УДК 004.412

К ВОПРОСУ О МЕТРИКАХ ТРУДОЁМКОСТИ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Евдокимов И.В., Байкалов И.С., Зуденков А.И., Радионов Т.В., Цирюльникова А.М.
ФГАОУ ВО «Сибирский федеральный университет», Красноярск, e-mail: evd-ivan@yandex.ru

Метрики программного обеспечения необходимы для безотказного функционирования системы. Каждый разработчик понимает их важность, но мало кто задумывался о зависимости между ними. Зачастую каждая новая метрика берёт своё начало от своей предшественницы и дополняется чем-то новым. Рассмотрев на примере три самых используемых метрики ПО – LOC (число строк кода), COCOMOII (трудоёмкость разработки ПО), Function Points (трудозатраты на разработку ПО), авторы оценивают их зависимость по построенным диаграммам, исходя из рассчитанных данных, взятых из трёх известных приложений – «BlaBlaCar», «Uber», «GetTaxi». Для подтверждения результата проводятся расчёты по трём выбранным метрикам для созданного авторами приложения «CollectiveRides» и строятся диаграммы, по которым отчётливо видна связь выбранных метрик.

Ключевые слова: cocomo, function points, loc, метрики программного обеспечения

CONSIDERING THE QUESTION OF THE METRICS OF THE LABORITY OF DEVELOPMENT OF MOBILE APPLICATION

Evdokimov I.V., Baykalov I.S., Zudenkov A.I., Radionov T.V., Tsiryulnikova A.M.
Siberian Federal University, Krasnoyarsk, e-mail: evd-ivan@yandex.ru

Software metrics are necessary for trouble-free system operating. Each developer understands their importance, but few people have thought about the dependence between them. Often, each new metric originates from its predecessor and is complemented by something new. Considering the example of the three most used software metrics – LOC (number of lines of code), COCOMOII (labouriousness of software development), Function Points (labor costs of softwate dewelopment) according to the authors' opinion, their dependence on the constructed diagrams is estimated based on the calculated data taken from Three well-known applications – BlaBlaCar, Uber, GetTaxi. To confirm the result, calculations are made for the three selected metrics for the «CollectiveRides» application created by the authors and diagrams are constructed for which the relationship of the selected metrics is clearly visible.

Keywords: cocomo, function points, loc, software metrics

В жизни люди часто сталкиваются с обработкой данных: ошибки системы могут привести к нарушению информационной безопасности, нанесению материального ущерба и так далее. Чтобы избежать всех этих нарушений, необходимо разработать программный продукт с учётом всех требований надёжности. Надёжность программного обеспечения – вероятность безотказной работы в течение некоторого периода времени [1]. Исходя из этого определения, нужно иметь возможность измерять качественные характеристики программного обеспечения (ПО) на протяжении всего цикла разработки. Выделяют следующие критерии оценки ПО: функциональность, надежность, эффективность, модифицируемость, мобильность [2].

Для определения критериев качества используют метрики. Под метрикой ПО [3] понимается система измерений, с помощью которой можно получить численное значение некоторого свойства программного обеспечения или его спецификаций.

В набор метрик входит [4]:

– анализ функциональных точек;
– количество строк кода;
– расчёт оценки стоимости разработки ПО;
– расчёт оценки трудоёмкости разработки;
– связность.

Целью авторов является анализ взаимозависимости метрик IT-проектов и, после получения определенных данных, сопоставление их с анализом взаимозависимости метрик разрабатываемого мобильного приложения «CollectiveRides» с целью определения совпадения данных анализов этих проектов или же их несовпадения и причин этого.

Рассмотрим самые значимые метрики из существующих [5], по мнению авторов, и перейдем к исследованию.

Количество строк кода (LOC with PERT): это число строк кода, исключая комментарии. Данная метрика находится в зависимости от определённого языка программирования, но, несмотря на это, она остаётся самой используемой. Точное число LOC можно узнать только после написания проекта, поэтому довольно сложно оценить размер кода программного продукта [6].

Осуществить такую оценку можно, используя способ анализа задач, который называется PERT [7].

Function Point (FP): эта метрика оценивает трудозатраты при разработке программного обеспечения. Под функциональными точками подразумеваются функциональные характеристики процессов разработки ПО. Анализируя их, нужно выполнить исследование предметной оценки, выявления границы программного продукта и функциональности разработки. Затем выполнить общий подсчёт и суммировать функциональные точки (FP) без учёта коэффициента выравнивания [8].

