

Информатика

УДК 004.021

И.Э. ТРОФИМОВА

(irina0223@yandex.ru)

Российский экономический университет им. Г.В. Плеханова

ВЫСОКОУРОВНЕВЫЕ ТРЕБОВАНИЯ И ИХ ДЕТАЛИЗАЦИЯ ДЛЯ КОМАНДЫ РАЗРАБОТКИ*

Рассматривается подход к детализации высокоуровневых требований заказчика с применением методологии Dynamics Systems Development Method, а именно пояснение понятия требований и их классификация, подход к формированию Users Stories («пользовательские истории») и их влияние на разработку и формирование технико-экономического обоснования.

Ключевые слова: *требования, детализация требований, Dynamics Systems Development Method, User Stories, разработка.*

В рамках разработки данной статьи ставится цель проанализировать подход к детализации высокоуровневых требований с применением одной из современной методологии, а также выбрать элементы методологии, которые будут использоваться автором при разработке программного продукта.

При разработке программного обеспечения важность хорошо понимаемого, приоритетного и согласованного набора требований очевидна. Однако попытка определить полный и детальный набор требований на старте разработки часто оказывается контрпродуктивно. Невозможно определить все подробные требования в самом начале долгого проекта. С течением времени бизнес-среда меняется; появляются новые требования и возможности. По мере продвижения проекта команда больше понимает потребности бизнеса. Слишком раннее определение подробных требований означает либо необходимость изменения спецификации позднее, что приводит к потере первоначальной работы, либо достижение изначально определенных требований и, как следствие, неспособность адекватно удовлетворить бизнес-потребность [1].

Для разработки таких систем предлагается взять за основу DSDM (Dynamic Systems Development Method), дополнить его элементами других методологий. В данной статье будет рассмотрен подход DSDM и элементы, которые используются автором. DSDM советует фиксировать требования на высоком уровне, на ранней стадии проекта. Постепенно по мере продвижения проекта появляются новые детали, которые намеренно уходят в разработку как можно позже.

Для дальнейшего рассмотрения методики формирования требований необходимо определить, что такое «Требование».

Самое простое определение требования – это услуга, функционал или функция, в которой нуждается пользователь. Требования могут быть функциями, ограничениями, бизнес-правилами или другими элементами, которые должны присутствовать для удовлетворения потребностей предполагаемых пользователей.

Проекты DSDM направлены на то, чтобы сформулировать требования таким образом, чтобы как можно дольше не привязывать их к конкретному решению [2].

Это связано с тем, что можно сохранить большую гибкость в том, каким образом в конечном итоге будет обеспечиваться решение, если требования будут выражаться как то, что должно быть достигнуто, а не как они будут удовлетворяться с технической точки зрения.

* Работа выполнена под руководством Попова А.А., кандидата технических наук, доцента кафедры информатики РЭУ им. Г.В. Плеханова.

Слишком рано выраженное решение может стать ограничивающим фактором в отношении того, что может быть достигнуто в установленные сроки и в рамках бюджета.

Категории потребностей

Успех любого решения является результатом двух аспектов [5] (рис. 1):

- что оно делает (функциональность, особенности);
- насколько хорошо он работает по заданным параметрам (нефункциональные атрибуты, критерии приемки, уровень сервиса).



Рис. 1. Категории требований

Функциональные требования

Функциональные требования выражают функционал или функцию и определяют то, что требуется, например:

- посетить сайт клиента;
- обеспечить удобство работы пользователя с программным приложением [6];
- получить место проведения конференции.

В требованиях не указано, каким образом решение будет физически достигнуто [3].

Нефункциональные требования

Нефункциональные требования определяют, насколько хорошо или на каком уровне должно функционировать решение. Они описывают такие атрибуты решения, как безопасность, надежность, доступность, отказоустойчивость, производительность и время отклика.

User stories («пользовательские истории», далее – US)

Пользовательская история – это требование, выраженное с точки зрения целей конечного пользователя. Пользовательские истории также могут называться «эпиками» (“Epics”), «темами» (“themes”) или «функциями», но все они имеют один и тот же формат.

Пользовательская история – это хорошо выраженное требование. Формат US стал самым популярным способом выражения требований в гибких методологиях разработки по ряду причин:

- US фокусируется на точке зрения роли, которую будет использовать решение или на которую оно окажет влияние;
- US определяет требование в языке, имеющем значение для этой роли;
- помогает прояснить истинную причину требования;
- помогает определить требования высокого уровня, не обязательно слишком рано вдаваться в детали низкого уровня.

Цели пользователя определяются, и ценность каждого требования для бизнеса сразу же учитывается в истории пользователя.

Формат US

Формат пользовательской истории выглядит следующим образом:

- Как (директору по маркетингу, инвестору и т. п.).

- Мне необходимо (улучшить обслуживание клиентов, получить отчет о моих инвестициях и т. п.).
- Чтобы достигнуть результата (удержания наших клиентов, понять, где сосредоточить внимание и т. п.).

Формат пользовательских рассказов помогает обеспечить, чтобы каждое требование фиксировалось в функционально-ориентированном, ценностно-ориентированном виде, а не в виде решения [4].

