

К вопросу оценки качества информационных систем

И.С. Калытюк, Г.А. Французова, А.В. Гунько

ФГБОУ ВО Новосибирский государственный технический университет, просп. Карла Маркса, д.20, Новосибирск, Россия

Аннотация – В данной статье обсуждается вопрос оценки качества информационных систем. Зачастую качество определяется как «соответствие требованиям» и «пригодность к использованию». «Соответствие требованиям» предполагает, что требования должны быть настолько чётко определены, что они не могут быть поняты и интерпретированы некорректно. Позже, на этапе разработки, производятся регулярные измерения разработанного продукта, для определения соответствия требованиям. Любые несоответствия должны рассматриваться как дефекты. «Пригодность к использованию» принимает во внимание требования и ожидания конечных пользователей продукта, которые ожидают, что продукт или предоставляемый сервис будет удобным для их нужд. Однако разные пользователи могут использовать продукт по-разному. Это означает, что продукт должен обладать максимально разнообразными вариантами использования. Как видно из вышеперечисленных общих определений качества, задача оценки является крайне сложной из-за многообразия интересов пользователей. Чаще всего невозможно предложить одну универсальную меру качества и приходится использовать ряд характеристик, охватывающих весь спектр предъявляемых требований. Наиболее близки к задачам оценки качества информационных систем модели качества программного обеспечения. В настоящее время используется несколько абстрактных моделей качества программного обеспечения. В работе рассматриваются некоторые из них – модель МакКола и модель Боема. Данные модели являются концептуальными и подходят для оценки качества любых информационных систем. Помимо рассмотрения концептуальных моделей, особое внимание обращается на оценку параметров уже готовых систем, таких как стоимость разработки, временные затраты, трудоёмкость. Существуют как линейные подходы, использующие простейшие формулы, так и различные модели, которые оперируют эмпирическими данными – например, *SLIM* и *COCOMO*. Эти модели часто используются в программных комплексах по типу *Duessa Estimate Easy UC* и *SoftStar SystemStar & Costar*, также упоминаемых в работе. Вдобавок необходимо упомянуть различные продукты по оценке размера кода и метрик сложности.

Ключевые слова: качество, модель МакКола, модель Боема, стоимость, трудоёмкость, временные затраты, *SLIM*, *COCOMO*.

ВВЕДЕНИЕ

Рассматривая определение качества информационной системы, можно столкнуться с большим количеством различных формулировок. В случае программных продуктов существует несколько интерпретаций качества. Они определяются на этапе разработки требований.

Основными определениями можно назвать два – определение *ISO* и *IEEE*. Определение, предлагаемое *ISO*, содержит следующее: качеством является полнота характеристик и свойств, относящихся к продукту или процессу, обеспечивающих удовлетворение потребностей. В случае *IEEE*, качество – это степень, при которой программный комплекс содержит в себе набор необходимых свойств.

Определение качества имеет смысл, если учитываются реальные условия использования информационных систем. Из этого следует, что качественные требования чётко зависят от области применения.

Целью настоящей работы является обсуждение способов оценки качества информационных систем. Структура данной работы следующая: в первом разделе представлена постановка задачи. Различные способы оценки качества представлены в разделе 2. Оценка параметров разработанных

систем и существующие программные комплексы в этой области описываются в разделах 3 и 4.

ПОСТАНОВКА ЗАДАЧИ

При создании информационных систем необходимо предусмотреть возможность измерения качества с помощью определённых критериев оценки, разработка которых предполагает решение следующих подзадач.

Для достижения поставленной цели общую задачу разделим на ряд подзадач.

1. Рассмотрение способов оценки качества систем.
2. Рассмотрение методов оценки параметров разработанной системы.
3. Рассмотрение программных комплексов, осуществляющих оценку параметров.

Для первой подзадачи требуется изучение существующих способов, которые используются для концептуальной оценки систем. Решение второй подзадачи – это определение методов для оценки таких параметров, как трудоёмкость, временные затраты, стоимость. В случае третьей подзадачи необходим обзор существующего программного обеспечения области оценки параметров разработанных систем.

СПОСОБЫ ОЦЕНКИ КАЧЕСТВА СИСТЕМ

В случае различных существующих стандартов оценки, таких как *ISO 9126* [1-4], предполагается разделение качества на совокупность не всегда однородных характеристик. При их использовании нет концептуальной основы для всех существующих программных комплексов.

Однако, существуют способы, которые позволяют оценить любую информационную систему. Например, Шломи Фиш выделяет следующие факторы высокого качества программных систем [5].

