Научная статья

УДК 004.415.2

ББК 32.972

К 11

DOI: 10.53598/2410-3225-2021-3-286-73-79

К вопросу о методе исследования пользовательских требований к программному обеспечению (Рецензирована)

Юлия Юрьевна Липко¹, Джульетта Абугалиевна Крымшокалова²,
Залина Асланбековна Шогенова³, Джабар Аскерович Лигидов⁴

¹ Южный федеральный университет, Таганрог, Россия.

^{2, 3, 4} Кабардино-Балкарский государственный университет им. Х.М. Бербекова,
Нальчик, Россия.

¹ jullipko@gmail.com

² dzhuletta.krym@bk.ru

³ shogenova.88@mail.ru

⁴ gzig999@gmail.com

Аннотация. Методы User Story все чаще используются в качестве основы артефактов проектирования требований при разработке программного обеспечения. На практике доказано, что метод User Story является более эффективным для описания основных целей системы. Но непрерывное управление работой программного обеспечения может быть особенно трудоемким и подверженным ошибкам, особенно при оценке качества или объема пользовательских историй и наблюдении за общей картиной системы. С другой стороны, эти модели были признаны эффективными инструментами коммуникации и анализа цели. В рамках данной работы рассмотрены и проанализированы методы выявления и представления требований к разработке программного обеспечения. В статье предлагается генеративный подход для создания диаграмм надежности на основе автоматизированного анализа пользовательских историй. Истории преобразуются в диаграммы, что позволяет разработчикам требований и пользователям проверять основные концепции и функциональные этапы, лежащие в основе историй, и обнаруживать искаженные или избыточные истории. Такие модели также открывают двери для автоматизированного систематического анализа.

Ключевые слова: пользовательские требования, информационные системы, программное обеспечение, жизненный цикл, методы User Story, пользовательские истории

Original Research Paper

On the question of a method for researching user requirements for software

Yuliya Yu. Lipko¹, Dzhulyetta A. Krymshokalova²,
Zalina A. Shogenova³, Dzhabar A. Ligidov⁴

¹ Southern Federal University, Taganrog, Russia.

^{2, 3, 4} Kabardino-Balkarian State University named after H.M. Berbekova, Nalchik,
Russia.

¹ jullipko@gmail.com

² dzhuletta.krym@bk.ru

³ shogenova.88@mail.ru

⁴ gzig999@gmail.com

Abstract. *The User Story methods (user stories) increasingly are used as the basis of requirements design artifacts in software development. In practice, it is proved that the User Story method is more effective for describing the main goals of the system. But continuous management of software operation can be particularly time-consuming and error-prone, especially when evaluating the quality or volume of user stories and observing the overall picture of the system. On the other hand, these models were recognized as effective tools for communication and goal analysis. Within the framework of this work, methods for identifying and presenting requirements for software development are considered and analyzed. In the article, we propose a generative approach for creating reliability diagrams based on automated analysis of user stories. Stories are converted into diagrams, which allow requirements developers and users to check the basic concepts and functional stages underlying the stories, and detect distorted or redundant stories. Such models also open the door for automated systematic analysis.*

Keywords: *user requirements, information systems, software, lifecycle, User Story methods, user stories*

Требования к программному обеспечению – это программная возможность, которой должна соответствовать система или системный компонент, или которыми она должна обладать, чтобы соответствовать контракту, стандарту, спецификации или другой официально установленной документации. Одна из самых главных проблем, с которой сталкиваются разработчики программного обеспечения, – это поделиться с заказчиком видением конечного продукта. Все заинтересованные стороны в проекте – разработчики, конечные пользователи, менеджеры программного обеспечения, менеджеры по работе с клиентами – должны достичь общего понимания того, каким будет продукт, и чем он будет заниматься, следовательно, нужны способы точно понимать клиентов при определении требований к программному продукту. Процесс современной разработки программного обеспечения нацелен на жизненный цикл проектирования информационных систем. На сегодняшний день существующие методики, технологии и стандарты так или иначе касаются и регламентируют этапы жизненного цикла как по функциональному наполнению, так и по содержанию. Процесс разработки программного обеспечения тесно связан с областью управления проектами, так как любой программный продукт является уникальным результатом.

