

Démo 5

Professeur : Michalis Famelis

GQM

Démonstrateur : Erick Raelijohn

1. Dire si les mesures suivantes sont internes ou externes :

R :

- | | |
|--------------------------------|---------|
| • Taille du programme | interne |
| • Productivité | externe |
| • Rentabilité | externe |
| • Taille de la spécification | interne |
| • Effort dans la conception | externe |
| • Durée de la conception | interne |
| • Utilisabilité d'un programme | externe |
| • Couplage | interne |

2.

Établissez un plan GQM (un seul objectif) pour la compagnie X en vous basant sur la fiabilité selon la norme ISO 9126.

- **Maturité** : fréquence des défaillances dues à des défauts du logiciel
- **Tolérance aux fautes** : aptitude à maintenir un niveau de service donné en cas de défaut du logiciel ou de violation de son interface
- **Possibilité de récupération** : capacité à rétablir son niveau de service et de restaurer les informations directement affectées en cas de défaillance; temps et effort nécessaire pour le faire

-----8<-----

G1 : Augmenter la fiabilité du logiciel

Q1 : Qu'est-ce qu'un module fiable?

Q2 : Y a-t-il un lien entre la taille et la fiabilité?

Q3 : Y a-t-il un lien entre le couplage et la fiabilité?

Q4 : Quel est l'effort relié à la fiabilité dans la construction du logiciel?

M1 : nb lignes de codes

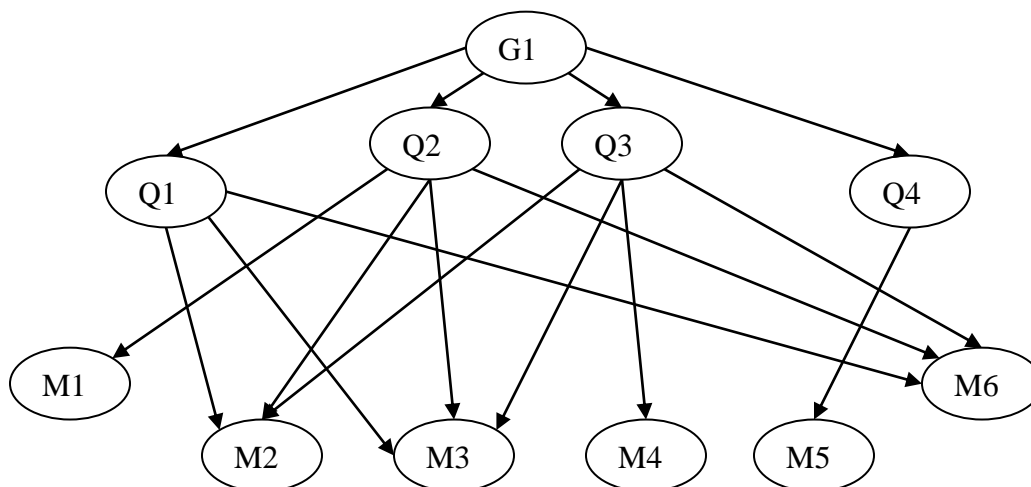
M2 : temps entre deux pannes

M3 : temps de test

M4 : mesure de couplage

M5 : temps consacré à la fiabilité

M6 : Nombre d'erreurs



3.

Utilisez le modèle de définition des objectifs pour formuler chacun des scenarios suivants :

1. On veut comprendre l'impact de l'augmentation du nombre de tests sur la maintenance du projet Ruby On Rails.
2. On veut rendre une page web plus lisible.
3. On ne sait pas si le paquet Debian texlive-latex-extra est prêt pour remise.
4. On désire assurer que le logiciel NLPX92 est capable de traiter de fichier du taille 2Toctets.
5. Notre logiciel doit être utilisable avec une plage braille (« refreshable Braille display »).

