

ВВЕДЕНИЕ

Для современного, активно развивающегося и высокотехнологичного информационного общества характерно существование множества автоматизированных информационных систем для различных сфер деятельности. И создание любой такой системы осуществляется в рамках соответствующего проекта. В свою очередь каждый проект начинается с формирования функциональных требований к нему, т.е. определения тех задач по обработке информации, которые должны быть реализованы в условиях проекта.

Как считают Элизабет Халл, Кен Джексон и Джереми Дик в [3], требования являются основой для любого проекта, в том числе и для создания любой информационной системы. Они определяют те потребности «заинтересованных сторон» (пользователей, клиентов, поставщиков, разработчиков и самого бизнеса), являющихся для них необходимыми и достаточными в рамках функциональных возможностей, которыми система должна впоследствии обладать, чтобы удовлетворить эти потребности.

Для того чтобы стать всем понятными, требования в большинстве случаев пишутся на «обычном» языке, что приносит проблемы другого рода: необходимость полностью и однозначно идентифицировать проблемы и зафиксировать потребности без использования профессиональных терминов или предварительных договоренностей, и это является весьма сложной задачей [3, с. 2].

Поэтому этап формирования требований для информационной системы является первичным и ключевым этапом ее создания.

Данное пособие предназначено для изучения некоторых вариантов процесса разработки и управления требованиями, а также показано, как можно качественно формировать требования Заказчика, ведя Заказчика от его «хотелок» к его мечте, необходимому Заказчику программному продукту.

Все разделы публикации снабжены картинками, моделирующими образы неких событий, процессов, сущностей (а точнее наиболее важных для нас их аспектов), происходящих в реальном мире. При раскрытии материала от раздела к разделу модели дополняются и детализируются, показывая динамику процесса проектирования.

Для более полного понимания процесса формирования требований в данном пособии рассматривается формирование требований к разрабатываемому ПО на примере **реального проекта: «Автоматизация процесса управления требованиями».**

Цели работы:

- Разобраться с ключевыми моментами процесса разработки Требований для создания ПО, на которых стоит акцентировать внимание.
- Научиться структурировать формировать и структурировать требования Заказчика.
- Использовать управленческие приемы, позволяющие аналитику общаться с заинтересованными лицами и эффективно продвигать результаты своего труда в проекте.

Теоретическая часть

МЕТОДЫ ОПРЕДЕЛЕНИЯ ТРЕБОВАНИЙ

Учебник : Лаврищева Е.М. , Петрухин В.А. стр 62

Пример

ПОДГОТОВКА СРЕДЫ ПРОЕКТА



«Скажи мне – и я забуду.
Покажи мне – и я запомню.
Вовлеки меня – и я научусь»»
Китайская народная мудрость.

Разработка требований и документирование проекта для сложных продуктов — это длительный процесс, к тому же очень кропотливый и требующий слаженной работы коллектива. Поэтому, с самого начала необходимо продумать и подготовить ландшафт (среду обитания), в которой этот процесс будет протекать

Цель данной группы работ: подготовить условия для качественного и эффективного взаимодействия команды проекта в рамках разработки и реализации требований к целевому продукту.

Практически любой ИТ проект начинается со сбора информации о деловых процессах, подлежащих автоматизации, среде их функционирования и т.п. Цель этого процесса заключается в том, чтобы все заинтересованные стороны договорились относительно общего толкования конечного результата проекта. Поскольку, участники проекта имеют различный уровень подготовки в области работы с информацией, то и материалы должны быть представлены в разнообразных (но заранее определенных) формах, доступных для понимания разными группами. Другими словами, с разной степенью детализации, уровня абстракции, применения нотаций и других характеристик. Поэтому, сразу разделим контент проекта, как минимум, на два домена:

1. Информация исходного видения с точки зрения потребителей будущего продукта, называемая «областью проблем». Основана на знании реального мира, в котором существует проблема

2. Информация о реализации – взгляд со стороны разработчиков, называемая «областью решения». Основана на знании домена технологий, в которой решение может быть реализовано.

Требования, хотя и в разных пропорциях, но собираются в течение всего проекта и, соответственно, все это время влияют на ход его реализации и конечный результат. Для того чтобы итоговый продукт максимально удовлетворял потребностям заказчика, а они, могут изменяться в процессе их реализации, необходимо на протяжении всего проекта поддерживать тесный и продуктивный диалог с потребителем конечного продукта. К этому тезису мы будем постоянно возвращаться в процессе изложения материала, поскольку он является ключевым. Исходя из вышесказанного, очень важно, чтобы оба домена проекта: «область проблем» и «область решения», изменялись синхронно и зависели друг от друга.

Постоянное взаимное общение представителей каждой из точек зрения, помогает эффективно бороться с проблемами неверных предположений сторонами:

- Проблема описанная заказчиком, может оказаться не той, в решении которой у него есть реальная нужда.
- Решение, описанное разработчиками, может отличаться от того, что они а самом деле создают.

На рисунке 3.1 изображена модель взаимодействия команды проекта в процессе формирования и использования контента проекта («области проблем» и «области решений»).

