

## G31 – Méthodologie de la production d'applications

### 1. Ingénierie de tests

#### Question 1 :

Le schéma présenté est une matrice fonctionnelle. Elle permet de « quantifier » de manière impérieuse l'importance d'une fonctionnalité en fonction de son importance aux yeux de l'utilisateur final du logiciel, ainsi que sa fréquence d'utilisation.

Une fonction avec une importance stratégique, pour reprendre les termes inscrits sur le schéma, avec une fréquence d'utilisation forte, aura une importance la plus élevée.

Les valeurs entrées dans la matrice sont choisies par le concepteur de celle-ci. Ici des valeurs faibles signifient une importance élevée. C'est au concepteur de choisir de manière réfléchie l'importance et les valeurs données à chaque croisement de la matrice.

#### Question 2 :

Le sujet de notre projet ne contenait aucune exigence de test. Nous avons donc « créé » notre propre panel de tests :

- Des tests fonctionnels afin de contrôler le bon fonctionnement des actions possibles par l'utilisateur,
- Des tests de performances, nous étions peu regardant sur ceux-ci, notre projet étant basé sur plateforme web, tant que l'on réussissait à accéder au site et naviguer sans trop de latence, le résultat nous semblait bon. Mais aussi que le serveur web tenait la charge face à de « nombreux » accès concurrents.

## 2. Les bonnes pratiques ITIL

### Question 3 :

Ce schéma est un diagramme de gestion de problèmes. Il indique l'ordre dans lequel le « niveau » des dysfonctionnements doit évoluer.

Lorsqu'un problème survient, un problème étant un dysfonctionnement dont on ne connaît pas la cause, celui-ci doit être investigué pour le transformer en erreur connu lorsque la cause est découverte. Mais il faut aussi opérer des changements afin d'y remédier et espérer ne plus avoir à rencontrer ce dysfonctionnement. Chaque incident, problème, erreur, est signalé par le biais d'un ticket ou d'un appel de la part de l'utilisateur qui remarque celui-ci.

### Question 4 :

Ce schéma est un diagramme de disponibilité qui représente la disponibilité / l'indisponibilité lors d'un incident d'une fonction.

Aucune valeur n'est entrée ici car ce diagramme ne représente que les étapes successives entre l'apparition d'un incident, sa résolution, et le temps moyen entre deux défaillances (la durée de disponibilité comme indiqué).

Pour connaître le coût de temps de chaque étape, opération, il faut, comme indiqué dans le cours « faire des mesures, monitorer et optimiser continuellement », ce qui permet aussi d'améliorer le rapport entre durée de disponibilité et durée de non-disponibilité.

### 3. La qualité du logiciel

#### Question 5 :

*SonarQube* est un outil intéressant et performant qui permet d'évaluer notre code et de suivre de bonnes pratiques dans l'écriture de code. Grâce à lui nous avons pu améliorer notre code en suivant ses recommandations.

Il faut aussi noter que nous avons rencontré des problèmes à causes de *quality gates* différentes selon les analyses et les langages utilisées : notre code avait passé les premiers tests, mais n'a plus passé les suivants (code sans modifications hormis celles proposées par *SonarQube*), problème résolu par un changement des paramètres des *quality gates*.

- Extrait en parti de la conclusion de notre étude sur la qualité du code.

#### Question 6 :

Dans cette citation de Bill Gates, celui-ci compare le développement d'un logiciel à la construction d'un avion. Dans sa comparaison, il confronte la mesure de la progression d'un développement logiciel par le biais de l'étude de ses lignes de codes, à celle de la construction d'un avion par la mesure de son poids.

Ce qui dans le cas d'un avion n'a aucun sens, ou du moins que dans de rares exceptions. La progression de la construction d'un avion se mesurerait plutôt par l'avancement de l'assemblage du fuselage, l'installation de l'électrique / électronique, de l'équipement intérieur. Mais en aucun cas son poids !

Bill Gates trouve donc, dans sa comparaison, la mesure de la progression du développement d'un logiciel par l'étude de ses lignes de codes, que cette méthode est peu utile et censée.