- Доступность программного обеспечения и его обновлений.
- Простота при определении версии продукта.
- Открытый исходный код при возможности.
- Отсутствие долгого обучения для возможности использования систем.
- Переносимость.
- Безопасность.
- Совместимость с предыдущими версиями.
- Множество путей поддержки пользователей.
- Небольшая ресурсоёмкость.

Широкую известность приобрела модель качества МакКола [6]. В ней характеристики разделяются на группы.

- Факторы – описание со стороны пользователей.
- Критерии – описание со стороны разработчиков.
- Метрики – количественное описание, измерение качества.

Было выделено 11 факторов, разбитые на 3 группы в зависимости от способа работы с программным комплексом. Получившуюся структуру представили графически в виде треугольника МакКола (Рис. 1).



Рис. 1. Треугольник МакКола

Критериями качества в данной модели являются уровни факторов в виде чисел. Они соответствуют поставленным целям разработки.

Для измерения факторов вводятся метрики качества. Они выражаются числами от 0 до 10. Выделены следующие метрики.

- Удобство проверки на соответствие стандартам.
- Точность вычислений, управления.
- Степень стандарта интерфейса.
- Полнота функциональности.
- Однородность принципов проектирования и документирования.
- Степень стандарта форматов данных.
- Устойчивость к ошибкам.
- Эффективность.
- Расширяемость.
- Диапазон потенциального использования.
- Аппаратная платформенная независимость.
- Полнота протоколирования ошибок.
- Модульность.
- Удобство использования.
- Защищённость.
- Самодокументированность.
- Простота использования.
- Программная платформенная независимость.
- Возможность соответствия продукта требованиям.
- Удобство обучения использованию.

Каждая из вышеуказанных метрик может повлиять на оценку факторов качества. Соответственно, каждый фактор является комбинацией значений метрик.

После создания модели МакКола, Боем предложил ещё одну модель [7]. В ней он расширил предшествующую модель, определив 20 атрибутов, которые включили в себя 11 факторов МакКола. Дополнительно были добавлены атрибуты ясности, удобства добавления изменений, документированности, понятности, адекватности, функциональности, универсальности, экономической эффективности и удобства добавления изменений.

Для оценки системы можно использовать все вышеперечисленные методы – они не привязаны к специфике системы. Для более подробной оценки всё же стоит воспользоваться метриками, выделенными в различных стандартах.

ОЦЕНКА ПАРАМЕТРОВ РАЗРАБОТАННЫХ СИСТЕМ

Отдельное внимание стоит обратить на оценку различных параметров информационных систем, таких как трудоёмкость, стоимость и временные затраты. Данная задача является достаточно сложной, существуют различные подходы к решению. Чаще всего требуется

комплексное рассмотрение нескольких подходов.

Самым простым методом является линейный подход [8]. Есть несколько формул, которые определяют стоимость разработки и трудоёмкость. Формула для стоимости разработки системы следующая:

$$S = T * W, \quad (1)$$

где T – количественная оценка трудозатрат в человеко-часах или человеко-месяцах (трудоёмкость); W – удельная стоимость.

В случае трудоёмкости также существует простая формула:

$$T = L * Q, \quad (2)$$

где L – размер исходного кода; Q – временная производительность.

Следовательно, стоимость разработки можно вычислить следующим образом:

$$S = L * Q * W, \quad (3)$$

где L – размер исходного кода; Q – временная производительность; W – удельная стоимость.

Данные простейшие формулы применяются в настоящее время достаточно часто. Однако существуют работы, в которых подобный метод критикуется. Фредерик Брукс в своей книге «Мифический человеко-месяц, или как создаются программные системы» [9] высказывал предположение, что код для решения задач со временем становится компактнее. В связи с этим, данная формула не всегда будет корректной.

Также, существует много способов оценки программных комплексов, использующих прошлый опыт (эмпирические данные). Самыми известными из них являются *SLIM*, *COCOMO*, *COCOMO II*.

SLIM стала первой нелинейной моделью, которая использовала эмпирические данные [10]. Трудоёмкость вычисляется таким образом:

$$T = \left(\frac{L}{C}\right) \times t_d^{-4}, \quad (4)$$

где L – размер проекта; C – фактор среды (учитывает производительность и уровень технологий); t_d – ограничение на срок поставки в годах.

Позже была предложена наиболее известная в настоящее время модель оценки стоимости *COCOMO* [11]. В её случае предлагается следующая формула для трудоёмкости:

$$T = a * L * b, \quad (5)$$

где L – размер проекта; a , b – константы в зависимости от режима модели.

