Module 6

Dans la Phase 5A : Réalisation technique du nouveau SI, approche modulaire, à quoi sert le découpage en modules dans la planification du développement logiciel?

Répartition du travail entre les développeur (ou entre équipes)

Identification claire des responsabilités de chaque développeur

Permet de gérer l’intégration des divers modules

Permet de planifier leurs tests (unitaires et d’intégration)

Dans la phase 5A : Réalisation technique du nouveau SI, en quoi consiste les tests?

Il s’agit de faire exécuter un programme afin de localiser des erreurs avant la livraison du système

Que teste-t-on?

L’exactitude des sorties du système (output)

|20:24| : Les maquettes permettent de se valider au niveau des affichage d’interfaces.

Le temps de réponse

|20:27| : C’est souvent les acteurs externes qui font planter le système.

|20:30| : Les DFD peut aider à trouver qui utilise les modules

La capacité de traiter le volume requis des données

|20:31| : nombre d’enregistrement, à quel moment le système écrase? Ex : 2000 ça va 4000 ça va 4100 ça écrase.

La capacité de recouvrement après un arrêt de fonctionnement

|20:32| : Avez-vous des logiciels en tète commerciaux qui ont des mécanismes de recouvrement. Ex : Excel Word.

La sécurité des données

|20:34| : par exemple, une des conditions du système fallait que sa passe par un audit de sécurité. Qui sont testé par des hackers.

La facilité d’utilisation de la documentation des manuels d’utilisation

|20:35| : Test d’utilisation, ex mettre un expert devant le logiciel et demander de l’utiliser sans aide.

Qu’est-ce que les tests permettent d’évaluer?

Les erreurs et leur impact

La conformité aux besoins

|20:42| : les besoins du TP1 sont couvert dans le TP2.

Les performances

|20:43| : performant selon les scénario utilisateurs externe

Indication de la qualité

|20:43| : révisé par un auditeur externe.

Qui test les systèmes?

Le développeur

Un testeur indépendant

Qu’elles sont les 2 types de tests que l’on veut faire durant la réalisation technique?

Le Test Exhaustif : On teste toutes les possibilités

Les Tests Sélectif : On choisit un chemin que l’on veut tester, une partie du programme.

Qu’elle est l’objectif de la conception des cas de test?

Élaborer divers ensembles de données (cas de test) pour découvrir le maximum d’erreurs …

… avec le minimum d’effort et de temps

… tout en essayant d’assurer la complétude de la recherche

Qu’elles sont les 2 méthodes / stratégie de tests vu dans ce module?

Test de boite blanche

Test de boite noire

Qu’elle est l’objectif des tests de boite blanche?

Garantir que toutes instructions et les conditions ont été exécutées au moins une fois.

Qu’est-ce qu’un test de boite blanche?

Considère l’intérieur du logiciel comme bien spécifié; vérifie que tous les chemins indépendants à l’intérieur d’un module, toutes les sorties de conditions et toutes les boucles ont été testés au moins une fois;

Utilise la structure de contrôle d’une spécification procédurale/modulaire (architecture du logiciel) pour dériver les cas de test

Qu’est-ce que la complexité cyclomatique?

??????

Qu’elle est l’objectif des tests de boite noire?

vérifier les spécifications fonctionnelles des modules du logiciel

Déterminer les données en entrée

Valeur spécifique, intervalle, collection, booléen

Tester des valeurs intermédiaires

Tester toutes les valeurs limites

La limite, juste en-dessous et au-dessus

Tester des valeurs exclues du domaine

Qu’est-ce qu’un test de boite noire?

Considère que l’output est bien spécifié et vérifie que l’implantation (le logiciel développé) réalise correctement tous les aspects de la spécification

Essaye d’identifier des fonctions incorrectes ou absentes, des erreurs d’interface, des erreurs de comportements ou de performance, des erreurs d’initialisation ou de terminaison

Qu’elles sont les 4 niveaux de tests?

(PROF : VOUS DEVEZ ABSOLUMENT CONNAITRE ÇA EN TANT QUE PROGRAMMEUR)

Test unitaire : Test un module technique de boite blanche et de boite noire, exemple de critère : 80% des instructions ont été testés

Test d’intégrations : Conception du SI, technique de boite noire, approche descendante, ascendante mixtes, remplacer les programmes appellant (moniteurs) et appelés (simulateurs)

Test de validation : test de performance, tests de robustesses (volume de données, mémoire, nombre de terminaux, / client) tests de dégradation, tests de sécurité.

4Test d’acceptation : vérification par l’usager, acceptation totale, conditionnel ou refus, tests alpha et beta.