

Методи інтелектуального аналізу даних в задачах проектування програмного забезпечення

Єсаулов Данило

Інститут прикладного системного аналізу, НТУУ «КПІ», УКРАЇНА, м. Київ,
пр. Перемоги, 37а, E-mail: esauldan@gmail.com

Abstract – In this article was highlighted the problem of assessing the quality and complexity of the software. Also highlighted the main principles and approaches to the computation complexity of software architecture. Main tasks set for a decision support system for refactoring.

Ключові слова – програмне забезпечення, рефакторинг, метрики, система підтримки прийняття рішень, об'єктно-орієнтоване програмування.

Задача створення якісного коду, який легко читається та підтримується розробниками є надзвичайно актуальною на сьогодні. Проте поняття складності коду є досі погано формалізованим і суб'єктивним.

Метою даної роботи є розробка системи підтримки прийняття рішень на основі метрик, яка б дозволяла виявляти складні місця в коді та надавала б практичні рекомендації щодо рефакторингу.

Застосування метрик програмного коду дозволяє розробникам і керівникам проектів оцінювати властивості створюваного або вже існуючого програмного забезпечення, прогнозувати обсяг робіт, давати кількісну характеристику тих чи інших проектних рішень, якість розроблених систем та їх частин, характеризувати складність або надійність програмного забезпечення.

Для визначення переліку необхідних для аналізу коду метрик та їх модифікацій необхідно сформулювати завдання, які з допомогою даних метрик планується вирішувати. В роботі пропонується виділити 3 групи завдань, які планується вирішувати за допомогою метрик:

- Класифікація програмного коду за його складністю
- Оцінка трудомісткості розробки програмного коду
- Профілювання програмного коду з метою автоматичного виявлення потенційно цікавих для аналізу фрагментів коду

Можна виділити наступні підходи до обчислення складності та якості вихідного коду:

- Кількісні метрики. Цей клас метрик зазвичай розглядається в першу чергу, зважаючи на простоту. Традиційно кількісні метрики обчислюються на основі початкового коду програм.

- Функціонально-орієнтовані метрики. Ці метрики побічно вимірюють програмний продукт і процес його розробки, при цьому розглядається не розмір, а функціональність або корисність продукту.

- Об'єктно-орієнтовані метрики. Даний клас використовується при оцінці програм, написаних на об'єктно-орієнтованих мовах програмування.

Задача, яка ставиться перед системою підтримки прийняття рішень в даній роботі, полягає в аналізі вихідного коду програмного продукту, виявленні найбільш слабких місць та формуванні практичних рекомендацій щодо рефакторингу.

Основна мета, яку переслідує рефакторинг є організація програмного коду таким чином, щоб він був готовий до внесення змін та впровадження нового функціоналу при мінімальних затратах. Також рефакторинг допомагає зробити код більш зрозумілим для розробників, що надає можливість передачі проекту іншим командам.

В роботі запропоновано використовувати наступні метрики: WMC (Weighted Methods Per Class), DIT (Depth of Inheritance Tree), NOC (Number of Children), CBO (Coupling between object classes), RFC (Response For a Class), LCOM (Lack of Cohesion in Methods). На основі їх обрахунку СППР робить перші висновки та класифікує програмний код за складністю та якістю. Наступним кроком йде виділення найбільш проблемних місць та надання практичних порад щодо зменшення складності та поліпшення якості коду.

Висновок

Результатом проведеної роботи є СППР, яка синтаксично аналізує код, надає розрахунки метрик вихідного коду та видає список порад щодо рефакторингу.

Література

1. Фаулер М. Рефакторинг. Улучшение существующего кода. / М. Фаулер, К. Бек, Д. Брант, У. Алдайк, Д. Робертс, Е. Гамма // Видавництво «Символ-Плюс», 2008. – 432с.
2. Фаулер М. Чистый код. / М. Фаулер // Видавництво «Пітер», 2010. – 464с.
3. Гамма Е. Приемы объектно-ориентированного проектирования. Паттерны проектирования. / Е. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес // Видавництво «Пітер», 2007. – 366с.
4. Lorenz M. Object-Oriented Software Metrics // M. Lorenz, J. Kidd // «Prentice Hall», 1994. – 146pp.