

Утверждение требований

Предмет: Управление требованиями

Утверждение требований позволяет удостовериться:

- в спецификации требований к ПО **должным образом описаны предполагаемые возможности** и характеристики системы, которые удовлетворяют потребности различных заинтересованных в проекте лиц;
- требования к ПО точно **отражают системные требования**, бизнес-правила и др.;
- **требования полные** и высококачественные;
- все требования **согласованы** друг с другом;
- требования **обеспечивают качественную основу** для дизайна и сборки ПО.

Утверждение требований может выполняться:

- после каждой процедуры сбора информации, ее анализа и документирования
- перед финальной версией спецификации выполняется официальная проверка спецификации требований к ПО

Включите в план проекта этапы утверждения требований в качестве отдельных задач.

Всякое исследование продукта ПО на предмет выявления проблем любым другим лицом, кроме его автора, называется **экспертной оценкой (peer review)**.

Пример плана проекта

При оценке уровня детализации многие руководители используют правило "40 часов", которое заключается в том, что все задачи должны быть выполнены в течение 40 часов. Но гораздо эффективнее будет включение задач по подготовке и документации.

Сбор требований, форма «счета к оплате» (40 часов)

Сбор требований, применение наличных денег (40 часов)

Сбор требований, счета к получению (40 часов)

Завершение и подпись требований (40 часов)

Но вам стоит рассмотреть план следующим образом:

8 встреч – требования к форме «счета к оплате», по 2 часа на каждую (16 часов)

Подготовка – 1 час на встречу (8 часов)

Документация результатов – 4 часа на встречу (32 часа)

Утверждение требований – 2 встречи по 2 часа

Виды неофициальных просмотров требований:

- проверка «за столом», когда вы просите **одного коллегу** исследовать ваш продукт;
- коллективная проверка, когда вы приглашаете **несколько коллег** для параллельной проверки продукта;
- критический анализ, когда **автор описывает создаваемый продукт** и просит его прокомментировать.

Официальная проверка представляет собой строго регламентированный процесс.

По его завершении **формируется отчет**, в котором указаны материал, рецензенты и мнение команды рецензентов о приемлемости продукта. **Главный результат**— совокупность всех найденных дефектов и поднятых вопросов.

Наиболее хорошо себя зарекомендовавшая себя форма официальной оценки называется **экспертизой**.

Начинайте экспертизу спецификации требований к ПО, когда требования разработаны еще примерно на 10%.

Экспертиза — это четко определенный многоэтапный процесс.

В нем участвует небольшая команда квалифицированных специалистов, которые тщательно **проверяют продукт на наличие дефектов** и изучают возможности улучшения.

Участники инспектирования:

Автор продукта. Это может быть аналитик, составивший документацию требований.

Автор любого предыдущего продукта или спецификации для проверки. В этой роли может выступить системный инженер или архитектор, который способен подтвердить, что в спецификации требований к ПО система описана достаточно детально. В экспертизе могут принимать участие представители клиента для того, чтобы подтвердить, что требования в спецификации требований к ПО описаны правильно и полно.

Участники проекта, основанную на проверяемом элементе.

Разработчик, тестировщик, менеджера проекта, а также составителя пользовательской документации. Они будут выявлять наличие проблем различных типов.

Требование, которое не поддается проверке, скорее всего сможет выявить тестировщик, а разработчик сумеет определить требования, которые технически невыполнимы.

Клиенты

Постарайтесь **ограничить команду шестью участниками** или меньше.

Большие команды могут легко увязнуть в **посторонних дискуссиях**, попытках решения проблем и дебатах о том, что считать ошибкой.

При этом **снижается темп проверки** и растет стоимость обнаружения каждого дефекта.

Роли экспертов

Все участники экспертизы, включая автора, ищут недостатки и возможности улучшения. Отдельные участники при этом играют различные роли.

Автор

Автор создает или поддерживает проверяемый продукт.

В роли автора спецификации требований к ПО, как правило, выступает аналитик, который осуществлял сбор требований клиента и писал спецификацию.

В ходе неофициальных проверок, таких, как критический анализ, именно автор часто возглавляет дискуссию.

Однако в экспертизе он принимает более пассивное участие.

Зачастую автору удается обнаружить ошибки, незамеченные другими участниками.

Координатор

Координатор, или руководитель проверки, планирует экспертизу совместно с автором, согласовывает особенности и организует совещания.