COCOMO II: рассматриваемая метрика оценивает трудоёмкость разработки ПО. Существуют две стадии оценки: начальная оценка и подробная оценка после проработки архитектуры. На обеих стадиях оценки нужно определить пять факторов масштаба.

Количество и значения множителей трудоёмкости отличаются для разных стадий оценки проекта. Для первого этапа необходимо оценить семь множителей трудоёмкости. Для второго этапа нужно определить 17 множителей трудоёмкости [9].

Сопоставим полученные данные с уже существующих популярных проектов, таких как «BlaBlaCar», «Uber» и «GetTaxi» (в настоящее время имеющий название Gett), чья тематика близка к «CollectiveRides». Данные взяты из крупнейшего веб-сервиса по хостингу IT-проектов GitHub [10] путем анализа исходного кода рассматриваемых проектов, ознакомиться и проанализировать которые можно на рис. 1, 2 и 3.

Проанализировав диаграммы, можно заметить линейную зависимость, выбранных авторами метрик ПО – чем больше строк кода, тем больше возрастают трудозатраты, а также трудоёмкость проекта.

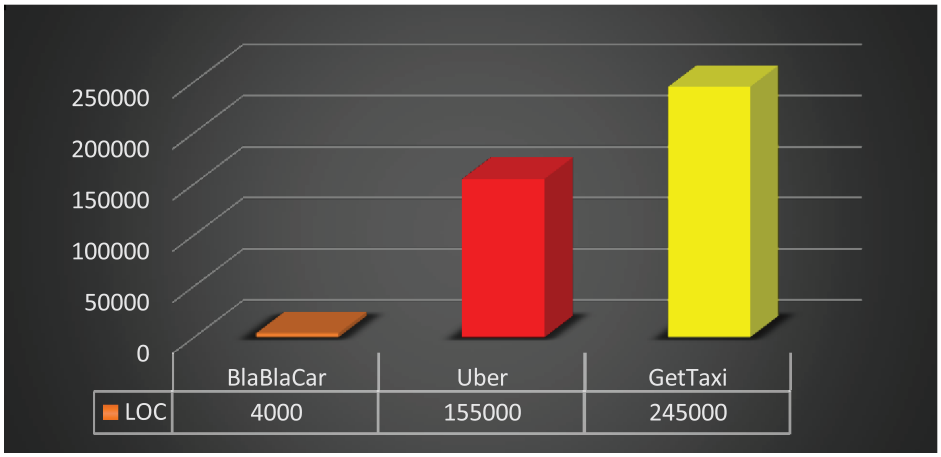

Рис. 1. График LOC рассматриваемых проектов

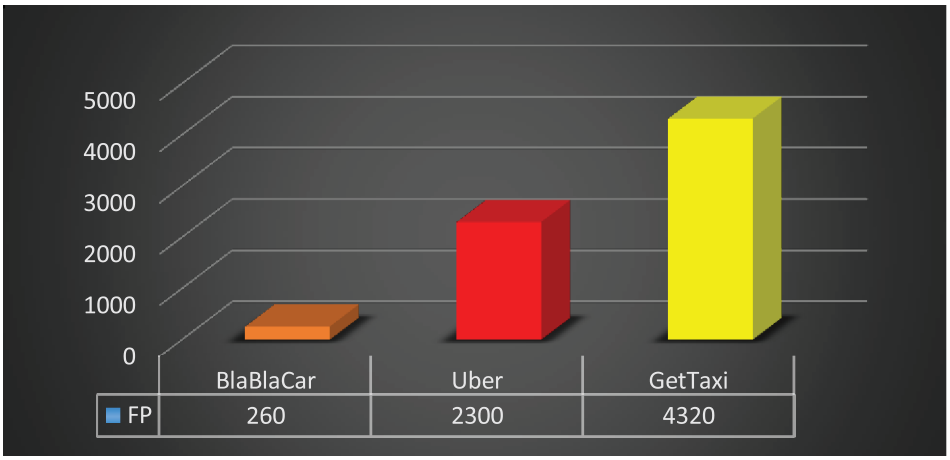

Рис. 2. График FP рассматриваемых проектов

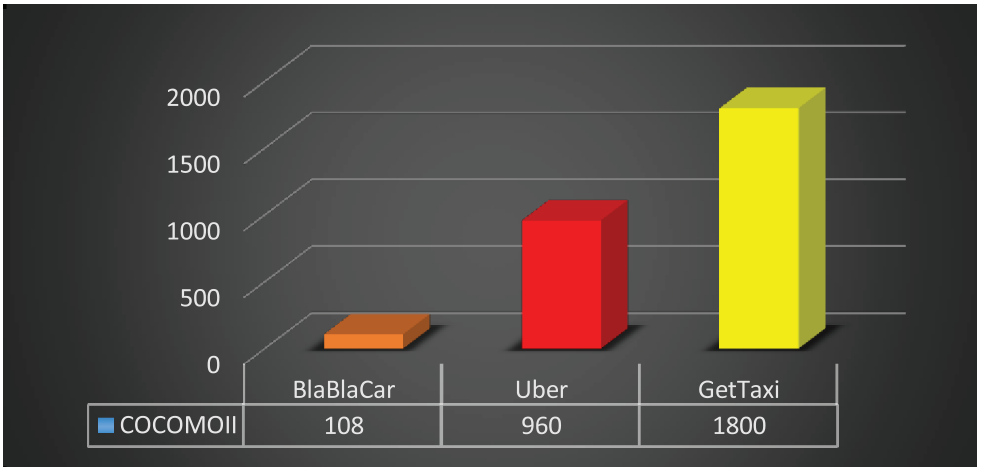

Рис. 3. График COCOMO II рассматриваемых проектов

Для проверки данной зависимости было реализовано мобильное приложение «CollectiveRides» для удобного, комфортного и, самое главное, дешевого передвижения для студентов от дома или общежития до университета и обратно. С помощью приложения пользователи могут быстро и без особых усилий собрать группу попутчиков, создать совместный чат для общения.

Методика LOC with PERT

