

doi: 10.32620/oikit.2019.86.14

УДК 681.3, 378.147

Л. Ф. Пудовкіна, В. В. Сіняєв

Застосування емпіричних моделей і метрик Холстеда для оцінювання якості додатків програмного забезпечення

*Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»*

Розглянуто застосування емпіричних моделей і метрик Холстеда для оцінювання якості програмного забезпечення. **Предметом** дослідження є методи вимірювання надійності програмного забезпечення і модель її розрахунку. **Мета** роботи – виконання перспективного напрямку подальших досліджень аналітичних та емпіричних моделей надійності функціонування програмного забезпечення. **Об'єктом** дослідження є процес оцінювання якості програмного забезпечення, що потребує великої кількості завдань. Це призводить до різноманітності підходів, методів і засобів. **Завдання:** здійснити порівняльний аналіз аналітичних та емпіричних моделей надійності функціонування та якості програмного забезпечення; описати моделі й методи для порівняльного аналізу цих моделей надійності функціонування програмного забезпечення; виконати апробацію і оцінювання ефективності використаних моделей і методів для порівняльного аналізу аналітичних та емпіричних моделей надійності функціонування програмного забезпечення. Використовуваними **методами** є розроблене програмне забезпечення, яке за допомогою метрик Холстеда і методів статичного аналізатора програмного коду, дозволяє оцінювати складність та якість програмних продуктів. Це дає можливість комплексно розглянути всі аспекти, пов'язані з аналітичними і емпіричними моделями. В результаті розроблено програмне забезпечення, яке за допомогою метрик Холстеда і методів статичного аналізатора програмного коду дає можливість оцінювати складність та якість програмних продуктів. Програмне забезпечення виконує такі функції: побудова графіків різних параметрів; виведення інформації з графіків у текстовий вигляд (із значеннями отриманого при експерименті). Лексичний аналізатор будує графіки, які відображають наступну інформацію про проаналізовані модулі: коментування; точний та апроксимований рівень якості; реальну та теоретичну довжину; інформативність; витрачені інтелектуальні зусилля. **Висновки.** Актуальність досліджень порівняльного аналізу аналітичних та емпіричних моделей надійності функціонування програмного забезпечення визначається тим фактом, що більшість програмних забезпечень ненадійні. Наукова новизна отриманих результатів полягає в такому: за допомогою порівняльного аналізу аналітичних та емпіричних моделей надійності функціонування програмного забезпечення вивчити детально моделі надійності і підвищити надійність програмного забезпечення.

Ключові слова: надійність програмного забезпечення; математична модель; емпіричне моделювання; функція надійності; інтенсивність відмов; модель даних; метрики Холстеда.

Вступ

В області надійності апаратури досягнуто рівня, коли вже створено ряд математичних методів, що дозволяють інженерам передбачати надійність його продукту. Ці математичні методи широко застосовують у багатьох інженерних областях.

Об'єкт пізнання досліджується емпіричними методами (спостереження, експеримент), встановлені факти є основою для побудови математичної моделі. Отримана система математичних рівнянь може досліджуватися аналітичними методами або за допомогою персонального комп'ютера. Проведено серію обчислювальних експериментів або комп'ютерних імітацій, і отримані результати зіставляють із результатами аналітичного дослідження математичної моделі і ек-

периментальними даними. Висновки враховують для поліпшення методики експериментального вивчення об'єкта дослідження, розвитку математичної моделі та вдосконалення комп'ютерної моделі.

Тому в цій статті буде розглянуто та проаналізовано порівняльний аналіз аналітичних та емпіричних моделей надійності функціонування і якості програмного забезпечення, щоб після порівняльного аналізу з'явилася можливість детально вивчити моделі надійності і підвищити надійність програмного забезпечення.

1 Основні показники надійності програмного забезпечення

Аналітичне моделювання [1] дозволяє проаналізувати тільки досить прості системи. Перевагою аналітичної моделі є те, що рішення можна аналізувати математичними методами. Недоліком аналітичних моделей є спрощення реальних ситуацій з метою отримання аналітичних рішень.

Моделі надійності [2] програм будуються на припущенні про те, що виявлення помилок є випадковою подією і тому має імовірнісний характер. Такі моделі призначені для оцінювання показників надійності програм і програмних комплексів у процесі тестування. Вони дають можливість прийняти обґрунтоване рішення про час проектування налагоджувальних робіт.

При побудові моделей використовують такі характеристики надійності програм.