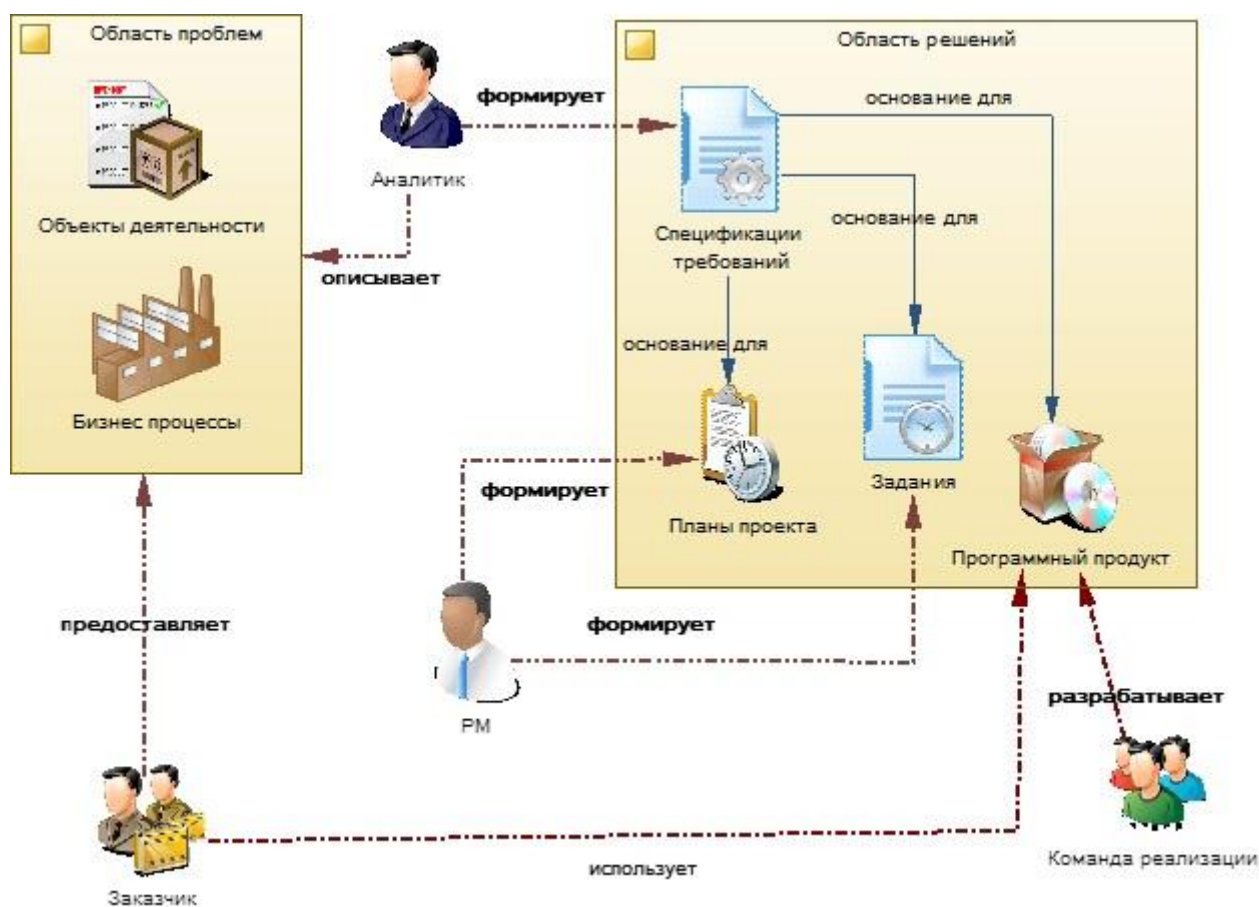


Рисунок 3.1 — модель процесса взаимодействия команды проекта

На этапе подготовки среды проекта, начинают формировать списки заинтересованных сторон. Важно не упустить никого, из тех — кого как-то касается процесс и(или) результат создания нового продукта. Заинтересованными лицами проекта могут быть не только будущие пользователи системы и команда проекта, но и различные должностные лица заказчика, инвесторы, эксперты по эргономики, чиновники стандартизирующих и регулирующих органов и т.п. Этот список может пополняться и уточняться в течении всего проекта, по мере формирования и уточнения требований.

Оформлять список удобно в виде таблиц, в которых помимо наименования заинтересованного лица, необходимо уточнить его цели, классифицировать его компетенцию и роль в проекте.

Ниже приведен пример использования таблиц, для формирования списка заинтересованных лиц проекта:

Таблица 2.1 – Совет ответственных за анализ требований		
Роль	ФИО ответственного лица	Заинтересованная сторона
Руководитель RAG	Сергеев П.А.	Руководитель проекта
Участник RAG	Прохоров И.Т.	Разработчик
Участник RAG		Разработчик
Участник RAG		Разработчик

Таблица 2.2 – Роли и обязанности процесса управления требованиями		
Роль/Группа	Перечень лиц	Основные обязанности
Бизнес - Аналитик	Семенова К.А.	Отвечает за моделирование бизнес-прецедентов, определяет рамки, в которых проводится бизнес-моделирование.
Системный - Аналитик	Гришаев Е.В.	Отвечает за внесение запросов на изменения в базу данных запросов на изменения от имени пользователей и заказчиков. Разрабатывает дизайн, создает прототип интерфейса пользователя.
Аналитическая группа		Разработка и поддержка процесса управления требованиями в проекте.
Группа тестирования		Верификация и <u>валидация</u> продукта.
Группа анализа требований		Анализ и внутреннее утверждение требований, <u>валидация</u> требований.

Таблица 2.3 – Перечень лиц, которые являются источниками информации для проекта			
ФИО	Организация	Должность	Заинтересованная сторона
Бондаренко И.О.	Организация 1	Технолог	Заказчик
<u>Ломачук И.Н</u>	Организация 1	Бухгалтер	Заказчик
Сергеев П.А.		Руководитель проекта	Исполнитель

Еще один целесообразный шаг для формирования ландшафта проекта, это публикация на его «витрине» — словаря (глоссария), который послужит вашей команде инструментом достижения консенсуса в толковании объектов и явлений предметной области. Это позволит заказчикам и разработчикам общаться на диспутах на «одном языке».

