УДК 004.023

МЕТОДИКА КОЛИЧЕСТВЕННО-КАЧЕСТВЕННОЙ ОЦЕНКИ ТЕСТОПРИГОДНОСТИ ПРОГРАММНЫХ СИСТЕМ НА РАННИХ ЭТАПАХ РАЗРАБОТКИ

З. Ф. Камальдинова, О. А. Затеева

TECHNOLOGY OF QUANTITATIVE AND QUALITATIVE ASSESSMENT OF SOFTWARE TESTABILITY IN THE EARLY STAGES OF DEVELOPMENT

Z. F. Kamaldinova, O. A. Zateeva

Аннотация. *Предмет*. Изложена проблема оценки тестопригодности требований, предъявляемых к разрабатываемому программному продукту на ранних этапах разработки проекта. Рассматривается возможность повышения качества формулировки требований за счет анализа степени их покрытия тестами. *Методы*. Предложена методика, предусматривающая разработку предварительного тест-плана и анализ покрытия тестами при формировании требований и оценке их сложности, важности, неопределенности и трудоемкости. *Результаты*. Предложенная методика была апробирована на данных реального проекта по созданию автоматизированной информационной системы. Описаны преимущества предложенной методики и область ее применения, представлены результаты ее реализации для анализа последствий недооценки проекта и недоработки требований на этапе их формирования. *Выводы.* Проведение количественно-качественной оценки тестопригодности программных систем на ранних этапах разработки позволяет снизить количество ошибок и сократить время на отладку на более поздних стадиях.

Ключевые слова: информационные технологии, оценка проекта, тестопригодность, требования, трудоемкость, стоимость проекта.

Abstract. *Background*. The paper raises the problem of assessing the testability of the requirements for the developed software product in the early stages of project development. *Materials and methods*. The article describes the methodology used to work out the requirements in the early stages of developing an IT project, namely, at the stage of estimating the cost and complexity of the project. The experiments of the methodology are based on the experience of working on projects, testing experiences and conducting initial assessment. *Results*. The article describes the advantages of the proposed methodology and its scope, presents an example of the use of this methodology, and formulates the consequences of underestimating the project and the deficiencies in the requirements at the stage of their formation. *Conclusions*. The use of the proposed methodology will further greatly simplify the process of development and testing, by saving time to clarify requirements.

Key words: information technologies, project evaluation, testability, requirements, labor intensity, project cost.

Введение

Формирование требований и предварительная оценка стоимости IT-проекта является одним из наиболее важных видов деятельности руководителя разработкой программных систем. При отсутствии адекватной и достоверной оценки требований невозможно обеспечить четкое планирование и управление проектом в дальнейшем [1–3].

Недооценка времени и затрат, необходимых для разработки продукта, приводит к недостаточной численности команды, работающей над проектом, коротким срокам, отведенным на разработку, и возможным штрафам для компании разработчика в случае нарушения установленного графика. С другой стороны, переоценка времени и затрат тоже имеет плохие последствия. Если для проекта выделено больше ресурсов, чем необходимо, и использование этих ресурсов не контролируется, то такой проект окажется более дорогостоящим, чем должен был быть при правильной оценке.

Однако наряду с недооцененностью проекта есть и другая проблема, связанная с тестопригодностью требований, предъявляемых к системе. Эта проблема возникает на этапе формирования требований, а выявляется часто только на этапе тестирования. В результате происходит застой в тестировании в ожидании уточнения и переформулировки требований заказчиком. Эта проблема встает особенно остро при реализации гибкого подхода к разработке программного обеспечения «Agile», код тестирования в жизненном цикле разработки программного обеспечения начинает играть одну из ключевых ролей.

Предлагается методика по реализации ранней оценки тестопригодности в процессе практической разработки программного обеспечения.

Материалы и методика

Agile (*agile software development*) – это семейство «гибких» подходов к разработке программного обеспечения, которая предполагает, что при реализации проекта не нужно опираться только на заранее созданные подробные планы. Важно ориентироваться на постоянно меняющиеся условия внешней и внутренней среды и учитывать обратную связь от заказчиков и пользователей. Это поощряет разработчиков и инженеров экспериментировать и искать новые решения, не ограничивая себя жесткими рамками и стандартами [4].

Большинство гибких методологий используют в разработке серии коротких циклов, называемых итерациями. Каждая итерация сама по себе выглядит как программный проект в миниатюре: планирование, анализ требований, проектирование, программирование, тестирование и документирование. При использовании подхода Agile определение тестопригодности необходимо проводить на каждой итерации разработки программного продукта.

