



СПБГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

Е. Л. Турнечкая

Источники требований

Конспект лекций

СПбГЭТУ «ЛЭТИ», 2023 г.

4.2. ИСТОЧНИКИ ТРЕБОВАНИЙ

4.2.1 Важность требований

Понятие требования определено в глоссариях международных стандартов и российских ГОСТов.

Требование – характеристика того, чем система или элемент конфигурации программного обеспечения должны обладать, чтобы быть приемлемыми для заказчика. (Глоссарий ГОСТ Р 51904-2002 «Программное обеспечение встроенных систем»)

Под требованиями будем понимать описание функциональных возможностей и ограничений, накладываемых на создаваемую программную систему. (Орлов С.А. «Программная инженерия»: учебник для вузов)

Требование – описание того, какие функции и с соблюдением каких условий должно выполнять приложение в процессе решения полезной для пользователя задачи. (Куликов С. С.. Тестирование программного обеспечения. Базовый курс: практик. пособие. Минск: Четыре четверти, 2015)

В нашем курсе мы придерживаемся следующего определения.

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

В литературе 20-летней давности, о пользователях, их задачах и полезных для них свойствах приложения в определении требования не было почти ничего сказано. Пользователь выступал некоей абстрактной фигурой, не имеющей отношения к приложению. В настоящее время такой подход не допустим, т.к. он не только приводит к коммерческому провалу продукта на рынке, но и многократно повышает затраты на разработку и тестирование.

Тестирующий в процессе тестирования требований:

1. участвует в проверке и анализе требований,
2. находит неясности и противоречия в требованиях,
3. предоставляет отзыв о функциях и удобстве использования будущего приложения.

Требования являются отправной точкой для определения того, что проектная команда будет проектировать, реализовывать и тестировать. Аргументом в пользу тестирования требований является то, что, по разным оценкам, в них зарождается более 50 процентов всех проблем с программным обеспечением. В итоге есть риск, что получится результат, показанный на рис.1.

