

 ARadzishevskiy 1 сентября в 09:05 Управление

# Практика формирования требований в ИТ проектах от А до Я. Часть 2 Цели и Потребности

Управление разработкой, Управление проектами, Управление продуктом, Бизнес-модели

С частью 1 можно ознакомиться, перейдя по [ссылке](#)

## IV ОПРЕДЕЛЯЕМ 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 по 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.**

Сейчас, в литературе доступен целый букет методик определения и формулирования целей. Есть даже такое направление — целеполагание, обучающее правилам постановки целей.

Цитата, приведенная в начале этого раздела, на мой взгляд, как нельзя лучше показывает роль целей в проекте по разработке Программного Обеспечения (ПО). Из моей практики, к целям, сформулированным на стадии инициирования проекта, возвращаются крайне редко. Формулирование целей в данном контексте — это скорее декларативный шаг провозглашения лозунгов, поднимающих командный дух и дающий команде старт для совместного обсуждения их ближайшего будущего. Чтобы предупредить критическую оценку своих высказываний со стороны знатоков целеполагания, сразу оговорюсь, что в дальнейшем мы будем формировать требования, детализирующие наши цели. Эти требования и будут 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

Поэтому, во-первых: Вы должны помочь представителям заказчика сформулировать цели, которые они хотят достичь в результате внедрения целевого продукта. Во-вторых: 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 функциональные возможности разрабатываемой системы — характеристики, не существенные для заказчика, но представляющие интерес для выведения Вами на рынок, разработанного целевого продукта.

Еще раз отдельно заострю ваше внимание на том, что при формулировании целей проекта, надо постараться заложить в них и свои собственные (команды реализации) стратегические интересы, которые Вы хотите достичь в результате своей деятельности в прое

А вот 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 на начальном этапе отменить часть излишних требований и сэкономить время на стадии определения границ проекта, но об этом чуть позже.

На рисунке 4.1 изображен процесс формирования целей создания информационной системы.

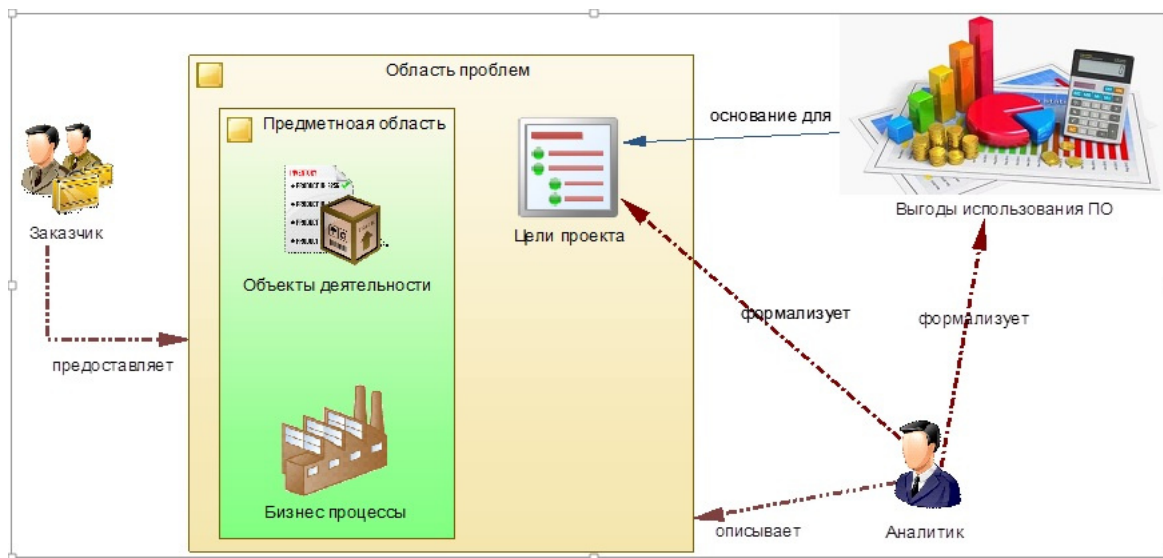


Рисунок 4.1 — модель процесса формирования целей

Отбросив общие цели процесса разработки требований, попытаемся выделить главные задачи, которые должна решить проектируемая нами учебная система «Управления требованиями», для удовлетворения, существующих потребностей заказчика. Поскольку в учебном проекте я выполняю и функцию аналитика, и имитирую деятельность заказчика, то приведу «наши совместные» размышления по формулированию каждой цели.

