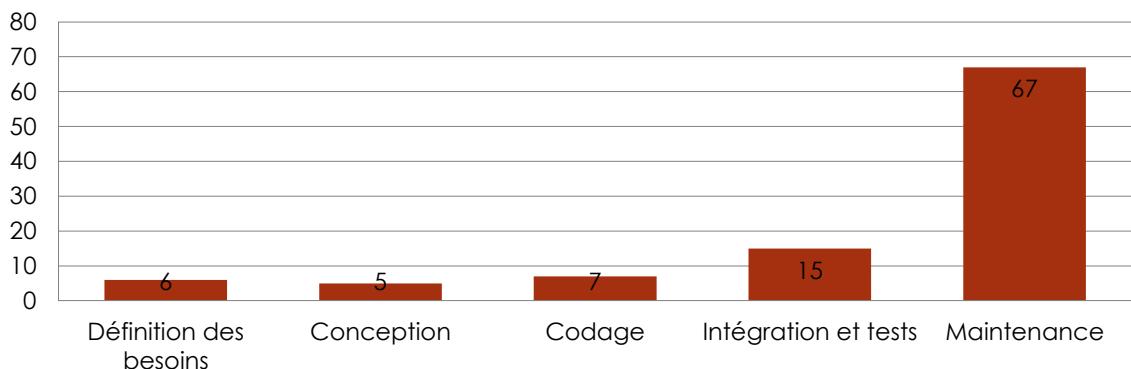
15

Walid DHOUIB – ISET SFAX

15

Répartition de l'effort de développement

Répartition de l'effort de développement



16

Walid DHOUIB – ISET SFAX

16

Les différentes catégories de logiciel

- ▶ Sur mesure (custom)
 - ▶ Pour un client spécifique
- ▶ Générique (generic)
 - ▶ Vendu sur le marché
 - ▶ e.g. un tableur (spreadsheet), un outil de base de données (database), un outil de traitement de texte (word processor)
- ▶ Embarqués (embedded)
 - ▶ Exécutent dans du matériel électronique isolé
 - ▶ machine à laver, télévision, lecteur DVD, téléphone mobile, magnétoscope, four à micro-ondes, réfrigérateur, joueur MP3, ...
 - ▶ Difficile à modifier

17

Walid DHOUIB – ISET SFAX

17

Qu'est-ce que le génie logiciel ?

- ▶ Définition
 - ▶ Le **processus** visant la résolution de problèmes posés par un client par le développement **systématique** et l'évolution de systèmes logiciels de **grande taille** et de **haute qualité** en respectant les contraintes de **coûts**, de **temps**, et autres.
- ▶ Definition
 - ▶ The process of solving customers' problems by the systematic development and evolution of large, high-quality software systems within cost, time and other constraints.

18

Walid DHOUIB – ISET SFAX

18

Qu'est-ce que le génie logiciel?

- ▶ Les quatre P du génie logiciel

- ▶ Personnel

- ▶ Qui produit le logiciel?

- ▶ Processus

- ▶ Comment le logiciel est-il produit?

- ▶ Projet

- ▶ La production réelle du logiciel

- ▶ Produit

- ▶ Tous les objets fabriqués pendant la production code source, exécutables, documentation, modèles de conception, cahier de charges, résultats de tests, mesures de productivité, ...

Risques et difficultés en génie logiciel

- ▶ Complexité et quantité des éléments à tenir en compte
- ▶ Incertitude concernant la technologie
- ▶ Incertitude concernant les exigences
- ▶ Incertitude concernant les compétences
- ▶ Adaptation face aux changements
- ▶ Déterioration du produit
- ▶ Risques politiques

Les parties prenantes dans le génie logiciel

1. Utilisateurs (users)

- ▶ Ceux qui se servent du logiciel

2. Clients (customers)

- ▶ Ceux qui paient pour le logiciel

3. Développeurs (developers)

- ▶ Ceux qui conçoivent le logiciel

4. Gestionnaires (managers)

- ▶ Ceux qui supervisent la production du logiciel

Tous ces rôles peuvent être remplis par la même personne

Maintenance de logiciels

► La maintenance est devenue une facette très importante du cycle de vie d'un logiciel.

- ▶ une enquête effectuée aux USA en 1986 auprès de 55 entreprises révèle que 53% du budget total d'un logiciel est affecté à la maintenance.
 - ▶ 34% maintenance évolutive (modification des spécifications initiales) ;
 - ▶ 10% maintenance adaptative (nouvel environnement, nouveaux utilisateurs) ;
 - ▶ 17% maintenance corrective (correction des bogues) ;
 - ▶ 16% maintenance perfective (améliorer les performances sans changer les spécifications) ;
 - ▶ 6% assistance aux utilisateurs ;
 - ▶ 6% contrôle qualité ;
 - ▶ 7% organisation/suivi ;
 - ▶ 4% divers.

Le cycle de vie d'un logiciel (1)

- ▶ Désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition.
- ▶ Le cycle de vie permet de détecter les erreurs au plus tôt et ainsi de maîtriser la qualité du logiciel, les délais de sa réalisation et les coûts associés.

23

Walid DHOUIB – ISET SFAX

23

Le cycle de vie d'un logiciel (2)

- ▶ C'est la description d'un processus couvrant les phases de:
 - ▶ **Création d'un produit,**
 - ▶ **Distribution sur un marché,**
 - ▶ **Disparition.**
- ▶ Le but de ce découpage est de
 - ▶ Maîtriser les risques,
 - ▶ Maîtriser au mieux les délais et les coûts,
 - ▶ Obtenir une qualité conforme aux exigences.
- ▶ On distingue deux types de cycle de vie
 - ▶ Le cycle de vie des produits s'applique à tous les types de produits, et peut être considéré comme un outil de gestion.
 - ▶ Le cycle de développement des logiciels s'insère dans le précédent, on l'appelle souvent abusivement cycle de vie des logiciels

24

Walid DHOUIB – ISET SFAX

24

Les étapes du cycle de vie d'un logiciel

- ▶ Définition des objectifs
- ▶ Analyse des besoins et faisabilité : besoins, contraintes, estimation de la faisabilité
- ▶ Spécifications ou conception générale : architecture générale
- ▶ Conception détaillée : définir précisément les sous-ensembles
- ▶ Codage (Implémentation ou programmation)
- ▶ Tests unitaires : sous-ensembles conformes aux spécifications
- ▶ Intégration : assurer l'interfaçage des différents éléments (modules) du logiciel
- ▶ Qualification (ou recette) : vérification de la conformité du logiciel aux spécifications initiales
- ▶ Documentation: informations nécessaires (utilisateur et développeur ultérieurs)
- ▶ Mise en production : déploiement sur le site du logiciel
- ▶ Maintenance : actions correctives, actions évolutives

Le cycle de vie permet de prendre en compte, en plus des aspects techniques, l'organisation et les aspects humains.

25

Walid DHOUIB – ISET SFAX

25

Les modèles de cycle de vie

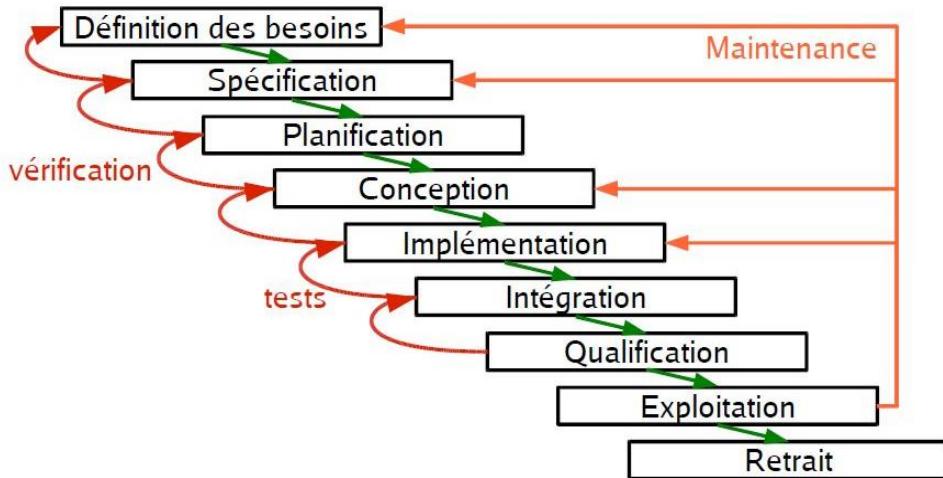
- ▶ Modèle en cascade
- ▶ Modèle en V
- ▶ MAQUETTAGE, PROTOTYPAGE
- ▶ DEVELOPPEMENT INCREMENTAL
- ▶ MODÈLE EN SPIRALE
- ▶ RAD

26

Walid DHOUIB – ISET SFAX

26

Modèle en cascade



27

Walid DHOUIB – ISET SFAX

27

Modèle de cycle de vie en V

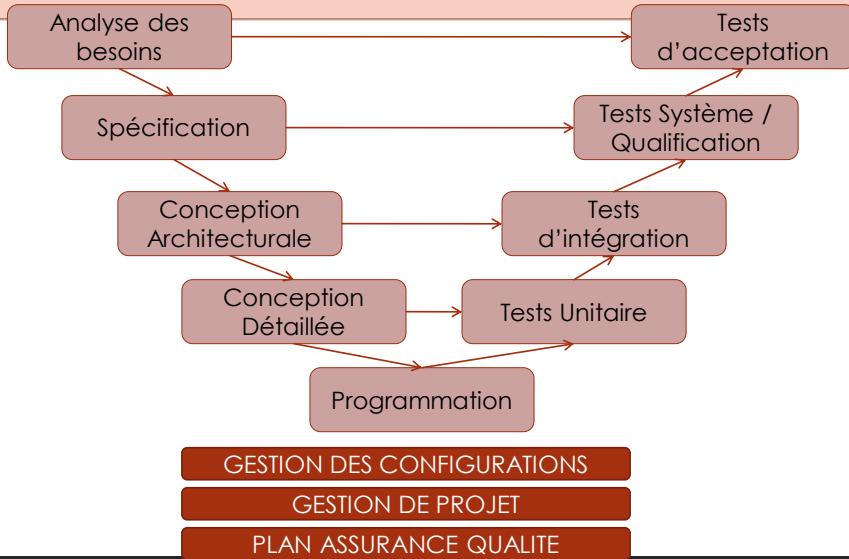
- ▶ Le plus connu et certainement le plus utilisé.
- ▶ Il s'agit d'un modèle en cascade dans lequel le développement et les tests du logiciel sont effectués de manière synchrone.
- ▶ Le principe de ce modèle est qu'avec toute décomposition doit être décrite la recomposition et que toute description d'un composant est accompagnée de tests qui permettront de s'assurer qu'il correspond à sa description.
- ▶ Ce modèle souffre toujours du problème de la vérification trop tardive du bon fonctionnement du système.