Опыт показывает, что использование глоссария экономит массу времени, а главное нервов, так как часто заказчики и исполнители тратят их на споры “каждый о своем”, возникающие только из-за разницы в понимании одного и того же термина. Например, использованное в заголовке слово “Ландшафт” может трактоваться, в зависимости от области применения, очень по-разному. Но в контексте нашего проекта добавим в Глоссарий такое определение:

Термин	Определение термина	Источник
Ландшафт	Это «система способов репрезентации, структурирования и символизирования окружающей среды проекта»	

Глоссарий ДОЛЖЕН пополняться в течение всего жизненного цикла проекта, поэтому и мы, по ходу развития нашего проекта в публикации, будем определять термины, которые могут вызвать различное толкование.

ЦЕЛИ, ПРОЕКТА



Цель не обязательно должна достигаться. Порой это просто направление двигаться дальше.

Брюс Ли.

Одним из основных признаков системы, отличающим ее от разрозненных компонентов, является подчиненность всей организации системы — некоторой цели. Проектная работа команды, представляет собой тоже некую систему и соответственно должна «идти на поводу» у какой-то цели. Потому, установив коммуникации между участниками проекта, начнем определять цели, которые мы хотим достичь в результате создания нового продукта.

Цель данной группы работ: определить основные задачи, которые ставят перед собой группы заинтересованных лиц, участвующих в проекте.

Формулирование целей в данном контексте — это скорее декларативный шаг провозглашения лозунгов, поднимающий командный дух и дающий команде старт для совместного обсуждения их ближайшего будущего.

Цель нужна для того, что в дальнейшем формировать требования, детализирующие наши цели. Эти требования должны соответствовать всем постулатам технологии SMART. SMART — методика оценки эффективности целеполагания по пяти критериям (конкретность, измеримость, достижимость, ориентация на результат, ограниченность по времени).

С точки зрения руководителя проекта правильно сформулированные цели должны позволить:

1. Идентифицировать продукт, который команда должна произвести в ходе реализации проекта, и блага, которые заказчик должен получить в результате его завершения;
2. Определить критерии успешности выполнения проекта в целом, с точки зрения основных заинтересованных лиц, в том числе не только заказчика, но и команды проекта.

Поэтому цель аналитика на данном этапе — обеспечить команду и заказчика именно этой информацией.

Определяя цели, важно учитывать два важных момента:

1. Заказчик чаще всего не в состоянии самостоятельно качественно сформулировать цели;
2. Цели в проекте у разных представителей заказчика, подрядчиков и исполнителей разные, а иногда и противоречащие друг другу.

При определении целей заказчика, совместно с его представителями, аналитик должен в первую очередь оперировать понятием выгоды, которую получит владелец от использования каждой рассматриваемой функции целевой системы. Поэтому, когда заказчик просит реализовать Вас какую-то возможность, приучите его сразу задуматься, а какой профит он с этого получит. Так Вы можете еще на начальном этапе отменить часть излишних требований и сэкономить время на стадии определения границ проекта

При формулировании целей заказчика, Вы должны учитывать и свои собственные стратегические цели развития, которые хотите достичь в результате выполнения работ. Например, это может быть попытка заложить в функциональные возможности разрабатываемой системы — характеристики, не существенные для заказчика, но представляющие интерес для вывода Вами на рынок, разработанного целевого продукта.

На рисунке 4.1 изображен процесс формирования целей создания информационной системы.

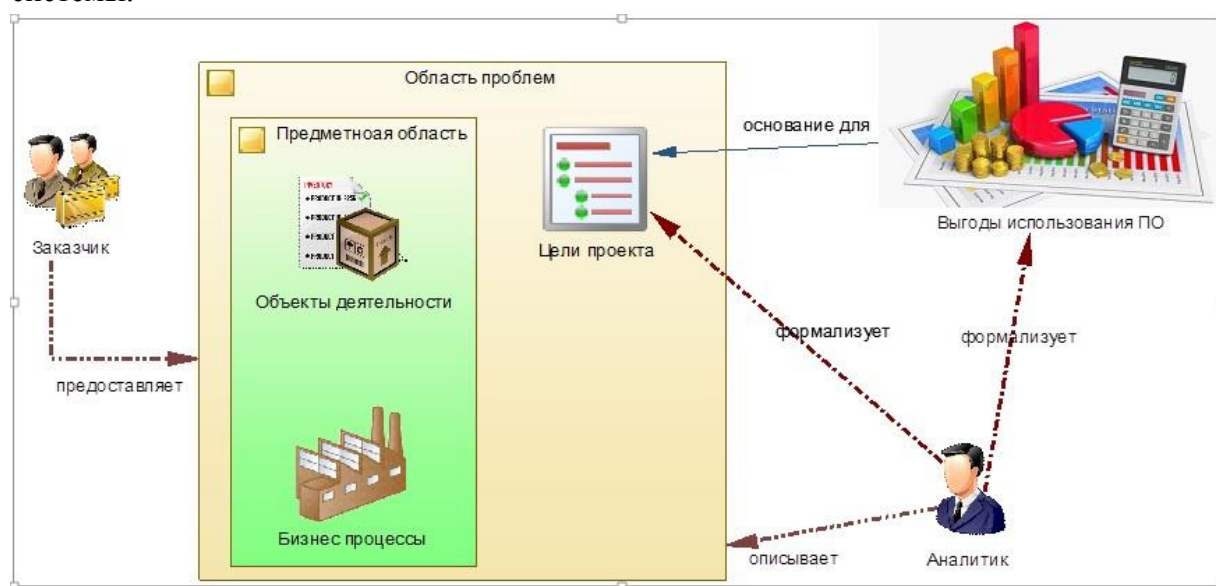


Рисунок 4.1 — модель процесса формирования целей

Отбросив общие цели процесса разработки требований, попытаемся выделить главные задачи, которые должна решить проектируемая нами учебная система «Управления требованиями», для удовлетворения, существующих потребностей заказчика.