Функція імовірності безвідмовної роботи $P(t)$ – ймовірність того, що відмова не виникає в межах заданого інтервалу часу від 0 до t , при заданих умовах експлуатації.

Функція імовірності відмови $Q(t)$ – ймовірність того, що протягом часу t відбудеться відмова програми як результат виявлення дії помилки в програмі. Таким чином,

$$Q(t) = (1 - P(t)) * 100\%, \quad (1)$$

де $Q(t)$ – ймовірність того, що протягом часу t відбудеться відмова програми як результат виявлення дії помилок;

$P(t)$ – ймовірність того, що помилки програми не виявляються на інтервалі часу від 0 до t ;

t – час.

Інтенсивність відмов $\lambda(t)$ – умовна щільність ймовірності часу від виникнення відмови програми за умови, що до моменту t відмови не було.

$$\lambda(t) = -\frac{\partial P(t)}{\partial t} / P(t) = \left| \frac{\partial Q(t)}{\partial t} \right| / P(t), \quad (2)$$

де $\lambda(t)$ – інтенсивність відмов;

t – час;

$P(t)$ – ймовірність того, що помилки програми не виявляються на інтервалі часу від 0 до t ;

dt – диференціал функції;

$Q(t)$ – ймовірність того, що протягом часу t відбудеться відмова програми як результат виявлення дії помилок.

2 Ефективність застосування емпіричних моделей даних

Емпірична модель [3] – різновид моделей, основу якої становлять результати аналізу деякого об'єму даних, отриманих у результаті експерименту, вимірювань або прогнозування. Емпіричну і аналітичну моделі можна використовувати для прогнозування оптимальних наборів даних для тестів продуктів будь-якого призначення.

Результатом такого аналізу, як правило, є виведення (створення) нових формул, рівнянь закономірностей, кореляційних залежностей, що описують зв'язок між розглянутими величинами. Результатом також можуть бути деякі масиви даних, що являють собою еталони, з якими надалі будуть порівнюватися отримані в процесі тестування дані. Але для емпіричних моделей необхідні додаткові процедури перевірки здійсненності цих умов.

Емпіричні моделі базуються на аналізі структурних особливостей програмних засобів. Емпіричні моделі часто не дають кінцевих результатів показників надійності, проте їх використання на етапі проектування програмного забезпечення є корисним для прогнозування потрібних ресурсів тестування, дозволяє виявляти зв'язок між складністю програмних засобів і його надійністю, уточнення планових термінів завершення проекту.

3 Оцінювання якості програм на основі метрик Холстеда

Інтегральна система оцінювання не тільки складності, але і якості програм у цілому є системою метрик, запропонованою Холстедом. Способи оцінювання та видів складності програмного продукту можна розділити на групи, показані на рисунку 1.

Статичні характеристики складності, спостерігаються на етапах розроблення програмного продукту і визначають їх трудомісткість.

Динамічні характеристики складності програмного продукту [4] виявляються в процесі його виконання.

Трудомісткість розроблення програмного продукту може оцінюватися двома способами:

- за інтегральними характеристиками складності, які визначають за зовнішніми параметрами додатка, в яких не враховують його внутрішню структуру (підхід «чорного ящика»);

- за структурними характеристиками складності, що враховують внутрішню структуру модулів і залежних від складності маршрутів (потоків управління), складності інформаційних зв'язків, потоків даних (підхід «білого ящика»).

Проблеми при використанні метрик коду. По-перше, всі відомі метрики коду є недостатньо значущими і точними. Вони не здатні забезпечити отримання об'єктивної картини про стан програмної системи, а лише видають показники, які обчислені за заданим алгоритмом. По-друге, процес вимірювання може бути штучно спотворений за рахунок того, що співробітники будуть «оптимізувати» свій програмний код так, щоб метрики видавали кращі результати.

Проте метрики є досить корисним інструментом у руках розробників і менеджерів проектів, що дозволяє виявити моменти виходу розробки на більш низький якісний рівень і розпізнати найбільш складні ділянки в системі.

Метрика Холстеда відноситься до метрик, які обчислюються на підставі аналізу числа рядків і лексичних елементів вихідного коду програми.

Основу метрики Холстеда складають чотири вимірювані характеристики програми:

NUOprtr (Number of Unique Operators) – число унікальних операторів програми, включаючи символи-роздільники, імена процедур і знаки операцій (словник операторів);

NUOprnd (Number of Unique Operands) – число унікальних операндів програми (словник операндів);

Noprtr (Number of Operators) – загальне число операторів у програмі;

Noprnd (Number of Operands) - загальне число операндів у програмі.