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).

Таким образом, мы должны охватить весь этот поток работ. И выполнять их желательно, как мы выяснили в предыдущей главе, придерживаясь единого стиля, используя единую «среду обитания» для всех процессов и участников проекта.

Итак, первый процесс, который мы будем выполнять при разработке 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 Карл</u> [5]

Эту информацию мы должны зафиксировать и обязательно предоставить для обсуждения всем заинтересованным лицам проекта, 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. Поскольку большая часть процесса разработки программных продуктов строится как раз вокруг спецификаций требований, следуя цели нашей системы:

### 2. Фиксировать перечень функциональных и системных требований, предъявляемых к разрабатываемому продукту. Добавь 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	<u>Вигерс Карл</u> [5]
Системное требование ( <u>System requirements</u> )	Абстрактное определение, того как система будет удовлетворять потребности пользователей	<u>Э.Халл</u> , <u>К.Джексон</u> [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 потребности 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 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.

## V УТОЧНЯЕМ ПОТРЕБНОСТИ ЗАКАЗЧИКА

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 этих потребностей, либо пересмотреть еще раз цели.

Еще одно важное предостережение: 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

Дале мы будем расширять и детализировать эту модель (см. рис. 5.1), продвигаясь от раздела к разделу в процессе формирования спецификаций требований.

## 1. ЗАДЕЙСТВУЕМ ПОЛЬЗОВАТЕЛЬСКИЕ ИСТОРИИ ДЛЯ ОПРЕДЕЛЕНИЯ ПОТРЕБНОСТЕЙ ЗАКАЗЧИКА

На мой взгляд, с самого начала проекта, очень удобно 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 владельцем процессов, составляет небольшие, в одно-два предложения, сценарии бизнес процессов, подлежащих автоматизации. Для каждой Пользовательской истории (англ. 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, 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. Я бы рекомендовал использовать такой подход и в больших проектах, в которых применяются профессиональные инструменты управления требованиями. Этот метод имеет не только практическое, но и большое психологическое значение, влияя на командный дух проекта. Пример Пользовательской истории показан на рисунке 5.1.

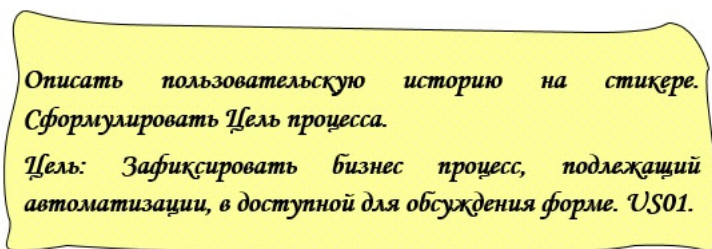


Рис. 5.1. Пользовательская история

Пользовательские истории также эффективно можно использовать и для приемочного тестирования.

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

Итак, 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 по-иному 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, входящую сферу их интересов, нам необходимо фиксировать этих «заинтересованных лиц» в нашей системе. Для развития любого проекта с 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, разработ алгоритмов, четко регламентирующих выполнение процессов, моделирование структуры объектов предметной области и т.п. Поэт окончив процесс фиксации основных Пользовательских историй, мы приступаем к моделированию процессов и объектов, реализу 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 од предложение, зафиксируем:

**US06. На основании подготовленного перечня автоматизируемых процессов, определить их приоритетность и очередности****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, и уникальный идентификатор, на который и ссылаются все вышеперечисленные артефакты. Записываем следующую Пользовательск историю:

**US07. На основании процессов, выявленных для автоматизации, сформулировать спецификации требования для реализаци 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 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 работу, а какие еще ожидают в очереди. А поскольку спецификации связываются и с Пользовательскими историями, то по цепочки сможем отслеживать и их состояние.