Пользовательские истории часто печатаются на физических карточках, для целей планирования и для того, чтобы помочь команде по разработке решений следить за прогрессом (табл. 1).

Таблица 1

Карточка пользовательской истории

Карточка US	
Идентификатор истории	12345
Имя истории	Заказ клиента
Описание	Как клиент, я должен сделать заказ, чтобы еду привезли с доставкой на дом
Подтверждение:	Примеры критериев приемки: Функциональные: – Могу ли я сохранить заказ и вернуться к нему позже? Да, могу. – Могу ли я изменить свой заказ до того, как оплачу его? Да, могу. – Могу ли я увидеть текущую общую стоимость того, что я выбрал до сих пор? Да, могу. Функциональный: доступность: – Могу ли я сделать заказ в любое время (24 часа в сутки или 24/7/365)? Да, могу. – Могу ли я просмотреть заказ в любое время (24 часа в сутки или 24/7/365) вплоть до доставки? Да, могу
Безопасность:	– Могут ли посторонние лица и другие клиенты иметь доступ к моему заказу? Нет, не могут. – Могу ли я не привязывать банковскую карту к приложению? Нет, не могу.

Эти критерии приемлемости определяют на высоком уровне критерии тестирования, которые будут подтверждать, что эта история пользователя работает так, как это требуется. Они не предназначены для полных тестовых сценариев, но будут использоваться для расширения в соответствующих тестовых сценариях и тестовых сценариев во время спринта, по мере необходимости.

Для Истории пользователя на самом высоком уровне критерии принятия могут быть использованы для определения целей проекта с использованием

Пользовательские истории могут быть связаны друг с другом, эту связь также необходимо отображать.

На стадии технико-экономического обоснования пользовательские истории должны представлять собой небольшое количество четких заявлений, которых достаточно для того, чтобы охватить проект, определить, стоит ли двигаться дальше, и установить вероятные затраты и выгоды, которые могут быть достигнуты.

Нефункциональные требования также могут возникнуть на этом этапе, однако ожидается, что они станут более четкими и подробными на протяжении всего проекта. Некоторые из наиболее критических требований могут быть очевидны с самого начала, когда цель проекта установлена, и их необходимо фиксировать, поскольку они могут ограничить некоторые из вариантов выбора для проекта [3].

Деятельность по удовлетворению потребностей в ходе развития проекта

В начале каждого спринта необходимо дополнительно исследовать пользовательские истории, выделенные для этого спринта. Истории пользователей разбиваются на более подробные, которые достаточно малы и понятны для работы команды. Детали разрабатываются только по одному спринту за раз, и таким образом можно управлять сложностью требований. Кроме того, мелкие детали выявляются только непосредственно перед созданием конкретного элемента решения. Это позволяет избежать траты времени на разработку деталей во всех областях наперед.

Таким образом, требования развиваются и возникают в процессе всего жизненного цикла проекта. Анализ детально проработанных требований необходимо намеренно оставлять как можно дальше по ходу проекта, чтобы избежать ненужной доработки и иметь возможность управлять сложностью. В связи с чем, на ранних стадиях проекта важно получить согласие на высокоуровневый базисный набор приоритетных требований. Это дает объем, направление и соответствующую степень контроля для того, чтобы развивать детали проекта, позволяя в то же время принимать изменения и контролировать их. При разработке программного продукта автор будет использовать User Stories для детализации требований, а также проектировать интерфейсы на начальных этапах развития проекта.

Литература

1. Вигерс К., Битти Дж. Разработка требований к программному обеспечению. 3-е изд. СПб.: БХВ, 2018.
2. Гибкая методология разработки приложений Dynamic Systems Development Method // WINFOX. [Электронный ресурс]. URL: <https://wnfx.ru/gibkaya-metodologiya-razrabotki-prilozheniy-dynamic-systems-development-method/> (дата обращения: 06.12.2020).
3. Детализация требований // IBM Knowledge Center. [Электронный ресурс]. URL: https://www.ibm.com/support/knowledgecenter/ru/SSYMRC_6.0.2/com.ibm.team.concert.doc/topics/s_calm_detailreqs.html (дата обращения: 06.12.2020).
4. Детализация требований // Энциклопедия по экономике. [Электронный ресурс]. URL: <https://economy-ru.info/info/195146/> (дата обращения: 06.12.2020).
5. Документирование требований: мелкие ошибки, порождающие крупные проблемы // Habr. [Электронный ресурс]. URL: <https://habr.com/ru/post/299930/> (дата обращения: 06.12.2020).
6. Попов А.А. Эргономика пользовательских интерфейсов в информационных системах. М.: РУСАЙНС, 2016.

IRINA TROFIMOVA

Plekhanov Russian University of Economics

TOP-LEVEL REQUIREMENTS AND THEIR DETAILING FOR THE DEVELOPMENT TEAM

The article deals with the approach to the detailing of the top-level requirements of the employer with the use of the methodology "Dynamics Systems Development Method", to be more exact the explanation of the concept of the requirements and their classification, the approach to the development of Users Stories and their influence on the development and formation of the technical and economical substantiation.

Key words: requirements, detailing of the requirements, Dynamics Systems Development Method, User Stories, development.