Рис. 1. Способи оцінювання та видів складності програмного продукту

Словник модуля:

$$n = \eta_1 + \eta_2, \quad (3)$$

де N – довжина програми;

η_1 – число унікальних різних операторів програми, включаючи символи-роздільники, знаки операцій та імена функцій (словник операторів);

η_2 – число унікальних імен операндів (словник операндів).

Інформаційна довжина програмного модуля:

$$N = \eta_1 * \log_2(\eta_1) + \eta_2 * \log_2(\eta_2). \quad (4)$$

Інформаційний обсяг модуля:

$$V = N * \log_2(\eta), \quad (5)$$

де V – обсяг програми.

Нехай $N1$ – число рядків (операторів), $N2$ – число операндів.

Рівень запису модуля (показник антиконцентрованості обчислень у модулі):

$$L = 2 * \eta^2 / (\eta^1 * N2), \quad (6)$$

де L – рівень запису модуля;

$N2$ – загальне число операндів у програмі.

Метрика Холстеда «інтелектуальності». За допомогою цієї метрики Холстед вимірював інтелектуальний зміст алгоритму, інваріантний відносно мов реалізації:

$$I = L * V, \quad (7)$$

де I – інформаційний зміст програми.

4 Аналіз модулів за допомогою програмного забезпечення визначення метрик Холстеда

Розроблено програмне забезпечення, яке за допомогою метрик Холстеда і методів статичного аналізатора програмного коду дозволяє оцінювати складність та якість програмних продуктів. Програмне забезпечення виконує такі функції:

- 1) побудова графіків різних параметрів;
- 2) виведення інформації з графіків у текстовий вигляд (із значеннями отриманих при експерименті).

Лексичний аналізатор будує графіки, які відображають таку інформацію про проаналізовані модулі: коментування; точний та апроксимований рівень якості (рисунок 2); реальну та теоретичну довжину (рисунок 3); інформативність; витрачені інтелектуальні зусилля.

Метрика Холстеда має виключно інформаційний характер, проте вона залишається однією з небагатьох, які дозволяють кількісно оцінити показник підтримуваних систем в майбутньому, при цьому даний показник має пряму кореляцію з якістю виробленого продукту.

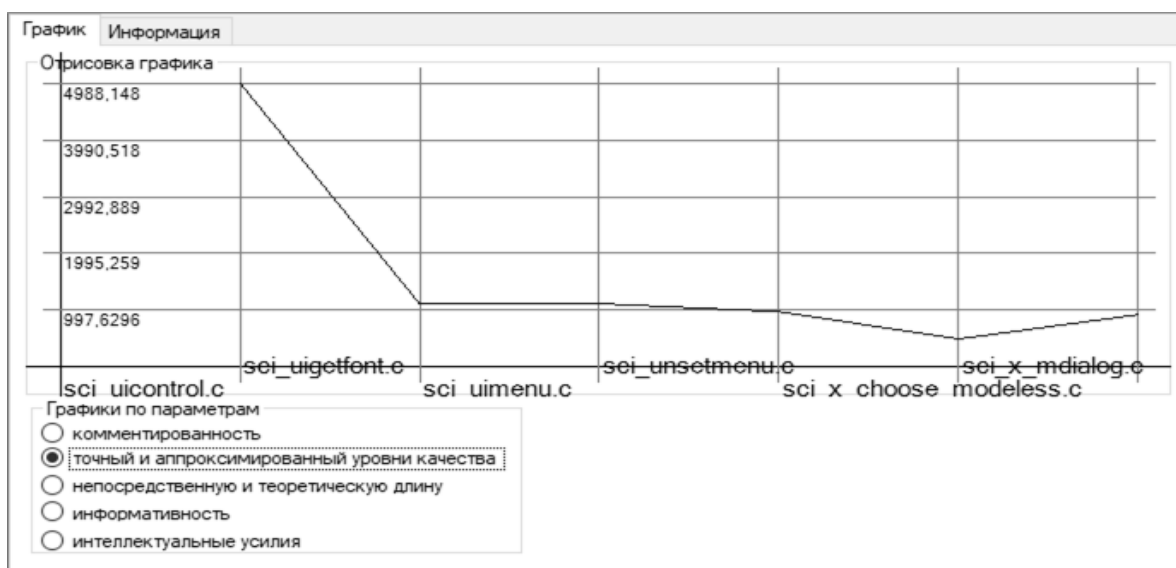


Рисунок 2 – Результат аналізу модуля на рівні якості

На рисунку 3 показано результат реальної та теоретичної довжини аналізу модулів за допомогою формул визначення метрик Холстеда.