28

Walid DHOUIB – ISET SFAX

28

Modèle en V



29

Walid DHOUIB – ISET SFAX

29

Cascade Vs V

- La représentation en V tient davantage compte de la réalité, le processus de développement n'est pas réduit à un enchaînement de tâches séquentielles.
- Elle montre que :
 - c'est en phase de **spécification** que l'on se préoccupe des procédures de qualification
 - c'est en phase de **conception globale** que l'on se préoccupe des procédures d'intégration
 - c'est en phase de **conception détaillée** que l'on prépare les tests unitaires
- Le modèle de cycle de vie en V permet d'anticiper sur les phases ultérieures de développement du produit. En particulier le modèle en V permet de commencer plus tôt :
 - Plan de tests de qualification,
 - Plan d'évaluation des performances,
- Le modèle en V comme celui en cascade conduisent à commencer plus tôt la documentation utilisateur.

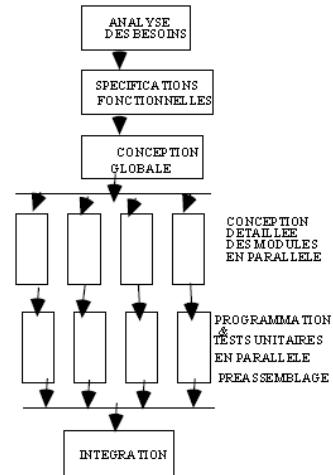
30

Walid DHOUIB – ISET SFAX

30

Cascade Vs V

- Les deux modèles permettent de développer parallèlement différents modules lorsque la phase de conception globale est validée



31

Walid DHOUIB – ISET SFAX

31

Cascade Vs V

- Ces modèles de développement permettent de **contrôler les rétro-actions** :
 - La vérification/validation par une *critique constructive* évite les retours arrière
 - Le cycle de vie met l'accent sur les phases *amont par rapport à la programmation*:
 - Spécification,
 - Conception.
- Toutefois, le modèle présenté est **parfois difficile à appliquer rigoureusement**
 - Il est quelquefois nécessaire de prendre en compte des changements importants dans les spécifications **dans une phase avancée du projet**
 - La durée imposée par le cycle de vie est parfois difficilement acceptée pour certains produits compétitifs** (exemple : logiciels micros....)

32

Walid DHOUIB – ISET SFAX

32

INSUFFISANCE

- ▶ L'interface utilisateur n'apparaîtra que fort tard. Il n'y a pas assez de bornes intermédiaires permettant de valider ce que sera la version finale du produit.
- Pour disposer plus tôt d'objets exécutables ou instrumentables pour les développeurs et pour les utilisateurs, d'autres modèles existent :
 - ▶ Maquettage, prototypage
 - ▶ Développement incrémental
- ▶ Des cycles de vie plus complets prennent en charge la totalité du développement du produit en tenant compte du cycle de décision et de l'analyse de risques.
 - ▶ Exemple du cycle de vie en spirale et de la méthode Merise.

33

Walid DHOUIB – ISET SFAX

33

MAQUETTAGE, PROTOTYPAGE

- ▶ Dans une industrie de fabrication on distingue
 - ▶ Maquette = Modèle réduit de l'objet
 - ▶ Prototype = Premier d'une série
- ▶ En développement de logiciel, il n'y a pas de production en série, mais on distingue :
 - ▶ Maquette ou prototype rapide
 - ▶ En amont du cycle de développement Analyse des besoins, Spécifications fonctionnelles.
 - ▶ Nature du produit : Interface et les fonctionnalités.
 - ▶ Prototype expérimental
 - ▶ Utilisé au niveau de la conception pour :
 - ▶ S'assurer de la faisabilité de parties critiques
 - ▶ Valider des options de conception
 - ▶ Prototype évolutif
 - ▶ La première version du prototype est l'embryon du produit final, on itère jusqu'au produit final

34

Walid DHOUIB – ISET SFAX

34

DEVELOPPEMENT INCREMENTAL

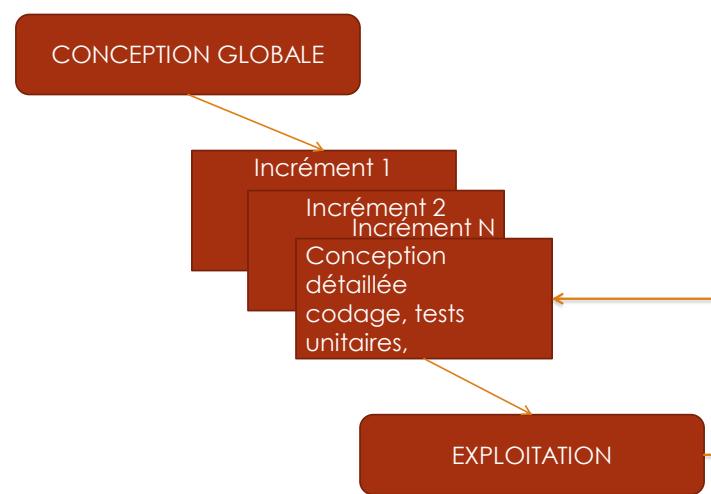
- ▶ Un logiciel peut être construit étape par étape
- ▶ Le logiciel est spécifié et conçu dans son ensemble.
- ▶ La réalisation se fait par incrément de fonctionnalités
- ▶ Chaque incrément est intégré à l'ensemble des précédents et à chaque étape le produit est testé, exploité et maintenu dans son ensemble.
- ▶ Accepter progressivement un logiciel par les utilisateurs

35

Walid DHOUIB – ISET SFAX

35

DEVELOPPEMENT INCREMENTAL



36

Walid DHOUIB – ISET SFAX

36

MODÈLE EN SPIRALE (BOEHM 1988)

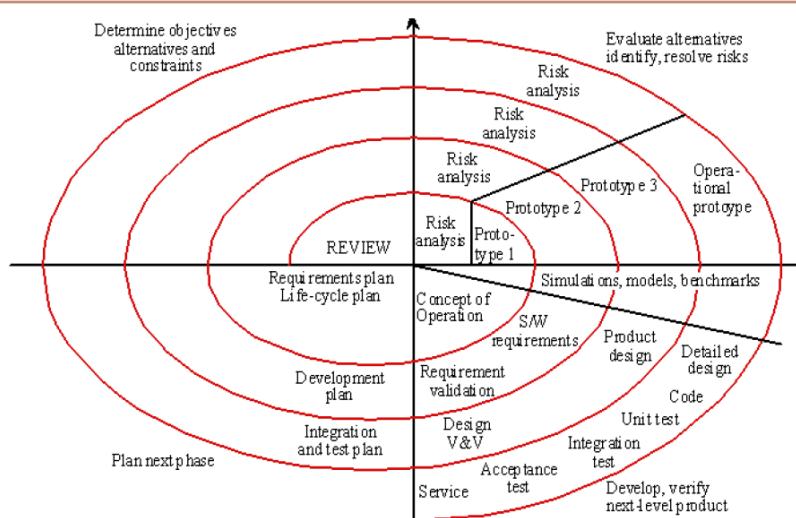
- ▶ Permet de réévaluer les risques en cours de développement
- ▶ Chaque cycle de la spirale se déroule en quatre phases
 - ▶ 1. détermination des objectifs du cycle, des alternatives pour les atteindre et des contraintes ; à partir des résultats des ces précédents , ou de l'analyse préliminaire des besoins;
 - ▶ 2. analyse des risques, évaluation des alternatives à partir de maquettage et/ou prototypage;
 - ▶ 3. développement et vérification de la solution retenue, un modèle « classique » (cascade ou en V) peut être utilisé ici ;
 - ▶ 4. revue des résultats et vérification du cycle suivant.

37

Walid DHOUIB – ISET SFAX

37

MODÈLE EN SPIRALE



38

Walid DHOUIB – ISET SFAX

38

19

Analyse des risques

► Risques humains:

- Défaillance du personnel ; surestimation des compétences
- Travailleur solitaire, héroïsme, manque de motivation

► Risques processus

- Pas de gestion de projet
- Calendrier et budget irréalistes ;
- Calendrier abandonné sous la pression des clients
- Composants externes manquants ;
- Tâches externes défaillantes ;
- Insuffisance de données
- Validité des besoins ;
- Développement de fonctions inappropriées
- Développement d'interfaces utilisateurs inappropriées

► Risques technologiques

- Produit miracle, "plaqué or";
- Changement de technologie en cours de route
- Problèmes de performance
- Exigences démesurées par rapport à la technologie
- Incompréhension des fondements de la technologie

39

Walid DHOUIB – ISET SFAX

39

Conditions d'application

- Le modèle en spirale s'applique essentiellement en interne , lorsque les clients et les fournisseurs font partie de la même entreprise, si l'analyse de risque démontre que le projet doit être continué, une équipe peut être réaffectée au projet.
- Alors que dans une relation client-fournisseur ordinaire, il y a eu signature de contrat et donc l'effort doit être estimé à l'avance. Le modèle en spirale ne peut donc s'appliquer. Ou bien il doit être adapté en signant des contrats partiels pour chaque itération.

40

Walid DHOUIB – ISET SFAX

40

Modèle RAD : "RAPID APPLICATION DEVELOPMENT "

- ▶ Ce modèle de développement tend à raccourcir le cycle de vie voire à le supprimer.
 - ▶ La phase de spécification/conception est remplacée par une phase de prototypage menée conjointement avec le client
 - ▶ La phase de prototypage débouche sur une interface validée par le client
 - ▶ L'outil génère des squelettes de fonctions , classes...
 - ▶ Le comportement de chaque objet de l'interface est ensuite décrit dans un langage approprié et ses fonctionnalités programmées.
 - ▶ Des soucis lors de la maintenance à cause du manque de conception inhérent à la démarche, une conception basée sur l'interface.
 - ▶ Récemment la méthode DSDM est apparue qui prend en compte ces remarques et structure l'approche RAD.
 - ▶ La méthode s'applique bien dans le cadre de petites applications de gestion, n'ayant pas de cycle de vie d'une trop longue durée.

