

TD n°1: Introduction

1 - Génie Logiciel : désigne l'ensemble des méthodes, des techniques et d'outils concourant à la production d'un logiciel de qualité avec maîtrise des coûts et délais.

Logiciel : est l'ensemble des programmes et de documents nécessaires à leur installation, utilisation, développement et maintenance.

2 - Les principaux facteurs de qualité d'un logiciel :

- Validité : (correction, justesse, conformité) est la capacité que possède un produit logiciel à remplir exactement ses fonctions, définies par le cahier des charges et les spécifications. Adéquation entre :
Le besoin effectif de l'utilisateur,
Les fonctions offertes par le logiciel.
- Fiabilité ou Robustesse : la robustesse (fiabilité, sûreté) est la capacité qu'offrent des systèmes logiciels à réagir de manière appropriée à la présence de conditions anormales (i.e. rien de catastrophique ne peut survenir, même en dehors des conditions d'utilisation prévues).
- Facilité d'utilisation : la facilité d'utilisation est la facilité avec laquelle des personnes présentant des formations et des compétences différentes peuvent apprendre à utiliser les produits logiciels et s'en servir pour résoudre des problèmes. Elle recouvre également la facilité d'installation, d'opération et de contrôle.
- Compatibilité : la compatibilité est la facilité avec laquelle des éléments logiciels peuvent être combinés à d'autres (un logiciel doit pouvoir interagir en synergie avec d'autres logiciels).
- Efficacité : l'efficacité est la capacité d'un système logiciel à utiliser le minimum de ressources matérielles, que ce soit le temps machine, l'espace occupé en mémoire externe et interne, ou la bande passante des moyens de communication (les logiciels doivent satisfaire aux contraintes de temps d'exécution).
- Portabilité : la portabilité est la facilité avec laquelle des produits logiciels peuvent être transférés d'un environnement logiciel ou matériel à l'autre (un même logiciel doit pouvoir fonctionner sur plusieurs machines).
- Réutilisabilité : la réutilisabilité est la capacité des éléments logiciels à servir à la construction de plusieurs applications différentes (80 % du code est du " tout venant " qu'on retrouve à peu près partout, 20 % du code est spécifique).

- **Maintenabilité** : la maintenabilité est le degré de facilité de la maintenance d'un produit logiciel.
 - **Extensibilité** : l'extensibilité est la facilité d'adaptation des produits logiciels aux changements de spécifications.
-
- **Intégrité** : aptitude d'un logiciel à protéger son code et ses données contre des accès non autorisés.
 - **Ponctualité** : la ponctualité est la capacité d'un système logiciel à être livré au moment désiré par ses utilisateurs, ou avant.

Donc, un bon logiciel bien fait est un logiciel :

correct (valide),
 fiable (robuste),
 avec un code réutilisable,
 compatible avec d'autres logiciels,
 efficace,
 portable,
 facile à utiliser,
 maintenable,
 ponctuel et
 extensible.

3 -

Le Logiciel professionnel est associé à une documentation pour informer le client des différentes étapes d'installation et d'utilisation et même des détails concernant le logiciel (le processus de développement) pour l'aider à bien comprendre le fonctionnement de ce dernier.

On distingue 3 sortes de documents :

1. Documentation concernant la spécification, l'architecture et la conception
2. Documentation technique (code, API, Algos, scripts d'installation et de configuration)
3. Manuel d'utilisation.

4 -

1. Faisabilité : (POURQUOI ?)
 - Pourquoi développer le logiciel ?

- Y a-t-il de meilleures alternatives ?
- Comment procéder pour faire ce développement ?
- Y a-t-il un marché pour le logiciel ?
- Quels moyens faut-il mettre en œuvre ? A-t-on le budget, le personnel, la matériel nécessaires ?

Activité	Etude préalable
Description	Etudier la faisabilité du projet, ses contraintes techniques (coût, temps, qualité) et les alternatives possibles.
Entrées	Problème à résoudre, objectifs à atteindre.
Sorties	OUI (la décision est prise de réaliser le logiciel) ou NON (le projet est abandonné).

2. Spécification : (QUOI ?)

Activité	Spécifier
Description	Décrire ce que doit faire le logiciel (comportement en boîte-noire). Décrire comment vérifier en boîte-noire que le logiciel fait bien ce qui est exigé.
Entrées	Client qui à une idée de ce qu'il veut. Exigences, désir, besoins... concernant le système permettant de résoudre le problème.
Sorties	Cahier des charges du logiciel (ou spécification du logiciel). Des procédures de validation. Version provisoire des manuels d'utilisation et d'exploitation du logiciel.