Согласно исследованиям [1], после внедрения метод гибкой разработки программного обеспечения получил широкое распространение в различных отраслях, превзойдя классическую водопадную модель. Среди гибких методов разработки требований все чаще используются пользовательские истории (англ. User Story) [2], вовлекающие потенциальных пользователей в процесс выявления требований путем написания их потребностей на естественном языке [3]. Текстовые шаблоны были предложены для пользовательских историй в форме «Как <тип пользователя> Я хочу <какой-то цели>, чтобы <по какой-то причине>». Истории пользователей были определены как эффективные при разработке «правильного программного обеспечения», поскольку они требуют детальной функциональной декомпозиции и повышают воспринимаемую производительность [4]. Однако слабо определенные истории могут вызвать различные проблемы при разработке гибкого программного обеспечения, такие как недостаточно документированные требования или игнорируемые нефункциональные требования, постоянная переориентация требований или потеря общей картины.

Работа в основном выполняется вручную, что может занять много времени и привести к ошибкам по мере увеличения количества историй [5]. Этот рост числа, как правило, приводит к размыванию концептуальных и технических знаний во всем отставании, так что могут потребоваться визуальные модели для привлечения заинтересованных сторон к выполнению задач по разработке требований [6]. В нескольких исследованиях изучалось распространение моделей UML, где их использование было особенно низким. Недавние исследования, посвященные более широкому определению

нию «концептуальных» моделей, показали более широкое распространение языков моделирования. Считаем, что диаграммы особенно полезны при общении между заинтересованными сторонами. Для этой конкретной цели диаграммы надежности были определены как эффективные визуальные инструменты для представления сценариев использования с точки зрения действующих лиц, границ, элементов управления и сущностей [7].

В последнее время широко изучались методы использования нейролингвистического программирования (НЛП) для нужд гибкой разработки программного обеспечения. Автор Казамайор провел большой обзор подходов к интеллектуальному анализу текста для разработки программного обеспечения [8, 9]. Большая часть работы на уровне требований была направлена на выявление, классификацию и оценку требований. В этом исследовании генерируются диаграммы надежности на основе автоматизированного анализа пользовательских историй с использованием естественного языка и предлагается сосредоточиться на создании моделей на основе текстовых описаний. Рассмотрим существующие подходы, которые анализируют текстовые ресурсы для поддержки задач разработки требований или создания моделей на основе историй.

Истории пользователей (User Story) – способ описания требований к разрабатываемой системе, сформулированных как одно или несколько предложений на повседневном или деловом языке пользователя. Главное действующее лицо User Story – это некий персонаж, который будет совершать какие-либо действия с тестируемым продуктом с учетом его потребностей. Персонаж сопровождается описанием проблем, которые он может решить с помощью программного продукта. Потребность представляет собой тезис в 1–2 предложения. Для одного пользователя может быть разработано несколько User Story.

В научно-образовательном журнале “The Standish Group” [10] отмечается, что из-за низкого качества требований к программным средствам, в особенности на ранних стадиях их жизненного цикла, возрастает количество критических факторов, которые приводят к провалу разрабатываемых проектов, и качество программных продуктов не соответствует потребностям пользователя. Процесс разработки программного обеспечения всегда имеет свои особенности и тонкости. Зная о них и правильно определив целевую аудиторию и ее потребности, специалист может провести проверку быстрее и качественнее.

Истории пользователей становятся все более популярными в последнее время при разработке программного обеспечения, но при неправильном определении могут возникнуть значительные расходы на разработку из-за неполной или неверной документации [9]. До недавнего времени было мало уделено внимания оценке эффективности пользовательских историй. В качестве примера возьмем шаблон Кона, где можно определить историю типа «*Как пользователь, я хочу иметь возможность создавать и редактировать профиль, чтобы меня узнавал веб-сайт*». В этом исследовании полагаемся на этот шаблон с дополнительной частью *so that* для создания моделей.