41

Walid DHOUIB – ISET SFAX

41

42

Deuxième leçon

- Définitions
 - Problèmes qualité logiciel



42

Logiciel : Définition

- Tout ou partie des programmes, procédures, règles et documentation associées à un système de traitement de l'information
- Programmes informatiques, procédures et éventuellement documentation et données associées concernant le fonctionnement d'un système informatique

ISO/IEC/IEEE 24765

43

Walid DHOUIB – ISET SFAX

43

Logiciel : Définition

- **Programmes** : les instructions qui ont été traduites en code source, qui ont été spécifiés, conçus, examinés, testés à l'unité et acceptés par les clients ;
- **Procédures** : les procédures utilisateur et autres processus qui ont été décrits (avant et après automatisation), analysés et optimisés;
- **Règles** : les règles, telles que les règles métier ou les règles de processus, qui vont être compris, décrit, validé, mis en œuvre et testé;
- **Documentation associée** : tous les types de documentation utiles aux clients, utilisateurs de logiciels, développeurs, auditeurs et mainteneurs. La documentation permet aux différents membres d'une équipe de mieux communiquer, examiner, tester et maintenir les logiciels. La documentation est définie et produite tout au long des étapes clés du cycle de vie du logiciel;
- **Données** : informations inventoriées, modélisées, standardisées et créées pour faire fonctionner le système informatique

44

Walid DHOUIB – ISET SFAX

44

Micrologiciel

- Combinaison d'un périphérique matériel et d'instructions informatiques ou de données informatiques qui résident en tant que logiciel en lecture seule sur le périphérique matériel.

ISO 24765

45

Walid DHOUIB – ISET SFAX

45

ERREURS, DÉFAUTS ET PANNE DU LOGICIEL

De nombreux termes sont utilisés pour décrire des problèmes avec un système logiciel. Par exemple:

- Le système s'est **bloqué (Crashed)** pendant la production.
- Le concepteur a fait une **erreur (Error)**.
- Après un examen, nous avons trouvé un **défaut (Defect)** dans le plan de test.
- J'ai trouvé un **bogue (Bug)** dans un programme aujourd'hui.
- Le système est **tombé en panne (Broke down)**.
- Le client s'est plaint d'un **problème (Problem)** avec un calcul dans le rapport de paiement.
- Une **panne (Failure)** a été signalée dans le sous-système de surveillance.

46

Walid DHOUIB – ISET SFAX

46

Types de conséquences graves des Bugs

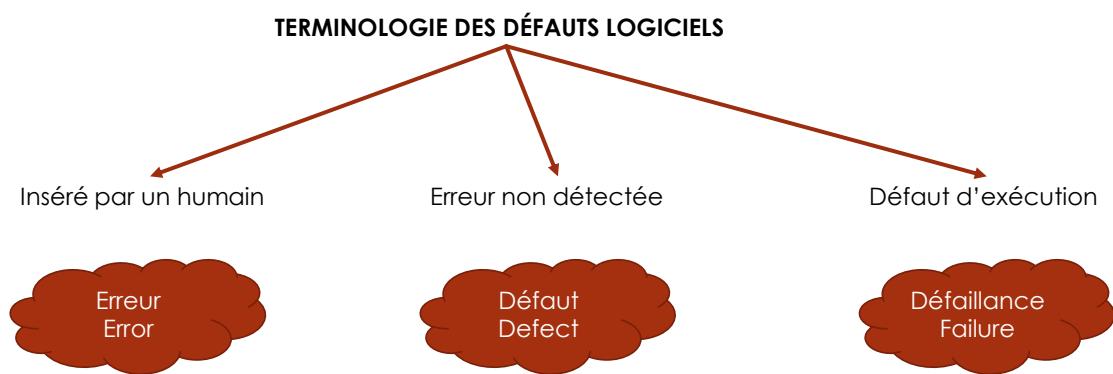
- ▶ Types de conséquences graves des Bugs
 - ▶ Physiques : Blessures ou mort
 - ▶ Financières : Coût de correction, impact sur les utilisateurs ou clients
- ▶ Les acteurs concernés
 - ▶ L'entreprise
 - ▶ Les utilisateurs finaux
- ▶ Un coût complexe difficile à calculer :
 - ▶ Temps de correction
 - ▶ Pénalités de retard de livraison
 - ▶ Impact commercial en terme de confiance et image de la société

47

Walid DHOUIB – ISET SFAX

47

TERMINOLOGIE RECOMMANDÉE POUR DÉCRIRE LES DÉFAUTS LOGICIELS



48

Walid DHOUIB – ISET SFAX

48

ERREUR, DÉFAUT ET DÉFAILLANCE

- **Erreur / Error (Mistake)** : Une action humaine qui produit un résultat incorrect (ISO 24765)
- **Défaut (Bug, faute) / Defect (Bug, Fault)**
 - Un problème (synonyme de défaut) qui, s'il n'est pas corrigé, pourrait entraîner l'échec d'une application ou produire des résultats incorrects. (ISO 24765)
 - Une imperfection ou une déficience dans un logiciel ou un composant du système qui peut empêcher le composant de remplir sa fonction, par ex. une définition de données ou une instruction de code source incorrectes. Un défaut, s'il est exécuté, peut entraîner la défaillance d'un logiciel ou d'un composant système (ISTQB 2011).
- **Défaillance / Failure** :
 - La fin de la capacité d'un produit à exécuter une fonction requise ou son incapacité à fonctionner dans les limites précédemment spécifiées (ISO 25010)
 - Ecart constaté du composant ou système par rapport au livrable, au service ou au résultat attendu (ISTQB 2011)

49

Walid DHOUIB – ISET SFAX

49

Incident

- Tout évènement arrivant pendant les tests qui requiert une vérification
- Divergence entre le résultat obtenu et le résultat attendu lors de l'exécution d'un test

50

Walid DHOUIB – ISET SFAX

50

CAS D'ERREURS, DE DÉFAUTS ET DE DÉFAILLANCES

- Cas 1: La pharmacie Alocal a ajouté une exigence logicielle à sa caisse enregistreuse pour empêcher les ventes de plus de 75 \$ aux clients qui doivent plus de 200 \$ sur leur carte de crédit de pharmacie. Le programmeur n'a pas entièrement compris la spécification et a créé une limite de ventes de 500 \$ dans le programme. Ce défaut n'a jamais causé de panne car aucun client pouvait acheter plus de 500 \$ d'articles étant donné que la carte de crédit de la pharmacie avait une limite de 400 \$.

51

Walid DHOUIB – ISET SFAX

51

CAS D'ERREURS, DE DÉFAUTS ET DE DÉFAILLANCES

- Cas 2: En 2009, un programme de fidélité a été mis en place auprès des clients d'un grand fournisseur de meubles. Les spécifications décrivent les règles commerciales suivantes: un client qui effectue un achat mensuel supérieur au montant moyen des achats mensuels pour tous les clients sera considéré comme un client privilégié. Le client privilégié sera identifié lors d'un achat et recevra immédiatement un cadeau ou une remise importante une fois par mois. Le défaut introduit dans le système (dû à une mauvaise compréhension de l'algorithme à mettre en place pour cette exigence) impliquait de ne prendre en compte que le montant moyen des achats en cours et non l'historique mensuel du client. Au moment de la panne du logiciel, la caisse enregistreuse identifiait beaucoup trop de clients privilégiés, ce qui entraînait une perte pour l'entreprise.

52

Walid DHOUIB – ISET SFAX

52

CAS D'ERREURS, DE DÉFAUTS ET DE DÉFAILLANCES

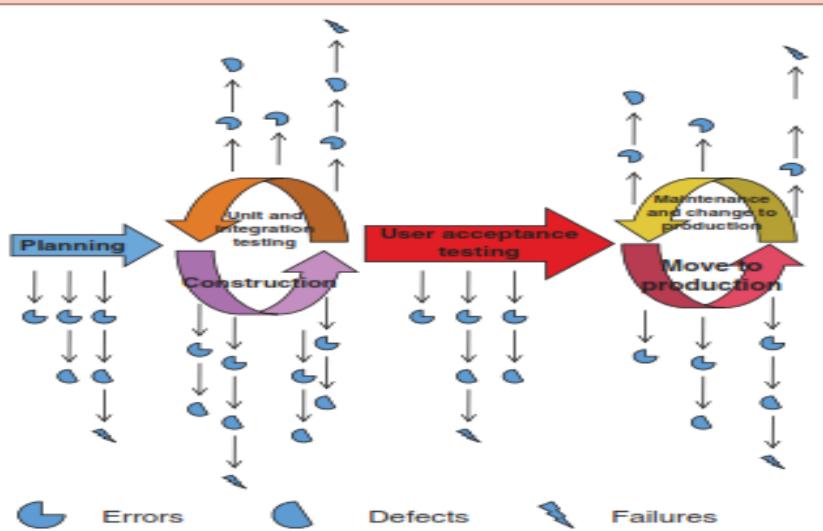
- Cas 3: Peter a testé le programme de Patrick quand Patrick était absent. Il a trouvé un défaut dans le calcul d'un plan d'épargne-retraite destiné à appliquer la nouvelle loi d'exonération fiscale pour ce type d'investissement. Il a retracé l'erreur jusqu'à la spécification du projet et a informé l'analyste. Dans ce cas, l'activité de test a correctement identifié le défaut et la source de l'erreur.

53

Walid DHOUIB – ISET SFAX

53

ERREURS, DÉFAUTS ET DÉFAILLANCES DANS LE CYCLE DE VIE DU LOGICIEL



54

Walid DHOUIB – ISET SFAX

54

De l'erreur à la défaillance

Erreur (Humain)



Défaut (Bug) dans le code, le résultat de l'erreur



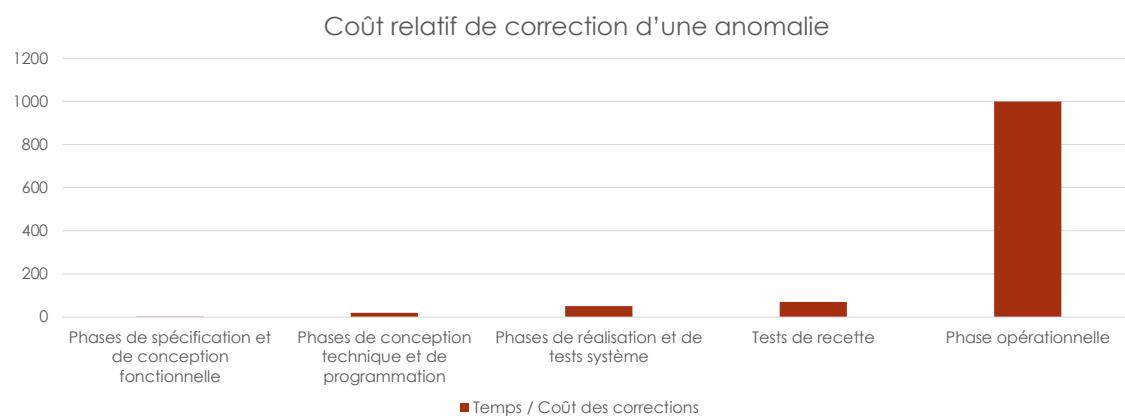
Défaillance résultat de l'exécution d'un défaut dans le code :
Le système fait ou ne fait pas ou fait différemment

55

Walid DHOUIB – ISET SFAX

55

Coût de correction des défauts



56

Walid DHOUIB – ISET SFAX

56

LES PROBLEMES DE QUALITE LOGICIEL

- Les erreurs peuvent survenir à n'importe quelle phase de développement logiciel tout au long du cycle de vie.
- Les défauts doivent être identifiés et corrigés avant qu'ils ne deviennent des défaillances.
- La cause des défaillances, défauts et erreurs doit être identifiée.