3. Conception : (COMMENT ?)

Activité	Concevoir
Description	Organiser le logiciel afin qu'il puisse satisfaire les exigences de la spécification. Faire les principaux choix techniques pour satisfaire les exigences de la spécification.
Entrées	Une spécification.
Sorties	Une description des décisions de conception. Des procédures de tests qui permettent de vérifier que les décisions de conception sont correctement implémentées en code source et qu'elles contribuent à satisfaire les exigences de la spécification.

4. Implantation : (COMMENT ?)

Activité	Coder et tester
Description	Écrire le code source du logiciel. Tester le comportement du code source afin de vérifier qu'il réalise les responsabilités qui lui sont allouées.
Entrées	Spécification, conception.
Sorties	Code source. Tests unitaire. Documentation.

5. Intégration :

Activité	Intégrer
Description	Assembler le code source du logiciel (éventuellement partiellement) et dérouler les tests d'intégration.

Entrées	Conception, code source, tests d'intégration (tests).
Sorties	Un rapport de tests d'intégration.

6. Validation :

Activité	Valider
Description	Construire le logiciel complet exécutable. Dérouler les tests de validation sur le logiciel complet exécutable.
Entrées	Logiciel complet exécutable à valider. Tests de validation.
Sorties	Rapport de tests de validation.

7. Maintenance :

Activités :

- Maintenance corrective : correction des bugs.
- Maintenance adaptative : ajuster le logiciel.
- Maintenance perfective, d'extension : augmenter / améliorer les possibilités du logiciel.

Productions :

- Logiciel corrigé ;
- Mises à jour ;
- Documents corrigés.

Il existe deux types de produits logiciels (génériques et sur mesure)

- La différence essentielle est que dans le développement de produits logiciels génériques, la spécification est possédée par le développeur du produit. Pour le développement de produits personnalisés (sur mesure), la spécification est détenue et contrôlée par le client.

Les implications sont importantes - le développeur peut rapidement décider de changer la spécification en réponse à un changement externe (par exemple un produit concurrent), mais, lorsque le client possède les spécifications, les changements doivent

être négociés entre le client et le développeur et peuvent avoir des conséquences contractuelles.

6 - Pour les utilisateurs de produits génériques, cela signifie qu'ils n'ont aucun contrôle sur la spécification de logiciels et ne peuvent donc pas contrôler l'évolution du produit. Le développeur peut décider d'inclure/exclure des fonctionnalités et changer l'interface utilisateur. Cela pourrait avoir des implications pour les processus métiers de l'utilisateur et ajouter des coûts de formation supplémentaires lorsque de nouvelles versions du système sont installées. Il peut également limiter la flexibilité du client à changer leurs propres processus métiers.

7 - Les quatre caractéristiques standards sont :

La maintenabilité (maintainability), la fiabilité (dependability or reliability), la performance ou l'efficacité (performance), et la utilisabilité (facile à utiliser, convivialité) ou acceptabilité (usability).

8 - D'autres attributs peuvent être inclus:

- La réutilisabilité (reusability) : peut être utilisé dans d'autres applications,
- la distribuabilité (distributability) : peut être distribué sur un réseau ou sur un ensemble de processeurs,
- la portabilité (portability) : peut fonctionner sur des plateformes multiples, et
- l'interopérabilité (interoperability) : peut travailler avec une large gamme de systèmes logiciels.

9 - La fiabilité et la performance des logiciels :

1. **Fiabilité :**

Correction, justesse, conformité : le logiciel est conforme à ses spécifications, les résultats sont ceux attendus.

Robustesse, sureté : le logiciel fonctionne raisonnablement en toutes circonstances, rien de catastrophique ne peut survenir, même en dehors des conditions d'utilisation prévues

10 - Quelques métriques:

- **MTBF** : Mean Time Between Failures
- **Disponibilité** (pourcentage du temps pendant lequel le système est utilisable) et Taux

d'erreur (nombre d'erreurs par KLOC)

2. **Performance** :

Les logiciels doivent satisfaire aux contraintes de temps d'exécution
Quelques métriques: Temps d'exécution.