|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| General Définitions (qualité) **Générale** : conformité avec les besoins, adéquation avec l’usage attendu, degré d’excellence, la valeur de quelque chose pour quelqu’un.  **ISO** : Ensemble des traits et des caractéristiques d’un produit logiciel portant sur l’aptitude à satisfaire des besoins exprimés ou implicites.  **IEEE** : Correspond au degré selon lequel un logiciel possède une combinaison d’attributs désirés.  **Crosby** : Correspond au degré selon lequel un client perçoit le degré de réponse d’un logiciel.  **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 Définitions **Assurance qualité** : Ensemble d’activités conçues pour évaluer le processus par lequel les produits sont développés ou fabriqués.  **Contrôle de qualité** : Ensemble d’activités conçues pour évaluer la qualité du produit – Sous ensemble de l’assurance qualité – Sujet du cours.  Qualité du logiciel Pourquoi évaluer la qualité? Achat d’un logiciel (alternatives) – Le logiciel délivre les fonctionnalités? – Dans quelle mesure y arrive-t-il? ISO/IEC 9126 (guide qualité) **Capacité** **fonctionnelle** : Ensemble d'attributs portant sur l'existence d'un ensemble de fonctions et leurs propriétés. Fonctions = besoins (exprimés ou implicites) satisfaits. **Sous-caractéristiques** : Aptitude – Capacité à faire des choses; Exactitude – Effets / Résultats justes ou convenus; Sécurité – Empêcher tout accès non autorisé aux données; Interopérabilité – Capacité à interagir avec des systèmes donnés. | **Fiabilité** : Ens. d'attributs portant sur l'aptitude à maintenir un niveau de service dans des conditions précises et pendant une période déterminée. **Sous-caract.** : Maturité; Tolérance aux fautes – Maintenir un niveau de service (après crash, panne, temps, etc.); Possibilité de récupération – Rétablir le niveau de service et la restauration des données.  **Convivialité** : Ens. d'attributs portant sur l'effort nécessaire pour l’utilisation. **Sous-caract.** : Facilité de compréhension, d’apprentissage et d’exploitation.  **Rendement (Efficience)** : Ens. d'attributs portant sur le rapport existant entre le niveau de service et la quantité des ressources utilisées, dans des conditions déterminées. **S-C.** : Comportement vis-à-vis du temps (temps de réponse/traitement) et des ressources (quantité de ressources utilisées).  **Maintenabilité** : Ens. d'attributs portant sur l'effort nécessaire pour faire des modifications données. **S-C** : Facilité d’analyse et de modification; Stabilité – Risque des effets inattendus des modifications; Facilité de test – Effort nécessaire pour valider le logiciel modifié.  **Portabilité** : Ens. d'attributs portant sur l'aptitude du logiciel à être transféré d’un environnement à l’autre. **S-C** : Facilité d’adaptation (Linux, Windows, iOS) et d’installation; Conformité aux règles (normes et conventions) de portabilité; Interchangeabilité. Processus d’évaluation (ISO 9126) **(1) Définition des exigences de qualité** : « *En tant que <qui>, je veux <quoi> afin de <pourquoi>* ». Utiliser les 6 caract. (guide qualité) pour spécifier les exigences.  **(2) La préparation de l’évaluation** : Sélection des métriques de qualité (correspondent aux caractéristiques énumérées) – Définition des taux de satisfaction (échelles de valeurs) – Définition des critères d’appréciation (préparation et compilation des résultats par caractéristiques)  **(3) La procédure d’évaluation** : Mesure (les métriques sont appliquées au produit, donnant des valeurs) – Notation (pour chaque valeur mesurée, une note de (de satisfaction) est attribuée) – Appréciation (un résultat global de l’évaluation est obtenu). Processus SCOPE ??? Les critères d'évaluation – La spécification de l'évaluation – Le plan de l'évaluation – Le rapport d'évaluation | Théorie de la mesure Définitions **Entité** : Objet – **Attribut** : Propriété – **Mesure** : processus où l’on affecte des symboles aux attributs. \*\*\* **Calcul** : indirecte – **Mesure** : direct   Types d’échelles **Nominale** : Aucun ordre. Les choix doivent être mutuellement exclusifs, ex : rouge, bleu, masculin, féminin, etc.  **Ordinale** : Ajout de l’ordre à l’échelle nominale, ex : petit, moyen, grand. – Transitivité : a < b et b < c, alors a < c. – Complétude : Il doit y avoir une relation d’ordre entre tous les choix.  **Intervalle** : Distance relative entre deux classes de mesures : soit *v1* et *v2* (deux valeurs successives), *v1*-*v2* = *distance*, Exemples : température (Celsius ou F\*).  **Ratio**: Zéro absolu possible, ex : Tempéra- ture en Kelvin, temps entre 2 fautes.  **Absolue** : Ne peut être mesuré que d’une seule façon. Consiste généralement à compter, ex : Nombre de voitures dans le stationnement, Nombre d’erreurs, etc. Transformations  |  |  |  | | --- | --- | --- | | **N** | Fonctions un à un. Ex : Rouge -> 1, bleu -> 2 | Mode  Fréquence  =, != | | **O** | Fonctions croi-ssantes qui pré-servent l’ordre | Médiane  Percentile  <, > | | **I** |  | Moy. Arith  Écart Type  +, -, \*, / | | **R** |  | Moy. Géo  Déviance Géo.  Tout | | **A** |  | Tout |   **Tendance centrale**  **Écart**  **Opérateurs** |
| Définir une mesure (exemples) **<E, R> vers <E, M, O>**  **Ex1** : Définir une mesure de visibilité pour un ensemble de classes. Savoir si une classe est plus visible (publique) qu’une autre.  **Ensemble (E)** : { C1,C2,C3 … Cn }; **Relation (R)** : est plus publique que; **Opérateur (O)** : >;  **Mesure (M)** M(C) = % des méthodes publiques.  **Ex2** : Définir une mesure de généralité pour un ensemble de classes. Savoir si une classe est plus générale (moins de méthodes abstraites) qu’une autre.  **E** : { C1,C2,C3 … Cn }; **R** : est plus générale que; **O** : >; **M** M(Ci) = nombres de méthodes abstraites dans Ci  Condition de représentation : si M(Cx) > M(Cy) alors Cx est plus général que Cy   Transformations (exemples) Une Audi coûte 2 fois le prix d’une Toyota. **Rép**. : oui, le prix est ratio.  Pour le prix de ton 5½ à Outremont, tu peux t’offrir un 5½ à Rosemont et un 3½ à Villeray. **Rép.** : oui. Le prix est ratio.    L’épisode d’aujourd’hui a débuté 2 fois plus tôt que la semaine passée.  **Rép**. : non, la date est en intervalles.  Il s’est écoulé 2 fois plus de temps que la dernière fois.  **Rép**. : oui, la durée est ratio.  Revoir  Démo 2 – 5  Démo 3 –  Démo 4 – | Mesure du produit logiciel Attributs internes **Spécification** : taille, réutilisation, modularité, fonctionnalité, redondance – **Conception** : t/r/m, couplage, fonctionnalité – **Code** : t/r/m/c/f, complexité algorithme, structuration du flot de contrôle. Taille physique (mesures) **LOC :** Lines of code  **NCLOC**: Non-Comment Lines Of Code (on ne compte pas les commentaires ni les lignes vides)  Densité des commentaires :  **Méthode Walston & Felix** : .  **D** : Taille doc; **L** : Taille du code (1k par unité). Taille (fonctionnalité) **Points de fonctions** : Mesurer la quantité de fonctionnalité – ;  UFC : ∑({Compter : entrées ext, sorties ext, req ext, fichiers int/ext} \* {facteurs de pondération})  TFC : facteur d’ajustement de UFC dans une plage de ±35% (0.65 <=TFC<=1.35).  **BANG** (fonctions) : ?  **BANG** (données) : Compter le nombre d’entités dans le diagramme Entités/Relations. | Mesure de la complexité **Complexité cyclomatique de McCabe** (3 façons de calculer): Couplage **R0** : Pas de couplage – **R1** : Données (communiquent avec des params) – **R2** : Signature (même params) – **R3** : Contrôle (X passe param qui influence Y) – **R4** : Variable globale commune – **R5** : Contenu (X modifie des données ou des instructions dans Y).  **Ex** : M2 et M4 ont 3 instances de couplage: : 2 instances de R3 (couplage de contrôle), 1 instance de R5 (couplage de contenu). Complexité du flux d’information |