DEVOIR SFFL

HAROUNI DJIHANE ***171731091986***
M1 SSI

EXERCICE 5 :

1-

Fautes:

1. **le calculateur est repris tel quel sur Ariane 5, sans aucune reprise des spécifications, ni tests complémentaires (les données d'entrées d'ARIANE 5 sont totalement différentes de celles d'ARIANE 4**

   Classification:

   Par nature: **accidentelle sans volonté de nuire.**
   Par le moment de leur création :**de développement (spécification et conception).**
   Par leur rapport avec le produit **:interne.**
   Par rapport à leur persistance :**permanente**.

2. **Une valeur de plus de 16 bits est affectée à la variable codée sur 16 bits.**

   Classification:

   Par nature: **accidentelle sans volonté de nuire.**
   Par le moment de leur création :**de développement (spécification et conception).**
   Par leur rapport avec le produit **:interne.**
   Par rapport à leur persistance :**permanente**.

3. **Aucun mécanisme de protection n'est prévu.**

   Classification:

   Par nature: **intentionnelle sans volonté de nuire.**
   Par le moment de leur création :**de développement (spécification et conception).**
   Par leur rapport avec le produit **:interne.**
   Par rapport à leur persistance :**temporaire**.

Defaillances:

1. **La destruction du lanceur**
   Classification:
   Suivant la cohérence : **cohérente.**
   Suivant la nature de la perturbation:**par arrêt**.
   Suivant la durée de la défaillance :**durable.**
   Suivant la gravité des conséquences de la défaillance : **catastrophique.**
   Suivant la fréquence de l'occurrence des défaillances: **probable.**

2. **La perte de trajectoire par conséquent le mécanisme d'autodestruction**
   Classification:
   Suivant la cohérence : **cohérente .**
   Suivant la nature de la perturbation:**par valeur.**
   Suivant la durée de la défaillance :**durable.**
   Suivant la gravité des conséquences de la défaillance: **critique.**
   Suivant la fréquence de l'occurrence des défaillances: **probable.**
3. **Absence d'un système de secours efficace (puisque le 2eme calculateur fonctionne de la même manière que le 1er**
   Classification:
   Suivant la cohérence :**cohérente**.
   Suivant la nature de la perturbation:**par arrêt**.
   Suivant la durée de la défaillance :**durable.**
   Suivant la gravité des conséquences de la défaillance: **indicative.**
   Suivant la fréquence de l'occurrence des défaillances: **rare.**
4. **L'exception est transmise au coeur du contrôle de vol**
   Classification:
   Suivant la cohérence :**cohérente**.
   Suivant la nature de la perturbation: **par valeur.**
   Suivant la durée de la défaillance :**durable.**
   Suivant la gravité des conséquences de la défaillance: **critique.**
   Suivant la fréquence de l'occurrence des défaillances: **rare.**

2- Les leçons qu'on peut tirer de ce cas :

- Si le logiciel peut induire des défaillances à grand risque , les tests doivent prendre une très grande importance et doit être testé sous toutes les conditions possibles.
- Si on compte inclure un système de secours , il doit être complémentaire au premier et non identique puisque cela ne changera rien si le 1er a échoué le 2eme fournira forcément les mêmes résultats .
- Donner l'importance et le temps nécessaire pour une bonne conception (la conception est une phase décisive dans le développement et qui permet d'éviter beaucoup de fautes et erreurs si elle est bien faite).
- La réalisation d'un bon cahier de charge et une bonne étude des langages à utiliser dans le développement du logiciel est nécessaire.
- Traiter les spécifications de chaque logiciel à part, ne pas reprendre des logiciels qui se ressemblent un peu avec des données différentes.

3- Les modèles les mieux adaptés pour le developpement des systemes industriel qui sont souvent complexe et demande beaucoup de temps

1. **Le modèle de développement avec prototypage** : permettant ainsi une meilleure interaction entre les phases de conception et díutilisation. Un prototype est une implantation partielle, physique et logique du produit en présentant toutes les interfaces externes. Les utilisateurs potentiels peuvent alors fournir un feedback ‡ l'équipe responsable du développement avant que commence le développement de la version finale du produit. Par cette approche, les utilisateurs et les concepteurs peuvent mutuellement se clarifier les réquisitions et les interprétations et assurer ainsi une bonne Évolution du projet.

2. **Le modèle de développement en spirale** : s'appuyant substantiellement sur le prototypage et la gestion des risques, le modèle de la spirale

(développé par Boehm, 1988) part du principe que le développement d'applications représente un **cycle itératif**, qui doit être répété jusqu'à ce que le but fixé soit atteint. Par une analyse régulière des risques et des contrôles réguliers du produit intermédiaire, le modèle en spirale diminue considérablement le risque d'échec lors des projets logiciels de grande taille.

3. **Le modèle de développement itératif :**le processus de développement est appliqué plusieurs fois,chaque itération augmente la quantité d'information.En gros c'est une amélioration du modèle en cascade.
   Avantages :
   -Gestion des risques importants lors des premières itérations.
   -Feed-back régulier des utilisateurs, développeurs et tests.
   -Complexité mieux gérée.
   -Exploitation des erreurs des itérations précédentes.

Remarque: On peut toujours combiner ces modèles ensemble et les adapter à notre projet , on a même la possibilité de les combiner avec le modèle en cascade ou modèle en v . Le but principal est d'éliminer ou réduire au maximum les défaillances au niveau de gravité dangereux et catastrophique.