

Nom et prénom : KANNE TAMIBE KOCHIAKE

Matricule : 17A0791FS

Introduction

Une métrique est un indicateur d'avancement ou de qualité des développements logiciels. Un logiciel est de qualité s'il a un nombre faible de bogues fonctionnels. La qualité structurelle d'un logiciel est donc la manière avec laquelle ces fonctionnalités sont implémentées et inclut entre autres la robustesse, la maintenabilité, la lisibilité ou encore l'évolutivité. Il existe probablement un lien entre les deux, mais pas nécessairement de causalité. Un logiciel avec un code de très mauvaise qualité pourrait tout à fait être jugé de très haute qualité par les utilisateurs. Ou à l'inverse, le résultat final d'un code d'une qualité extrêmement haute pourrait être jugé par les utilisateurs comme étant de mauvaise qualité. C'est la mesure de la qualité structurelle du code qui nous intéresse dans le présent document.

La mesure de la qualité d'un code peut en partie être mesurée automatiquement par des outils d'analyse statique du code source. Ces outils sont disponibles pour la plupart des langages et proposent de nombreuses métriques et conventions standards dont les principales seront décrites dans le présent document. L'outil d'analyse utilisé dans ce devoir est : Halstead. Ainsi, les résultats de cette analyse sont les suivants :

Halstead Metrics Analysis Report for App.java

File name : App.java

Operators :

Name	Count
Package	1
Import	3
Public	2
Class	1
Static	1
Void	1

Throws	1
.	10
;	5
{}	113
()	509
[]	5

Operands:

Name	Count
Fr	29
Dispo	14
setDispo	8
Create	2
Sleep	7
3000	7
findByldArticle	2
findLastEntryld	2
newFig	15
Taille	5
setTaille	2
Obj	12
typeObj	9
Libelle	3
Collection	4
Auteur	2
Editeur	3
Categorie	3
Desc	6
Update	2
Delete	2
findFicheArticle	1
promptRechercheMultiCriteres	1

Values :

No of Distinct Operators (n1)	652.0
No of Distinct Operands (n2)	2756.0
Total No of Operators (N1)	12.0
Total No of Operands (N2)	177.0
Program length	189.0

Program Vocabulary	3408.0
Estimated Length	37591.93139253169
Purity Ratio	198.89910789699306
Volume	1429.2638181777827
Difficulty	20.936865021770682
Program Effort	29924.30364168883
Programming Time	1662.4613134271574

Copyright © Specksboy Inc. 2014-15

Dans les tableaux ci-dessus, nous voyons :

- ❖ Le nombre d'opérateurs distincts (n1), le nombre d'opérandes distincts (n2) ;
- ❖ Le nombre total d'opérateurs (N1), le nombre total d'opérandes (N2) ;
- ❖ La longueur du code ou programme, la difficulté, le taux d'effort, le temps d'exécution du programme, le volume et bien d'autres.

Conclusion

En conclusion, analyser ce code nous a permis de calculer sa complexité en utilisant les métriques d'Halstead (1977) qui base également ses calculs sur les opérateurs et expressions et de comprendre que faire un contrôle du niveau de complexité cyclomatique en continu pendant le développement permet d'obtenir une meilleure conception du code.