У процесі проведення експериментів отримані результати (табл. 1 – 2):

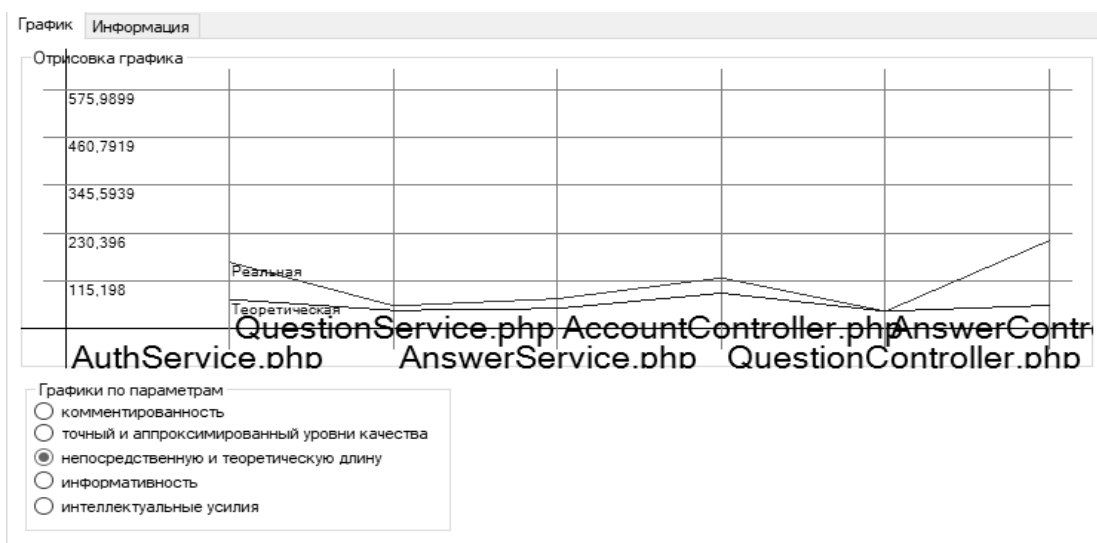


Рисунок 3 – Результат аналізу модуля на реальну і теоретичну довжину

Таблица 1

Результаты анализа модулей

№ з/п	Назва модуля	Коментованість	Реальна довжина	Теоретична довжина	Точний рівень якості	Апроксимований рівень якості
1	sci_uicontrol	0,0392	2461	1308,8635	4988,14814	4988,14814
2	sci_uigetfont	0,0392	1079	474,5420	1101,2105	1101,2105
3	sci_uimenu	0,1145	826	575,9898	1110,7894	1110,7894
4	sci_unsetmenu	0,1007	821	264,5725	969	969
5	sci_x_choose_model	0,0787	485	326,9987	486,6206	486,6206
6	sci_x_mdialog	0,0711	858	427,0575	914,5882	914,5882
Найбільше значення		0,1145	2461	1308,8635	4988,14814	4988,14814

Таблица 2

Результаты анализа модулей

№ з/п	Назва модуля	Інформативність	Інтелектуальні зусилля	Кількість операцій	Кількість операндів
1	sci_uicontrol	93295257,7689	3,7495	1499	962
2	sci_uigetfont	7655545,1367	6,3129	652	427
3	sci_uimenu	6108989,6348	4,9511	491	335
4	sci_unsetmenu	4599354,9387	4,8983	498	323
5	sci_x_choose_modeless	1421345,2575	6,0023	289	196
6	sci_x_mdialog	4960922,5894	5,9307	520	338
Найбільше значення		93295257,7689	6,3129	1499	962

Найбільшого значення набув модуль:

- з коментування: `sci_uimenu`;
- з реальної, теоретичної довжини; точного та апроксимованого рівня якості; інформативності; кількості операцій, операндів: `sci_uicontrol`;
- з інтелектуального зусилля: `sci_uigetfont`;

Висновок

Проведено порівняльний аналіз аналітичних та емпіричних моделей надійності функціонування та якості програмного забезпечення. Описано їх моделі й методи. Виконано апробацію і оцінювання ефективності використаних моделей і методів.

Абсолютно надійних програм не існує, оскільки абсолютний ступінь надійності не може бути теоретично доведений і, отже, є недосяжним. Однак важливо знати, наскільки надійним є конкретне програмне забезпечення.