Диаграммы надежности были введены Розенбергом и Стивенсом [10] для выражения упрощенного сотрудничества. Подобные диаграммы устраняют разрыв между вариантами использования и подробными эксплуатационными спецификациями. Этот тип диаграмм повторно использует набор стандартных конструкций и стереотипов UML, изменяя их семантику. Они опираются на четыре основные конструкции, а именно: Субъект (то есть пользователь); Граница (то есть Интерфейс между пользователем и системой); Сущность (то есть объект домена); Управление (то есть все, что находится между Границами и Сущностями). Диаграммы надежности также должны соответствовать набору простых правил, поскольку субъект может быть связан только

с границами, сущности могут быть связаны только с элементами управления и элементы управления могут быть связаны с чем угодно, кроме субъектов.

Считаем, что подход при использовании метода пользовательских историй стоит изучить, поскольку использование диаграмм надежности для проверки достоверности пользовательских историй имеет множество преимуществ. Во-первых, эти диаграммы предназначены для преодоления разрыва между анализом требований и детальным проектированием, когда бизнес-логика варианта использования или пользовательской истории может быть разложена на более простые процессы. Во-вторых, определяются и моделируются основные точки взаимодействия с пользователями. В-третьих, концепции домена также точно определены и могут быть сопоставлены с доменом приложения. Другими словами, создавая модели на основе пользовательских историй, можно выполнять структурированный анализ, например, проверка соответствия с другими артефактами моделирования, выявление проблем с качеством в пользовательских историях или расхождений в словаре.

Пользовательские требования описывают требования непосредственных пользователей программы. Чаще всего пользовательские требования описываются с помощью вариантов использования. Варианты использования могут быть описаны как в текстовом виде, так и с помощью языка UML. Пользовательские требования могут быть собраны в ходе анализа бизнес-процессов организации, интервью с заказчиком, в ходе проведения круглых столов, демонстрации предварительной версии программного обеспечения и т.д. При организации общения с заказчиком правильным будет придерживаться нескольких простых правил:

- четко формулировать цель встречи и обсуждаемые вопросы;
- заранее определить состав участников встречи, оповестить их о месте, времени и продолжительности встречи;
- документировать принятые в ходе встречи решения. Пользовательские требования подчинены бизнес-требованиям, а значит, каждое из них должно иметь как минимум одно связанное с ним бизнес-требование.

Когда речь заходит о требованиях и управлении требованиями в разработке программного обеспечения, то сразу же встает вопрос о листах спецификации – документах, в которых записываются функции, требуемые производителем программного обеспечения или поставщиком, а затем проверяются и удаляются. Однако может возникнуть ситуация, когда требования программного обеспечения изменятся во время разработки, и команде разработчиков нужно адаптироваться и соответственно должны быть по-другому распределены задачи между ними. Именно по этой причине в последние годы были разработаны гибкие методы управления проектами, и поэтому они работают с другими вспомогательными средствами.

На рисунке 1 приведен пример Scrum-проекта с отставанием. Scrum – наиболее популярная Agile-методология в области программного обеспечения. Используя данный метод, большинство компаний улучшили качество результатов и продуктивность. Scrum – это Agile-метод разработки, который определяется как итеративный, инкрементный и эмпирический процесс управления проектом и контроля его выполнения.

Отставание продукта (доменное отставание) является ядром Scrum-проекта, потому что он собирает требования, которые должно выполнить программное обеспечение. Это могут быть функции или ошибки, например, которые команда должна реализовать или исправить. Обычно для описания этих требований используются так называемые пользовательские истории. Пользовательские истории – это один из самых быстрых способов документирования требований клиента (цель документирования состоит в том, чтобы оперативно и без затрат реагировать на возникающие изменения). Для одного пользователя может быть разработано несколько User Story.