Прежде всего, следует обратиться к доступным источникам, описывающим проблемы и решения автоматизируемой предметной области. Согласно SWEBOK (Software Engineering Body of Knowledge), анализ требований Requirement Process состоит из следующих основных составляющих:

- Requirements Elicitation (Извлечение требований),
- Requirements Analysis (Анализ требований в узком смысле),
- Requirements Specification (Специфицирование требований),
- Requirements Validation (Проверка требований).

Извлечение требований - это сбор информации о задачах, целях и проблемах заказчика, которые он собирается решать при помощи создаваемого (целевого) продукта. На основании этих пожеланий будут сформулированы некие потребности заказчика (требования пользователей), пригодные для выполнения дальнейших работ по их реализации.

Добавим в глоссарий термин:

Термин	Определение термина	Источник
Требования пользователей (<u>user requirements</u>)	<u>описание</u> цели и задачи, которые пользователям позволит решить система.	<u>Вигерс</u> Карл [5]

Эту информацию мы должны зафиксировать и обязательно предоставить для обсуждения всем заинтересованным лицам проекта, с возможностью их дополнения и изменения. Откуда вытекает первая цель нашей системы:

1. Фиксировать перечень потребностей пользователей, которые они желают удовлетворить при помощи целевого продукта.

На основании утвержденных с заказчиком потребностей, подлежащих автоматизации (домен проблем), мы будем формировать спецификации требований к разрабатываемому продукту (домен решений), используя известные приемы, которые будут описаны ниже. Поскольку большая часть процесса разработки программных продуктов строится как раз вокруг спецификаций требований, следующая цель нашей системы:

2. Фиксировать перечень функциональных и системных требований, предъявляемых к разрабатываемому продукту. Добавим небольшое уточнение: в том числе фиксировать покрытие пожеланий пользователей (области проблем) системными требованиями (областью решений).

Согласуем термин требования в Глоссарии:

Термин	Определение термина	Источник
Функциональные требования (<u>functional requirements</u>)	Функциональность ПО, которую разработчики должны построить, чтобы пользователи смогли выполнить свои задачи в рамках бизнес-требований	Вигерс Карл [5]
Системное требование (<u>System requirements</u>)	Абстрактное определение, того как система будет удовлетворять потребности пользователей	Э.Халл, К.Джексон [6]

Спецификации требований станут в нашей системе каркасом, связывающим все части проекта от потребности заказчика (поскольку, они явно связаны со спецификациями) до реализованных функций в целевом продукте. Для этого, на основании спецификаций, мы будем выставлять задания исполнителям на проектирование, разработку, тестирование, развертывание, пилотное внедрение и т.д. Разработчики в коде укажут ссылку на задачу. По задаче можно выяснить требования и т.д. в реверсном направлении к потребностям пользователя. Опираясь на вышеизложенные постулаты, сформулируем следующую цель:

3. На основании спецификации требований выставлять задания исполнителям, направленным на их реализацию в конечном продукте.

Поскольку факт выполнения задания, по сути, выполняет продвижение функциональности продукта к запланированным характеристикам, нам необходимо фиксировать в системе все действия, предпринятые исполнителями, по задачам. Таким образом, мы сможем в разрезе времени и каждого отдельного требования отслеживать приращение функциональности целевого продукта. Следовательно, следующую цель можно сформулировать так:

4. Фиксировать факт выполнения заданий исполнителями.

Практически в каждом проекте, по ходу его реализации, происходят изменения и уточнения требований. Это отдельный раздел в дисциплине «Анализ требований» и не упомянуть этот процесс в нашем проекте мы не можем, а потому еще одна цель:

5. Управлять изменениями требований.

Теперь, когда мы разобрались с основными целями, перейдем к формулированию потребностей заказчика, которые удовлетворят поставленные цели.

Теперь, когда мы разобрались с основными целями, перейдем к формулированию потребностей заказчика, которые удовлетворят поставленные цели.

Формирование потребностей заказчиков



Из всех возможных способов совершенствования процесса разработки ПО наибольшее преимущество за формулированием требований. Карл Вигерс

Дисциплина «Системный анализ», для борьбы со сложностью, предлагает использовать такие основные приемы, как абстракция и декомпозиция, позволяющие раскладывать требования по уровням представления. Следуя этим принципам, в основе пирамиды анализа располагаются Цели, например, описанные в предыдущем разделе. Поднимаясь вверх, мы раскладываем их на более мелкие элементы, собирая пожелания заказчика к функциональности разрабатываемого продукта.

Цель данной группы работ: собрать максимально полные и точные сведения о потребностях заказчика, которые они хотят удовлетворить при помощи целевого продукта.

При сборе потребностей заказчика необходимо помнить: если потенциальные пользователи системы излагают Вам какие-то пожелания, которые мы не можем связать ни с одной из целей, то следует уточнить у заказчика, необходимость автоматизации этой потребности, либо пересмотреть еще раз цели.

ВАЖНО : не пытайтесь описывать потребности пользователей в терминах программного продукта, это плохая практика. Избегайте таких выражений как: «выбор в меню», «нажатие кнопки», «внесение в поле» и т.п., а используйте например такие: «выбор варианта», «выполнение процесса», «определение значения» и т.п.



