

Управление требованиями

Почему требования - это важно

- Недостаточный объем информации, поступающей от пользователей
- требования, сформулированные не полностью
- кардинальное изменение требований

Влекут:

Проект выполняется **не вовремя** и не в рамках **бюджета**

Нигде более, как на стадии сбора требований, так тесно не связаны интересы всех заинтересованных в проекте лиц с успехом проекта.

Заинтересованные лица:

Заказчики, пользователи, аналитики требований, разработчики, тестеры, менеджер по проекту,

Основные возникающие проблемы:

Разработчики не умеют, спокойно и профессионально собирать требования

У клиентов не хватает терпения, не понимают важность этого процесса (пример параграф, что тут собирать и так все понятно). Пример аутсорсинга

Пример из книги первый

Время общения = времени работы с компьютером

Что бывает из-за недостаточного внимания:

- Клиент не досказал = разработчик не догадался (пример)
- Клиент прерывает работу из-за необходимости изменения, когда не хватает основного функционала
- Приходится модифицировать систему из-за того что она не выполняет задачи о которых мы говорили в начале разработки (документировать требования, вести протоколы встречи, пример тс)

Результат

Получилось:

вы получите отличный продукт, восхищенных заказчиков и удовлетворенных разработчиков (повезло)

Не получилось:

непонимание, разочарование и разногласия, которые подрывают веру в продукт и его ценность. Репутация

Все заинтересованные лица должны быть вовлечены в составление этого документа

Из-за чего возникают проблемы:

- Проблемы возникают из-за несовершенства способов сбора, документирования, модификации и согласования требований к ПО
- Предполагаемая функциональность
- Ошибочные, несогласованные предположения (пример НУЦ)
- Недостаточно определенных требований
- Бессистемного изменения процесса

Что бывает, из-за несовершенства процесса:

- Образ и **границы проекта** никогда не определены ясно.
- Заказчики **очень заняты**, чтобы работать с аналитиками и программистами над требованиями.
- **Заместители пользователей** - менеджеры по продуктам, по разработке, менеджеры пользователей или маркетологи - вызываются говорить от имени клиентов, но не точно представляют их потребности.
- Требования существуют только в головах «экспертов», работающих в вашей организации, и никогда не фиксируются в письменном виде.
- Заказчики настаивают, чтобы критиковались все требования, без учета их приоритетов.
- Разработчики получают двусмысленную и неполную информацию, поэтому при кодировании им приходится делать предположения.
- Взаимодействие между разработчиками и заказчиками ограничивается внешним видом пользовательского интерфейса и не затрагивает того, что же действительно клиенты собираются делать с помощью приложения.

Что бывает, из-за несовершенства процесса:

- Образ и **границы проекта** никогда не определены ясно.
- Заказчики **очень заняты**, чтобы работать с аналитиками и программистами над требованиями.
- **Заместители пользователей** - менеджеры по продуктам, по разработке, менеджеры пользователей или маркетологи - вызываются говорить от имени клиентов, но не точно представляют их потребности.
- Требования существуют только в головах «экспертов», работающих в вашей организации, и никогда не фиксируются в письменном виде.
- Заказчики настаивают, чтобы критиковались все требования, без учета их приоритетов.
- Разработчики получают двусмысленную и неполную информацию, поэтому при кодировании им приходится делать предположения.
- Взаимодействие между разработчиками и заказчиками ограничивается внешним видом пользовательского интерфейса и не затрагивает того, что же действительно клиенты собираются делать с помощью приложения.

- Ваши заказчики подписывают требования и затем постоянно изменяют их.
- Проект разрастается, когда вы принимаете изменения требований, график при этом нарушается, потому что никаких дополнительных ресурсов не выделяется и никакие функции не удаляются.
- Запросы на изменение требований теряются по пути: ни вы, ни ваши клиенты не знают статус каждого запроса.
- Заказчики настаивают на определенной функциональности, которую разработчики и создают, однако эти возможности системы клиентам не нужны.
- Технические требования хороши, а вот заказчики нет.

- Ваши заказчики подписывают требования и затем постоянно изменяют их.
- Проект разрастается, когда вы принимаете изменения требований, график при этом нарушается, потому что никаких дополнительных ресурсов не выделяется и никакие функции не удаляются.
- Запросы на изменение требований теряются по пути: ни вы, ни ваши клиенты не знают статус каждого запроса.
- Заказчики настаивают на определенной функциональности, которую разработчики и создают, однако эти возможности системы клиентам не нужны.
- Технические требования хороши, а вот заказчики нет.