-----8<-----

4. Pour chaque scenario de la question précédente élicitez un plan GQM.

1. On veut comprendre l'impact de l'augmentation du nombre de tests sur la maintenance du projet Ruby On Rails.
 - a. How well is the behaviour of the existing code tested
 - i. Metric: McCabe
 - ii. Metric: number of tests per module
 - b. What is the testing effort
 - i. Metric: time per test
 - ii. Metric: comment density of test
2. On veut rendre une page web plus lisible.
 - a. Quelle est la lisibilité actuelle?
 - b. Quelle est la lisibilité requise?
 - c. Quel est l'effort d'augmenter la lisibilité
3. On ne sait pas si le paquet Debian texlive-latex-extra est prêt pour remise.
 - a. Quelle est la fiabilité actuelle?
 - b. Quelle est la stabilité requise?
 - c. Quelle est l'échelle de temps?
4. On désire assurer que le logiciel NLPX92 est capable de traiter de fichier du taille 2Toctets.
 - a. What is the available processing power (puissance de traitement)
 - b. How efficient is the used algorithm
 - i. Algorithmic complexity
 - ii. Cyclomatic complexity
 - c. What is the complexity of the input (2TB might just be XML, and when it become binary –i.e., compressible, then the processing is easier)
 - i. Type of file
 - ii. Compressibility
5. Notre logiciel doit être accessible.
 - a. How accessible is it already?
 - b. How easy is it to add support for new hardware
 - i. Metrics about maintainability (coupling)
 - c. What are available resources (because some accessibility devices might require additional computation –eg speech synthesis)

5.

La compagnie DEVLOG veut mettre sur pied un programme de mesure pour évaluer le rendement des logiciels qu'elle produit. En tant que responsable du service qualité, proposer un tel programme en utilisant la technique GQM et en vous basant sur les définitions du rendement de la norme ISO9126.

G1 : Améliorer le rendement

Q1 : Quel est le lien entre la mémoire utilisée et la taille du programme.

Q2 : Quel est le lien entre le temps d'exécution et la complexité d'un programme.

Q3 : Est-ce que la vitesse d'exécution est influencée par le couplage d'un programme.

Q4 : Est-ce que la mémoire utilisée par un programme est influencée par son couplage.

Q5 : Qu'est-ce qu'un programme efficace.

M1 : Mémoire maximale utilisée. (Q1,4,5)

M2 : Mémoire moyenne utilisée.

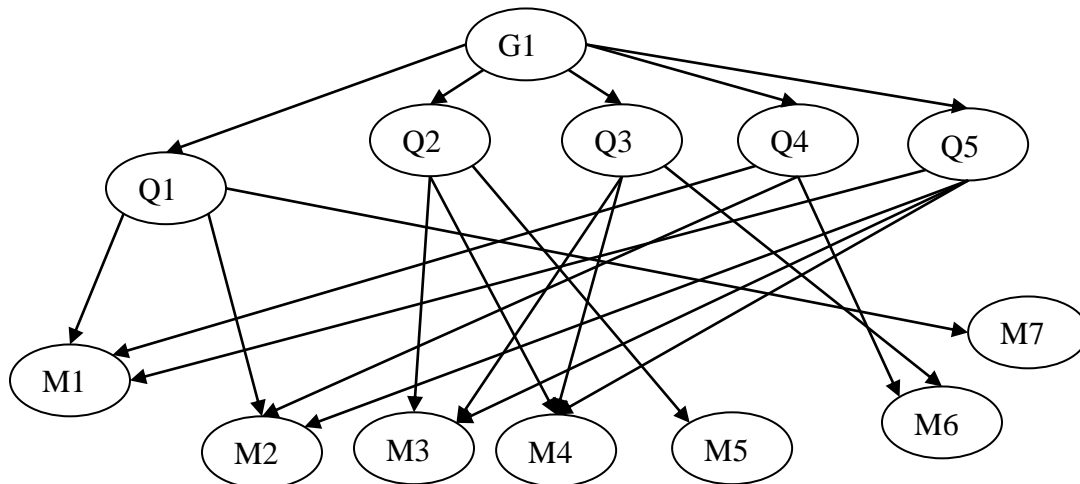
M3 : Temps total de l'exécution.

M4 : taille de l'entrée.