Рисунок 5.1 — модель процесса формализации потребностей заказчика

Далее будет показан процесс расширения и детализации этой модели (см. рис. 5.1), продвигаясь от раздела к разделу в процессе формирования спецификаций требований.

ОПРЕДЕЛЕНИЯ ПОТРЕБНОСТЕЙ ЗАКАЗЧИКА

Для описания пользовательских условий можно использовать метод описания «Пользовательских историй». Этот прием хорошо описан в методиках гибкого программирования Scrum, Канбан [10]. Суть его состоит в том, что команда, совместно с владельцами процессов, составляет небольшие, в одно-два предложения, сценарии бизнес процессов, подлежащих автоматизации. Для каждой Пользовательской истории (User Story)

определяется цель, которая должна быть достигнута в результате ее выполнения. Если цель сформулировать сложно, следует задуматься о целесообразности использования этого сценария в общем процессе.

Важно осознавать, что формирование Пользовательских историй подразумевает непосредственное вовлечение будущих потребителей разрабатываемого продукта в процесс его создания, делая их соавторами (или соучастниками в случае провала). И этот факт нужно обязательно использовать, путем донесения до сознания заказчика, необходимости разделения ответственности за качество создаваемого продукта.

Пользовательские истории удобно писать на стикерах (карточках), наклеенных на доску, к которой имеет постоянный доступ вся команда проекта. Эта площадка обычно используется для проведения митингов – совместного обсуждения этапов реализации процесса автоматизации объявленных бизнес сценариев. Оформление Пользовательских историй на стикере, гарантирует, что она не станет слишком большой.

При прохождении Пользовательской историей жизненного цикла: формирования, обсуждения, реализации, тестирования и т.д., стикеры с их описаниями переклеивают на разные «дорожки», соответствующие этапу разработки. Таким образом, в небольших проектах, при помощи описанного приема, управляют не только содержанием проекта верхнего уровня, но и процессом его реализации.

Этот метод имеет не только практическое, но и большое психологическое значение, влияющее на командный дух проекта.

Исходя из выявленных выше целей, начнем собирать и публиковать пользовательские истории (Функциональные требования).

нам необходимо разработать Программный Продукт, который обеспечит поддержку следующих бизнес процессов:

US01. Описать пользовательскую историю на стикере. Сформулировать Цель процесса.

Цель: Зафиксировать бизнес процесс, подлежащий автоматизации, в доступной для обсуждения форме.

«US01» — в данном примере — это идентификатор Пользовательской истории, состоящий из префикса, указывающего на тип артефакта проекта (User Story) и его порядкового номера. Добавим в глоссарий термин:

Термин	Определение термина	Источник
Бизнес-требования (<u>business requirements</u>)	<u>набор</u> активностей, которые преобразуют несколько видов входных характеристик в выход, имеющий ценность для потребителя	М. Хаммер, Дж. Чампи

Как правило, стиль изложения и содержание пользователями своих пожеланий к разрабатываемому продукту, мягко говоря, не идеален. Даже показав эти пожелания другим членам команды, мы сталкиваемся с непониманием ими сути проблемы. Поэтому,

каждую пользовательскую историю желательно обсудить с заинтересованными участниками и попытаться сформулировать ее понятно и односложно. Запишем это так:

US02. Обсудить пользовательскую историю с участниками проекта. При необходимости внести изменения.

Цель: Ознакомить команду проекта со сценарием бизнес процесса. Согласовать его.

Для дальнейшего использования Пользовательской истории, нам необходимо зафиксировать ее в системе. Например, связав Пользовательскую историю с соответствующими требованиями, мы сможем трассировать ее реализацию в конечном продукте через цепочку: Пользовательская история → Требование, созданное по ней → Задания, выставленные по требованию и т.д. Формулируем:

US03. Опубликовать в системе, согласованные Пользовательские истории.

Цель: Зафиксировать в системе перечень утвержденных Бизнес-требований, для их учета и обработки.

Поскольку Пользовательские истории формулируются автором или группой авторов и представляют их взгляд на проблемы, входящие в сферу их интересов, нам необходимо фиксировать этих «заинтересованных лиц» в нашей системе. Для развития любого проекта очень важно определить круг лиц, так или иначе заинтересованных в получении целевого продукта. На основании этих данных мы будем согласовывать требования, определять роли пользователей и т.п.

US04. На основании Пользовательских историй, определить основных заинтересованных лиц проекта.

Цель: Зафиксировать в системе перечень участников и заинтересованных лиц проекта.

Пользовательская история не дает нам основания сразу приступать к написанию кода программы, поскольку это всего лишь определенный взгляд на процессы, по сути – декларации или наброски. Сначала потребуется их уточнение, детализация, разработка алгоритмов, четко регламентирующих выполнение процессов, моделирование структуры объектов предметной области и т.п. Поэтому, окончив процесс фиксации основных Пользовательских историй, мы приступаем к моделированию процессов и объектов, реализующих их. Попутно определяем их связи и информационные потоки. Сформулируем это так:

US05. На основании Пользовательских историй, определить основные функциональные требования к разрабатываемой системе.

Цель: Зафиксировать в системе перечень бизнес процессов, подлежащих автоматизации.

После определения полной функциональности разрабатываемого продукта, часто приходит осознание того, что весь этот перечень реализовать в установленные сроки, имеющимися ресурсами — не реально. И тогда, нужно разделить функции на: жизненно необходимые для поддержания процесса и функций, которые нужны, но пока без них можно обойтись. Сжав эти размышления в одно предложение, зафиксируем:

US06. На основании подготовленного перечня автоматизируемых процессов, определить их приоритетность и очередность.

Цель: Определить текущие границы проекта.