**US08. На основании спецификации требования, сформировать задания исполнителям. В ходе исполнения заданий измени состояния требований.****Цель: На основании требования, обеспечить процесс реализации проекта в целевом продукте.**

Задание, выставленное исполнителю, становятся 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. Закрытия 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 «прозре Они начинают отчетливо понимать, что хотели не “это” или не совсем “это”. Возникает необходимость вносить изменения в требо Несмотря на кажущуюся простоту — это очень кропотливый процесс, требующий проведения изменений сквозь все артефакты пр Иногда, приходится вносить изменения от самого начала (фиксации Пользовательской истории) и далее по всем этапам, формируя 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 удоб 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.

## 2. ИСПОЛЬЗУЕМ ВИЗУАЛИЗАЦИЮ ТРЕБОВАНИЙ ДЛЯ ОБСУЖДЕНИЯ ИХ С ЗАКАЗЧИКОМ

Еще один эффективный прием дисциплины «Системный анализ» — моделирование. Эрик Эванс в книге [13] позиционирует модел «язык, на котором говорят все члены группы разработчиков, а также могут ... общаться со специалистами в предметной области б переводчика». Поэтому чем лучше мы подберем набор моделей для взаимодействия с заказчиком и членами команды проекта, те 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 т в относительно свободной форме, тем самым, добавив наглядности.

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

Термин	Определение термина	Источник
Модель (Model)	это 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.