57

Walid DHOUIB – ISET SFAX

57

LES SOURCES D'ERREUR LOGICIEL

- Beizer (1990) [BEI 90] fournit une étude qui a combiné les résultats de plusieurs autres études pour nous fournir une indication sur l'origine des erreurs.
 - 25% structurel;
 - 22% de données;
 - 16% de fonctionnalités implémentées;
 - 10% construction / codage;
 - 9% d'intégration;
 - 8% des exigences / spécifications fonctionnelles;
 - 3% de définition / tests en cours d'exécution;
 - 2% architecture / design;
 - 5% non spécifié.

58

Walid DHOUIB – ISET SFAX

58

LES 8 CATEGORIES DES ERREURS LOGICIEL

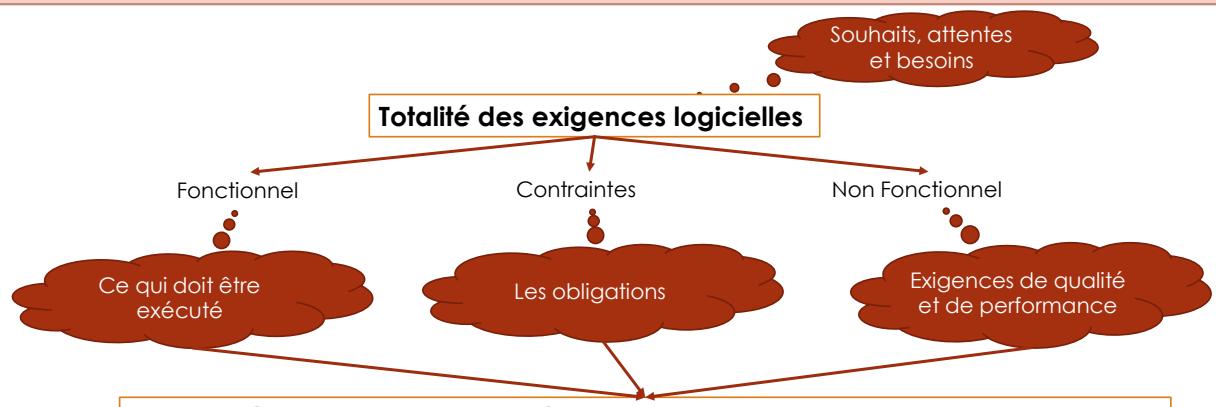
- 1) problèmes de définition des exigences;
- 2) maintenir une communication efficace entre le client et le développeur;
- 3) écarts par rapport aux spécifications;
- 4) les erreurs d'architecture et de conception;
- 5) erreurs de codage (y compris le code de test);
- 6) non-respect des processus / procédures en vigueur;
- 7) examens et tests inadéquats;
- 8) erreurs de documentation.

59

Walid DHOUIB – ISET SFAX

59

DETERMINATION DES EXIGENCES LOGICIELLES



- Connaissance du domaine et règles métier
- Environnement organisationnel et opérationnel
- Entretiens, brainstorming et réunions animées
- Observation, histoires utilisateurs et scénarios
- Classification et documentation des exigences

60

Walid DHOUIB – ISET SFAX

60

1- PROBLÈMES DE DÉFINITION DES EXIGENCES

Pour bien comprendre le besoin il faut :

- identifier les parties prenantes (c'est-à-dire les acteurs clés) qui doivent participer à l'élaboration des exigences;
- gérer les réunions;
- techniques d'interview permettant d'identifier les différences entre les souhaits, les attentes et les besoins réels;
- une documentation claire et concise des exigences fonctionnelles, des exigences de performance, des obligations et des propriétés des futurs systèmes;
- appliquer des techniques systématiques d'élicitation des besoins;
- gérer les priorités et les changements (par exemple, les changements aux exigences).

61

Walid DHOUIB – ISET SFAX

61

1- PROBLÈMES DE DÉFINITION DES EXIGENCES

Une exigence est dite de bonne qualité lorsqu'elle répond aux caractéristiques suivantes:

- | | |
|---|--|
| <ul style="list-style-type: none"> ▪ Correct; ▪ Achevée; ▪ Faisable (réaliste, possible); ▪ Concis (simple, précis); ▪ Clair pour chaque groupe de parties prenantes (par exemple, le client, l'architecte système, les testeurs, et ceux qui maintiendront le système); ▪ Cohérent; ▪ Vérifiable et testable; | <ul style="list-style-type: none"> ▪ Sans ambiguïté, (même interprétation de l'exigence de la part de toutes les parties prenantes); ▪ Nécessaire (répond au besoin d'un client); ▪ Indépendant de la conception; ▪ Indépendant de la technique de mise en œuvre; ▪ Peut être rattaché à un besoin commercial; ▪ Unique. |
|---|--|

62

Walid DHOUIB – ISET SFAX

62

2- PROBLEMES DE MAINTIEN DE COMMUNICATIONS ENTRE LE CLIENT ET LE DÉVELOPPEUR

- Mauvaise compréhension des instructions du client;
- Le client veut des résultats immédiats;
- Le client ou l'utilisateur ne prend pas le temps de lire la documentation qui lui est adressée;
- Mauvaise compréhension des changements demandés aux développeurs lors de la conception;
- L'analyste arrête d'accepter les modifications pendant la phase de définition des exigences et de conception, étant donné que pour certains projets 25% des spécifications auront changé avant la fin du projet.

63

Walid DHOUIB – ISET SFAX

63

POUR MINIMISER LES ERREURS

- Prendre des notes à chaque réunion et distribuer le compte rendu à toute l'équipe du projet;
- Revoir les documents produits;
- Être cohérent avec votre utilisation des termes et développer un glossaire des termes à partager avec toutes les parties prenantes;
- Informer les clients du coût de modification des spécifications;
- Choisissez une approche de développement qui vous permet d'accepter les changements en cours de route;
- Numéroter chaque exigence et mettre en place un processus de gestion du changement.

64

Walid DHOUIB – ISET SFAX

64

3- ÉCARTS PAR RAPPORT AUX SPÉCIFICATIONS

Cette situation se produit lorsque le développeur interprète incorrectement une exigence et développe le logiciel en fonction de sa propre compréhension. Cette situation crée des erreurs qui ne peuvent malheureusement être détectées que plus tard dans le cycle de développement ou lors de l'utilisation du logiciel.

Les autres types d'écart sont:

- Réutiliser le code existant sans procéder aux ajustements adéquats pour répondre aux nouvelles exigences;
- Décider d'abandonner une partie des besoins en raison de contraintes budgétaires ou de temps;
- Initiatives et améliorations introduites par les développeurs sans vérification avec les clients.

65

Walid DHOUIB – ISET SFAX

65

4 - ERREURS D'ARCHITECTURE ET DE CONCEPTION

Des erreurs peuvent être insérées dans le logiciel lorsque les concepteurs traduisent les exigences des utilisateurs en spécifications techniques exemple :

- Une vue d'ensemble incomplète du logiciel à développer;
- Rôle peu clair de chaque composant de l'architecture logicielle;
- Données primaires non spécifiées et classes de traitement de données;
- Une conception qui n'utilise pas les algorithmes appropriés pour répondre aux exigences;
- Une séquence de processus métier ou technique incorrecte;
- Mauvaise conception des critères métier ou des règles de processus;
- Une conception qui ne remonte pas aux exigences;
- Omission des statuts de transaction qui représentent correctement le processus du client;
- L'incapacité de traiter les erreurs et les opérations illégales, qui permet au logiciel de traiter des cas qui n'existaient pas dans le secteur d'activité du client - on estime que jusqu'à 80% du code du programme traite des exceptions ou des erreurs.

66

Walid DHOUIB – ISET SFAX

66

5 - ERREURS DE CODAGE

Les erreurs de programmation typiques sont:

- Choix inappropriate du langage de programmation et des conventions;
- Ne pas aborder la façon de gérer la complexité dès le début;
- Mauvaise compréhension / interprétation des documents de conception;
- Abstractions incohérentes;
- Erreurs de boucle et de condition;
- Erreurs de traitement des données;
- Traitement des erreurs de séquence;
- Absence ou mauvaise validation des données lors de leur entrée;

67

Walid DHOUIB – ISET SFAX

67

5 - ERREURS DE CODAGE (SUITE)

- Mauvaise conception des critères de règles métier;
- Omission des statuts de transaction nécessaires pour représenter véritablement le processus client;
- Défaut de traitement des erreurs et des opérations illégales, qui permet au logiciel de traiter des cas qui n'existaient pas dans le secteur d'activité du client;
- Mauvaise affectation ou traitement du type de données;
- Erreur de boucle ou interférence avec l'indice de boucle;
- Manque de compétences pour gérer les nids extrêmement complexes;
- Problème de division entière;
- Mauvaise initialisation d'une variable ou d'un pointeur;
- Code source qui ne remonte pas à la conception;
- Confusion concernant un alias pour les données globales

68

Walid DHOUIB – ISET SFAX

68

6 - NON-CONFORMITÉ AVEC LES PROCESSUS / PROCÉDURES ACTUELS

Les logiciels non documentés entraîneront tôt ou tard les problèmes suivants:

- Lorsque les membres de l'équipe logiciel ont besoin de coordonner leur travail, ils auront des difficultés à comprendre et à tester des logiciels mal documentés ou non documentés.
- La personne qui remplace ou maintient le logiciel n'aura que le code source comme référence.
- SQA trouvera un grand nombre de non-conformités (par rapport à la méthodologie interne) concernant ce logiciel.
- L'équipe de test aura des difficultés à élaborer des plans et des scénarios de test, principalement parce que les spécifications ne sont pas disponibles.

69

Walid DHOUIB – ISET SFAX

69

7 - EXAMENS ET TESTS INADÉQUATS

Le but des revues et tests de logiciels est d'identifier et de vérifier que les erreurs et les défauts ont été éliminés du logiciel. Si ces activités ne sont pas efficaces, le logiciel livré au client sera probablement sujet à l'échec.