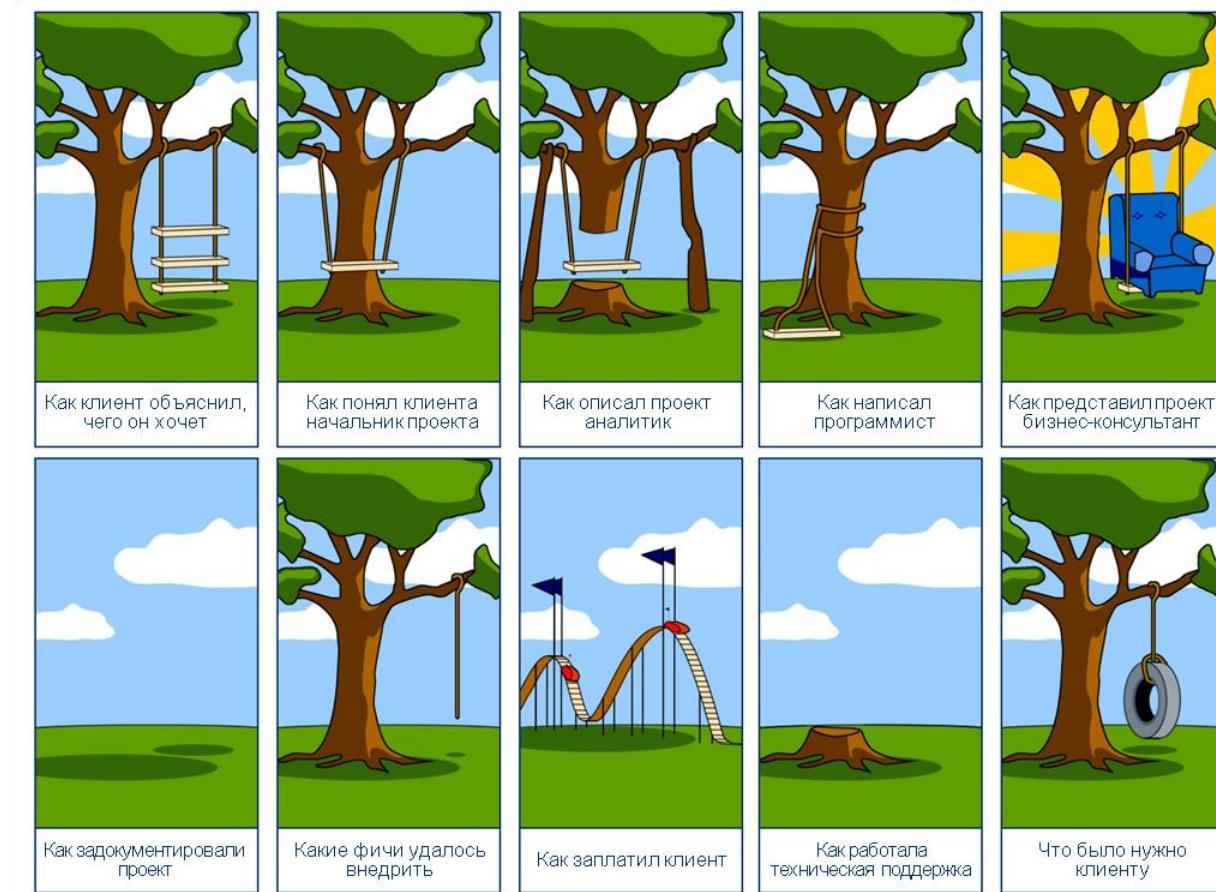


Рисунок 1 – Важность требований

На первой фазе проекта клиент пытался объяснить желания менеджеру проекта на словах, без составления документации. Менеджер понял, как смог, и передал информацию аналитику для составления технического задания. Программист реализовал задуманное и представил результат для передачи клиенту. Бизнес-консультант приукрасил результаты. Документирование реализованного программного продукта также не сделано. Единственным совпадением в результате и пожелании заказчика было дерево и веревка. Стоимость этого проекта для клиента была очень большой, но результат не совпал с ожидаемым клиентом. Поэтому проект провален. Эта история показывает важность получения первичных требований.

4.2.2 Первичные и детальные требования

Первичные требования документируют желания и потребности заказчика и должны быть составлены на языке, понятном заказчику.

Требования формулируют, ЧТО система делает. Например, возможно такое требование для приложения в финансовой сфере: клиенту предоставлены возможности добавления, снятия денег со счета, а также просмотр состояния счета.

Детальные требования документируют требования в специальной форме. Например, данные о лицевых счетах будут сохранены в БД под управлением СУБД POSTGRESS.

Источниками требований выступают пользователи, заказчики и разработчики программной системы. Как правило, на этапе сбора требований работают аналитики, тестировщики могут им помогать.



Рисунок 2 – Источники требований

4.2.3 Техники выявления требований

Для выявления требований применяют множество техник. Рассмотрим их в контексте взаимодействия с пользователем (рис.3).



Рисунок 3 – Техники сбора требований

Контактные методы

Интервью. Самый универсальный путь выявления требований, заключающийся в общении аналитика и представителя заказчика. Интервью может проходить в классическом понимании этого слова: беседа в виде «вопрос-ответ», в виде переписки и т.п. Главным здесь является то, что ключевыми фигурами выступают двое – интервьюируемый и интервьюер.

Работа с фокусными группами. Может выступать как вариант «расширенного интервью», где источником информации является не одно лицо, а группа лиц. Они, как правило, представляют целевую аудиторию пользователей, например, группа из отдела бухгалтерии или приемной комиссии вуза.

Анкетирование позволяет автоматически собрать и обработать огромное количество ответов от огромного количества респондентов. Ключевым фактором успеха является правильное составление анкеты, правильный выбор аудитории и правильное преподнесение анкеты.