Рис. 1. Scrum-проект с отставанием

Fig. 1. Backlog scrum project

Анализируя некоторый набор целей пользователей по отношению к разрабатываемому продукту и разбивая их, например, по интересам, по ожиданиям от продукта, по частоте использования и по уровню подготовки, появляются такие кластеры пользователей, которые можно использовать как роли в User Story.

Используем пользовательские истории на основе шаблонов в качестве входных данных, которые будут проанализированы с помощью библиотек для создания дерева зависимостей. Затем все дерево разбивается на помеченные слова, которые анализируются в соответствии с лингвистическими правилами для создания графа именования. Полученный граф обрабатывается для идентификации элементов с привязкой. Рассматриваются вероятности того, что два слова имеют друг с другом некоторую связь, которая затем преобразуется в набор объектов диаграммы надежности. Диаграммы надежности создаются с использованием библиотеки PlantUML с открытым исходным кодом. PlantUML – это набор инструментов Java для текстовых UML-диаграмм. Диаграммы задаются на удобочитаемом текстовом языке, и затем на основе этого описания создается визуальное представление.

Исходя из вышеизложенного при рассмотрении схем представления требований, можно сделать следующие выводы:

- UML-диаграмма прецедентов может использоваться для представления требований, наглядно показывая основные функциональные возможности взаимодействия заинтересованных лиц с проектируемой системой, без детализации их выполнения. Однако неподготовленному пользователю такая диаграмма покажется сложной и непонятной;
- User Story отлично подходит для разработки поведенческих сценариев тестирования, программы и методики приемочных испытаний;
- User Story позволяет указать нефункциональные требования с точным заданием значений нужных показателей, например, времени отклика на событие, наработки на отказ и пр.

Основные результаты и выводы

Использование пользовательских историй в качестве метода исследования начальных требований широко распространено в промышленных проектах в соответствии с принципами гибкой разработки программного обеспечения. За последние годы было предложено множество шаблонов, от неограниченно свободного формата до очень строгих. Тем не менее первоначальное предложение Кона «Как <тип пользователя> Я хочу <какую-то цель>, чтобы <какая-то причина>», вероятно, оставалась наиболее часто используемым шаблоном для историй. В данном исследовании были рассмотрены оценки качества пользовательских историй и создания формальной модели для выполнения структурированного анализа.

Приведенные диаграммы надежности предназначены для выражения и проверки поведенческих сценариев с точки зрения действующих лиц, границ, элементов управления и сущностей. Тогда сопоставление между историями пользователей и диаграммами надежности становится естественным, делая акцент на концепциях домена, пользовательских интерфейсах и пошаговых действиях. Благодаря обработке естественного языка (Методы НЛП), можно идентифицировать слабо определенные или неоднозначные истории, извлекая основные концепции и процессы, а также выделяя связи между всеми этими элементами и разными способами. В качестве среднесрочной цели планируется провести проверку подхода, вручну написав диаграммы надежности промышленного проекта и проверив точность между сгенерированными и написанными вручную диаграммами, и актуальность историй, которые были идентифицированы как плохо определенные нашим инструментом. На следующем этапе, в качестве эффективного средства коммуникации между заинтересованными сторонами на различных уровнях, разработчики и нетехнические пользователи будут участвовать в оценке полезности процесса генерации. Этот элемент можно рассматривать в качестве дополнительного ресурса для поддержки проектирования решений и непрерывной эволюции программного обеспечения.