Когда все процессы, классы, хранилища и прочие объекты разрабатываемых требований спроектированы, необходимо всю эту информацию «нанизать» на единый стержень, связывающий между собой элементы: в нотациях, документах описаниях и прочих артефактах. Таким стержнем обычно выступают спецификации требований. Они фиксируются в виде структурированных данных, имеют уникальный идентификатор, на который и ссылаются все вышеперечисленные артефакты. Записываем следующую Пользовательскую историю:

US07. На основании процессов, выявленных для автоматизации, сформулировать спецификации требования для реализации их в продукте. Зарегистрировать требования в системе, соотнести с Пользовательскими историями.

Цель: Зарегистрировать в системе: контент, концептуальное наполнение содержания проекта, формулировки сути автоматизируемых сервисов, обеспечивающих базис для разработки продукта, его тестирования и передачи в эксплуатацию.

Когда мы разработаем спецификации, мы получим полный список низкоуровневых требований к системе с подробным описанием их реализации. На основании этого списка мы сможем выставлять задания или цепочки заданий исполнителям на проектирование, разработку, тестирование, развертывание, пилотное внедрение и т.д. При выставлении задания по требованию, его состояние в системе может автоматически меняться. Таким образом, в списке спецификаций мы сможем увидеть, какие требования уже взяты в работу, а какие еще ожидают в очереди. А поскольку спецификации связываются и с Пользовательскими историями, то по цепочки мы сможем отслеживать и их состояние.

US08. На основании спецификации требования, сформировать задания исполнителям. В ходе исполнения заданий изменить состояния требований.

Цель: На основании требования, обеспечить процесс реализации проекта в целевом продукте.

Задание, выставленное исполнителю, становится доступными ему в системе. Поскольку задание выставлено по требованию, исполнитель может ознакомиться с его спецификацией, а также спецификациями всех связанных с ним требований.

Завершив работы, исполнитель фиксирует отчет о выполнении, «закрывая» свои задачи. Закрытие цепочки задач по требованию автоматически инициирует изменение его состояния.

US09. Зафиксировать факт выполнения задания исполнителем. Изменить состояние требования.

Цель: На основании факта выполнения задания исполнителем, обеспечить актуализацию процесса выполнения проекта.

Часто, после разработки первых прототипов целевого продукта, пользователей, ознакомившихся с его работой, посещает «прозрение». Они начинают отчетливо понимать, что хотели не “это” или не совсем “это”. Возникает необходимость вносить изменения в требования. Несмотря на кажущуюся простоту — это очень кропотливый процесс, требующий проведения изменений сквозь все артефакты проекта. Иногда, приходится вносить изменения от самого начала (фиксации Пользовательской истории) и далее по всем этапам, формирования требования.

US10. Внести изменения в спецификации требования для реализации в продукте. Изменить состояние требования. Синхронизировать с Пользовательской историей при необходимости.

Цель: Изменить содержание контента, обеспечивающего базис для разработки продукта, его тестирования и передачи в эксплуатацию.

Поскольку процесс управления Требованиями связан с постоянным мониторингом степени их выполнения, необходимо иметь удобный инструмент, позволяющий отбирать массивы требований по разным срезам их свойств, включая текущие состояния.

US11. Предоставить полный или ограниченный по разным срезам перечень требований с текущим состоянием.

Цель: Получить данные о текущем состоянии процесса разработки целевого продукта проекта.

В ходе продвижения проекта мы можем: добавлять новые Пользовательские истории; раскладывать утвержденные истории, детализируя их в группе, непосредственно занимающейся ее реализацией.

ВИЗУАЛИЗАЦИЮ ТРЕБОВАНИЙ ДЛЯ ОБСУЖДЕНИЯ ИХ С ЗАКАЗЧИКОМ

Еще один эффективный прием дисциплины «Системный анализ» — моделирование. Эрик Эванс в книге [13] позиционирует модель как: «язык, на котором говорят все члены группы разработчиков, а также могут ... общаться со специалистами в предметной области без переводчика». Поэтому чем лучше мы подберем набор моделей для взаимодействия с заказчиком и членами команды проекта, тем более тесным и продуктивным будет наше сотрудничество и, соответственно, тем более полно и качественно мы сможем удовлетворить их потребности.

Для наглядности можно отображать пользовательские истории в виде моделей, представляющих набор связанных функций — сценариев на диаграммах Бизнес коммуникаций. В отличие от канонических диаграмм, здесь можно отобразить объекты разных типов в относительно свободной форме, тем самым, добавив наглядности.

Дадим определение модели, поскольку это один из ключевых элементов проектирования:

Термин	Определение термина	Источник
Модель (Model)	<u>это</u> упрощение; это такая интерпретация реальности, при которой из явления извлекаются существенные для решения задачи аспекты, а лишние детали игнорируются	Эрик Эванс [13]

Пример модели формирования Пользовательских историй показан на рисунке 5.2.