Тестопригодность в тестировании – это степень пригодности формального требования для создания теста (с последующим документированием в виде тест-кейса) и для последующего выполнения теста с целью определения факта реализации данного требования [5]. Обычно под тестопригодностью понимают оценку требований с позиции тестирования и так называемого test driven development (TDD). TDD является передовой техникой разработки программного обеспечения, основывающейся на повторении кратких циклов разработки: изначально пишется тест, который покрывает желаемое изменение, а далее пишется сам код, позволяющий пройти данный тест, и в конце проводится рефакторинг нового кода к соответствующим стандартам [6].

Однако все эти методы носят формальный характер и относятся к организации процесса разработки, мы же попробуем применить эти модели и методы к сбору требований на самых ранних этапах разработки программного обеспечения. При этом перспективным является применение современных технологий адаптивного управления ресурсами с учетом человеческого фактора [7–9].

На основе существующей практики ранней оценки требований к программному обеспечению можно выделить три методики, которые могут применяться на начальных этапах жизненного цикла IT-проекта.

Первая методика основана на оценке требований по трем показателям: сложность, неопределенность и важность. Назовем ее *CUV* (*complexity*, *uncertainty, value*) моделью требований. Указанные критерии для более удобного представления сводятся в обобщенную таблицу.

Отличие предложенной методики от TDD заключается в том, что предварительный тест-план (и соответствующие тест-кейсы) не является основой для дальнейшей разработки, а служит лишь как вспомогательный инструмент для уточнения формулировки требований. Планирование и управление проектом производится классически на основе согласованного с заказчиком технического задания, в соответствии с принятыми стандартами разработки и ожиданиями заказчика. При этом реализация раннего планирования процедуры тестирования и контроля качества программного обеспечения позволяет добиться преимущества по сравнению с классическим подходом.

Модель CUV составляется представителями команды разработчиков совместно с заказчиком для полного покрытия и оценки всех функциональных требований к системе. В первом столбце таблицы располагается «Раздел» системы, к которому относятся функциональные требования (например, требование «Скрытие пароля звездочками» относится к разделу «Авторизация»). Разделом может быть, например, интерфейс, база данных, интеграция и т.д. Во второй столбец таблицы заносят «Требования». К требованиям относятся те функции, которые нужно реализовать в системе для удовлетворения потребностей заказчика.

Затем проводится качественная оценка требований. Неопределенность выражает степень непонимания того, как нужно реализовать тот или иной функционал. Высокая оценка неопределенности может говорить о том, что заказчик недостаточно подробно и ясно описал свои требования к реализуемым функциям системы. Сложность – это показатель того, насколько тяжело будет реализовать какое-либо требование (возможно, очень сложный алгоритм или просто очень большой объем работ). Важность – это оценка того, насколько нужна и важна рассматриваемая функция в системе в целом.

Для оценки качественных показателей используют три уровня области значений: *L* (Low) – низкий, *A* (Average) – средний, *H* (High) – высокий. Оценку производят совместно заказчик и исполнитель, при этом каждый из них оценивает свои показатели. Важность оценивается заказчиком, так как он расставляет приоритеты разработки нового функционала для достижения своих целей, а сложность и неопределенность оцениваются исполнителем с точки зрения разработки программного продукта.

Пример оценки требований для типовой системы контроля успеваемости в образовательном учреждении среднего образования представлен в табл. 1.

Таблица 1

Оценка требований

Раздел	Требования	*C*	*U*	*V*
Интерфейс	Вывод списка предметов для учащегося	*L*	*A*	*H*
	Вывод текущей успеваемости	*L*	*L*	*H*

Если в оценке одного требования по модели CUV выявляются хотя бы два показателя *H* (High), это означает необходимость более глубокой проработки этого требования: изменения формулировки, детализации или декомпозиции.

Вторая методика раннего анализа требований к программному обеспечению состоит в проведении экспертной оценки трудоемкости (в человеко-месяцах, неделях или днях). Эта оценка проводится командой разработки на основе изучения и анализа временных и человеческих ресурсов, необходимых для успешной разработки системы в срок с учетом критических путей. Пример оценки трудоемкости представлен в табл. 2.

Таблица 2

Оценка трудоемкости

Раздел	Требования	Трудоемкость (*T*), чел.-дней
Интерфейс	Вывод списка предметов для учащегося	3
	Вывод текущей успеваемости	2

На основе оценок трудоемкости рассчитывают общее время, требуемое для выполнения проекта. При этом добавляют дополнительное время на проверку реализованных функций отделом тестирования (QA), менеджмент и возможные непредвиденные обстоятельства.