Семинары позволяют группе людей очень быстро обменяться информацией. Они хорошо сочетаются с техниками интервью, анкетированием, прототипированием и моделированием.

Мозговой штурм позволяет за минимальное время сгенерировать большое количество идей, которые в дальнейшем можно не спеша рассмотреть с точки зрения их использования для развития проекта.

Методы отслеживания требований

Следующая группа методов основана на бесконтактной форме.

Наблюдение. Может выражаться в буквальном наблюдении за некоторыми процессами, или во включении аналитика в эти процессы в качестве участника. С одной стороны, наблюдение позволяет увидеть то, о чём могут умолчать интервьюируемые, анкетируемые и представители фокусных групп.

Анализ и изучение документов. Позволяет выявить требования в регламентирующих деятельность организации документах.

Самостоятельное описание. Является техникой фиксации и формализации требований. Важно не пытаться самому «придумать требования за заказчика», а в спокойной обстановке самостоятельно обработать собранную информацию и аккуратно оформить её для дальнейшего обсуждения и уточнения.

Методы, основанные на моделировании.

Такие методы показывают динамику связи между «бизнес-процессом и взаимодействием» и «техническим процессом и взаимодействием».

Взаимодействие с бизнес-процессами рассмотрим на примере: «договор на закупку формируется отделом закупок, визируется бухгалтерией и юридическим отделом...». Взаимодействие на основе технических процессов иллюстрирует следующий пример: «платёжное поручение генерируется модулем “Бухгалтерия”, шифруется модулем “Безопасность” и передаётся на сохранение в модуль “Хранилище”»).

Прототипирование. Состоит в демонстрации и обсуждении промежуточных версий продукта. Например, дизайн страниц сайта может быть сначала представлен в виде картинок – датафреймов, и затем свёрстан.

4.2.4 Функциональные и нефункциональные требования

По классификации Карла Вигерса различают функциональные и нефункциональные требования.

Функциональные требования описывают поведение системы и функции, которые она должна выполнять.

Нефункциональные требования относят к характеристикам системы и ее внешнего окружения.

1. Функциональные требования показывают, ЧТО должна выполнять программа, в них устанавливают границы разработки проекта. Например, требуется распознать текст. В функциональных требованиях четко прописывают рукописный текст или машиночитаемый, с шаблонных документов или написанный в произвольной форме. Для понимания потенциальными пользователями зафиксированных в техническом задании требований используют use case – пользовательские сценарии. При проверке функциональных требований проводят системное, интеграционное тестирование, используют E2E и API тестирование.

Пример функциональных требований: конвертация форматов файлов, распознавание текстов, редактирование фотографий, вычисление данных по введенным значениям, регистрация в личном кабинете, онлайн-покупки товаров, возможности редактирования текстов, обмена сообщениями и т.д.

2. Под нефункциональными требованиями понимают свойства системы, которыми она должна обладать при реализации своего поведения: надежность, масштабируемость, удобство использования. Они фиксируют, каким образом будет работать система. Нефункциональные требования могут описать только квалифицированные технические специалисты или разработчики системы. Проверку качества проводят с помощью нагрузочного и юзабилити тестирования. Обязательно проводят нагрузочное тестирование и тестирование безопасности.

Примерами нефункциональных требований могут быть следующие: требования по скорости загрузки веб-страницы, размеру шрифта в названии диалогового окна, времени непрерывной работы системы до первого отказа при одновременной работе 100 пользователей.

Список источников

1. Тестирование программного обеспечения. Базовый курс: практик. пособие. / С. С. Куликов. — Минск: Четыре четверти, 2015. С. 294.
2. Орлов С.А. Программная инженерия. Технологии разработки программного обеспечения: учебник. — СПб: Питер, 2020. С.640.
3. Статья. Тестирование документации к программным продуктам URL: <https://habr.com/ru/articles/346290/>
4. Справочник QA-инженера. URL <https://sergeygavaga.gitbooks.io/>