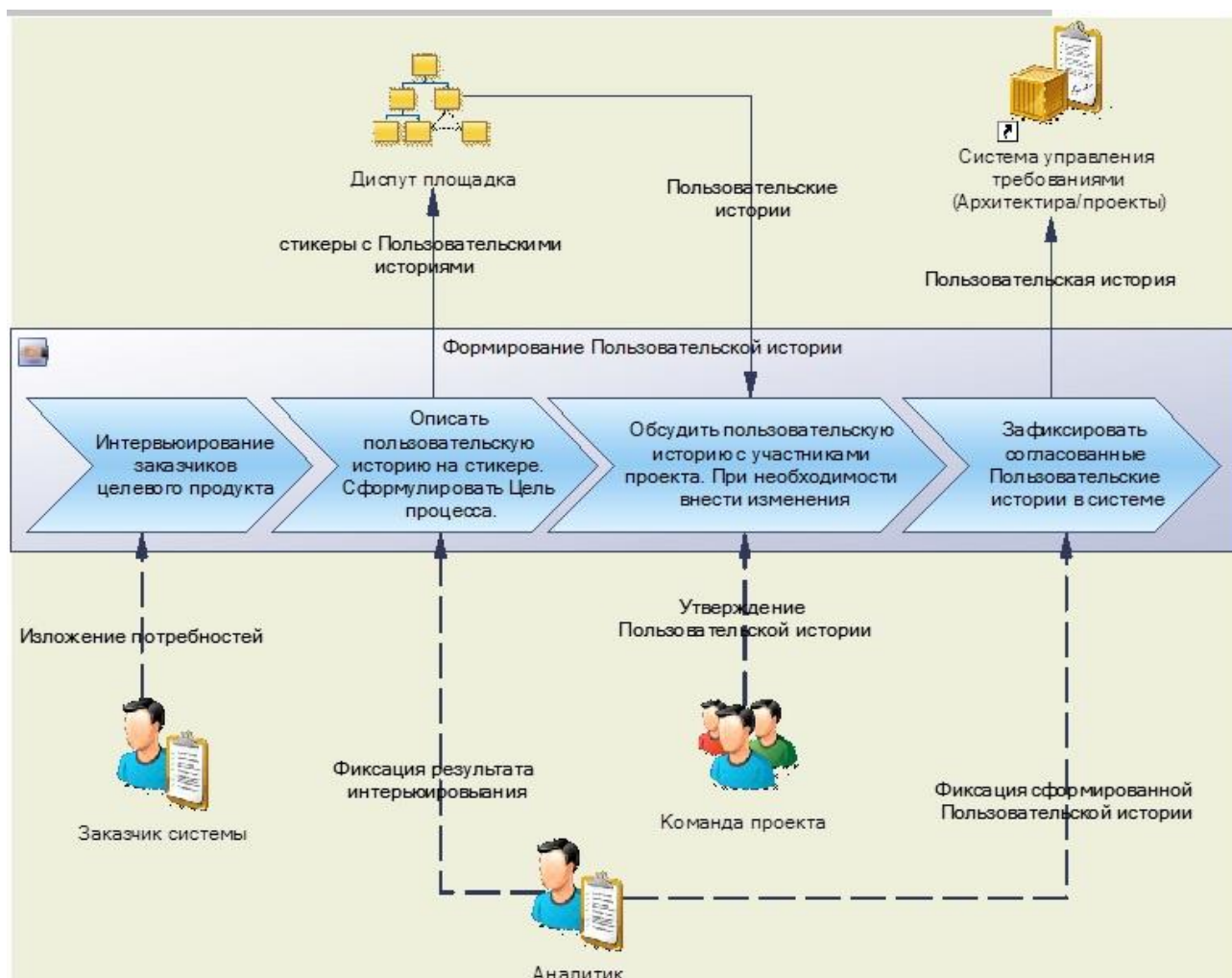


Рис. 5.2. Сценарий формирования Пользовательской истории

На рисунке мы наглядно видим процессы, их последовательность и место выполнения, а также ресурсы.

С помощью таких рисунков заказчикам легче представлять предметную область и обсуждать требования к разрабатываемому продукту. Но, передавая заинтересованным лицам диаграммы подобного типа, следует сопровождать их устным, а лучше письменным описанием. Например, для презентации, приведенной выше диаграммы, можно использовать такой текст:

В центральной части рисунка изображен бизнес сценарий «Формирование Пользовательской истории». Сценарий состоит из четырех процессов, отображенных в той последовательности, в которой они должны выполняться. В нижней части изображены

исполнители процессов, пунктирной линией обозначено их воздействие, а надписью определена роль. В верхней части диаграммы представлены места дислокации процессов, а линиями обозначены потоки данных, которые через них проходят.

ИТОГИ ПРОЦЕССА ОПРЕДЕЛЕНИЯ ПОТРЕБНОСТЕЙ ЗАКАЗЧИКА

Пользовательские истории, с одной стороны являются великолепным механизмом для выявления пользовательских требований, а с другой, их можно позиционировать, как основной инструмент влияния заказчика на разработку программного обеспечения. Проговаривание Пользовательских историй с заинтересованными лицами проекта на диспутах, побуждают людей многократно «проигрывать» в уме разные сценарии их работы, и не только те которые используются в настоящее время, но и те которые им хотелось воплотить в будущем.

В методологиях гибкого проектирования — Пользовательская история используется, как быстрый способ документировать требования клиента, без необходимости разрабатывать обширные формализованные документы и впоследствии тратить ресурсы на их поддержание. Но такой подход может успешно использоваться только для небольших или несложных проектов, не связанных с «закрученными» бизнес процессами, которые не планируются развивать и поддерживать.

Можно использовать Пользовательские истории, как первый шаг в процессе сбора и формализации потребностей заказчика, для получения полного перечня бизнес задач, которые могут быть автоматизированы путем создания программного продукта. Далее эти потребности должны быть детализированы и трансформированы в функции целевой системы.

Порядок выполнения работы:

1. Получить задание у преподавателя, или предложить свою тему, согласовав ее с преподавателем.
2. Сформировать группу, для выполнения проекта. Группа должна включать 3-4 человека.

Цель этого этапа: подготовить условия для качественного и эффективного взаимодействия команды проекта в рамках разработки и реализации требований к целевому продукту.