Третья методика основана на комбинации первых двух с дополнительным анализом тестопригодности (CUVQA), предлагаемая впервые в данной статье. Суть предложения состоит в составлении предварительного тест-плана и соотнесении его пунктов с соответствующими оценками CUV и трудоемкости для анализа степени возможного покрытия тестами. Данная методика основана на обобщении результатов организации и проведения тестирования программного обеспечения авторами на практике и предназначена для повышения эффективности оценки трудоемкости IT-проектов на ранних стадиях разработки.

Зачастую на практике, на этапе тестирования, выявляется, что требования из-за недостатка или неточности формулировки, слишком обширного характера формулировки или, наоборот, излишней конкретизации очень сложно проверить. Поэтому предлагается в момент, когда эти требования формулируются, создавать некий прототип тест-плана и относительно него проводить анализ покрытия требований.

Тест-план представляет собой документ, описывающий весь объем работ по тестированию, начиная с описания тестируемых объектов, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения [10–12].

После составления тест-плана необходимо составить таблицу соответствия требованиям (здесь ограничимся только списком функций, обязательных к тестированию, табл. 3). В таблице приведены указания на пункты тест-плана без детального описания ожидаемого и фактического результатов.

Таблица 3

Пример сопоставления требований к тест-плану

Раздел	Требования	Тест-план
Интерфейс	Вывод списка предметов для учащегося	Проверка дневника
	Вывод текущей успеваемости	Проверка дневника

Таким образом, проверяется, все ли требования могут быть протестированы. Если какому-то требованию нет соответствующего пункта тест-плана, то это требование считается критическим. Такое требование нужно уточнять, делить на несколько других требований (декомпозировать) или, наоборот, обобщать формулировки.

Проведение комплексной совместной оценки сложности, важности, неопределенности, трудоемкости и тестопригодности требований при формировании их перечня позволяет повысить адекватность и достоверность планирования IT-проекта на ранних стадиях разработки программного обеспечения.

Результаты

Предложенная методика CUVQA может быть использована руководителями проектов и аналитиками вручную, а также реализована в подсистеме автоматизированного анализа требований в составе системы планирования проектов или менеджера задач (task tracker).

Для анализа результативности этой методики было проведено исследование результатов ее применения в проекте одной из Самарских IT-компаний на крупном государственном проекте. На основе данных о возникающих ошибках и их устранению (баг-репортов) была получена следующая статистика по заведенным ошибкам за последние месяцы работы (рис. 1).

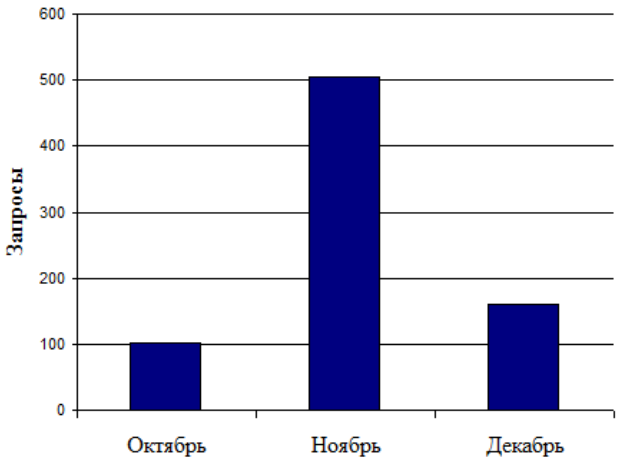

Рис. 1. Статистика по заведенным ошибкам за прошедшие месяцы

При проведении анализа ошибок рассматривался декабрь 2018 г., так как именно в этом месяце тестировался большой блок нового функционала. За месяц тестирования была обнаружена 161 ошибка, из которых 124 ошибки

было заведено по новому функционалу. Проведя анализ поставленных задач, было выявлено, что 27 ошибок возникли из-за некорректной формулировки требований к разрабатываемой системе.

В соответствии с методикой CUVQA были проработаны критические требования, при тестировании которых выявлялись ошибки. Более детальный анализ выявил, что некоторые требования, сформулированные на ранних этапах проекта и связанные со сложным отображением результатов (таблицы, списки, группы), потребовали уточнения в процессе тестирования, и как следствие, переработки пользовательского интерфейса системы и логики работы приложения. Например, возникли трудности с отображением разных элементов категорий в одном списке (строки «иностранный язык», «английский язык», французский язык»), что вызвало дублирование.

