



جامعة الموصل  
كلية علوم الحاسوب والرياضيات

# بناء أداة تقييس برمجيات جافا الكيانية

مريم حسين مرعي

رسالة ماجستير

البرمجيات

بإشراف

م. د. محمد عبدالغني طه الدباغ

## الخلاصة

إن من أهم الاهداف الرئيسية لهندسة البرمجيات هو تطوير أنظمة برمجيات بجودة عالية، وان هذه الجودة لا تتحقق الا إذا تم بناء الانظمة وفقاً لمعايير ومقاييس معينة، فبدون تطبيق المقاييس لا يمكن تحديد جودة أي نظام. ومع ظهور البرمجيات الكيانية بقيت مشكلة تعدد المقاييس تورق المطورين اذا لا يوجد مقاييس موحدة يمكن تطبيقها على جميع مراحل تطوير البرمجيات الكيانية.

تم في هذه الرسالة بناء أداة (MMT) Measuring Metrics Tool التي تقوم بتحديد امكانية التزام أو عدم التزام المبرمجين بالتصميم المقترح للنظام قيد التطوير وذلك من بمقارنة النتائج التي سيتم حسابها عن طريق تطبيق مقاييس تصميم البرمجيات الكيانية MOOD و MEMOOD و QMOOD، اختصاراً لـ (Metrics for Object Oriented Design ، Maintainability) (Quality Model for Object و Estimation Model for Object Oriented Design) (Oriented Design) على الشيفرات البرمجية للغة جافا لعدد من الحالات الدراسية (5 حالات دراسية) وبمستويات متفاوتة من المهارات البرمجية (8 مستويات من المهارات البرمجية التي توزعت بين خريجي البكالوريوس والماجستير وأدوات التوليد التلقائي للشيفرة)، وكان الغرض من تحديد هذا الالتزام لمعرفة اذا ما كان هنالك حرية للمبرمجين في اجراء بعض التغييرات والاضافات تبعاً لخبراتهم البرمجية.

بعد تطبيق النماذج الثلاثة تبين من النتائج أن هنالك حرية للمبرمجين في التغيير على التصميم بشرط ان لا يكون التغيير جوهرياً يؤدي الى خلل في عمل النظام. ان هذه الحرية المتاحة للمبرمجين حققت نتائج أفضل، فعلى سبيل المثال من نتائج النموذج MOOD ارتفعت نسبة الوصول للحدود القياسية لأحد المبرمجين الى 25% عن حالة الالتزام التام بالتصميم وارتفعت بنسبة 37.5% عن حالة التوليد التلقائي للبرنامج. وبينت النتائج امكانية ان تكون هنالك تغذية راجعية من مرحلة البرمجة الى مرحلة التصميم في حال كانت نتائج مرحلة البرمجة أفضل سواء بالوصول الى الحدود القياسية أو بتجاوز مشكلة او خلل في التصميم. وتبين أيضاً من خلال النتائج تأثر نتائج النموذجين MOOD و QMOOD تأثراً واضحاً بتباين خبرات ومهارات المبرمجين وتبعاً للتغييرات الثانوية التي يتحكمون بها. بينما لم يكن هنالك تبايناً ملحوظاً لنتائج نموذج MEMOOD لتأثره بالتغييرات الهيكلية التي لا يتحكم بها المبرمجون. تبين أيضاً من النتائج ان هنالك علاقة ارتباط وثيقة بين النموذجين MOOD و QMOOD لأن جزء من عوامل نموذج MOOD تدخل في حسابات مقاييس نموذج QMOOD وتؤثر فيه تأثيراً مباشراً بصورة سلبية أو ايجابية. أخيراً فقد تم استعمال لغة جافا في بيئة عمل "Eclipse" لتطوير الأداة المقترحة.

**University of Mosul  
College of Computer Sciences  
And Mathematics**



# **Constructing of Java Object Oriented Software Metrics Tool**

**Mariam Hussein Marie**

**M.Sc./Thesis  
Software**

**Supervised By**

**Dr. Mohammad Abdulghani Taha**

## Abstract

One of the main objectives of software engineering is to develop high-quality software systems. This quality is mainly achieved by adhering certain standards and criteria that govern stability, maintainability and other quality characteristics. Because of that, system's quality cannot be determined without applying standards. Even with the rise of Object-Oriented Programming, the varieties of standards are still considered as problematic for software developers.

In this work, the MOOD, MEMOOD and QMOOD metrics were selected to be applied to Java object-oriented code on a number of case studies with a variety of programming skill levels.

The aim of applying these metrics at the programming phase is to find the adherence of programmers to the design of the system under development, and also to check if they had the freedom to make changes, modifications or additions to system's design

After the three models were applied it was deduced that the programmers possessed enough freedom to make changes to the design. provided that the changes are not substantial and does not lead to a defect or deviation in the system's operations.

The freedom that the programmers enjoyed led to much better results. For example, the results of the MOOD model for one programmer increased the achievement of standard ranges by (25%) than "programmer1" and (37.5%) than the "auto generated" codes. The obtained results set forth that there could be reverse feedback from the programming stage to the design stage if the results of the programming stage were found to be better either in achieving the standard ranges or by bypassing a problem or a defect in the design.

The obtained results also indicate that both the MOOD and QMOOD models are clearly influenced by the points of view, skill levels, and secondary changes that the programmers control. In contrast, the MEMOOD model results were not affected by the structural changes controlled by the programmers.

also, the results clearly show a close correlation between the MOOD and QMOOD models since the some of the former's factors are used in the latter and have a direct positive or negative effect on each other.

Finally, the "Eclipse" integrated development environment for Java was used during the development of the proposed tool.