


Требования



“**Требование** (requirement) - условие,
которое включает обязательные для
выполнения критерии”



Так клиент
объяснил, чего он
хочет



Так клиента понял
менеджер проекта



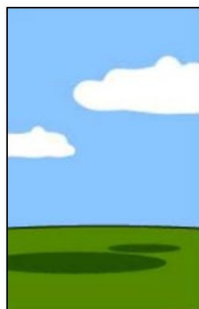
Так аналитик
описал проект



Так программист
реализовал проект



Так проект был
прорекламирован
консультантами



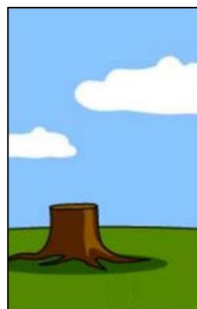
Так проект был
задокументирован



Так проект был
сдан в
эксплуатацию



В такую сумму
проект обошёлся
заказчику



Так работала
техническая
поддержка



Что на самом деле
было нужно
клиенту

Рисунок 2.2.b — Типичный проект с плохими требованиями

Святослав Куликов “Тестирование ПО. Базовый курс.”

Продуктная документация

используется проектной командой во время разработки и поддержки продукта

План проекта, тест-план

Требования к продукту (PRD) и функциональные спецификации (FSD, SRS)

Архитектура и дизайн

Тест-кейсы и тестовые наборы (test suites)

Технические спецификации

Проектная документация

используется не только на стадии разработки, но и на более ранних и поздних стадиях (например, на стадии внедрения и эксплуатации)

Продуктная документация

Пользовательская и
сопроводительная документация

Маркетинговая документация
(**market requirements
document, MRD**)

Источники и пути выявления требований

Интервью	Работа с фокусными группами	Анкетирование
Семинары и мозговой штурм	Наблюдение	Прототипирование
Анализ документов	Моделирование процессов и взаимодействий	Самостоятельное описание