Анализ статистики установил, что использование предложенной методики позволило бы снизить количество такого рода ошибок, связанных с некорректной формулировкой требований, на 81,5 %. Все данные, представленные в исследовании, были сведены в табл. 4.

Таблица 4

Анализ ошибок до и после применения методики

Требование	Число ошибок, заведенных из-за некорректной формулировки требования (без CUVQA)	Число ошибок, прогнозируемых после использования CUVQA
Реализация нового экрана чатов для пользователей	10	2
Реализация нового экрана рабочих программ школ	11	2
Реализация нового экрана управления производственным календарем	4	0
Реализация нового экрана настройки требований к паролям	2	1
Общее число ошибок	27	5

Обсуждение

Предложенная методика CUVQA позволяет сформировать полную, адекватную и достоверную оценку требований, предъявляемых к разрабатываемому программному продукту. Во время оценки тестопригодности происходят проработка и уточнение требований, что очень помогает отделу тестирования во время последующей проверки продукта на соответствие требованиям. Также уточнение требований позволяет повысить адекватность оценки трудоемкости IT-проекта, и, как следствие, его стоимости.

Сочетание количественных и качественных оценок требований, а также распределение работ по подготовке проекта между участниками разной специализации (представителей как исполнителя, так и заказчика) позволяет снизить

неопределенность и влияние человеческого фактора и повысить объективность планирования проекта с учетом особенностей используемых ресурсов.

Отличие предложенной методики от широко распространенных методологий Agile и TDD заключается в том, что в ней сохраняются преимущества обоих подходов: техническое задание представляется для заказчика в традиционной форме перечня требований, однако при их формулировке учитываются последующие сложности тестирования и риски выявления рассогласований трактовки. При этом она не требует кардинального перестроения жизненного цикла IT-проекта.

Проведенный эксперимент показал хорошие результаты применения методики CUVQA на практике. Ее внедрение позволяет прорабатывать требования к системе на этапе оценки стоимости и сроков разработки проекта. В дальнейшем это значительно упрощает процесс разработки и тестирования за счет экономии времени на уточнение требований.

Выводы

Предложена методика, предусматривающая разработку предварительного тест-плана и анализ покрытия тестами при формировании требований и оценке их сложности, важности, неопределенности и трудоемкости. Проведение количественно-качественной оценки тестопригодности программных систем на ранних этапах разработки позволяет снизить количество ошибок и сократить время на отладку на более поздних стадиях.

Библиографический список

1. *Бебрыш, Т. Н.* Оценка экономической эффективности разработки программного продукта : метод. указания к выполнению экономического раздела дипломного проекта / Т. Н. Бебрыш. – Абакан : Сиб. федер. ун-т., 2009. – 34 с.
2. *Dejaeger, K.* Data Mining Techniques for Software Effort Estimation: A Comparative Study / K. Dejaeger, W. Verbeke, D. Martens, B. Baesebs // IEEE Transactions on Software Engineering. – 2012. – Vol. 38, iss. 2. – P. 375–397. – URL: http://www.computer.org/csdl/trans/ts/2012/02/tts2012020375-abs.html (дата обращения: 10.11.2018).
3. *Jargensen, M.* A Systematic Review of Software Development Cost Estimation Studies / M. Jargensen, M. Shepperd // IEEE transactions on software engineering. – 2007. – Vol. 33, № 1.
4. *Петров А.* Agile, scrum, kanban: в чем разница и для чего использовать? / А. Петров. – URL: https://rb.ru/story/agile-scrum-kanban/ (дата обращения: 03.12.2018).
5. ISTQB Certification Foundation Level Terms 7 // Software Testing And Programming. – URL: http://www.articlesontesting.com/2011/10/istqb-certification-foundation-level_1551.html (дата обращения: 03.12.2018).
6. Тест знаний Test-driven development (TDD) // Quizful. – URL: http://www.quizful.net/test/test-driven-development (дата обращения: 05.12.2018).
7. *Ivaschenko, A.* Time-based regulation of auctions in P2P outsourcing / A. Ivaschenko, A. Lednev // Proceedings of the 2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT) (17–20 November). – Atlanta, Georgia, USA, 2013. – P. 75–79
8. *Ivaschenko, A.* Multi-agent solution for business processes management of 5PL transportation provider / A. Ivaschenko // Lecture Notes in Business Information Processing. – N.Y. : Springer International Publishing, 2014. – Vol. 170. – P 110–120.
9. *Иващенко, А. В.* Управление взаимодействием персонала предприятия в многоакторной интегрированной информационной среде / А. В. Иващенко // Программные продукты и системы. – 2012. – № 3. – С. 18–22.