- Les revues ne couvrent qu'une très petite partie des livrables intermédiaires du logiciel;
- Les revues n'identifient pas toutes les erreurs trouvées dans la documentation et le code logiciel;
- La liste des recommandations issues des revues n'est pas mise en œuvre ou suivie de manière adéquate;
- Les plans de test incomplets ne couvrent pas correctement l'ensemble des fonctions du logiciel, laissant des parties non testées;
- Le plan de projet n'a pas laissé beaucoup de temps pour effectuer des revues ou des tests. Dans certains cas, cette étape est raccourcie car elle est coincée entre le codage et la livraison finale. Les retards dans les premières étapes du projet ne signifient pas toujours que la date de livraison sera prolongée, au détriment des tests appropriés;
- Le processus de test ne rapporte pas correctement les erreurs ou les défauts constatés;
- Les défauts détectés sont corrigés, mais ne sont pas soumis à des tests de régression adéquats (c'est-à-dire, un nouveau test du logiciel corrigé complet) par la suite.

70

Walid DHOUIB – ISET SFAX

70

8 - ERREURS DE DOCUMENTATION

Il a été reconnu que la documentation obsolète ou incomplète des logiciels utilisés dans une organisation est un problème courant. Peu d'équipes de développement aiment passer du temps à préparer et à examiner la documentation.

Outre la classification des types d'erreurs, il est important de comprendre la courbe de fiabilité typique des logiciels. La figure 1 décrit la courbe de fiabilité du matériel informatique en fonction du temps. Cette courbe s'appelle une courbe en forme de U ou de baignoire. Il représente la fiabilité d'un équipement, comme une voiture, tout au long de son cycle de vie.

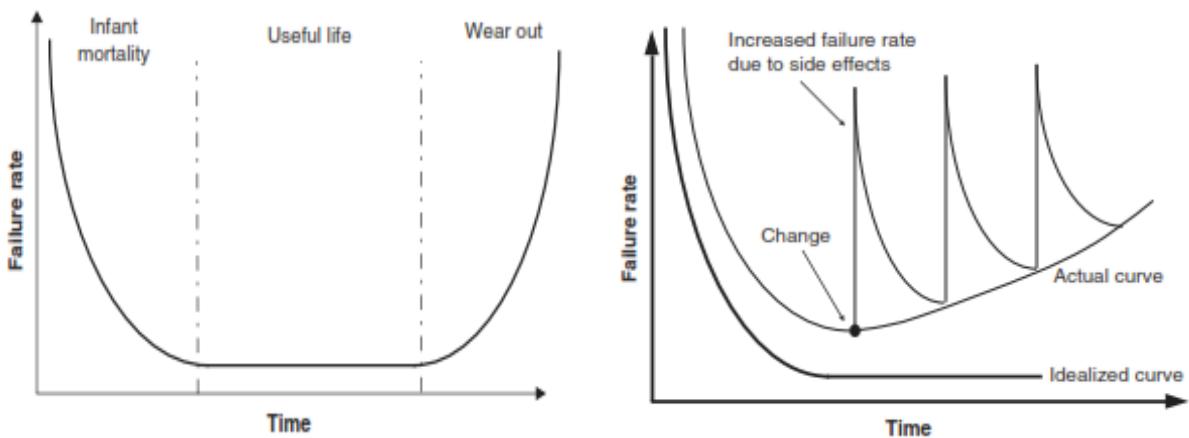
En ce qui concerne les logiciels, la courbe de fiabilité ressemble davantage à ce que montre la figure 2. Cela signifie que la détérioration du logiciel se produit avec le temps en raison, entre autres, de nombreux changements dans les exigences.

71

Walid DHOUIB – ISET SFAX

71

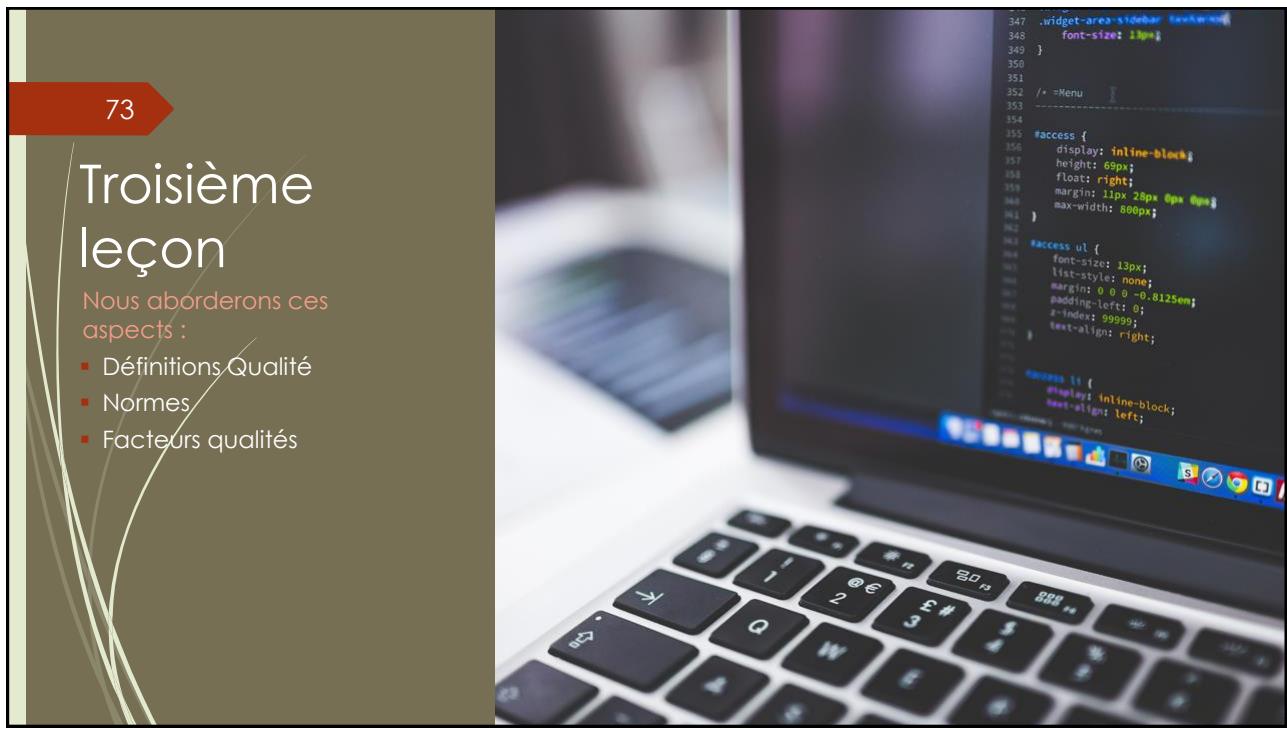
8 - ERREURS DE DOCUMENTATION



72

Walid DHOUIB – ISET SFAX

72



73

Troisième leçon

- Définitions Qualité
 - Normes
 - Facteurs qualités

73

QUALITÉ : DÉFINITION

- **Qualité** : Aptitude d'un ensemble de caractéristiques (physiques, sensorielles, comportementales, temporelles, ergonomiques, Fonctionnelles) *intrinsèques d'un objet (produit, service, processus, système, organisme, logiciel...)* à satisfaire des exigences (source : l'ISO 9000 : 2015)
 - **Exigence** : besoin ou attente formulé, généralement implicite ou obligatoire

74

Walid DHOUIB – ISET SFAX

74

QUALITÉ LOGICIEL : DÉFINITION

Qualité du logiciel :

- Conformité aux exigences logicielles établies; la capacité d'un logiciel à satisfaire les besoins déclarés et implicites lorsqu'il est utilisé dans des conditions spécifiées (ISO 25010).
- La mesure dans laquelle un produit logiciel répond aux exigences établies; cependant, la qualité dépend de la mesure dans laquelle ces exigences établies représentent avec précision les besoins, les désirs et les attentes des parties prenantes (IEEE 730)

75

Walid DHOUIB – ISET SFAX

75

Normalisation ISO

- ▶ ISO : International Standard Organisation
- ▶ C'est une fédération mondiale d'organismes nationaux de normalisation ayant pour but de contribuer au développement de la normalisation et produire des normes internationales dans les domaines industriels et commerciaux appelées normes ISO afin de simplifier les échanges de produits et/ou de services entre pays
- ▶ L'ISO n'a aucun pouvoir réglementaire ou législatif
- ▶ ISO Online - ISO sur Internet <http://www.iso.org>

76

Walid DHOUIB – ISET SFAX

76

Normalisation IEC ou CEI

- ▶ IEC : International Electrotechnical Commission en anglais
- ▶ CEI : Commission électrotechnique internationale en français
- ▶ C'est une organisation internationale de normalisation chargée des domaines de l'électricité, de l'électronique, de la compatibilité électromagnétique, de la nanotechnologie et des techniques connexes.
- ▶ Elle est complémentaire de l'Organisation internationale de normalisation (ISO), qui est chargée des autres domaines.

77

Walid DHOUIB – ISET SFAX

77

Normes IEEE

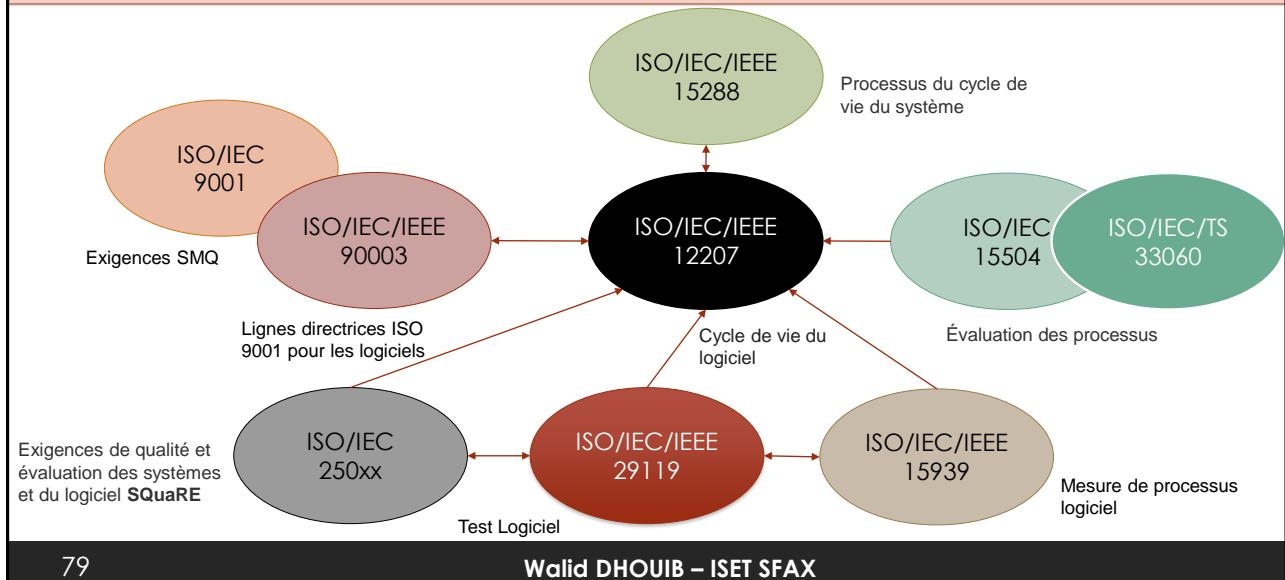
- ▶ IEEE : Institute of Electrical and Electronics Engineers
- ▶ en français l'**« Institut des ingénieurs électriciens et électroniciens »**, est une association professionnelle
- ▶ L'IEEE compte plus de 400 000 membres et possède différentes branches dans plusieurs parties du monde
- ▶ L'IEEE est constituée d'ingénieurs électriciens, d'informaticiens, de professionnels du domaine des télécommunications, etc.
L'organisation a pour but de promouvoir la connaissance dans le domaine de l'ingénierie électrique (électricité et électronique).