M5 : Complexité du programme (McCabe)

M6 : Couplage du programme (CBO)

M7 : taille du programme en LOC



6. Étude du cas Motorola

Buts :

G1 = Amélioration de la planification du projet

G2 = Augmentation de la détection des erreurs

G3 = Augmenter la fiabilité

G4 = Diminution de la densité des défauts

G5 = Amélioration du service à la clientèle

G6 = réduire le coût de la non-conformité

G7 = Augmenter la productivité logiciel

Définitions :

Erreur : problème détecté lors de la révision de la phase où il a été introduit.

Défaut : problème détecté après la révision de la phase où il a été introduit.

Faute : Erreur ou défaut.

Goal 1 : Amélioration de la planification du projet

Question 1.1 : Quelle est l'exactitude de l'estimation des échéanciers ?

Métrique 1.1 : Durée réelle du projet / Durée estimée du projet

Question 1.2 : Quelle est l'exactitude de l'estimation de l'effort ?

Métrique 1.2 : Effort réel du projet / effort estimé

Goal 2 : Augmentation de la détection des erreurs

Question 2.1 : Quelle est l'efficacité de la détection des fautes avant le déploiement ?

Métrique 2.1 : Nombre de défaut pré-déploiement / Nombre de défauts total

Question 2.2 : Quelle est l'efficacité de détection d'erreur à chaque phase du développement.

Métrique 2.2 : Nombre d'erreurs dans la phase i / nombre d'erreurs+défauts de la phase i

Goal 3 : Augmenter la fiabilité

Question 3.1 : Quel est le taux de défaillance et commence il change en fonction du temps.

Métrique 3.1 : Nb de défaillances / Temps d'exécution

Goal 4 : Diminution de la densité des défauts

Question 4.1 : Quel est le nombre de fautes et comment il se compare au nombre de défaut

Métrique 4.1a : nombre de fautes causées par l'ajout de code / taille du code ajouté

Métrique 4.1b : nombre de défauts causés par l'ajout de code / taille du code ajouté

Question 4.2 : Quel est la quantité de défauts contenus dans un programme livré aux clients par rapport à sa taille ?

Métrique 4.2a : Nombre de défauts livrés / taille du programme en Assembleur

Métrique 4.2b : Nb de défauts livrés causés par l'ajout de code/taille du prog en Asm

Question 4.3 : Quelle est la quantité de défauts trouvés par les clients en fonction de la taille du programme ?

Métrique 4.3a : nb défauts trouvés par les clients / taille du programme en Assembleur

Métrique 4.3b nb défauts trouvés par les clients et qui sont causés par l'ajout de code

Taille du programme en Assembleur

Goal 5 : Amélioration du service à la clientèle

Question 5.1 : Quel est le nombre de problèmes nouvellement ouverts durant le mois ?

Métrique 5.1 : nb total des nouveaux problèmes post-déploiement ouvert durant le mois

Question 5.2 : Quel est le nombre de problèmes ouverts à la fin du mois ?

Métrique 5.2 : nb total des problèmes qui sont restés ouverts à la fin du mois.

Question 5.3 : Quel est l'âge moyen des problèmes ouverts à la fin du mois ?

Métrique 5.3 : temps total entre maintenant et l'ouverture des problèmes restés ouverts

Nb de problèmes encore ouverts à la fin du mois

Question 5.4 : Quel est l'âge moyen des problèmes qui ont été fermés durant le mois ?

Métrique 5.4 : Total des âges des problèmes fermés durant le mois

Total des problèmes fermés durant le mois

Goal 6 : Réduire le coût de la non-conformité

Question 6.1 : Quel est le coût relié à l'enrayement des problèmes post-déploiement au cours du mois ?

Métrique 6.1 : Coût en dollars associé à l'enrayement des problèmes post-déploiement au cours du mois ?

Goal 7 : Augmenter la productivité logicielle

Question 7.1 : Quelle a été la productivité du développement de projet basé sur la taille du programme ?

Métrique 7.1a : taille du programme en Assembleur / effort de développement

Métrique 7.1b : taille de l'ajout de code / effort de développement