Что такое требования?

Требования это:

В IEEE Standard Glossary of Software Engineering Terminology (1990)

Требование - это:

- условия или возможности, необходимые пользователю для решения проблем или достижения целей;
- условия или возможности, которыми должна обладать система или системные компоненты, чтобы выполнить контракт или удовлетворять стандартам, спецификациям или другим формальным документам;
- документированное представление условий или возможностей для пунктов 1 и 2.

Заказчики часто думают, что это:

Концепция продукта для разработчиков (бывает наоборот, сделайте нам бесплатно)

Разработка интерфейса пользователя

Для чего нужны:

Требования нужны в частности для того, чтобы Разработчик мог определить и согласовать с Заказчиком временные и финансовые перспективы проекта автоматизации.

Уровни требований:

- Бизнес-требования
- Требования пользователей
- Функциональные требования



Бизнес-требования (business requirements)

содержат высокоуровневые цели организации или заказчиков системы. Как правило, их высказывают те, кто финансируют проект, покупатели системы, менеджер реальных пользователей, отдел маркетинга или «ясновидец» (специалист по системам машинного зрения). В этом документе объясняется, почему организации нужна такая система, то есть описаны цели, которые организация намерена достичь с ее помощью. Бизнес-требования можно в форме документа об образе и границах проекта.

Пример:

- ускорение и автоматизация процесса обработки заказов с момента поступления его на производство и до момента сдачи заказа клиенту;
- Повышение качества обслуживания клиентов

Требования пользователей (user requirements)

описывают цели и задачи, которые пользователям позволит решить система. К отличным способам представления этого вида требований относятся варианты использования, сценарии и таблицы «событие — отклик». Таким образом, в этом документе указано, что клиенты смогут делать с помощью системы.

Пример использования:

«Сделать заказ» для заказа билетов на самолет, аренды автомобиля, заказа гостиницы через Интернет.

Найти документы

Функциональные требования (functional requirements)

Определяют функциональность ПО, которую разработчики должны построить, чтобы пользователи смогли выполнить свои задачи в рамках бизнес-требований. Иногда именуемые требованиями поведения (behavioral requirements), они содержат положения с традиционным «должен» или «должна».

Пример:

«Система должна по электронной почте отправлять пользователю подтверждение о заказе».

Системные требования (system requirements)

Обозначают высокоуровневые требования к продукту, которые содержат многие подсистемы, то есть система (IEEE, 1998с). Говоря о системе, мы под понимаем программное обеспечение или подсистемы ПО и оборудования. Люди — часть системы, поэтому определенные функции системы могут распространяться и на людей.

Пример:

Система автоматизировала скучное добавление точного количества вещества в ряд мензурок. Из требований для всей системы мы вывели производные функциональные требования к ПО, чтобы посылать сигналы оборудованию для перемещения распределяющих вещество насадок, считывания показаний датчиков положения и включения/выключения насоса.

Бизнес-правила (business rules)

включают корпоративные политики, правительственные постановления, промышленные стандарты и вычислительные алгоритмы. Бизнес-правила не являются требованиями к ПО, потому что они находятся снаружи границ любой системы ПО. Однако они часто налагают ограничения, определяя, кто может выполнять конкретные варианты использования, или диктовать, какими функциями должна обладать система, подчиняющаяся соответствующим правилам. Иногда бизнес-правила становятся источником атрибутов качества, которые реализуются в функциональности. Следовательно, вы можете отследить происхождение конкретных функциональных требований вплоть до соответствующих им бизнес-правил.

Пример:

Постановление о работе со счетами-фактурами в электронном виде
Порядок хранения банковской документации

Software requirements specification, SRS

Функциональные требования документируются в *спецификации требований к ПО*, где описывается так полно, как необходимо, ожидаемое поведение системы.

Спецификация требований к ПО используется:

- при разработке,
- тестировании,
- гарантии качества продукта,
- управлении проектом
- связанных с проектом функциях.

Атрибуты качества (quality attributes)

В дополнение к функциональным требованиям спецификация содержит нефункциональные, где описаны цели и атрибуты качества. *Атрибуты качества (quality attributes) представляют собой дополнительное описание функций продукта, выраженное через описание его характеристик, важных для пользователей или разработчиков.*

Пример:

- легкость и простота использования, легкость перемещения, целостность, эффективность и устойчивость к сбоям;
- ограничения дизайна и реализации. *Ограничения (constraints) касаются выбора* возможности разработки внешнего вида и структуры продукта (Брендбук)

Пример:

Бизнес-требование может выглядеть так:

«Продукт позволит пользователям исправлять орфографические ошибки в тексте эффективно». На коробке CD-ROM указано, что проверка грамматики включена как характеристика, удовлетворяющая бизнес-требования.