Практика розроблення програмного забезпечення передбачає пріоритет завдання забезпечення надійності над завданням її оцінювання. Перш ніж забезпечувати надійність, слід навчитися її вимірювати. Але для цього потрібно мати практично прийнятну **одиницю вимірювання надійності функціонування програмного забезпечення і формули її розрахунку**.

Список літератури

- 1 Майерс, Г. Д. Надежность программного обеспечения / Г. Дж Майерс. – М.: Мир, 1980. – 360 с.
- 2 Липаев, В. В. Надежность программных средств / В. В. Липаев. – М.: Синтег, 1998. – 232 с.
- 3 Майерс, Г. Д. Искусство тестирования программ / Г. Дж Майерс. – М.: Финансы и статистика, 1982. – 176 с.
- 4 Гласс, Р. Руководство по надежному программированию / Р. Гласс. – М.: Финансы и статистика, 1982. – 460 с.

References

- 1 Majers, G. D. Nadezhnost` programmnoy obespecheny`ya [Software reliability] / G. Dzh Majers. – Moscow: My`r, 1980, 360 p.
- 2 Ly`paev, V. V. Nadezhnost` programmnykh sredstv [Software Reliability] / V. V. Ly`paev. – Moscow: Sy`nteg, 1998, 232 p.
- 3 Majers, G. D. Y`skusstvo testy`rovany`ya program [The art of software testing] / G. Dzh Majers. – Moscow: Fy`nansy y` staty`sty`ka, 1982, 176 p.
- 4 Glass, R. Rukovodstvo po nadezhnomu programmy`rovany`yu [A guide to reliable programming] / R. Glass. – Moscow: Fy`nansy y` staty`sty`ka, 1982, 460 p.

Надійшла до редакції 10.10.2019, розглянута на редколегії 20.10.2019

Applying Empirical Models and Halstead Metrics to evaluate the quality of application software

The application of empirical models and Halsted metrics to evaluate software quality is considered. The **subject** of the study are methods of measuring the reliabil-

ity of software and its calculation model. The **purpose** of the work is to perform a promising direction for further research of analytical and empirical models of software reliability. The **object** of the study is the process of evaluating the quality of the software, which requires a large number of tasks. This leads to a variety of approaches, methods and tools. **Objectives:** to carry out comparative analysis of analytical and empirical models of software reliability and quality; describe models and methods for benchmarking these software reliability models; to test and evaluate the effectiveness of the models and methods used for the comparative analysis of the analytical and empirical models of the software's reliability. The methods used are software developed that, using Halsted metrics and static code analyzer methods, allows you to evaluate the complexity and quality of software products. This makes it possible to comprehensively consider all aspects related to analytical and empirical models. As a result, software has been developed that, using Halsted metrics and static code analyzer methods, makes it possible to evaluate the complexity and quality of software products. The software performs the following functions: graphing various parameters; outputting information from graphs to text view (with values obtained during the experiment). The lexical analyzer builds graphs that display the following information about the modules analyzed: commenting; accurate and approximate quality level; real and theoretical length; informative; spent intellectual effort. **Conclusions.** The relevance of comparative analysis of analytical and empirical models of software reliability is determined by the fact that most software is unreliable. The scientific novelty of the obtained results is as follows: by means of comparative analysis of analytical and empirical models of reliability of software functioning, to study in detail the models of reliability and to increase the reliability of software.

Keywords: software reliability; mathematical model; empirical modeling; reliability function; failure rate; data model; Halsted metrics.

Відомості про авторів:

Пудовкіна Лариса Федорівна – кандидат технічних наук, доцент кафедри «Інженерії програмного забезпечення», Національний аерокосмічний університет ім. М.Є. Жуковського «Харківський авіаційний інститут», Харків, Україна, e-mail: l.pudovkina@khai.edu.

Сіняєв Вадим Вікторович – студент 667п2 групи, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна, e-mail: vadim2702@gmail.com, ORCID: 0000-0002-4731-6489.

About the Authors

Pudovkina Larisa – Candidate of Engineering Sciences, Associate Professor of the Software Engineering Department, National Aerospace University named after N. E. Zhukovsky “Kharkiv Aviation Institute”, Kharkiv, Ukraine, e-mail: l.pudovkina@khai.edu.

Sinyaev Vadim – student of 667п2 group, National Aerospace University named after N. E. Zhukovsky “Kharkiv Aviation Institute”, Kharkiv, Ukraine, e-mail: vadim2702@gmail.com, ORCID: 0000-0002-4731-6489.