78

Walid DHOUIB – ISET SFAX

78

Normes ISO



79

Walid DHOUIB – ISET SFAX

79

Norme ISO 9001 : 2015

- Elle spécifie les exigences relatives au système de management de la qualité lorsqu'un organisme:
 - a) doit démontrer son aptitude à fournir constamment des produits et des services conformes aux exigences des clients et aux exigences légales et réglementaires applicables, et
 - b) vise à accroître la satisfaction de ses clients par l'application efficace du système, y compris les processus pour l'amélioration du système et l'assurance de la conformité aux exigences des clients et aux exigences légales et réglementaires applicables.
- Toutes les exigences de la présente Norme internationale sont génériques et prévues pour s'appliquer à tout organisme, quels que soient son type ou sa taille, ou les produits et services qu'il fournit.

80

Walid DHOUIB – ISET SFAX

80

7 Principes de l'ISO 9000 : 2015

- ▶ Orientation Client
- ▶ Responsabilité de la direction ou LEADERSHIP
- ▶ Implication du Personnel
- ▶ Approche Processus et l'approche système
- ▶ Amélioration Continue
- ▶ Prise de décision fondée sur des preuves
- ▶ Gestion des relations avec les parties intéressées (client, fournisseur, personnel, sous-traitant, partenaire, banque, assurance...)

81

Walid DHOUIB – ISET SFAX

81

Chapitres de la norme ISO 9001 : 2015

- I. Domaine d'application
- II. Références normatives
- III. Termes et définitions
- IV. Contexte de l'organisme
- V. Leadership
- VI. Planification
- VII. Support
- VIII. Réalisation des activités opérationnelles
- IX. Evaluation des performances
- X. Amélioration

82

Walid DHOUIB – ISET SFAX

82

Norme ISO/IEC/IEEE 90003-2018

Ingénierie des logiciels

- ▶ Dérivée de la norme de management de la qualité ISO 9001 : 2015
- ▶ Concerne l'ingénierie des logiciels
- ▶ Fournit des lignes directrices pour l'application de l'ISO 9001 : 2015 à l'acquisition, à la fourniture, au développement, à l'exploitation et à la maintenance de logiciels informatiques et de services de support associés.
- ▶ Elle n'ajoute et ne modifie d'aucune autre manière les exigences de la norme ISO 9001: 2015.

83

Walid DHOUIB – ISET SFAX

83

ISO/IEC/IEEE 12207:2017

Ingénierie des systèmes et du logiciel — Processus du cycle de vie du logiciel

- ▶ ISO / IEC / IEEE 12207: 2017 fournit également des processus qui peuvent être utilisés pour définir, contrôler et améliorer les processus du cycle de vie des logiciels au sein d'une organisation ou d'un projet
- ▶ Les processus, activités et tâches de ce document peuvent également être appliqués lors de l'acquisition d'un système qui contient un logiciel, seul ou conjointement avec ISO / IEC / IEEE 15288: 2015, Systèmes et génie logiciel - Processus du cycle de vie du système
- ▶ Dans le contexte de ce document et de l'ISO / CEI / IEEE 15288, il existe un système créés par l'homme, qui utilise peu ou pas de logiciel à ceux dans lesquels le logiciel est le principal intérêt. Il est rare de rencontrer un système complexe sans logiciel, et tous les systèmes logiciels nécessitent des composants physiques du système (matériel) pour fonctionner, soit dans le cadre du système logiciel d'intérêt, soit comme système ou infrastructure habilitante.

84

Walid DHOUIB – ISET SFAX

84

ISO/IEC/IEEE 15288:2015

Ingénierie des systèmes et du logiciel — Processus du cycle de vie du système

- ▶ ISO / IEC / IEEE 15288: 2015 établit un cadre commun de descriptions de processus pour décrire le cycle de vie des systèmes créés par l'homme. Elle définit un ensemble de processus et la terminologie associée d'un point de vue technique. Des ensembles sélectionnés de ces processus peuvent être appliqués tout au long du cycle de vie pour gérer et exécuter les étapes du cycle de vie d'un système.
- ▶ ISO / IEC / IEEE 15288: 2015 fournit également des processus qui prennent en charge la définition, le contrôle et l'amélioration des processus de cycle de vie du système utilisés dans une organisation ou un projet. Les organisations et les projets peuvent utiliser ces processus lors de l'acquisition et de la fourniture de systèmes.

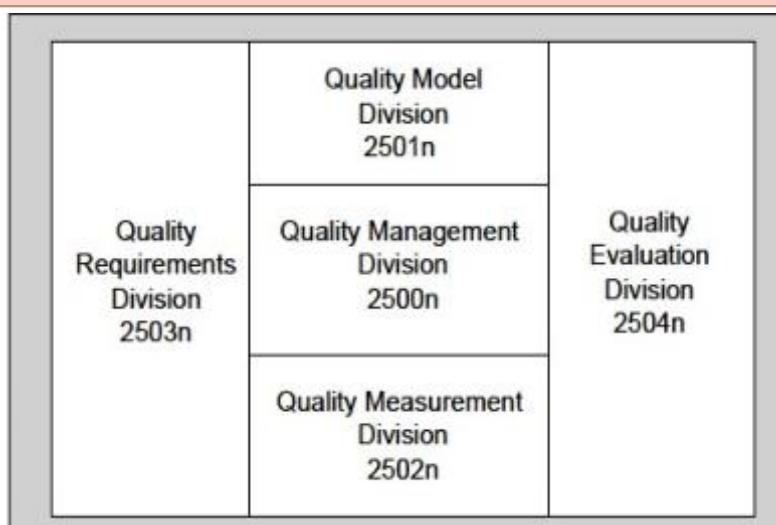
85

Walid DHOUIB – ISET SFAX

85

La série des normes ISO 25000

La série des normes ISO 25000 relative aux exigences de qualité et évaluation des logiciels « **SQuaRE** » appelées aussi divisions.



86

Walid DHOUIB – ISET SFAX

86

Norme ISO/IEC 25010:2011

Ingénierie des systèmes et du logiciel — Exigences de qualité et évaluation des systèmes et du logiciel (SQuaRE) — Modèles de qualité du système et du logiciel

- ▶ Précédemment : ISO/IEC 9126-1:2001
- ▶ ISO / IEC 25010: 2011 définit :
 - ▶ Un modèle de qualité de fonctionnement 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.
 - ▶ Un modèle de qualité de produit composé de huit caractéristiques (qui sont ensuite subdivisées en sous-caractéristiques) qui se rapportent aux propriétés statiques du logiciel et aux propriétés dynamiques du système informatique.
- ▶ Le modèle est applicable aux systèmes informatiques et aux produits logiciels.

87

Walid DHOUIB – ISET SFAX

87

Structure utilisée pour le modèle qualité



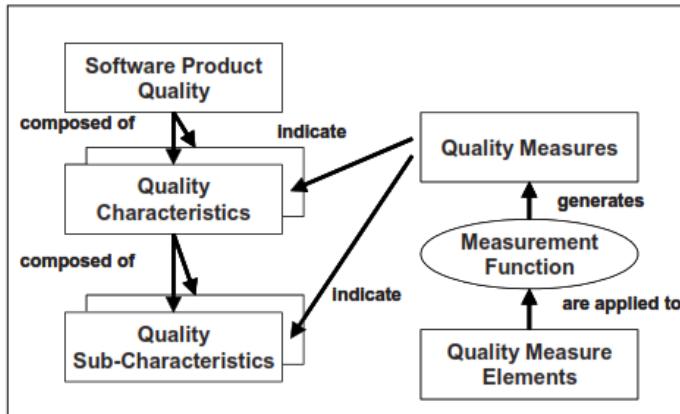
Figure 2 — Structure used for the quality models

88

Walid DHOUIB – ISET SFAX

88

Relations entre l'ISO 25010 et l'ISO 2502n



89

Walid DHOUIB – ISET SFAX

89

Modèles de qualité dans la série SQuaRE

- ▶ Modèle de qualité en cours d'utilisation
- ▶ Modèle de qualité du produit
- ▶ Modèle de qualité des données dans ISO / CEI 25012.

90

Walid DHOUIB – ISET SFAX

90

Modèles de qualité dans la série SQuaRE

- ▶ Ces modèles de qualité servent ensemble de cadre pour garantir que toutes les caractéristiques de qualité sont considérés.
- ▶ Ces modèles fournissent un ensemble de caractéristiques de qualité pertinentes pour un large éventail de parties prenantes, telles que: développeurs de logiciels, intégrateurs de systèmes, acquéreurs, propriétaires, mainteneurs, sous-traitants, professionnels de l'assurance et du contrôle de la qualité et utilisateurs.
- ▶ L'ensemble complet des caractéristiques de qualité de ces modèles ne sera pas pertinent pour toutes les parties prenantes.
- ▶ Chaque catégorie de parties prenantes doit être représentée dans l'examen et la prise en compte de la pertinence des caractéristiques de qualité dans chaque modèle avant de finaliser l'ensemble des caractéristiques de qualité qui seront utilisées

91

Walid DHOUIB – ISET SFAX

91

Exigence de qualité en cours d'utilisation

(basées sur ISO/IEC 25010, 25022, 25030)

Système/Logiciel Qualité en cours d'utilisation

Efficacité

Efficiency

Satisfaction

Absence des risques

Couverture contextuel

Efficacité

Efficiency

Utilité

Confiance

Plaisir

Confort

Réduction des risques économiques

Réduction des risques liés à la santé et à la sécurité

Réduction des risques environnementaux

Complétude du contextuel

Flexibilité

92

Walid DHOUIB – ISET SFAX

92

Modèle de qualité du logiciel en cours d'utilisation

- ▶ La qualité en usage est la mesure dans laquelle un produit ou un système peut être utilisé par des utilisateurs spécifiques pour répondre à leurs besoins afin d'atteindre des objectifs spécifiques
- ▶ Le modèle de qualité d'utilisation définit cinq caractéristiques liées aux résultats de l'interaction avec un système : Efficacité, efficience, satisfaction, absence de risque et couverture du contexte
- ▶ Chaque caractéristique peut être attribuée à différentes activités des parties prenantes, par exemple, l'interaction d'un opérateur ou la maintenance d'un développeur.
- ▶ Ce modèle concerne le résultat de l'interaction du produit quand il est utilisé dans un contexte particulier.
- ▶ Ce modèle est applicable au système personne-machine complet, y compris les systèmes informatiques et les logiciels.