Следует заметить, что a является константой смены режима, что приводит к скачкообразному изменению трудозатрат. Тем не менее, данный рост не увеличивает временные затраты на проект. Временные затраты определяются по следующей формуле:

$$F = 2,5 * T * k, \quad (6)$$

где T – трудозатраты, k – константа.

Из данной формулы видно, что константа k может позволить не увеличивать временные затраты. В целом, в *COCOMO* очень многое зависит от подбора констант.

Продолжением классической модели стала *COCOMO II* [12]. Было предложено оценивать размер не только количеством строк кода, но и функциональными и объектными точками. Также стал учитываться уровень зрелости процесса разработки. Несмотря на это, основные принципы классической *COCOMO* были сохранены. Формулы были усложнены и модель стала более гибкой.

СУЩЕСТВУЮЩИЕ ПРОГРАММНЫЕ КОМПЛЕКСЫ ПО ОЦЕНКЕ ПАРАМЕТРОВ РАЗРАБОТАННЫХ СИСТЕМ

Рассматривая оценку параметров разработанного программного обеспечения, необходимо упомянуть о уже существующих продуктах в этой области. Чаще всего данные программные комплексы рассчитывают метрики, основываясь на исходный код и *UML*-диаграммы.

Самым распространённым программным обеспечением являются средства для подсчёта размера кода. Данное программное обеспечение не ограничивается нахождением количества строк. Также оно выводит дополнительные параметры, которыми пренебрегают при вычислении без использования дополнительных программ. Из наиболее известных можно выделить *SLOCCount* [13] и *Code Counter Pro* [14].

SLOCCount является простейшим бесплатным продуктом. Он ориентирован на *UNIX*-системы, но поддержка *Windows* также присутствует. Из преимуществ выделяется самое большое количество поддерживаемых языков, что делает его универсальным продуктом. Недостатком же является поддержка только базовых моделей первой версии *COCOMO*.

Из коммерческих продуктов стоит обратить внимание на *Code Counter Pro*. Основным достоинством является детализация отчётов, которой сопутствует довольно развитый графический интерфейс. Недостатком можно считать вычисление только базовых характеристик в сравнении с бесплатными продуктами.

В случае метрик сложности существуют такие программные комплексы, как *Verisoft Complexity Measures Tool* [15], *Eclipse Metrics Plugin* [16].

Verisoft Complexity Measures Tool – известный коммерческий проект, который поддерживает расчёт различных метрик: цикломатической сложности, Холстеда, индекс сопровождаемости. Плюсом является графический интерфейс и формирование подробных отчётов. Из минусов можно

отметить поддержку только таких языков, как C++ и Java.

Из бесплатных продуктов можно рассмотреть *Eclipse Metrics Plugin*. Преимуществом является поддержка всех вышеперечисленных метрик, включая объектно-ориентированные. Недостатком же является то, что он работает в связке со средой разработки *IDE Eclipse*.

Отдельное внимание стоит обратить на оценку экономических параметров. Средства для их оценки работают непосредственно с трудоёмкостью, сроками реализации и стоимостью информационных систем. Сложность данных инструментов зачастую больше, чем у инструментов вычисления *SLOC* и метрик сложности.

Duversa Estimate Easy UC [17] работает с диаграммами прецедентов и актёров. На их основе рассчитывается трудоёмкость проекта. Плюсом данного программного комплекса является возможность импорта данных для оценки из всех известных форматов документов. Также, помимо трудоёмкости вычисляется число объектных точек, после чего проводится вторая оценка трудоёмкости при использовании *COCOMO II*. Минусом же является то, что пользователь сам должен проводить сравнение данных оценок из-за частого их несовпадения.

Рассматривая инструменты на основе методик *COCOMO*, следует обратить внимание на *SoftStar SystemStar & Costar* [18]. В него включены такие функциональные возможности, как например поддержка модели *REVIC*, возможность полного анализа в виде отчётов. Однако присутствует и недостаток – калибрование на основе существующих проектов отсутствует.

ЗАКЛЮЧЕНИЕ

В работе рассмотрены существующие понятия качества информационных систем. Выделяются различные способы оценки качества систем, которые могут быть полезны при изменении систем и являются концептуально более общими. Также рассматривались методы оценки параметров информационных систем на основе различных формул и программное обеспечение, реализующее данные вычисления.

Программные комплексы оценки параметров разработанной системы могут быть использованы после подробного исследования и тестирования. Как видно из возможностей рассмотренных средств, все они используют разные методики.

Таким образом, для более объективной оценки качества информационных систем рекомендуется одновременное применение нескольких инструментов.

