**Требования**

**Что такое «требование»**

Как мы только что рассмотрели в главе, посвящённой жизненному циклу тестирования, всё так или иначе начинается с документации и требований.

Требование — описание того, какие функции и с соблюдением каких условий должно выполнять приложение в процессе решения полезной для пользователя задачи.

Тестирование программного обеспечения. Базовый курс. 2-е издание.

**Важность требований**

Требования являются отправной точкой для определения того, что проектная команда будет проектировать, реализовывать и тестировать. Элементарная логика говорит нам, что если в требованиях что-то «не то», то и реализовано будет «не то», т.е. колоссальная работа множества людей будет выполнена впустую.

Описывая важность требований, подчёркивается, что они:

* Позволяют понять, что и с соблюдением каких условий система должна делать.
* Предоставляют возможность оценить масштаб изменений и управлять изменениями.
* Являются основой для формирования плана проекта (в том числе плана тестирования).
* Помогают предотвращать или разрешать конфликтные ситуации.
* Упрощают расстановку приоритетов в наборе задач.
* Позволяют объективно оценить степень прогресса в разработке проекта.

Вне зависимости от того, какая модель разработки ПО используется на проекте, чем позже будет обнаружена проблема, тем сложнее и дороже будет её решение. А в самом начале («водопада», «спуска по букве v», «итерации», «витка спирали») идёт планирование и работа с требованиями.

Если проблема в требованиях будет выяснена на этой стадии, её решение может свестись к исправлению пары слов в тексте, в то время как недоработка, вызванная пропущенной проблемой в требованиях и обнаруженная на стадии эксплуатации, может даже полностью уничтожить проект.

В общем случае документацию можно разделить на два больших вида в зависимости от времени и места её использования.

**Продуктная документация**(product documentation, development documentation) используется проектной командой во время разработки и поддержки продукта. Она включает:

* **План проекта** (project management) и в том числе тестовый план (test).
* **Требования к программному продукту**(product requirements document)и функциональные спецификации (functional specifications document, FSD; software requirements specification, SRS).
* **Архитектуру и дизайн**(architecture and design).
* **Тест-кейсы и наборы тест-кейсов**(test cases, test suites).
* **Технические спецификации** (technical specifications), такие как схемы баз данных, описания алгоритмов, интерфейсов и т.д.

**Проектная документация**(project documentation) включает в себя как продуктную документацию, так и некоторые дополнительные виды документации и используется не только на стадии разработки, но и на более ранних и поздних стадиях (например, на стадии внедрения и эксплуатации).

* **Пользовательскую и сопроводительную документацию** (user and accompanying documentation), такую как встроенная помощь, руководство по установке и использованию, лицензионные соглашения и т.д.
* **Маркетинговую документацию** (market requirements document, MRD),которую представители разработчика или заказчика используют как на начальных этапах (для уточнения сути и концепции проекта), так и на финальных этапах развития проекта (для продвижения продукта на рынке).

**Источники и пути выявления требований:**

* Интервью, опросы, анкетирование.
* Мозговой штурм, семинар.
* Наблюдение за производственной деятельностью, «фотографирование» рабочего дня.
* Анализ нормативной документации.
* Анализ моделей деятельности.
* Анализ конкурентных продуктов.
* Анализ статистики использования предыдущих версий системы.
* Совещание.
* Use case.

**Анкетирование**

Данный способ подразумевает под собой составление листа-опросника (анкеты, брифа), который может содержать открытые (требуют от опрашиваемого сформулировать его ответ) и закрытые (требуют от опрашиваемого выбрать ответ из предложенных вариантов) вопросы.

Анкетирование используется для того, чтобы подтвердить или детализировать ранее известные требования, выбрать параметры для решений.

Самым известным примером анкетирования может быть “Бриф на разработку сайта” — анкета содержащая список основных требований и информацию о будущем сайте.

Преимущества:

1. Высокая скорость получения результатов.
2. Сравнительно небольшие материальные затраты.

