niversité Mohammed V Souissi
:ole Nationale Supérieure d'Informatique
d'Analyse des Systèmes – ENSIAS-

Année universitaire 2008-2009
Mr A. IDRI

Examen Economie du Génie Logiciel
Durée :1h
Documents non autorisés

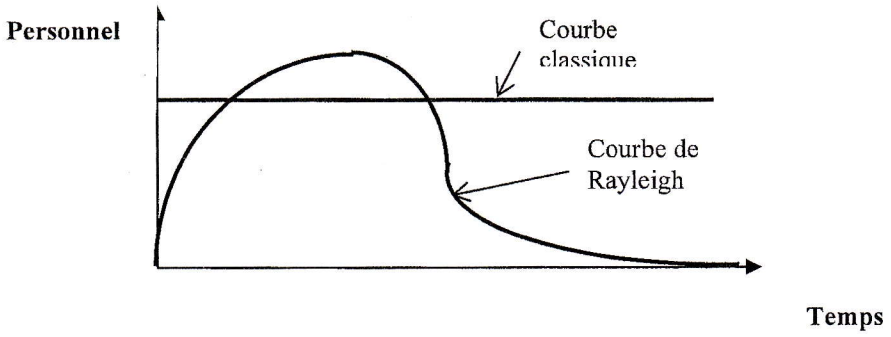
1- Expliquer le processus de mise au point d'un modèle d'estimation du coût adoptant un arbre de décision ID3. Citer les avantages et les inconvénients de cette approche.

2- Expliquer l'effet d'une compression ou d'un allongement du temps de développement d'un projet logiciel dans le modèle COCOMO intermédiaire

3- Discuter l'effet du code réutilisé sur l'évaluation de l'effort de développement d'un logiciel par le modèle COCOMO. Illustrer votre réponse par un exemple.

4- La mesure de McCabe V(G) permet de mesurer la complexité structurelle d'un code. Donner la définition et la signification du nombre V(G) ? Le nombre V(G) est-il suffisant pour représenter la complexité d'un code ? Justifier

5- Le modèle de Putnam adopte la courbe de Rayleigh pour la répartition de l'effort total sur tout le cycle de développement d'un logiciel. Expliquer et citer les avantages de cette approche sur celle adoptant une courbe classique (voir figure ci-dessous).



Eléments de réponse examen EGL 2009

1- Expliquer le processus de mise au point d'un modèle d'estimation du co0t adoptant un arbre de décision ID3. Citer les avantages et les inconvénients de cette approche

Le processus consiste en un ensemble d'étapes :

- Détermination de la population de projets qui seront utilisés dans la construction de l'arbre avec les attributs pris en compte
- Construire l'arbre de décision en Appliquant l'algorithme ID3 qui consiste à choisir à chaque itération l'attribut qui divise le mieux la population en entrée (autrement dit qui donne lieu à des classes de projets homogènes), en construire un nœud et répéter l'algorithme récursivement sur les classes résultantes jusqu'à arriver aux feuilles de l'arbre qui représenteront les estimations de coûts des projets logiciels.
- Introduire le projet sujet à l'estimation du coût par la racine de l'arbre. Selon les valeurs d'attributs de ce projet il va parcourir l'arbre avant d'aboutir à une feuille qui donnera une estimation du coût de ce projet.

Les avantages et inconvénients de cette démarche sont les suivants :

Avantages	Inconvénients
• méthode est non paramétrique ; elle ne postule aucune hypothèse a priori sur la distribution des données • résistante aux données atypiques ; le modèle de prédiction est non linéaire. • Lorsque la base d'apprentissage est de taille importante, elle présente des propriétés similaires aux algorithmes des plus proches voisins	• incapacité, avec les algorithmes classiques (C4.5, CART, CHAID, etc.), à détecter les combinaisons de variables ; ceci est dû au principe de construction pas à pas de l'arbre, entraînant une certaine « myopie ». • nécessité de disposer d'un échantillon d'apprentissage de grande taille. • Instabilité de l'arbre

2- Expliquer l'effet d'une compression ou d'un allongement du temps de développement d'un projet logiciel dans le modèle COCOMO intermédiaire

Rappel (Attribut Projet) : Required Development Schedule (SCED)/ Délai de développement requis

Cette note mesure la contrainte liée au calendrier imposé à l'équipe du projet de développement logiciel. Les notations sont définies en termes de pourcentage de décélération ou d'accélération du temps par rapport au calendrier nominal du projet, représentant une quantité donnée d'effort. Calendriers accélérés ont tendance à produire:

- plus d'efforts dans les premières phases pour éliminer les risques et affiner l'architecture
- plus d'efforts dans les phases ultérieures pour accomplir davantage de tests et de documents en parallèle.

