



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

# تقييم جودة التصميم البرمجي باستخدام تعلم الآلة: أداة تصنيف ذكية لمخططات الاصناف

رسالة مقدمة

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

من قبل

عمر رعد يونس سعيد

بإشراف

أ.م.د. أشرف عبد المنعم عبد المجيد ذنون

## المستخلص

تمثل مرحلة التصميم في دورة حياة تطوير البرمجيات الركيزة الأساسية التي يُبنى عليها نجاح النظام واستقراره وقابليته للصيانة والتطوير المستقبلي. غير أنّ غياب أدوات تقييم آلية دقيقة لمخططات UML Class Diagrams) تؤدي إلى تراكم العيوب البنوية واكتشافها في مراحل متأخرة، مما يرفع تكلفة الإصلاح ويقلل وثوقية النظام. ومع تزايد تعقيد الأنظمة البرمجية الحديثة، تصبح الحاجة ملحة إلى إطار عملي قادر على قياس جودة التصميم بشكل موضوعي، وإصدار مؤشرات كمية دقيقة تدعم فرق التطوير في اتخاذ قرارات تقنية مبكرة وفعّالة.

جاءت هذه الدراسة لمعالجة هذه الفجوة من خلال تطوير أداة CDQC Tool، وهي أداة ذكية لتحليل مخططات الفئة وتقييمها آلياً وفق مقاييس كمية ومعايير تصميمية واضحة. يعتمد النهج المقترح على استخراج جميع العناصر البنوية من فئات وسمات وعمليات وعلاقات من ملفات XML بعد ذلك تُعاد صياغة البيانات المستخرجة لتوليد متجه خصائص شامل يُمثل كل صنف، ثم يُعدّى إلى مجموعة من نماذج التعلم الآلي الخاضع للإشراف تشمل الانحدار وجسّي، شجرة القرار، الغابة العشوائية، آلة المتجهات الدائمة، وخوارزمية الجار الأقرب، إضافةً إلى تقنيتي التصويت الصلب والتصويت الناعم لتعزيز دقة التصنيف وتحسين موثوقية النتائج.

أُجري التقييم التجريبي على مجموعة بيانات مكوّنة من ٤٢٠ عيّنة ٣٤ سمة تصميمية. وأظهرت النتائج أن نموذج التصويت الصلب حقق دقة ٩٤,٦٤٪ في حين تراوحت دقة النماذج الفردية بين ٨٩,٢٩٪ و٩٢,٨٦٪. والذي حققها التصويت اللين فقد حقق دقة ٩٢,٨٦٪. كما تولد الأداة تقارير تحليلية متقدمة تُبين حالة كل فئة (جيدة/غير جيدة) مع تقديم مؤشرات تفصيلية تتعلق حماية السمات والعمليات ومدى الالتزام بمبدأ الفتح/الإغلاق (OCP)، إلى جانب توصيات عملية قابلة للتنفيذ لتحسين التصميم وتقليل احتمالية ظهور العيوب في المراحل اللاحقة من التطوير.

أثبتت دراسات الحالة على مشاريع مفتوحة المصدر بمستويات مختلفة من تعقيد قدرة الأداة على توفير تقييم دقيق وموضوعي، وإصدار تغذية راجعة فورية تسهل اكتشاف العيوب البنوية مبكراً، مما يقلل الكاليف المترتبة على إصلاحها لاحقاً. كما أبرزت النتائج فاعلية دمج تقنيات التعلم

الآلي مع تحليل البنيوي القائم على XML كمنهجية متكاملة تدعم تحسين جودة التصميم وتمنح فرق التطوير وسيلة موثوقة لاخاذ قرارات مبنية على بيانات كمية دقيقة.

من خلال هذه المساهمة، تقدم الدراسة منظوراً جديداً في مجال هندسة البرمجيات يعتمد على الدمج بين التحليل الآلي للمخططات والتقنيات الذكية، ما يعزز جودة النظم البرمجية، ويرفع قدرتها على الصيانة والتطوير المستقبلي، ويمنح فرق التطوير أداة عملية تسهم في نجاح المشاريع البرمجية على المدى الطويل.

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



# **Software Design Quality Assessment Using Machine Learning: An Intelligent Classification Tool for Class Diagram**

**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  
Omar Raad Younus Saeed**

**Supervised by  
Dr. Ashraf Abdulmunim Abdulmajeed  
Assistance Prof.**

---

**2025 A.D.**

**1447 A.H**

## Abstract

The design phase in the software development life cycle represents the cornerstone upon which the success, stability, maintainability, and future scalability of a system are built. However, the lack of precise automated tools for evaluating UML Class Diagrams often leads to the accumulation of structural flaws, which are typically discovered at later stages, thereby increasing the cost of remediation and reducing system reliability. With the growing complexity of modern software systems, there is an urgent need for a practical framework capable of objectively assessing design quality, providing accurate quantitative indicators to support development teams in making early and effective technical decisions.

This study addresses this gap by developing CDQC Tool, an intelligent system for automated analysis and evaluation of UML class diagrams based on quantitative metrics and well-defined design criteria. The proposed approach extracts all structural elements including classes, attributes, methods, and relationships—from XML files. These extracted data are then processed to generate a comprehensive feature vector representing each class, which is subsequently fed into a suite of supervised machine learning models, including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and k-Nearest Neighbor (KNN), in addition to both hard and soft voting techniques to enhance classification accuracy and improve reliability.

The experimental evaluation was conducted on a dataset comprising 420 samples and 34 design features. Results demonstrated that the hard voting model achieved an accuracy of 94.64%, while the accuracies of individual models ranged between 89.29% and 92.86%, with the soft voting model also achieving 92.86%. Furthermore, the tool generates advanced analytical reports indicating the status of each class (Good/Not Good), along with detailed metrics

on attribute and method protection levels, adherence to the Open/Closed Principle (OCP), and actionable recommendations for design improvement and reducing the likelihood of defects in later development stages.

Case studies on open-source projects with varying levels of complexity confirmed the tool's ability to provide precise, objective evaluations and deliver immediate feedback that facilitates early detection of structural flaws, significantly reducing remediation costs. The results also highlighted the effectiveness of integrating machine learning techniques with XML-based structural analysis as a comprehensive methodology for improving design quality, equipping development teams with a reliable, data-driven decision-making tool. Through this contribution, the study offers a novel perspective in software engineering, emphasizing the synergy between automated diagram analysis and intelligent techniques, thereby enhancing software system quality, increasing maintainability and scalability, and providing development teams with a practical solution that supports the long-term success of software projects.