Требования пользователей могут содержать задачи (варианты использования):

«Найдите орфографическую ошибку» или «Добавьте слово в общий словарь».

Функциональные требования:

- поиск и выделение слова с ошибкой,
- отображение диалогового окна с фрагментом текста, где это слово находится
- замена слова с ошибкой корректным вариантом по всему тексту.

Атрибут качества

Легкость

простота использования (usability)

как раз и определяет значение словом «эффективно» в бизнес-требованиях.

В спецификации требований к ПО не должно быть:

деталей дизайна или реализации (кроме известных ограничений),
данных о планировании проекта;
сведений о тестировании;
описание среда разработки;
ограничения бюджета;
Руководство пользователя;
требования для выпуска продукта и продвижения;
его в поддерживаемую среду.

Все это относится к требованиям к проекту, но не к продукту,

Разработка и управление требованиями картинка



Разработка требований состоит:

Извлечение (elicitation), анализ (analysis), документирование (specification) и утверждение (validation).

В эти подэтапы входят все действия, включающие сбор, оценку и документирование требований для ПО, в том числе:

- идентификация классов пользователей для данного продукта;
- выяснение потребностей тех, кто представляет каждый класс пользователей;
- определение задач и целей пользователей, а также бизнес-целей, с которыми эти задачи связаны;
- анализ информации, полученной от пользователей, чтобы отделить задачи от функциональных и нефункциональных требований, бизнес - правил, предполагаемых решений и поступающих извне данных;
- установление относительной важности атрибутов качества;
- установление приоритетов реализации;
- документирование собранной информации и построение моделей;
- просмотр спецификации требований, который позволяет удостовериться в том, что запросы пользователей всеми понимаются одинаково, и устранение возникших проблем до передачи документа разработчикам.

Управление требованиями

Включает выработку и поддержание взаимного согласия с заказчиками по поводу требований к разрабатываемому ПО». Соглашение воплощается в спецификации (в письменной форме) и моделях.

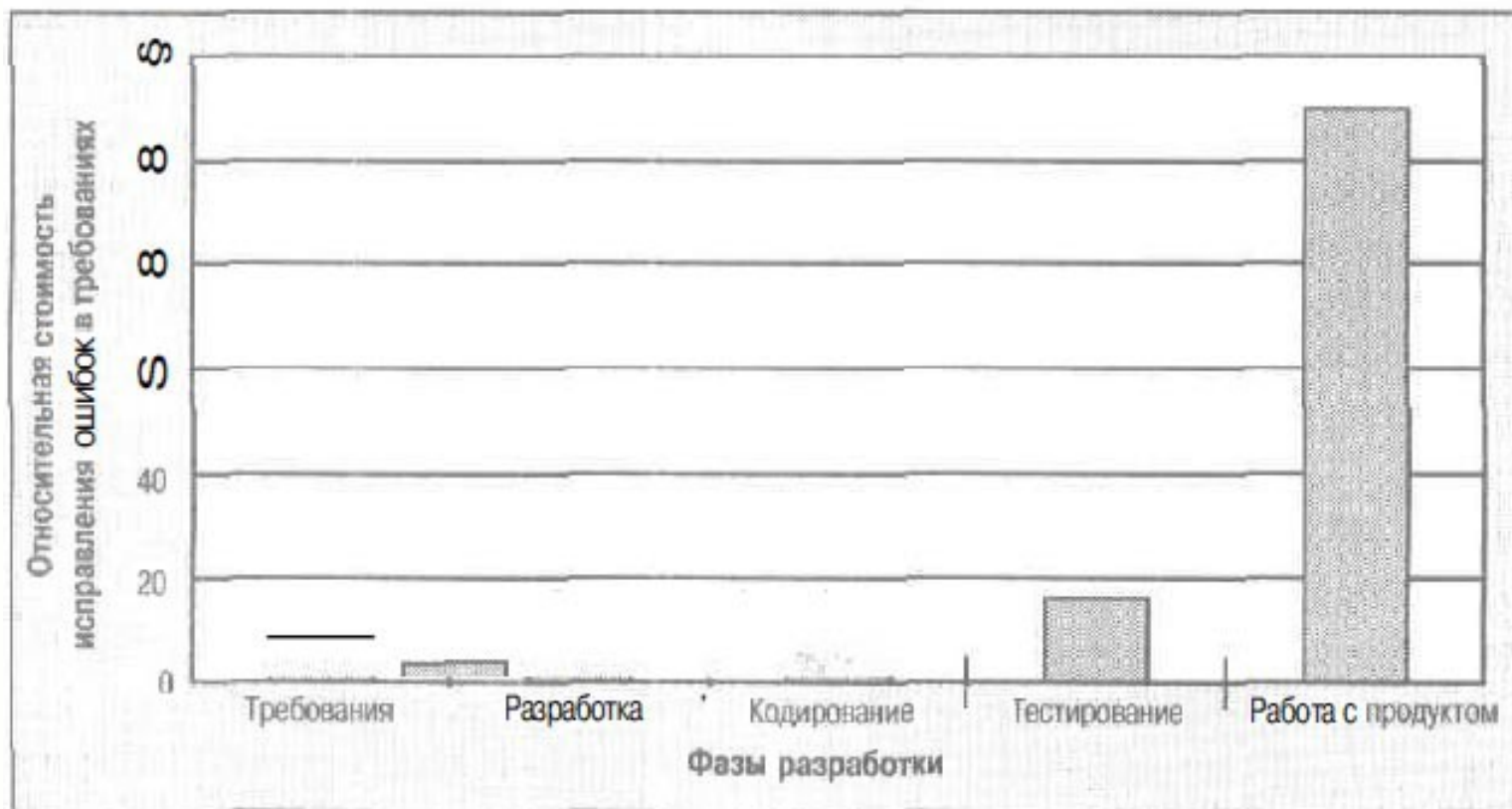
Пользователи и разработчики должны принять задокументированные требования и высказаться за создание этого продукта.

