



وزارة التعليم العالي والبحث العلمي
جامعة الموصل
كلية علوم الحاسوب والرياضيات
قسم البرمجيات

تصميم نموذج فعال لتحسين قابلية الصيانة

رسالة مقدمة

الى مجلس كلية علوم الحاسوب والرياضيات في جامعة الموصل
كجزء من متطلبات نيل شهادة ماجستير علوم في
البرمجيات

من قبل

غسان خليل ابراهيم الخاتوني

بإشراف

أ. د. شهباء ابراهيم خليل الطائي

المستخلص

تعد جودة البرمجيات أمراً ذو أهمية كبيرة في مجال هندسة البرمجيات، وذلك نظراً لزيادة حجم وتعقيد الأنظمة البرمجية في الوقت الحاضر وازدياد المنافسة بين شركات البرمجيات على استحصال رضى الزبون. ولذلك، يجب على مهندسي البرمجيات بناء أدوات هندسة البرمجيات لتوفير الدعم الضروري في مراحل بناء وتطوير البرمجيات، والتي من ضمنها أدوات قياس البرمجيات (Software Measurement Tools)، حيث توفر هذه الأدوات الكثير من الوقت والجهد اللازم لأجراء عمليات حساب المقاييس، فضلاً عن إعطاء تصور واضح لمهندسي البرمجيات عن الجودة بشكل عام.

في هذه الرسالة تم بناء وتصميم نموذج فعال لتحسين قابلية الصيانة للبرمجيات (Efficient Model For Maintainability EMFM)، والذي يتم بواسطته تحليل الشفرة البرمجية للبرمجيات المكتوبة بلغة بايثون او جافا وفق اسلوب التحليل الثابت Static Analysis لاستخراج مجموعة من المقاييس وحساب مقاييس مؤشر قابلية الصيانة Maintainability Index MI على مستوى الاصناف classes بشكل تفصيلي وعلى مستوى النظام ككل. حيث تم قياس قابلية الصيانة للبرمجيات التي تحتوي على شفرات مستنسخة code clone التي تؤدي الى زيادة تكلفة صيانة البرامج وانتشار الأخطاء بشكل أكبر. لمعالجة الاضرار الناتجة عن الشفرات المستنسخة تم اضافة منهجية الى النموذج المقترح للكشف عنها وازالتها او التقليل من الاضرار الناتجة عنها من خلال استخدام تقنيات اعادة البناء Refactoring.

تم تنفيذ النموذج EMFM واختباره على مجموعة من الملفات (5 Case Study) التي تم الحصول عليها من GitHub، حققت هذه الملفات بنسخ من الشفرات المستنسخة، إذ استطاع النموذج الكشف عنها ومن ثم اعادة بناءها وحساب مقاييس مؤشر قابلية الصيانة، حيث اظهرت النتائج ان القيم لمقاييس مؤشر قابلية الصيانة قد زادت مع تحسن في قابلية القراءة والفهم للشفرات البرمجية والذي يؤدي الى تقليل من تكلفة الصيانة والى زيادة جودة البرامج، ويوفر النموذج نظرة سريعة لمهندسي البرمجيات عن تكلفة صيانة البرامج. تم برمجة النموذج باستخدام لغة python تحت بيئة Microsoft Windows 10، وقد تم اختبار النموذج على عدة برمجيات (5 Case Study) مكتوبة بلغة بايثون أيضاً واثبت الاختبار سهولة الاستخدام ودقة النتائج في حساب جميع المقاييس المستخدمة في العمل والمتضمنة مقاييس عدد الاسطر البرمجية ومقاييس هلستيد (LOC Metrics, Halsted Metrics) وتم حساب مؤشر قابلية الصيانة Maintainability Index، كما وأجريت مناقشة النتائج والاحصاءات التي تم توصل اليها.

**Ministry of Higher Education and
Scientific Research
University of Mosul
College of Computer Science and
Mathematics
Department of Software**



Designing an Efficient Model For Improve Maintainability

**A Thesis Submitted to the Council of the College of
Computer Science and Mathematics
University of Mosul
as a Partial Fulfillment of Requirements
for the Degree of Master of Science
in Software**

**By
Ghassan Khaleel Ibraheem Al Khatouny**

**Supervised by
Prof. Dr. Shahbaa Ibraheem Khaleel**

2023 A.D.

1445 A.H.

Abstract

Software quality is of great importance in the field of software engineering, due to the increasing size and complexity of software systems nowadays and the increasing competition between software companies for customer satisfaction. Therefore, software engineers must build software engineering tools to provide the necessary support in the software construction and development stages, including Software Measurement Tools, as these tools save a lot of time and effort needed to perform metrics calculations, as well as give a clear visualization. For software engineers about quality in general.

In this thesis, an effective model for improving the maintainability of software (EMFM) was built and designed, through which the code of software written in Python or Java is analyzed according to the Static Analysis method to extract a set of metrics and calculate the Maintainability Index (MI) metrics. At the class level in detail and at the system level as a whole. The maintainability of software that contains clone code was measured, which leads to an increase in the cost of software maintenance and a greater spread of errors. To address the damage resulting from cloned codes, a methodology has been added to the proposed model to detect them and remove them or reduce the damage resulting from them through the use of refactoring techniques.

The EMFM model was implemented and tested on a set of files (5 Case Studies) obtained from GitHub. These files were injected with copies of the cloned codes. The model was able to detect them and then reconstruct them and calculate maintainability index metrics. The results showed that the values for the metrics The maintainability index has increased with an improvement in the readability and understanding of software codes, which leads to reducing the cost of maintenance and increasing the quality of the software. The model provides software engineers with a quick overview of the cost of software maintenance. The model was programmed using the Python language under the Microsoft Windows 10 environment. The model was tested on several programs (5 Case Studies) also written in Python. The test proved the ease of use and the accuracy of the results in calculating all the metrics used in the work, including (LOC Metrics, Halsted Metrics,) The Maintainability Index was calculated, and the results and statistics obtained were discussed.