Координатор распределяет проверяемый материал между участниками за несколько дней до совещания. Он отвечает за своевременное начало совещаний, поощрение вклада каждого участника и управление дискуссией, которая должна быть направлена на выявление дефектов, а не на решение проблем. Кроме того, в его обязанности входит информировать менеджеров или того, кто занимается сбором результатов различных экспертиз, о результатах.

Читатель

Один из проверяющих выступает в роли читателя. На совещании он перефразирует положения спецификации — по одному требованию за раз. Затем другие участники указывают на потенциальные дефекты и проблемы. Выражая требование своими словами, читатель дает трактовку требованиям, которая может отличаться от интерпретации других членов команды.

Секретарь

Секретарь, или клерк, использует стандартные формы для документирования возникающих вопросов и дефектов, выявленных в ходе совещания.

Секретарь должен вслух прочесть свои записи, чтобы убедиться в их точности.

Другие участники инспектирования должны помочь ему зафиксировать суть каждой проблемы таким образом, чтобы автор смог четко уяснить природу и местоположение каждой проблемы.

Входные критерии

Вы готовы к экспертизе документа о требованиях в том случае, если он удовлетворяет определенным предварительным условиям.

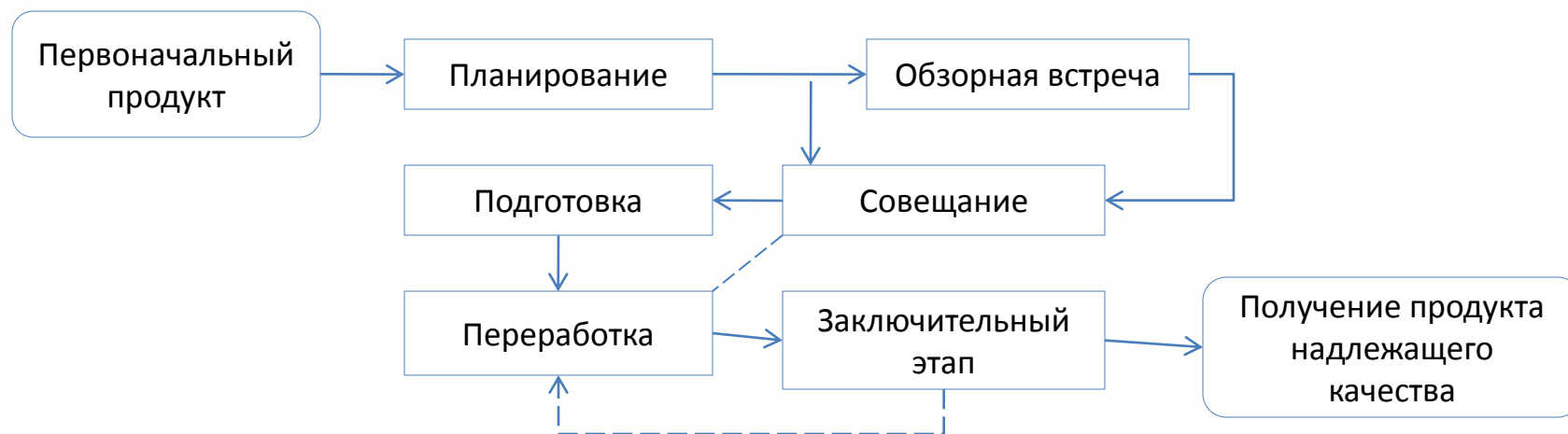
Набор входных критериев (entry criteria) четко обрисовывает авторам то, как следует подготовиться к инспектированию.

Примеры входных критериев для составления документации по требованиям:

- документ должен соответствовать шаблону;
- в документе выполнена проверка правописания;
- автор просмотрел документ на наличие ошибок в макете;
- номера строк следует напечатать в документе, чтобы облегчить
- ссылки на определенные элементы в течении совещания;
- все нерешенные вопросы помечаются значком «TBD» (необходимо определить);

Этапы экспертизы

Экспертиза— это многоэтапный процесс.



Планирование

Экспертизу совместно планируют автор и координатор.

Они определяют:

- состав участников,
- материалы, которые проверяющие должны получить до начала первого совещания,
- количество совещаний, необходимых для охвата всего материала

Количество обнаруженных дефектов очень зависит от темпов работы.