К действиям по управлению требованиями относятся:

- определение основной версии требований;
- просмотр предлагаемых изменений требований и оценка вероятности воздействия каждого изменения до его принятия;
- включение одобренных изменений требований в проект установленным способом;
- согласование плана проекта с требованиями;
- обсуждение новых обязательств, основанных на оцененном влиянии изменения требований;
- отслеживание отдельных требований до их дизайна, исходного кода и вариантов тестирования;
- отслеживание статуса требований и действий по изменению на протяжении всего проекта.

Недоработанные требования

Основное следствие проблем с требованиями — переделка того, что, как вы думаете, уже готово. На это расходуется от 30 до 50% общего бюджета разработки, а ошибки в требованиях стоят от 70 до 85% стоимости переделки.



Недостаточное вовлечение пользователей

Заказчики зачастую не понимают, почему так важно тщательно собирать требования и обеспечить их качество.

Разработчики не всегда придают значение вовлечению пользователей в процесс, из-за того что среди них больше фанатов написания кода, а не любителей возиться с клиентами или же потому что они считают, что все уже знают и потребностях пользователей.

В любом случае трудно добраться до людей, которые непосредственно будут иметь дело с продуктом, выразители мнения пользователей не всегда понимают, что тем нужно в реальности.

Разрастание требований пользователей

Так как требования тщательно прорабатываются и их объем со временем увеличивается, проект часто выходит за установленные рамки как по срокам, так и по бюджету.

Первоначально принятые планы не всегда основаны на реалистичном понимании размера и сложности требований, ограничение же на модификацию только усиливает проблемы.

Чтобы управлять границами требований, нужно:

- уточните бизнес-цели проекта, основы его образа, рамки, ограничения, критерии успеха и ожидаемую пользу;
- Оцените, как предполагаемые характеристики или изменения требований отразятся на связанной с ними структуре.

Изменения зачастую критически важны для успеха, однако они всегда имеют цену.

Двусмысленность требований

- Пользователь имеет возможность интерпретировать одно и то же положение по-разному.
- У нескольких читателей требований возникает разное представление о продукте.

Пример:

Двусмысленность зачастую проистекает из неточности и плохой детализации требований, в результате чего разработчикам приходится заполнять возникающие пробелы собственными силами.

Двусмысленность ведет и к формированию различных ожиданий у заинтересованных лиц:

- Клиенты удивляются «Что же это получилось?»
- Разработчики же впустую тратят время, занимаясь не теми проблемами.
- А тестеры готовятся к проверке не тех особенностей поведения системы.