Dans le tableau suivant, la compression du temps de 75% est jugée très faible. Un étirement du délai 160% est classé très haut. **Etirer ou compresser revient à ajouter ou diminuer l'effort**. Ceci est épargné pour une équipe plus petite, pour laquelle le délai est généralement équilibré par le besoin de mener les fonctions administratives projet sur une plus longue période de temps. La nature de cet équilibre est en cours d'étude. SCED est le seul facteur coût utilisé pour décrire l'effet de la compression/expansion du délai pour l'ensemble du projet.

Facteur	Très bas	Bas	Normal	Haut	Très haut	Extrêmement haut
SCED	1,23	1,08	1.00	1.04	1.1	

3- Discuter l'effet du code réutilisé sur 1'évaluation de l'effort de développement d'un logiciel par le modèle COCOMO. Illustrer votre réponse par un exemple.

Le code qui est repris et intégré dans le nouveau développement contribue également à la charge. Le code préexistant, qui est traité comme une boîte noire, est appelé code réutilisé. Le code préexistant qui est traité comme une boîte blanche et est modifié pour être réutilisé, est appelé code adapté. La taille effective du code réutilisé et adapté est ajustée pour obtenir son équivalence en nouveau code. Le code ajusté est

appelé lignes de code source équivalente (ESLOC). L'ajustement est basé sur l'effort supplémentaire qu'il faut pour modifier le code et l'inclure dans le produit.

Le modèle traite le dimensionnement de la réutilisation (à partir des points de fonction ou des lignes de source de Code), de la même façon dans le modèle Early Design et le modèle Post-Architecture.

La décision de réutilisation dépend de l'indicateur AAF.

Le traitement de COCOMO de la réutilisation du logiciel utilise un modèle d'estimation non linéaire, (Voir équations suivantes). Cela implique l'estimation de la quantité de logiciels à être adapté et les trois facteurs de degrés de modification : le pourcentage de modification de la conception (DM), le pourcentage de code modifié(CM), et le pourcentage de l'effort d'intégration requis pour intégrer le logiciel adapté ou réutilisé (GI).

AAF = (0,4 x DM) + (0,3 x CM) + (0,3 x IM)

En effet, le code réutilisable est très souvent modifié, même lorsque celui-ci devait –être une boite noire. Or les petites modifications dans le produit réutilisé génèrent des coûts importants et disproportionnés. C'est principalement à cause de deux facteurs: le coût de la compréhension du logiciel devant être modifié, et le coût relatif de la vérification des interfaces du module. (Extrait du document modèle COCOMO.pdf, dans le dossier de préparation)

4- **La mesure de McCabe V(G) permet de mesurer la complexit6 structurelle d'un code. Donner la définition et la signification du nombre V(G) ? Le nombre V(G) est-il suffisant pour représenter la complexité d'un code ? Justifier**

V(G) d'un code source est le nombre cyclomatique du graphe de flux de contrôle correspondant au code. V(G)= Nb arcs – Nb nœuds + 1

la complexité cyclomatique n'est pas suffisante à renseigner sur la complexité du code. En effet, le nombre de lignes de code joue aussi un rôle. Par exemple, deux programmes de même complexité cyclomatique mais dont l'un ferait 100 lignes et l'autre 10000 lignes ne pourraient intuitivement pas être considérés de même complexité. (extrait du document complexité des logiciels.pdf, dans le dossier de préparation)

5- **Le modèle de Putnam adopte la courbe de Rayleigh pour la répartition de l'effort total surtout 1e cycle de développement d'un logiciel. Expliquer et**

citer les avantages de cette approche sur celle adoptant une courbe classique (voir figure ci-dessous).

A partir des résultats obtenus ci-dessus et d'observation des projets réels, Putnam établi une relation entre la charge totale, le nombre d'instruction et la durée du développement. L'effort à fournir est en relation inverse à la puissance 4 du temps de développement. Ce qui signifie que par exemple un raccourcissement du temps de développement de 10% suppose une augmentation de la charge totale de 40%. Dans le modèle de Putnam, l'augmentation de la durée diminue la charge tandis que son raccourcissement l'accroît.

Parmi les avantages de cette approche on cite la simplicité avec laquelle il est calibré. La plupart des organismes de logiciel peuvent rassembler facilement taille, effort et durée (temps) pour des projets passés.