Список литературы

1. Вигерс К. Разработка требований к программному обеспечению: пер. с англ. Москва: Русская редакция, 2004. 576 с.
2. Рогозов Ю.И., Свиридов А.С. Метод построения интегральной модели жизненного цикла информационной системы // Телекоммуникации. 2011. № 3. С. 8–17.
3. Магуайр М.К., Херст С.Дж. Редизайн страниц корпоративной Интернет-сети Столичной полицейской службы. RSEHF (ранее HUSAT), Университет Лафборо, 2001.
4. Use-case_20. URL: https://www.researchgate.net/publication/301704896_Use-case_20
5. Басс Дж.М. Артефакты и адаптация гибких методов в крупномасштабных программах разработки оффшорного программного обеспечения // Информационные и программные технологии. 2016. Т. 75. С. 1–16.
6. Дизайн визуальных обозначений 2.0: к понимаемым пользователем техническим обозначениям требований / П. Кэйр, Н. Генон, П. Хейманс, Д.Л. Муди // 21-я международная разработка требований IEEE. Конференция (RE). 2013. С. 115–124.
7. Петре М. Uml на практике: в трудах Междунар. конф. по разработке программного обеспечения. Сер. ICSE'13. Пискатауэй, Нью-Джерси: IEEE Press, 2013. С. 722–731.
8. Систематизация моделей жизненного цикла информационных систем в рамках схемы J. Zachman / Ю.И. Рогозов, С.А. Бутенков, А.С. Свиридов, Н.С. Горбань // Известия ЮФУ. Технические науки. 2008. № 1 (78). С. 68–72.
9. Магуайр М.С. Контекст использования в рамках деятельности по использованию // Международный журнал человеко-компьютерных исследований. 2001. № 55 (4). С. 453–484.

10. The Standish Group: научно-образовательный журнал. URL: <https://www.standishgroup.com>

11. Систематический обзор литературы по методам и задачам разработки гибких требований / И. Инаят, С.С. Салим, С. Марчак, М. Данева, С. Шамширбанд // Компьютеры в поведении человека. 2015. Т. 51. С. 915–929.

References

1. Wiegers K. Software requirements: transl. from English. Moscow: Russian Edition, 2004. 576 p.
2. Rogozov Yu.I., Sviridov A.S. Method of constructing an integral model of the information system life cycle // Telecommunications. 2011. No. 3. P. 8–17.
3. Maguire M.K., Hurst S.J. Redesign of the pages of the corporate Internet network of the Metropolitan Police Service. RSEHF (formerly HUSAT), Loughborough University, 2001.
4. Use-case_20. URL: https://www.researchgate.net/publication/301704896_Use-case_20
5. Bass J.M. Artefacts and agile method tailoring in large-scale offshore software development programmes // Information and Software Technology. 2016. Vol. 75. P. 1–16.
6. Visual notation design 2.0: towards user-comprehensible RE notations, RE 2013 / P. Caire, N. Genon, P. Heymans, D.L. Moody // Requirements Engineering, Proceedings of the 21st IEEE International Requirements Engineering Conference. 2013. P. 115–124.
7. Petre M. Uml in practice: in the proceedings of the International Conference on Software Development. Ser. ICSE'13. Piscataway, New Jersey: IEEE Press, 2013. P. 722–731.
8. Systematization of life cycle models of information systems within the framework of the J. Zachman scheme / Yu.I. Rogozov, S.A. Butenkov, A.S. Sviridov, N.S. Gorban // News of SFU. Technical Sciences. 2008. No. 1 (78). P. 68–72.
9. Maguire M.S. Context of use within usability activities // International Journal of Human-Computer Studies. 2001. No. 55 (4). P. 453–484.
10. The Standish Group: Scientific and Educational Journal. URL: <https://www.standishgroup.com>
11. A systematic literature review on agile requirements engineering practices and challenges / I. Inayat, S.S. Salim, S. Marczak, M. Daneva, S. Shamshirband // Computers in Human Behavior. 2015. Vol. 51. P. 915–929.

Авторы заявляют об отсутствии конфликта интересов.

The authors declare no conflicts of interests.

Статья поступила в редакцию 28.07.2021; одобрена после рецензирования 28.08.2021; принята к публикации 29.08.2021.

The article was submitted 28.07.2021; approved after reviewing 28.08.2021; accepted for publication 29.08.2021.

© Ю.Ю. Липко, Д.А. Крымшохалова, З.А. Шогенова, Д.А. Лигидов, 2021