10. QALight. – URL : https://qalight.com.ua/baza-znaniy/test-plan/ (дата обращения: 04.12.2018).
11. *Лапыгин, Ю. Н.* Экономическое позиционирование : учеб. пособие / Ю. Н. Лапыгин, В. Е. Крылов, А. П. Чернявский. – М. : Эксмо, 2009. – 256 с.
12. *Зырянов, М.* Риск-менеджмент в ИТ-службе / М. Зырянов. – URL: https://www.osp.ru/cio/2014/09/13042946/ (дата обращения: 06.12.2018).

References

1. Bebrysh T. N. *Otsenka ekonomicheskoy effektivnosti razrabotki programmnogo produkta: metod. ukazaniya k vypolneniyu ekonomicheskogo razdela diplomnogo proekta* [Evaluation of economic efficiency of software development: method. instructions for the implementation of the economic section of the diploma project]. Abakan: Sib. feder. un-t., 2009, 34 p.
2. Dejaeger K., Verbeke W., Martens D., Baesebs B. *IEEE Transactions on Software Engineering*. 2012, vol. 38, iss. 2, pp. 375–397. Available at: http://www.computer.org/csdl/trans/ts/2012/02/tts2012020375-abs.html (accessed Nov. 10, 2018).
3. Jargensen M., Shepperd M. *IEEE transactions on software engineering*. 2007, vol. 33, no. 1.
4. Petrov A. *Agile, scrum, kanban: v chem raznitsa i dlya chego ispol'zovat'?* [Agile, scrum, kanban: what is the difference and what to use?]. Available at: https://rb.ru/story/agile-scrum-kanban/ (accessed Dec. 03, 2018).
5. *Software Testing And Programming*. Available at: http://www.articlesontesting.com/2011/10/istqb-certification-foundation-level_1551.html (accessed Dec. 03, 2018).
6. *Quizful*. Available at: http://www.quizful.net/test/test-driven-development (accessed Dec. 05, 2018).
7. Ivaschenko A., Lednev A. *Proceedings of the 2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT) (17–20 November)*. Atlanta, Georgia, USA, 2013, pp. 75–79
8. Ivaschenko A. *Lecture Notes in Business Information Processing*. New York: Springer International Publishing, 2014, vol. 170, pp 110–120.
9. Ivashchenko A. V. *Programmnye produkty i sistemy* [Software and systems]. 2012, no. 3, pp. 18–22.
10. *QALight*. Available at: https://qalight.com.ua/baza-znaniy/test-plan/ (accessed Dec. 04, 2018).
11. Lapygin Yu. N., Krylov V. E., Chernyavskiy A. P. *Ekonomicheskoe pozitsionirovanie: ucheb. posobie* [Economic positioning: tutorial]. Moscow: Eksmo, 2009, 256 p.
12. Zyryanov M. *Risk-menedzhment v IT-sluzhbe* [Risk management in it service]. Available at: https://www.osp.ru/cio/2014/09/13042946/ (accessed Dec. 06, 2018).

Камальдинова Зульфия Фаисовна
кандидат технических наук,
и.о. заведующего межвузовской
кафедрой информационных
и развивающих образовательных систем
и технологий,
Самарский государственный
технический университет
(Россия, г. Самара,
ул. Молодогвардейская, 244)
E-mail: kamal_zzz@mail.ru

Kamal'dinova Zul'fiya Faisovna
candidate of technical sciences,
acting head of sub-department
of information and developing
educational systems and technologies,
Samara State Technical University
(244 Molodogvardeyskaya street,
Samara, Russia)

Затеева Ольга Александровна
студентка,
Самарский государственный
технический университет
(Россия, г. Самара,
ул. Молодогвардейская, 244)
E-mail: oborisova95@mail.ru

Zateeva Ol'ga Aleksandrovna
student,
Samara State Technical University
(244 Molodogvardeyskaya street,
Samara, Russia)

УДК 004.023
Камальдинова, З. Ф.
Методика количественно-качественной оценки тестопригодности программных систем на ранних этапах разработки / З. Ф. Камальдинова, О. А. Затеева // Модели, системы, сети в экономике, технике, природе и обществе. – 2018. – № 4 (28). – С. 134–142.