93

Walid DHOUIB – ISET SFAX

93

Modèle de qualité du logiciel en cours d'utilisation

- ▶ Efficacité : précision et exhaustivité avec lesquelles les utilisateurs atteignent les objectifs spécifiés
- ▶ Efficience : ressources dépensées en fonction de l'exactitude et de l'exhaustivité avec lesquelles les utilisateurs atteignent leurs objectifs
- ▶ la satisfaction : degré de satisfaction des besoins des utilisateurs lorsqu'un produit ou système est utilisé dans un contexte d'utilisation spécifié
- ▶ Utilité : degré auquel un utilisateur est satisfait de la réalisation perçue d'objectifs pragmatiques, y compris les résultats de l'utilisation et les conséquences de l'utilisation
- ▶ Confiance : degré auquel un utilisateur ou une autre partie prenante a confiance qu'un produit ou un système se comportera comme prévu
- ▶ Plaisir : degré auquel un utilisateur obtient du plaisir de répondre à ses besoins personnels
- ▶ Confort : degré auquel l'utilisateur est satisfait du confort physique

94

Walid DHOUIB – ISET SFAX

94

Modèle de qualité du logiciel en cours d'utilisation

- ▶ Absence de risque: degré auquel un produit ou un système atténue le risque potentiel pour la situation économique, la vie humaine, la santé ou l'environnement
- ▶ Réduction des risques économiques : degré auquel un produit ou un système réduit le risque potentiel pour la situation financière, le fonctionnement efficace, la propriété commerciale, la réputation ou d'autres ressources dans les contextes d'utilisation prévus
- ▶ Réduction des risques pour la santé et la sécurité : degré auquel un produit ou un système réduit le risque potentiel pour les personnes dans les contextes d'utilisation prévus
- ▶ Réduction des risques environnementaux : degré auquel un produit ou un système atténue le risque potentiel pour la propriété ou l'environnement dans les contextes d'utilisation prévus

95

Walid DHOUIB – ISET SFAX

95

Modèle de qualité du logiciel en cours d'utilisation

- ▶ Couverture du contexte : degré auquel un produit ou un système peut être utilisé avec efficacité, efficience, absence de risque et satisfaction dans les deux contextes d'utilisation spécifiés et dans des contextes au-delà de ceux initialement explicitement identifiés
- ▶ Complétude du contexte : degré auquel un produit ou un système peut être utilisé avec efficacité, efficience, sans risque et satisfaction dans tous les contextes d'utilisation spécifiés
- ▶ Flexibilité : degré auquel un produit ou un système peut être utilisé avec efficacité, efficience, absence de risque et satisfaction dans des contextes au-delà de ceux initialement spécifiés dans les exigences

96

Walid DHOUIB – ISET SFAX

96

Modèle de qualité du produit logiciel

- ▶ Le modèle de qualité du produit classe les propriétés de qualité du produit système / logiciel en huit caractéristiques.
- ▶ Ces caractéristiques et sous-caractéristiques fournissent une terminologie cohérente pour spécifier, mesurer et évaluer la qualité des systèmes et produits logiciels.
- ▶ Ils fournissent également un ensemble de caractéristiques de qualité qui peut être comparée aux exigences qualité initiales aux fins d'exhaustivité.

97

Walid DHOUIB – ISET SFAX

97

Exigences de qualité du produit (Partie 1)

(basées sur ISO/IEC 25010, 25023, 25030)

Système/Logiciel Qualité du produit

La capacité fonctionnelle

- Complétude fonctionnelle
- Exactitude fonctionnelle
- Pertinence fonctionnelle

Les performances

- Comportement temporel
- Utilisation des ressources
- Capacité

Compatibilité

- Coexistence
- Interopérabilité

L'utilisabilité

- L'aptitude à la reconnaissance
- Facilité d'apprentissage
- L'opérabilité
- Protection contre les erreurs de l'utilisation
- Esthétique de l'interface utilisateur
- Accessibilité

98

Walid DHOUIB – ISET SFAX

98

Exigences de qualité du produit (Partie 2)

(basées sur ISO/IEC 25010, 25023, 25030)

Système/Logiciel Qualité du produit

Robustesse

Sécurité

Maintenabilité

Portabilité

- Maturité
- Disponibilité
- Tolérance aux pannes
- La récupérabilité

- Confidentialité
- Intégrité
- Non-répudiation
- La responsabilité
- Authenticité

- Modularité
- La réutilisabilité
- L'analysabilité
- La possibilité de modification
- testabilité

- L'adaptabilité
- Facilité d'installation
- Remplaçabilité

99

Walid DHOUIB – ISET SFAX

99

La capacité fonctionnelle

► La complétude fonctionnelle (Functional completeness)

- L'ensemble des fonctions couvrent toutes les tâches spécifiées et les objectifs de l'utilisateur
- Ex : L'utilisateur doit pouvoir réaliser toutes les actions attendues, de la recherche à l'achat final, en passant par l'ajout dans un panier.

► L'exactitude fonctionnelle (Functional correctness)

- Le produit ou système met à disposition les résultats corrects avec un degré de précision nécessaire
- Ex : Si je cherche des livres des années 80, j'ai bien la liste des livres des années 80.

► La pertinence fonctionnelle (Functional appropriateness)

- Les fonctions facilitent l'accomplissement de tâches et d'objectifs précis
- Ex : L'ajout d'un article dans le panier se fait sur les pages attendues, l'utilisateur se voit proposer des articles souvent achetés en même temps et l'enchaînement des actions fait tout son sens.

100

Walid DHOUIB – ISET SFAX

100

Les performances

► Le comportement temporel (Time behaviour)

- ▶ Les temps de réponse, de traitement et de débits d'un produit ou d'un système, lors de l'exécution de ses fonctions, satisfont aux exigences
- ▶ Ex : La recherche de produits doit afficher les résultats en moins d'une seconde.

► L'utilisation des ressources (Resource utilization)

- ▶ Les quantités et types de ressources utilisés par un produit ou un système, dans l'exercice de ses fonctions, satisfont aux exigences
- ▶ Ex : Je souhaite qu'un cluster seul puisse supporter une charge de 1 000 utilisateurs en simultané.

► La capacité (Capacity)

- ▶ Les limites maximales d'un produit ou d'un paramètre du système répondent aux exigences
- ▶ Ex : Je souhaite que mon système utilise au maximum 50Go de données par mois

101

Walid DHOUIB – ISET SFAX

101

La compatibilité

► La co-existence

- ▶ Un produit peut exécuter ses fonctions de manière efficace tout en partageant un environnement et des ressources en communs avec d'autres produits, sans impact négatif sur tout autre produit
- ▶ Ex : L'application de e-commerce doit co-exister avec l'application de back-office permettant de faire les comptes de l'entreprise qui tourne sur le même serveur.

► L'interopérabilité (Interoperability)

- ▶ Deux ou plus de systèmes, produits ou composants peuvent échanger des informations et utiliser les informations échangées
- ▶ Ex : L'application de e-commerce doit dispatcher la commande aux services de livraison.

102

Walid DHOUIB – ISET SFAX

102

L'utilisabilité (partie 1)

► L'aptitude à la reconnaissance (Appropriateness recognizability)

- Les utilisateurs peuvent reconnaître qu'un produit ou un système est adapté à leurs besoins
- Ex : Les utilisateurs utilisent l'application pour acheter des articles dont ils ont besoin.

► La facilité d'apprentissage (Learnability)

- Un produit ou un système peut être utilisé par des utilisateurs spécifiques pour atteindre des objectifs précis d'apprentissage de l'utilisation du produit ou du système avec efficacité, efficience, absence de risque et de satisfaction dans un contexte d'utilisation spécifique
- Ex : L'utilisateur est accompagné tout le long de son parcours sur l'ensemble du site, il comprend comment aller jusqu'à l'achat.

► L'opérabilité (Operability)

- Un produit ou un système a des attributs qui le rend facile à utiliser et à contrôler
- Ex : Je peux contrôler l'intérieur de mon panier, retirer des articles ajoutés par erreur, visualiser le prix final après éventuelles réductions.

103

Walid DHOUIB – ISET SFAX

103

L'utilisabilité (partie 2)

► La protection contre les erreurs utilisateur (User error protection)

- Un produit ou un système est protégé contre les erreurs issues des utilisateurs
- Ex : Si l'utilisateur saisit une chaîne de caractères invalides, l'application doit le détecter et retourner un message d'erreur.

► L'esthétique de l'interface utilisateur (User interface aesthetics)

- Une interface utilisateur permet une interaction agréable et satisfaisante pour l'utilisateur
- Ex : Les résultats des tests UX/UI montrent les informations importantes pour l'utilisateur de manière claire et agréable.

► L'accessibilité (Accessibility)

- Un produit ou un système peut être utilisé par des personnes ayant le plus grand nombre de caractéristiques et de capacités pour atteindre un objectif spécifique dans un contexte d'utilisation spécifique
- Ex : Les résultats des tests UX/UI montrent que l'application est suffisamment simple pour pouvoir être utilisée par des personnes de plus de 60 ans.

104

Walid DHOUIB – ISET SFAX

104

La robustesse

- ▶ **La maturité (Maturity)**
 - ▶ Un système, un produit ou un composant répond aux besoins de fiabilité en fonctionnement normal
 - ▶ Ex : L'application est stable et opérationnelle.
- ▶ **La disponibilité (Availability)**
 - ▶ Un système, un produit ou un composant est opérationnel et accessible au besoin
 - ▶ Ex : L'application est accessible en ligne 24/24, 7/7, à un taux de disponibilité de 99,9%.
- ▶ **La tolérance aux pannes (Fault tolerance)**
 - ▶ Un système, un produit ou un composant fonctionne comme prévu malgré la présence de pannes matérielles ou logicielles
 - ▶ Ex : Si un serveur tombe, un autre prend le relai en moins d'une seconde. Les erreurs logicielles ne font pas tomber le matériel.
- ▶ **La récupérabilité (Recoverability)**
 - ▶ En cas d'interruption ou de défaillance, un produit ou un système peut récupérer les données directement affectées et rétablir l'état souhaité du système
 - ▶ Ex : Le serveur ayant pris le relai a gardé les articles dans les paniers de tous les utilisateurs actuellement connectés.