LOC with PERT: для расчёта количества строк понадобятся 3 эксперта в данной области, которые знакомы с темой проекта и представляют его трудозатраты (табл. 1).

Таблица 1

Оценки экспертов

№ эксперта	L_i (KLOC)	H_i (KLOC)	M_i (KLOC)
1	3	6	4,5
2	3	7	5
3	5	7	6

$$s = \frac{1}{n}\sum_{i=1}^{n}\frac{L_i + H_i + 4M_i}{6}. \qquad (1)$$

Применяя метод оценки LOC with PERT, программа «CollectiveRides» выполним оценку проекта по формуле (1):

$$s = \frac{1}{n}\sum_{i=1}^{n}\frac{L_i + H_i + 4M_i}{6} = \frac{1}{3}\sum_{i=1}^{3}\frac{L_i + H_i + 4M_i}{6} =$$

$$= \frac{1}{3}(4{,}5 + 5 + 6) = 5{,}16\,KLOC.$$

Методика Function Point

1. Определение типа оценки

Разберем разработку программы «CollectiveRides», которая будет разработана в среде Android Studio 2.3 на Java.

2. Определение области оценки и границ продукта

Рассмотрим все планируемые функции, такие как авторизация, чат, поиск попутчиков, вызов такси и ориентация по карте.

Границы продукта (рис. 4) [11]:

3. Подсчет функциональных точек, связанных с данными

Для оценки количества невыровненных функциональных точек (UFP) необходимо использовать матрицу сложности данных (табл. 2) [11, 12].

Таблица 2

Матрица сложности данных

	1–19 DET	20–50 DET	50 + DET
1 RET	Low	Low	Average
2–5 RET	Low	Average	High
6 + RET	Average	High	High

Оценка данных в UFP подсчитывается различными способами для внутренних логических файлов (ILFs) и для внешних интерфейсных файлов (EIFs) (табл. 3) в зависимости от их сложности [13].

Для разрабатываемого проекта, по предварительной оценке, сложность ILFs и EIFs оказалась такой, что окно авторизации имеет 5 невыровненных точек, окно чата и доступ к местоположению – 10 невыровненных точек, окно с функциями поиска попутчиков и вызова такси – 10 невыровненных точек, а окно с картой – 5.