На этом этапе следует:

- сформировать список ответственных за формирование и анализ требований;
- описать роли и обязанности процесса управления требованиями;

3. Определить и описать проблему, которая подлежит автоматизации.

Цель данного этапа: изучить бизнес процесс заказчика, определить место проблемы в этом процессе.

На этом этапе следует:

- тщательно образом изучить бизнес процессы заказчика, терминологию, применяемую в данном бизнесе, а при необходимости - даже сленг людей, которые выступают на этом рынке в качестве исполнителей и пользователей.

- выявление значимых проблем, определение того, насколько хорошо они решаются при текущем положении дел и как взаимодействуют друг с другом.
- выделить из бизнес процесса, изучить и подробно описать проблему, подлежащую автоматизации.
- **определить перечень лиц, которые являются источниками требований;**
- определить входные данные и данные, которые необходимо получить на выходе разрабатываемого программного продукта.
- сбор требований. Это процесс взаимодействия с лицами, формирующими требования.
- **начинать** формирование словаря предметной области.

4. Сформировать цель разрабатываемого ПО.

Цель данного этапа: определить основные функции разрабатываемого ПО, которые ставят перед собой группы заинтересованных лиц, участвующих в проекте.

5. Постановка задачи.

Цель данного этапа: - подробно описать основные потребности заказчика, которые необходимо автоматизировать, и порядок их обработки.

На этом этапе следует:

- выявить потенциальных пользователей будущей системы и подробно описать их потребности.
- определение ключевых абстракций, характеризующих словарь предметной области (выделение сущностей предметной области, необходимых для работы разрабатываемого программного продукта - словарь сущностей предметной области) [13 стр. 46]
- определение механизмов управления абстракциями (описание действий системы для обработки этих сущностей – словарь по Абботу).
- описание сущностей предметной области и действия системы по их обработке – объектно-ориентированный словарь.

6. Формирование требований.

Цель данного этапа: - бесформенный набор потребностей преобразовать в логически связанные группы требований.

На этом этапе следует:

- классифицировать требования. На этом этапе бесформенный набор потребностей преобразуется в логически связанные группы требований.
- определить и описать необходимые входные данные и данные, которые необходимо получить на выходе при автоматизации каждого требования;
- описать ограничения и правила, согласно которым должны выполняться требования.
- разрешение противоречий (требования лиц, занятых в процессе формирования требований, могут быть противоречивыми. На данном шаге определяются и разрешаются противоречия такого рода);
- расстановка приоритетов при рассмотрении недостаточно хорошо решаемых потребностей, поскольку чаще всего решить сразу все проблемы невозможно.

Варианты заданий

Оформление отчета

По результатам выполнения указанных задач оформляется отчет, отражающий ход выполнения работы и содержащий описание полученных результатов на всех этапах работы.

К оформлению отчета предъявляются следующие требования:

- требования к содержанию - указаны в разделе «Порядок выполнения работы».
- требования к оформлению- определяются соответствующими стандартами (СТО ИРНИТУ 005- 2015 СИСТЕМА МЕНЕДЖМЕНТА КАЧЕСТВА. Учебно-методическая деятельность. Оформление курсовых проектов (работ) и выпускных квалификационных работ технических специальностей")

Литература

В настоящее время доступно большое количество литературы, посвященное Управлению требованиями. Большую часть произведений можно разделить на четыре группы:

1. Предоставление теоретической информации, описывающей ту или иную фундаментальную «тяжёлую» методологию, например: RUP или SADT. Эти методологии чрезвычайно эффективны и полезны, применительно к крупным проектам. Но их использование требует длительного процесса обучения, включая тренинги либо практику в компаниях, в которых они уже успешно применяются. Классическим произведением для аналитиков, из этой группы можно считать [1], [2].
2. Изложение академической точки зрения на процессы из области управления требованиями. Лично мне очень помогли книги [4], [5].
3. Экскурс в различные технологии в виде сравнительных обзоров. На мой взгляд, книга [6], самое удачное произведение для аналитиков из этой группы, хотя ее можно отнести и к группе академических произведений. О выборе методики для использования в различных проектах, компактно и доходчиво описано в статье [9].
4. Практические идеи, представленные в виде «лёгких» методологий, чаще всего предназначенных для небольших проектов, например: ICONIX, процесс Экстремального программирования (eXtreme Programming) [7], [10] и т.д. Такие методологии можно относительно быстро начать эффективно использовать в небольших командах.

СПИСОК ИСТОЧНИКОВ

1. Якобсон А., Буч Г., Рамбо Дж. – «Унифицированный процесс разработки ПО» (2004)
2. Дэвид А. Марк и Клемент МакГоуэн – «Методология структурного анализа и проектирования SADT»
3. Коберн — «Современные методы описания функциональных требований» (2002)
4. Леффингуэлл Дин, Уидрих Дон — «Принципы работы с требованиями к ПО» (2002)
5. Карл И. Вигерс – «Разработка требований к программному обеспечению» (2002)
6. Элизабет Халл, Кен Джексон, Джереми Дик — «Разработка и управление требованиями практическое руководство пользователей» (2005)
7. Скотт Амблер – «Гибкие технологии: экстремальное программирование и унифицированный процесс разработки» (2005)
8. Гринфилд Джек, Кейн Шорт — «Фабрики разработки программ» (2007)
9. Алистэр Коуберн — «Каждому проекту своя методология»
10. Вольфсон Борис- «Гибкие методологии разработки»
11. Лешек А. — «Анализ требований и проектирование систем»
12. Фримен Эрик, Фримен Элизабет — «Паттерны проектирования» (2011)
13. Эванс Эрик — «Предметно-ориентированное Проектирование» (2011)
14. ГОСТ 34.602-89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»