«Золочение» продукта

Под «золочением» понимают такие ситуации, когда разработчики добавляют функции, которых нет в спецификации, но им кажется, что это понравится пользователям.

Пользователи иногда требуют функции или элементы интерфейса которые выглядят отлично, но не представляют особой ценности для продукта.

Все, что вы захотите добавить, стоит времени и денег, поэтому постарайтесь осознать ценность своевременного выпуска продукта.

Пример:

Минимальная спецификация

Иногда сотрудников отдела маркетинга или менеджеров охватывает искушение создать урезанный вариант спецификации, как, например набросок концепций продукта на салфетке.

Они ожидают, что разработчики «**нарастят мясо**» на основе этих набросков, пока проект развивается.

Пропуск классов пользователей

Большинство продуктов предназначены для нескольких групп пользователей, которые могут применять различные наборы функций с разной частотой и имеют опыт работы с ПО самого широкого диапазона.

Если вы не определили важные классы пользователей для вашего продукта заранее, некоторые потребности клиентов не будут учтены.

Небрежное планирование

«Я кое-что придумал для нового продукта. Когда вы сможете это сделать?»

Не отвечайте на подобный вопрос, пока больше не узнаете о проблеме. Неопределенные, недетализированные требования порождают слишком оптимистические оценки: они «выходят боком», когда возникает перерасход.

Неподготовленная оценка звучит как обязательство для слушателя.

Выгоды разработки спецификации требований:

- меньше дефектов в требованиях;
- меньше переделок;
- меньше ненужных функций;
- ниже стоимость модификации;
- быстрее разработка;
- меньше разобщенности;
- меньше расползания границ;
- меньше беспорядка в проекте;
- точнее оценки тестирования;
- выше удовлетворение заказчиков и разработчиков.

Характеристики отдельных положений спецификации требований

Полнота

Каждое требование должно полно описывать функциональность, которую следует реализовать в продукте. То есть оно должно содержать всю информацию, необходимую для разработчиков, чтобы тем удалось создать этот фрагмент функциональности.

Пример:

Корректность

Каждое требование должно точно описывать желаемую функциональность. Для соблюдения корректности необходима связь с источниками требований, например с пожеланиями пользователей.

Пример:

Осуществимость

Необходима возможность реализовать каждое требование при известных условиях, ограничениях системы и операционной среды.

Чтобы не придумывать недостижимые положения, обеспечьте взаимодействие разработчиков с маркетологами и аналитиками требованиями на период всего извлечения требований.

Пример:

Необходимость

Каждое требование должно отражать возможность, которая действительно необходима пользователям или которая нужна для соответствия внешним системным требованиям или стандартам.

Пример:

Назначение приоритетов

Назначьте приоритеты каждому функциональному требованию, характеристике или варианту использования, чтобы определить, что необходимо для каждой версии продукта.

Пример:

Недвусмысленность

Все читатели требований должны интерпретировать их одинаково, но естественный язык зачастую грешит многозначностью. Пишите документацию просто, кратко и точно, применяя лексику, понятную пользователям.

Пример:

Проверяемость

Попробуйте разработать несколько тестов или примените другие приемы для проверки, например экспертизу или демонстрации, чтобы установить, действительно ли в продукте реализовано каждое требование.

Пример:

Характеристики спецификации требований

Недостаточно получить прекрасные отдельные положения. Набор требований, составляющий спецификацию должен отвечать характеристикам, описанным в следующих разделах

1. Полнота

Никакие требования или необходимые данные не должны быть пропущены.

2. Согласованность

Согласованные требования не конфликтуют с другими требованиями такого же типа или с высокоуровневыми пользовательскими, системными или бизнес-требованиями.

3. Способность к модификации

Необходимо обеспечить возможность переработки требований, если понадобится, и поддерживать историю изменений для каждого положения. Для этого все они должны быть уникально помечены и обозначены, чтобы вы могли ссылаться на них однозначно. Каждое требование должно быть записано в спецификации только единожды.

Трассируемость

Трассируемость, или возможность для анализа, можно реализовать как в направлении назад, к первоисточникам, так и вперед, к элементам дизайна и исходному коду, который его реализует, а также к вариантам использования, которые позволяют проверить корректность, реализации.

Вам никогда не удастся создать спецификацию, в которой бы про являлись все эти атрибуты. Однако если вы будете помнить о них при написании и просмотре требований, вы создадите более качественный документ, а значит, и продукт.