105

Walid DHOUIB – ISET SFAX

105

La sécurité

- ▶ **La confidentialité (Confidentiality)**
 - ▶ Un produit ou un système garantit que les données ne sont accessibles qu'aux personnes autorisées à y avoir accès
 - ▶ Ex : Le gérant de l'application a des accès à des données que les autres utilisateurs ne peuvent pas obtenir de quelque façon que ce soit.
- ▶ **L'intégrité (Integrity)**
 - ▶ Un système, un produit ou un composant empêche l'accès non autorisé à des programmes informatiques ou des données, ou la modification de ceux-ci
 - ▶ Ex : Seuls les accès via l'application de e-commerce peuvent modifier le contenu du panier des utilisateurs.
- ▶ **La non-réputation (Non-reputation)**
 - ▶ Des actions ou des événements peuvent être prouvés avoir eu lieu, de sorte que les événements ou les actions ne peuvent être réputées plus tard
 - ▶ Ex : La validation de l'achat et du paiement ne peuvent être contestés.
- ▶ **La responsabilité (Accountability)**
 - ▶ Les actions d'une entité peuvent être tracées à l'entité de façon unique
 - ▶ Ex : Chaque action sur l'interface contient l'identité de son auteur.
- ▶ **L'authenticité (Authenticity)**
 - ▶ Preuve de l'identité d'un sujet ou d'une ressource
 - ▶ Ex : L'application garanti que c'est bien cet utilisateur qui a passé la commande.

106

Walid DHOUIB – ISET SFAX

106

La maintenabilité (1)

► La modularité (Modularity)

- Un système ou un programme informatique est composé de composants distincts, de sorte qu'un changement sur un composant a un impact minimal sur les autres composants
- Ex : L'ajout d'une nouvelle brique d'analyses des tendances des achats n'impose pas la refonte totale de l'application.

► La réutilisabilité (Reusability)

- Un composant ou une brique peut être utilisé dans plusieurs systèmes ou dans la création d'autres composants ou briques
- Ex : Le module de choix de date peut être utilisé partout, aussi bien dans le profil de l'utilisateur pour sa date de naissance que pour choisir sa date de livraison à la fin de son panier.

► L'analysabilité (Analysability)

- Efficacité avec lequel il est possible d'évaluer l'impact sur un produit ou un système d'une modification envisagée à une ou plusieurs de ses parties, ou de diagnostiquer un produit pour détecter des défaillances ou des causes de défaillance, ou identifier des parties à modifier
- Ex : Si je veux modifier la fonctionnalité d'ajout d'un article dans le panier, je sais exactement tout ce que cela impacte. Les erreurs intervenues lors de son utilisation sont consultables dans un journal qui me permet d'y remédier.

107

Walid DHOUIB – ISET SFAX

107

La maintenabilité (2)

► La possibilité de modification (Modifiability)

- Un produit ou un système peut être modifié efficacement sans introduire des défauts ou dégrader la qualité du produit existant
- Ex : Le code est suffisamment bien construit et toute évolution n'entraîne pas de régression dans les fonctionnalités existantes.

► La testabilité (Testability)

- Efficacité et efficience avec lequel des critères de test peuvent être établis pour un système, un produit ou un composant et des tests peuvent être effectués pour déterminer si ces critères ont été respectés
- Ex : Je dispose de différentes façons pour tester efficacement le code, que ce soit avec des tests unitaires, d'intégration, de bout en bout ou de performance.

108

Walid DHOUIB – ISET SFAX

108

La portabilité

► L'adaptabilité (Adaptability)

- Un produit ou un système peut être adapté de manière efficace et efficiente à des matériels, logiciels ou autres environnements opérationnels ou d'utilisation différents ou en constante évolution
- Ex : Le site web peut être utilisé depuis un smartphone, peu importe son OS et sa résolution.

► La facilité d'installation (Installability)

- Efficacité et efficience avec lequel un produit ou un système peut être installé et/ou désinstallé avec succès dans un environnement spécifié
- Ex : L'application peut être déployée sur une machine peu importe son OS ou sa configuration en moins de quelques minutes, voire secondes.

► La remplaçabilité (Replaceability)

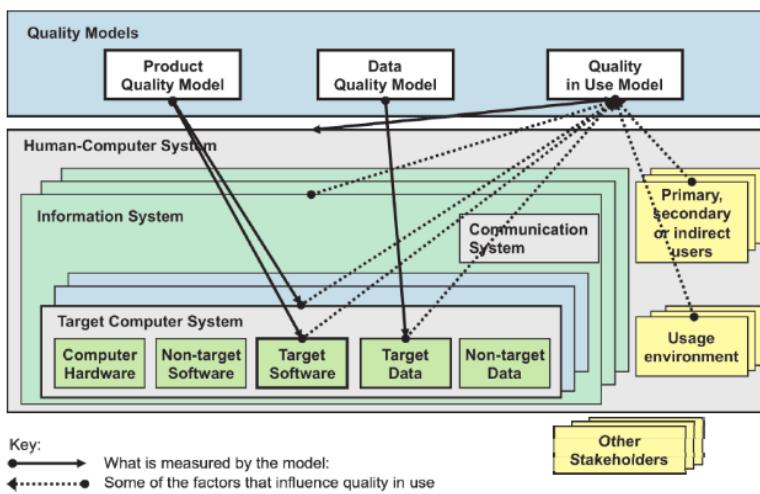
- Un logiciel ou un produit peut en remplacer un autre spécifiquement pour le même usage dans le même environnement
- Ex : Je dois pouvoir remplacer la sous application d'achat par une autre conçue dans une toute autre technologie sans grands impacts sur le reste de l'écosystème.

109

Walid DHOUIB – ISET SFAX

109

L'objectif des modèles de qualité



110

Walid DHOUIB – ISET SFAX

110

L'objectif des modèles de qualité

- ▶ Le modèle de qualité du produit se concentre sur le système informatique cible qui comprend le produit logiciel cible, et le modèle de qualité d'utilisation se concentre sur l'ensemble du système homme-ordinateur qui comprend le système informatique cible et le produit logiciel cible.
- ▶ Le système informatique cible comprend également du matériel informatique, des produits logiciels non cibles, des données non cibles et des données cibles, qui font l'objet du modèle de qualité des données.
- ▶ Le système informatique cible est inclus dans un système d'information qui peut également inclure un ou plusieurs systèmes informatiques et systèmes de communication, tels qu'un réseau local et Internet.
- ▶ Le système d'information fait partie d'un système homme-ordinateur plus large (tel qu'un système d'entreprise, un système intégré ou un système de contrôle à grande échelle) et peut inclure des utilisateurs et l'environnement d'utilisation technique et physique. L'endroit où est jugée la limite du système dépend de la portée des exigences ou de l'évaluation et de l'identité des utilisateurs.

111

Walid DHOUIB – ISET SFAX

111

Utilisation d'un modèle qualité

- ▶ Les modèles de qualité du produit et de qualité d'utilisation sont utiles pour spécifier les exigences, établir des mesures et effectuer des évaluations de la qualité.
- ▶ Les caractéristiques de qualité définies peuvent être utilisées comme liste de contrôle pour assurer un traitement complet des exigences de qualité, fournissant ainsi une base pour estimer l'effort et les activités qui seront nécessaires pendant le développement des systèmes.
- ▶ Les caractéristiques du modèle de qualité d'utilisation et du modèle de qualité du produit sont destinées à être utilisées comme un ensemble lors de la spécification ou de l'évaluation de la qualité du système informatique ou du produit logiciel.

112

Walid DHOUIB – ISET SFAX

112

Utilisation d'un modèle qualité

- ▶ Il n'est pratiquement pas possible de spécifier ou de mesurer tous sous-caractéristiques de toutes les parties d'un grand système informatique ou d'un produit logiciel.
- ▶ il n'est généralement pas pratique de spécifier ou de mesurer la qualité d'utilisation pour tous les scénarios possibles de tâches utilisateur.
- ▶ L'importance relative des caractéristiques de qualité dépendra des buts et objectifs de haut niveau du projet. Par conséquent, le modèle doit être adapté avant utilisation dans le cadre de la décomposition des exigences afin d'identifier les caractéristiques et sous-caractéristiques les plus importantes, et les ressources allouées entre les différents types de mesures en fonction des buts et objectifs des parties prenantes pour le produit.

113

Walid DHOUIB – ISET SFAX

113

La qualité du point de vue de différentes parties prenantes

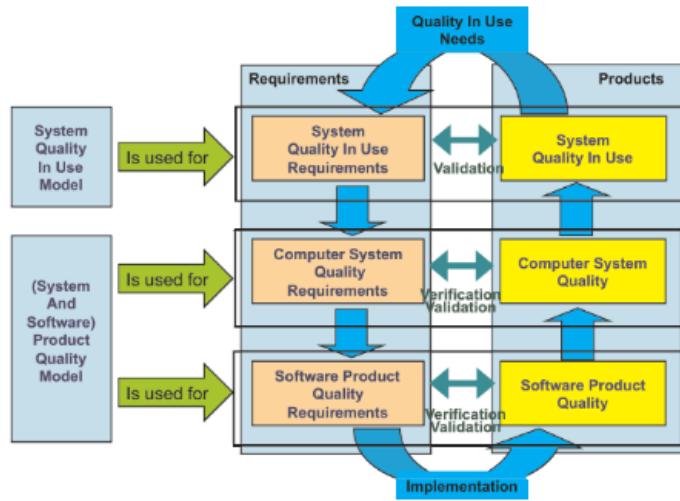
- ▶ Les modèles de qualité fournissent un cadre pour la collecte des besoins des parties prenantes.
- ▶ Les parties prenantes comprennent les types d'utilisateurs suivants:
 1. Utilisateur principal: personne qui interagit avec le système pour atteindre les objectifs principaux.
 2. Utilisateurs secondaires qui fournissent une assistance, par exemple
 - a) fournisseur de contenu, gestionnaire / administrateur système, responsable de la sécurité;
 - b) mainteneur, analyseur, porteur, installateur.
 3. Utilisateur indirect: personne qui reçoit la sortie, mais n'interagit pas avec le système.

114

Walid DHOUIB – ISET SFAX

114

Qualité système et logiciel dans le modèle de cycle de vie



115

Walid DHOUIB – ISET SFAX

115