Чем медленнее изучается спецификация требований к ПО, тем **больше дефектов удастся выявить.**

Поскольку ни одна команда не может тратить бесконечное количество времени на проверку требований, **выберите подходящий темп**, принимая во внимание риск пропуска важных дефектов.

При выборе темпа работы учитывайте следующие факторы:

- дату предыдущей экспертизы;
- объем текста на каждой странице;
- сложность спецификации;
- риск не заметить ошибку;
- насколько критично для успеха проекта проверить весь этот материал;
- квалификацию автора спецификации требований к ПО.

Обзорная встреча

Во время обзорной встречи **автор описывает исходные данные материала**, предназначенного для проверки, предположения, которые он сделал, и его личные цели.

Если все проверяющие хорошо знакомы с проверяемым материалом, вы можете пропустить эту стадию.

Подготовка

До совещания каждый из **экспертов исследует продукт, чтобы определить возможные дефекты и возникающие вопросы**; для этого они используют контрольные **списки типичных дефектов** и другие приемы анализа. До 75% дефектов, обнаруживаемых при экспертизе, выявляются на этапе подготовки, поэтому этот этап пропускать нельзя

Совещание

На совещании **читатель знакомит членов команды с каждым требованием** спецификации — по одному за раз, перефразируя их своими словами.

По мере обсуждения дефектов и других проблем, **секретарь заносит их в форму**, которая для автора требований становится руководством к действию.

Совещание **не должно длиться более двух часов**, поскольку работу уставших людей нельзя назвать продуктивной (запланировать еще одно совещание).

В заключение совещания команда принимает решение: следует ли принять документ без изменений, с незначительными поправками или указать на необходимость значительной переработки.

Переработка

Практически на каждом этапе работы, связанным с контролем качеством, **выявляются определенные дефекты.**

После совещания автору следует отвести определенное время на **переработку требований.**

Заключительный этап

На этом заключительном этапе экспертизы координатор или специально назначенный член команды работает с автором, чтобы убедиться, что все поднятые **проблемы разрешены и все ошибки исправлены** соответствующим образом.

Заключительный этап завершает процесс инспектирования, на этом этапе координатор определяет, **удовлетворяют ли выходные критерии экспертизы.**

Выходные критерии

В ходе экспертизы определяются выходные критерии (exit criteria), которые необходимо одобрить, прежде чем координатор объявит о том, что проверка закончена.

Вот перечень стандартных критериев:

- все возникшие вопросы были сняты;
- все изменения, внесенные в документ и соответствующие продукты, были внесены корректно;
- в измененном документе проверено правописание;
- все проблемы, помеченные «TBD», разрешены, при этом фиксировались данные по разрешению каждого, дата и исполнитель;

Контрольные списки дефектов

Для того чтобы помочь экспертам обнаружить типичные ошибки в инспектируемых продуктах, **разработайте контрольный список дефектов** для каждого типа документа о требованиях, создаваемого в вашей компании.

С помощью таких контрольных списков вы привлекаете внимание проверяющих к **часто возникающим проблемам с требованиями.**

Сократите списки, сообразуясь с нуждами вашей компании, и измените их в соответствии с тем, какие **проблемы в ваших требованиях повторяются чаще всего.**

Вы можете попросить некоторых инспекторов при подготовке использовать **разные поднаборы** контрольных списков (под систему).

Пример контрольного списка

1. Является ли конкретный вариант использования продукта автономной и отдельной задачей?
2. Ясно ли сформулирована цель или измеряемое значение для конкретного варианта использования продукта?
3. Очевидно ли, кто получит преимущества от этого варианта использования?
4. Написан ли вариант использования на базовом уровне, без ненужных деталей, связанных с дизайном и реализацией?
5. Все ли предполагаемые альтернативные направления задокументированы?
6. Все ли известные условия исключения задокументированы?
7. Есть ли какие-либо последовательности действий, которые можно разделить на отдельные варианты использования?
8. Все ли этапы каждого направления записаны в ясной, недвусмысленной и полной форме?
9. Каждый ли исполнитель и стадия варианта использования продукта соответствуют выполнению конкретной задачи?
10. Осуществимо ли и поддается ли проверке каждое альтернативное направление, определенное в варианте использования?
11. Должным ли образом структурируют вариант использования выходные и входные условия?