ЛИТЕРАТУРА

- [1] ISO/IEC 9126 Software engineering — Product quality — Part 1: Quality model, 2001.
- [2] ISO/IEC 9126 Software engineering — Product quality — Part 2: External metrics, 2001.
- [3] ISO/IEC 9126 Software engineering — Product quality — Part 3: Internal metrics, 2001.
- [4] ISO/IEC 9126 Software engineering — Product quality — Part 4: Quality in use metrics, 2001.
- [5] What Makes Software High-Quality? – Shlomi Fish's Homesite: [Электронный ресурс]. – Режим доступа: <https://www.shlomifish.org/philosophy/computers/high-quality-software/>, свободный (дата обращения: 10.04.2021).
- [6] McCall J., Richards P., Walters G. Factors in Software Quality. Technical Report CDRL A003, US Rome Air Development Centre, vol. 1, 1977.
- [7] Boehm B.W., Brown J.R., Kaspar H., Lipow M., MacLeod G.J., Merritt M.J. Characteristics of Software Quality. TRW Series of Software Technology, Amsterdam, North Holland, 1978, 166 p.
- [8] Стефанова Н.А., Курбангелдыев Д. Оценка стоимости разработки программного обеспечения. Актуальные вопросы современной экономики. 2020. №1. С. 67-72.
- [9] Brooks F.P. The mythical man-month: essays on software engineering. Addison-Wesley Longman Publishing Co., Inc., MA, Boston, 1995, 322 p.
- [10] Putnam L.H. A General Empirical Solution to the Macro Software Sizing and Estimating Problem // IEEE Transactions on Software Engineering, 1978, no. 4(4), pp. 34–360.
- [11] Boehm B.W. Software Engineering Economics in *IEEE Transactions on Software Engineering*, 1984, vol. SE-10, no. 1, pp. 4–21.
- [12] Boehm B.W. et al. The COCOMO 2.0 Software Cost Estimation Model: A Status Report. American Programmer, 1996, no. 9(7), pp. 2–17.
- [13] SLOCCount: <https://dwheeler.com/sloccount/>, (дата обращения: 20.04.2021).
- [14] Code Counter Pro – Free download and software reviews – CNET Download: https://download.cnet.com/Code-Counter-Pro/3000-2383_4-10173501.html, (дата обращения: 20.04.2021).
- [15] Verisoft – Code Complexity Measures Tool: https://www.verifysoft.com/en_code_complexity.html, (дата обращения: 22.04.2021).
- [16] Eclipse Plugins, Bundles and Products – Eclipse Marketplace: <https://marketplace.eclipse.org/content/eclipse-metrics#group-details>, (дата обращения: 22.04.2021).
- [17] Estimate Easy UC – Duversa Software Informer: <https://estimate-easy-uc.software.informer.com/>, (дата обращения: 24.04.2021).
- [18] SystemStar Software Estimation Tool based on COCOMO II and COSYSMO 2, from Softstar Systems: <http://softstarsystems.com/>, (дата обращения: 24.04.2021).



Иван Сергеевич Кальтюк, аспирант кафедры автоматике НГТУ. Основное направление научных исследований: разработка и исследование систем сбора и анализа данных. E-mail: ivankalytyuk@yandex.ru



Андрей Васильевич Гунько, к. т. н., доцент кафедры автоматике НГТУ. Основное направление научных исследований: разработка автоматизированных систем сбора и обработки результатов. E-mail: gun@ait.cs.nstu.ru



Галина Александровна Французова, д. т. н., профессор кафедры автоматике НГТУ. Основное направление научных исследований: синтез систем экстремального регулирования. E-mail: frants@ac.cs.nstu.ru

Статья поступила 20.05.2021.

On the Issue of Assessing the Quality of Information Systems

I. S. Kalytyuk, G. A. Frantsuzova, A.V. Gunko
Novosibirsk State Technical University, Novosibirsk, Russia

Abstract. This paper discusses the issue of assessing the quality of information systems. Often, quality is defined as "compliance" and "suitability for use". "Compliance" implies that the requirements must be so clearly defined that they can't be understood and interpreted incorrectly. Later, during the development phase, regular measurements of the developed product are made to determine compliance with the requirements. Any inconsistencies should be treated as defects. "Suitability for use" takes into account the requirements and expectations of the end users of the product, who expect that the product or service provided will be convenient for their needs. However, different users may use the product in different ways. This means that the product should have the most diverse use cases.