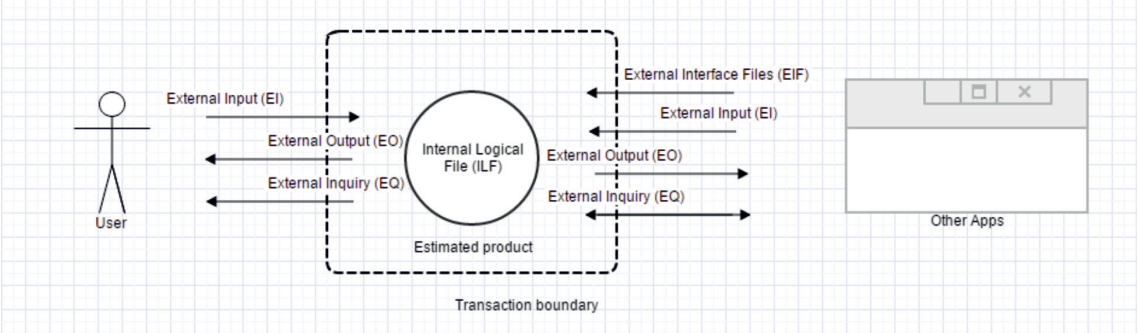

Рис. 4. Границы продукта в методе функциональных точек

Таблица 3

Оценка данных в UFP для ILFs и EIFs [11]

Сложность данных	Количество UFP (ILF)	Количество UFP (EIF)
Low	7	5
Average	10	7
High	15	10

4. Подсчет функциональных точек, связанных с транзакциями

Для оценивания сложности транзакций FP служат матрицы, которые представлены на табл. 4 и табл. 5.

Таблица 4

Матрица сложности внешних входных транзакций (EI) [11]

EI	1–4 DET	5–15 DET	16 + DET
0–1 FTR	Low	Low	Average
2 FTR	Low	Average	High
3 + FTR	Average	High	High

Таблица 5

Матрица сложности внешних выходных транзакций и внешних запросов (EO & EQ) [11]

EO & EQ	0–1 DET	6–19 DET	20 + DET
0–1 FTR	Low	Low	Average
2–3 FTR	Low	Average	High
4 + FTR	Average	High	High

Оценка транзакций в UFP может быть получена из матрицы сложности транзакций в UFP (табл. 6).

Таблица 6

Сложность транзакций в UFP [11]

Сложность транзакций	Количество UFP (EI & EQ)	Количество UFP (EO)
Low	3	4
Average	4	5
High	6	7

В приложении авторов, по расчетам архитектора, будет 5 EI транзакции уровня Low, 4 EO уровня Low и 2 EQ транзакции уровня Average. Следовательно, приблизительное количество невыровненных функциональных точке для данного расчета будет: 15, 16, 8.

5. Определение суммарного количества UFP

Общий объем продукта в UFP определяется путем суммирования по всем информационным объектам (ILF, EIF) и элементарным операциям (транзакциям EI, EO, EQ) [11]:

$$UFP = \sum_{ILF} UFP_i + \sum_{EIF} UFP_i + \\ + \sum_{EI} UFP_i + \sum_{EO} UFP_i + \sum_{EQ} UFP_i. \qquad (2)$$

Для разбираемого примера

$$UFP = (5+10+5)+10+ \\ +(5\cdot 3)+(4\cdot 4)+(4\cdot 2)=69.$$

Метод анализа функциональных точек ничего не говорит о трудоемкости разработки оцененного продукта [14]. Если у компании нет статистики, то для оценки трудоемкости и сроков проекта можно использовать метод COCOMO II [15].

Методика COCOMO II

Разберем предварительную оценку

Формула для оценивания трудоемкости [16]:

$$PM = EAF \times A \times SIZE^E, \qquad (3)$$

где

$$E = B + 0{,}01 \times \sum_{j=1}^{5} SF_j \qquad (4)$$

$B = 0{,}91$; $A = 2{,}94$;

SF_j – фактор масштаба;

SIZE – объем программного продукта в тысячах строк исходного текста;
EM_j – множители трудоемкости, $n = 7$;
EAF – произведение выбранных множителей трудоемкости:

$$EAF = \prod_{j=1}^{n} EM_j . \quad (5)$$

По формулам (3), (4) и (5) имеем

$$E = B + 0,01 \times \sum_{j=1}^{5} SF_j = 0,91 + 0,01 \times$$

$$\times \left(3,72 + 2,03 + 4,24 + 2,19 + 4,68\right) = 1,0786,$$

$$EAF = \prod_{j=1}^{n} EM_j = \prod_{j=1}^{7} EM_j =$$

$$= 1 \times 1,22 \times 1 \times 1 \times 1 \times 1,10 \times 1,14 = 1,53,$$

$$PM = EAF \times A \times SIZE^E =$$

$$= 1,53 \times 2,94 \times 5,16^{1,0786} = 26,4 \text{ чел.×мес.}$$

Заключение

Исходя из полученных данных, можно оценить трудозатраты, а также сложность разработки рассматриваемого проекта. Количество строк кода приложения «CollectiveRides» будет составлять примерно 5160. Также по методике Functional Points были составлены функциональные точки и спрогнозирована сложность каждой из них, рассчитан объём продукта в невыровненных функциональных точках, равный 69. Трудоёмкость разрабатываемого проекта была вычислена по методике COCOMO II – время разработки составило 8 месяцев.

По результатам, полученным в ходе исследования, можно предположить, что рассмотренные метрики находятся в тесной связи – от количества строк кода зависят трудозатраты и трудоёмкость ПО. Вероятнее всего, этот вывод будет справедлив для большинства IT-проектов.

Исследование зависимости метрик может быть полезно при расчете затрат на какой-либо разрабатываемый IT-проект. Если известны данные одной метрики, то можно по ним приблизительно рассчитать данные для других метрик, зная их зависимость.

Список литературы

1. Дроботун Е.Б. Надежность программного обеспечения. Виды критических ошибок. // Информационно-вычислительные технологии в науке. Секция 1. 2009 [Электронный ресурс]. – URL: http://www.ivtn.ru/2009/confs/enter/paper.php?p=1173 (дата обращения: 22.05.2017).

2. Евдокимов И.В. Информационные технологии учета методического обеспечения образовательного процесса // Проблемы социально-экономического развития Сибири. – 2012. – № 4 (10). – С. 9–14.

3. Евдокимов И.В., Боярчук Н.Я. Особенности стратегического планирования развития регионов Севера Восточной Сибири // сб.: Ценности и интересы современного общества: мат-лы Междунар. науч.-практич. конф. 2013. – С. 268–271.

4. Романов В.Ю. Анализ объектно-ориентированных метрик для проектирования архитектуры программного обеспечения // International Journal of Open Information Technologies. – 2014. – № 3. – С. 11–17.

5. Брукс Питер. Метрики для управления ИТ-услугами (Metrics for IT Service Management). Серия: Библиотека IBS, переводчик В. Первушина. – М.: Издательство «Альпина Паблишер», 2008. – 288 с.

6. Vu Nguyen, Sophia Deeds-Rubin, Thomas Tan, Barry Boehm. A SLOC Counting Standard, Center for Systems and Software Engineering, University of Southern California. – 2007. – 15 p.

7. Miller R.W. Schedule, Cost, and Profit Control with PERT – A Comprehensive Guide for Program Management // McGraw-Hill. – 1963. – 227 p.

8. Engineering Function Points and Tracking System, Software Technology Support Center, Retrieved on May 14, 2008. – 159 p.

9. Barry Boehm, et al. «Software cost estimation with COCOMO II». Englewood Cliffs, NJ:Prentice-Hall, 2000. – 544 p.

10. The world's leading software development platform GitHub. Main page [Электронный ресурс]. – URL: https://github.com/ (дата обращения: 19.04.2017).

11. Архипенков С. Лекции по управлению программными проектами [Электронный ресурс]. – URL: http://www.arkhipenkov.ru/resources/sw_project_management.pdf (дата обращения: 13.06.2017).

12. Евдокимов И.В. Менеджмент качества и управление развитием системы обработки экспертной аналитики // Труды Братского государственного университета. Серия: Экономика и управление. – 2015. – Т. 1. – С. 212–219.

13. Function Point Counting Practices Manual, Release 4.3, IFPUG, 2009. – 150 p.

14. Евдокимов И.В., Макеев В.В., Кокташев В.В. Экономическое обоснование эффективности ИТ-проектов в регионе Крайнего Севера на основе метода Function Points // Международный журнал гуманитарных и естественных наук. – 2017. – Т. 2. № 3. – С. 141–146.

15. Periyasamy and Aditi Ghode Department of Computer Science University of Wisconsin-La Crosse La Crosse, WI 54601 Cost Estimation using extended Use Case Point (e-UCP) Model, IEEE 2009. – 11 p.

16. Садовский И.Д. Применение модели COCOMO II для оценки разработки Программного обеспечения в Windows проектах // Экономика и бизнес: теория и практика. – 2016. – № 10. – С. 102–106.