Организация и завершенность

1. Корректны ли все ли внутренние перекрестные ссылки на другие документы?
2. Написаны ли все ли требования на одном и том же соответствующем уровне детализации?
3. Обеспечивают ли требования адекватную основу для дизайна?
4. Указан ли приоритет реализации каждого требования?
5. Определены ли все внешние интерфейсы оборудования, ПО и взаимодействия?
6. Определены ли все алгоритмы, соответствующие функциональным требованиям?
7. Включены ли в спецификацию требований к ПО все известные потребности клиента или системы?
8. Отсутствует ли в требовании какая-либо необходимая информация? Если да, помечена ли она значком «TBD»?
9. Задokumentировано ли ожидаемое поведение системы для всех предполагаемых ошибочных условий?

Корректность

1. Конфликтуют ли какие-либо требования с другими требованиями или повторяют их?
2. Написано ли каждое требования ясным, точным и недвусмысленным языком?
3. Можно ли проверить каждое требование с помощью тестирования, демонстрации, просмотра или анализа?
4. Не выходит ли какое-либо требование за границы проекта?
5. Нет ли в каком-либо требовании тематических или грамматических ошибок? Можно ли реализовать все требования, не выходя за рамки установленных ограничений?

Атрибуты качества

1. Все ли задачи, связанные с производительностью, сформулированы соответствующим образом?
2. Все ли соображения, касающиеся охраны труда и безопасности, сформулированы соответствующим образом?
3. сформулированы соответствующим образом?

Возможность отслеживания

1. Каждое ли требование идентифицировано уникально и корректно?
2. Прослежено ли каждое функциональное требование до более высокого уровня (например, требования к системе или вариант использования)?

Особые проблемы

1. Все ли требования можно отнести к именно к требованиям, а не к решениям разработки или реализации?
2. Идентифицированы ли все функции, зависящие от времени, и определены ли их критерии?
3. Были ли рассмотрены соответствующим образом все проблемы, связанные с интернационализацией?

Типичные проблемы при просмотре требований

Большой объем документации

Перспективу тщательной проверки спецификации требований к ПО, состоящей из **нескольких сотен страниц**, можно смело назвать ужасающей.

Вам покажется заманчивым **пропустить проверку** и сразу приступить к конструированию — тоже не самый мудрый выбор.

Даже в спецификации среднего размера, как правило, **все эксперты тщательно проверяют первую часть**, несколько наиболее стойких изучат середину, но маловероятно, чтобы кто-то дошел до конца.

Решение:

- Чтобы избежать перегрузок, **проверяйте документ по мере его разработки**, не ждите, когда он будет закончен.
- **Определите области повышенного риска** (места где сложная логика), которые требуют тщательной экспертизы — для менее важных фрагментов хватит и неофициального просмотра.
- Попросите определенных инспекторов **начать с различных частей документа**, это позволит окинуть каждую страницу свежим взглядом.

Большая команда экспертов

Многие участники проекта и клиенты хотят заниматься требованиями, поэтому **список желающих** может оказаться весьма длинным. Однако **чем больше команда экспертов, тем труднее составить график совещаний**, тем больше на них посторонних разговоров и тем труднее прийти к согласию по каждому вопросу.

Решение:

- четко представьте себе, чью точку зрения представляет каждый участник. Тактично **отклоните участие лиц**, готовых отстаивать точку зрения, которая уже представлена.
- соберите несколько небольших групп для параллельной проверки спецификации требований к ПО и **объедините их контрольные списки дефектов в один**, удалив все повторы.

Географическое разделение инспекторов

Все больше компаний разрабатывают продукты, **привлекая команды, разъединенные географически**. В этом случае проверки проводить труднее.

Видеоконференции могут стать эффективным решением, но в телеконференциях вы не можете воспринимать язык тела и мимику других участников, как при личных встречах.

Решение

В этом случае рецензенты используют **функции текстового редактора, чтобы ввести свои комментарии в проверяемый документ**. Каждый комментарий помечается инициалами эксперта, таким образом, любой может ознакомиться с мнением других рецензентов.

Тестирование требований

Разработка вариантов тестирования поможет **выявить множество проблем** с требованиями, даже если вы не станете выполнять тесты.

При написании вариантов тестирования (функциональность) кристаллизуется ваше понимание того, как система **должна себя вести при определенных условиях**.