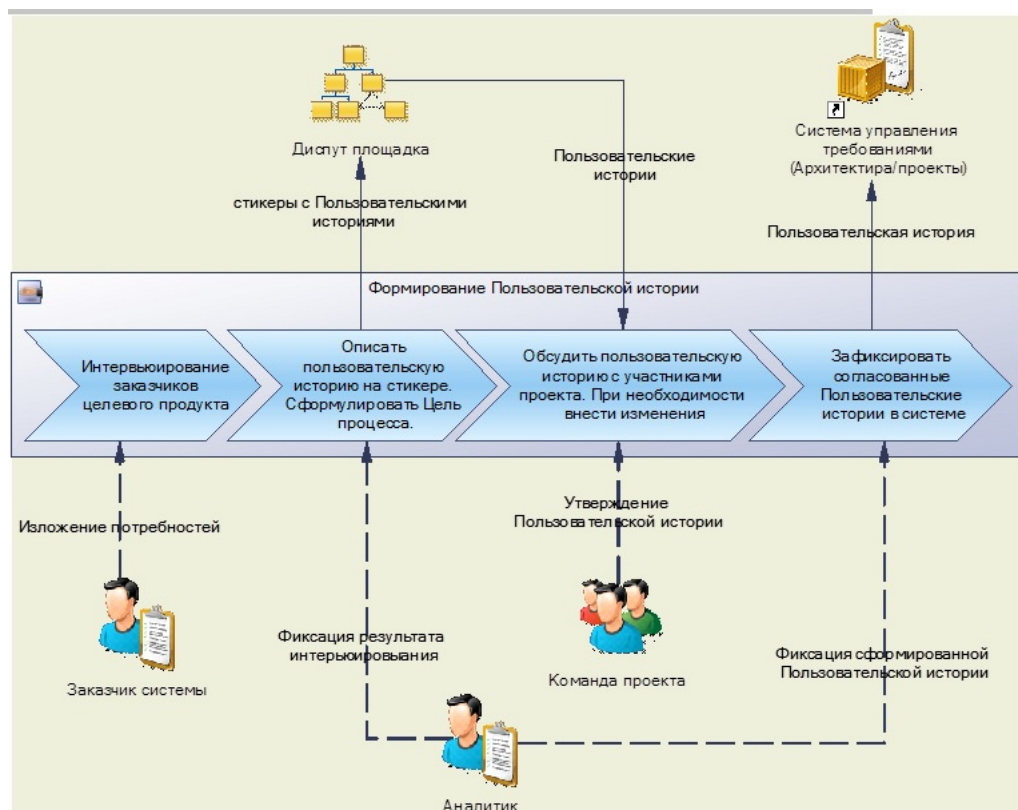


Рис. 5.2. Сценарий формирования Пользовательской истории

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

С помощью таких рисунков заказчик легче представляет 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 четырёх 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 представлен места дислокации процессов, а линиями обозначены потоки данных, которые через них проходят.

### 3. ПОДВЕДЕМ 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 хот 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 со «закрученными» бизнес процессами, которые не планируются развивать и поддерживать.

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


В следующей части мы будем выявлять функции системы и определять границы проекта [ссылка](#)

#### ▼ Список литературы

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 создание автоматизированной системы»

**Метки:** формирование требований, проектирование, требования, управление проектами, менеджмент проектов

↑ +11 ↓ 75 👁 3,5k 💬 2

 **7,0** Карма **69,8** Рейтинг **35** Подписчики

Алексей Радзишевский @ARadzishevskiy  
СЕО, Менеджер проектов, Системный аналитик

Поделиться публикацией    

**ПОХОЖИЕ ПУБЛИКАЦИИ**

9 февраля 2016 в 15:53

**43 полезных сервиса для управления проектами. Без эпитетов**

↑ +26 👁 254k 📌 729 💬 40



14 сентября 2012 в 12:40

**Успешный проект менеджмент без купюр**

↑ +25 👁 19,4k 📖 161 💬 69

24 августа 2011 в 16:33

**Комплексная система управления проектами для малого и среднего бизнеса**

↑ +2 👁 11,3k 📖 23 💬 23

## Комментарии 2

 **pastuh9090** 01.09.17 в 14:02 🗨 📖 📌 ⬆

Хотелось бы больше ссылок на ГОСТы, как это всё укладывается в тему ЕСПД предприятия?

 **ARadzishevskiy** 01.09.17 в 14:45 🗨 📖 📌 🔄 ⬆

В данной публикации упор сделан не на оформление, а именно на суть процесса формирования требований. Ссылки на ГОСТ\_ы будут в ч посвященной разработке спецификаций. А также они приведены в статье «О качестве требований в ИТ проектах, начистоту (с позиции команды разработки)», на которую есть ссылка в начале. первой части.

Только [полноправные пользователи](#) могут оставлять комментарии. [Войдите](#), пожалуйста.

### ИНТЕРЕСНЫЕ ПУБЛИКАЦИИ

**Паттерны поведения пользователей**

↑ +5 👁 557 📖 7 💬 2

**36 материалов о нейросетях: книги, статьи и последние исследования**

↑ +5 👁 518 📖 28 💬 1

**Конференция ISDEF: развиваем не продукт, а бизнес**

↑ +12 👁 230 📖 3 💬 1

**Передача экстренных данных в системе ЭРА-ГЛОНАСС** 🏷

↑ +15 👁 2,4k 📖 5 💬 52

**Браузер браузеру рознь, или Будни отдела корпоративной инфраструктуры**

↑ +8 👁 737 📖 10 💬 3

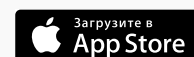
Аккаунт

Разделы

Информация

Услуги

Приложения

[Войти](#)[Публикации](#)[О сайте](#)[Реклама](#)[Регистрация](#)[Хабы](#)[Правила](#)[Тарифы](#)

[Компании](#)

[Помощь](#)

[Контент](#)

[Пользователи](#)

[Соглашение](#)

[Семинары](#)

[Песочница](#)

[Конфиденциальность](#)



© 2006 – 2017 «ТМ»

[Служба поддержки](#)

[Мобильная версия](#)