As can be seen from the above general definitions of quality, the task of evaluation is extremely difficult due to the diversity of user interests. Most often, it is impossible to offer a single universal measure of quality and you have to use a number of characteristics that cover the entire range of requirements. Software quality models are the closest to the tasks of assessing the quality of information systems. Currently, several abstract models of software quality are used. The paper considers some of them – the McCall model and the Boehm model. These models are conceptual and are suitable for evaluating the quality of any information systems.

In addition to the consideration of conceptual models, special attention is paid to the evaluation of the parameters of ready-made systems, such as the cost of development, time costs, and labor intensity. There are both linear approaches that use the simplest formulas, and various models that operate on empirical data – for example, SLIM and COCOMO. These models are often used in software packages such as Duessa Estimate Easy UC and SoftStar SystemStar & Costar, also mentioned in this paper. In addition, it is necessary to mention various products for evaluating code size and complexity metrics.

Keywords: quality, McCall model, Boehm model, cost, labor intensity, time cost, SLIM, COCOMO.

REFERENCES

- [1] ISO/IEC 9126 Software engineering — Product quality — Part 1: Quality model, 2001.
- [2] ISO/IEC 9126 Software engineering — Product quality — Part 2: External metrics, 2001.
- [3] ISO/IEC 9126 Software engineering — Product quality — Part 3: Internal metrics, 2001.
- [4] ISO/IEC 9126 Software engineering — Product quality — Part 4: Quality in use metrics, 2001.
- [5] What Makes Software High-Quality? – Shlomi Fish's Homesite. Available at: <https://www.shlomifish.org/philosophy/computers/hi-gh-quality-software/> (accessed 10 April 2021).
- [6] McCall J., Richards P., Walters G. Factors in Software Quality. Technical Report CDRL A003, US Rome Air Development Centre, vol. 1, 1977.
- [7] Boehm B.W., Brown J.R., Kaspar H., Lipow M., MacLeod G.J., Merritt M.J. Characteristics of Software Quality. TRW Series of Software Technology, Amsterdam, North Holland, 1978, 166 p.
- [8] Stefanova N.A., Kurbangeldyev D. Estimating the cost of software development. Current issues of the modern economy, 2020, no. 1, pp. 67-72.
- [9] Brooks F.P. The mythical man-month: essays on software engineering. Addison-Wesley Longman Publishing Co., Inc., MA, Boston, 1995, 322 p.
- [10] Putnam L.H. A General Empirical Solution to the Macro Software Sizing and Estimating Problem // IEEE Transactions on Software Engineering, 1978, no. 4(4), pp. 34–360.
- [11] Boehm B.W. Software Engineering Economics in IEEE Transactions on Software Engineering, 1984, vol. SE-10, no. 1, pp. 4–21.
- [12] Boehm B.W. et al. The COCOMO 2.0 Software Cost Estimation Model: A Status Report // American Programmer, 1996, no. 9(7), pp. 2–17.
- [13] SLOCCount. Available at: <https://dwheeler.com/sloccount/> (accessed 20 April 2021).
- [14] Code Counter Pro – Free download and software reviews – CNET Download. Available at: https://download.cnet.com/Code-Counter-Pro/3000-2383_4-10173501.html (accessed 20 April 2021).

- [15] Verisoft – Code Complexity Measures Tool. Available at: https://www.verifysoft.com/en_code_complexity.html (accessed 22 April 2021).
- [16] Eclipse Plugins, Bundles and Products – Eclipse Marketplace. Available at: <https://marketplace.eclipse.org/content/eclipse-metrics#group-details> (accessed 22 April 2021).
- [17] Estimate Easy UC – Duvevsa Software Informer. Available at: <https://estimate-easy-uc.software.informer.com/> (accessed 24 April 2021).
- [18] SystemStar Software Estimation Tool based on COCOMO II and COSYSMO 2, from Softstar Systems. Available at: <http://softstarsystems.com/> (accessed 24 April 2021).



Ivan Sergeevich Kalytyuk, postgraduate student of the Department of Automation, NSTU. The main direction of scientific research: development and research of data collection and analysis systems.
E-mail: ivankalytyuk@yandex.ru
Novosibirsk, prosp. Karl Marx, 20



Galina Frantsuzova, Doctor of Technical Sciences, Professor, Department of Automation, NSTU. The main direction of scientific research: the synthesis of systems of extreme regulation.
E-mail: frants@ac.cs.nstu.ru
Novosibirsk, prosp. Karl Marx, 20



Andrey Vasilievich Gunko, Ph.D., Associate Professor of the Department of Automation, NSTU. The main direction of scientific research: the development of automated systems for collecting and processing results.
E-mail: gun@ait.cs.nstu.ru
Novosibirsk, prosp. Karl Marx, 20

The paper has been received on 20/05/2021.