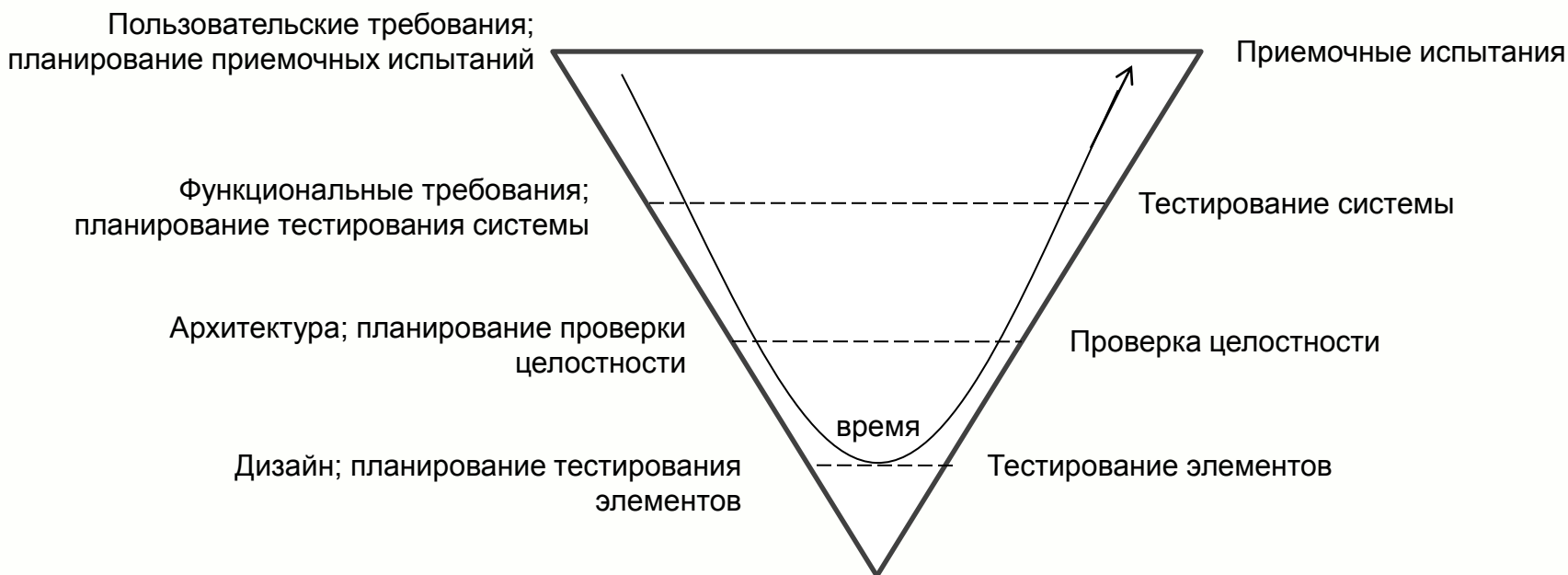
Когда начинать тестирование?

Начинайте планирование тестирования и разработку вариантов **тестирования на ранней стадии проекта**, это позволит обнаружить многие ошибки вскоре после их появления.

А также **предотвратит дополнительный ущерб**, наносимый ими, и снизятся затраты на тестирование и обслуживание.

V-образная модель разработки ПО

Действия, связанные с **тестированием и разработкой проекта** выполняются параллельно.



Эта модель указывает, что:

- приемочные испытания основывается на требованиях пользователей
- тестирование системы — на функциональных требованиях
- тестирование целостности — на архитектуре системы

В процессе разработки требований вы **не можете проводить никаких тестов**, так как никакого ПО еще нет.

Однако **концептуальные варианты тестирования**, созданные на основе требований, позволят выявить ошибки, неясности и пропуски данных в спецификации требований к ПО и проанализировать модели задолго до того, как разработчики приступят к написанию кода.

Примера -вариант использования «Просмотр заказа» (View Order)

Концептуальные **варианты тестирования** могут быть следующими:

1. пользователь вводит номер заказа, который он хочет просмотреть, этот заказ существует, пользователь разместил заказ. Ожидаемый результат: отображение заказа детально;
2. пользователь вводит номер заказа, который он хочет просмотреть, этот заказ не существует. Ожидаемый результат: появляется сообщение: «Сожалею, заказ не найден»
3. пользователь вводит номер заказа, который он хочет просмотреть, этот заказ существует, но он размещен не этим пользователем. Ожидаемый результат: появляется сообщение: «Сожалею, это не ваш заказ».

В идеале разработчик пишет функциональные требования, а тестировщик — варианты тестирования на основе одних и тех же материалов— пользовательских требований.