Недостатки:

1. Данный метод не подходит для выявления неявных требований.
2. При составлении опросника физически невозможно учесть все необходимые вопросы.

**Интервью**

Этот метод известен многим в качестве своего рода беседа “по душам” с заинтересованным лицом, тет-а-тет.

Необходимо задавать открытые вопросы для получения информации и закрытые для того, чтобы подтвердить или опровергнуть конкретные варианты требований.

Данный способ применяется, в основном, для получения информации по какой-либо конкретной теме и/или для уточнения требований.

Многим этот способ может показаться достаточно легким, но это не так. Провести хорошее интервью достаточно сложно. Вы должны гибко реагировать на реакцию интервьюируемого и, в случае необходимости, изменять порядок заготовленных вопросов или их формулировку. Не забудьте включить диктофон во время интервью или вести заметки.

Из плюсов:

1. Возможность задавать вопросы в произвольной последовательности.
2. Возможность использовать вспомогательный материал.
3. Анализ невербальной реакции опрашиваемого человека, позволит сделать дополнительный вывод о достоверности его ответов.

Из минусов:

1. Интервью отнимает достаточно много времени и сил.
2. Дополнительной сложностью является получение одинаковых ответов от интервьюируемых.

**Анализ нормативной документации**  
Данная методика может быть использована при наличии в организации документации, которая может помочь в определении потребностей заказчика. Примеры документации включают в себя: регламенты, описания процессов, структура организации, спецификации продукта, различные процедуры, стандарты и инструкции, шаблоны документов и т.д.

Выявленные требования являются основой для дальнейшего анализа и должны быть детализированы. Данная методика применима, например, при автоматизации устоявшихся в организации регламентированных бизнес процессов.

Плюс:

* Быстрое получение информации.

Минус:

* Данный способ не применим при наличии в компании только базовых документов (или при их полном отсутствии) или, если в компании заказчика не поддерживается актуальность документации.

**Мозговой штурм**

Мозговой штурм — наиболее часто используемый метод получения требований, которые связаны с новыми или плохо изученными направлениями деятельности организации заказчика или функциями системы.

Он позволяет собрать множество идей от различных заинтересованных лиц (стейкхолдеров) в кратчайшие сроки и практически бесплатно.

Во время мозгового штурма участники «накидывают» любые идеи, касающиеся решения данной проблемы.

С помощью этой методики можно проработать несколько различных вариантов решения заданной проблемы, а также разрешить конфликты требований.

Плюсы:

* Позволяет генерировать множество (в том числе и нестандартных) вариантов решений, а также позволяет участникам развивать идеи друг друга.

Минусы:

* Участники мозгового штурма должны быть мотивированы на идеи.
* Трудно применим в распределенных командах.

**Совещание**

Совещание — встреча, ориентированная на обсуждение конкретных вопросов, которые были определены и озвучены участникам заранее.

На такие встречи привлекаются люди, которые придерживаются различных точек зрения по текущей проблеме и могут помочь описать требования, основываясь на взглядах с разных сторон. В процессе совещания уточняется общий список требований, выявляются скрытые требования и решаются конфликты требований.

Совещания являются одной из ключевых практик в Agile, т.к. в них участвуют все заинтересованные в развитии проекта и решении проблемы стороны.

Плюсы:

* Позволяет развить и детализировать требования, определить приоритеты.

Недостатки:

* Сложности в организации встречи, если команда географически разделена, могут возникнуть трудности с присутствием всех необходимых людей на совещании.
* Консенсус необязательно будет достигнут.

**Use case**

Use cases или варианты использования позволяют собрать и сформировать функциональные требования от лица участников. Диаграммы вариантов использования определяют границы решения и показывают связи с внешними системами и участниками.

Метод позволяет детализировать требования с точки зрения пользователей, а также помогает уточнить и систематизировать функционал, который требуется реализовать.

Плюсы:

* Позволяет проработать все варианты развития сценария (основной и альтернативные сценарии).

Минусы:

* Метод не применим для сбора нефункциональных требований.