Рисунок 2.2.d — Основные техники сбора и выявления требований

Уровни и типы требований

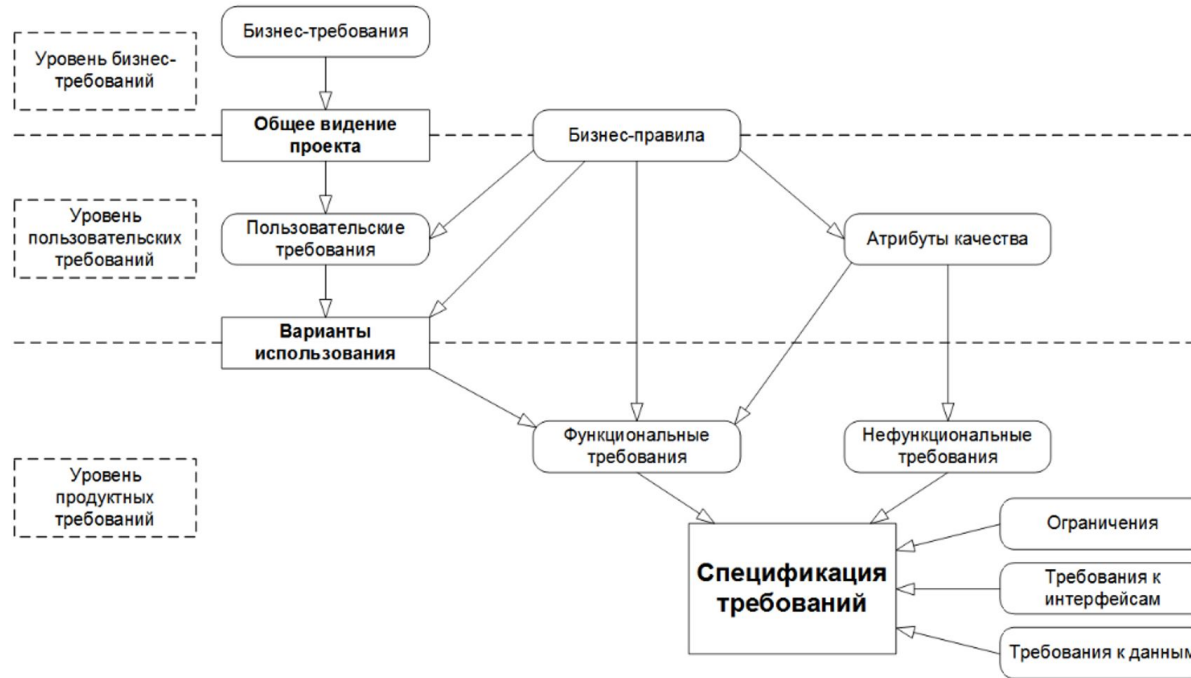


Рисунок 2.2.е — Уровни и типы требований



Неявные требования

1. Законы, регламенты, инструкции
2. Список задач, существующие тесты и баг-репорты
3. Руководство пользователя
4. Реклама продукта
5. Интервью с командой и заказчиками
6. Чаты и email-переписка
7. Прототип, дизайн-макет
8. Конкурентный анализ, личный опыт



<https://software-testing.ru/library/around-testing/requirements/3567-requirements>



Свойства требований

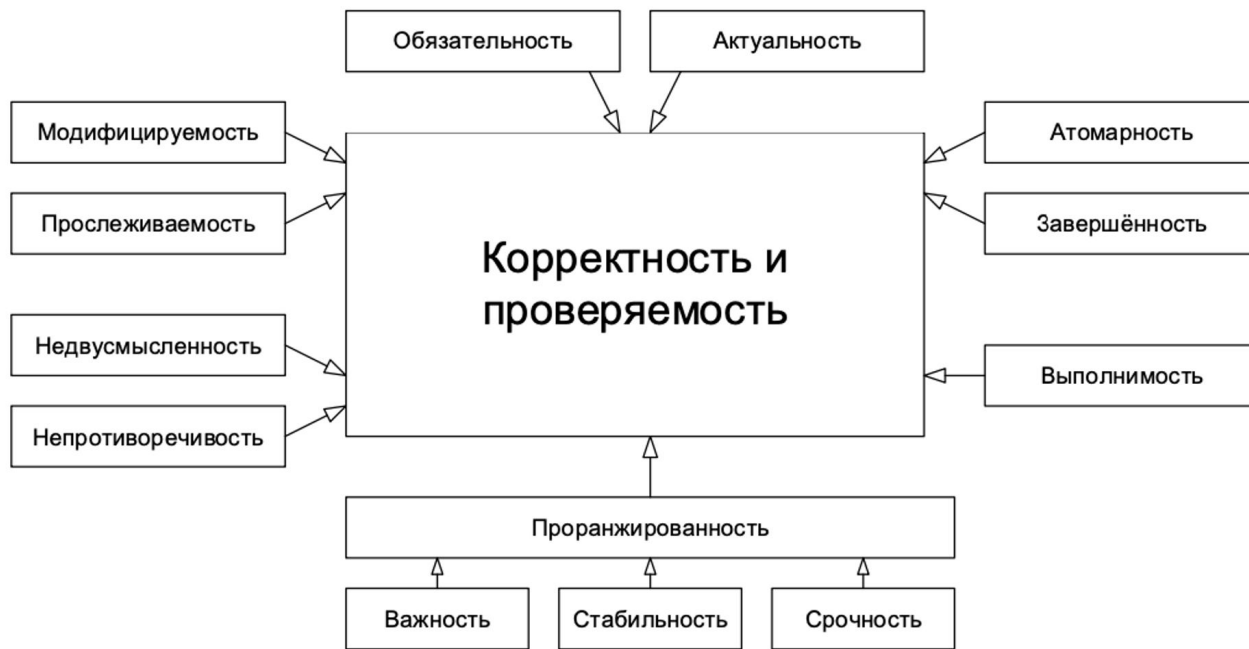


Рисунок 2.2.f — Свойства качественного требования



1

Завершенность (completeness)

Требование является полным и законченным с точки зрения представления в нём всей необходимой информации, ничто не пропущено по соображениям «это и так всем понятно».

Примеры

1. Пароли должны храниться в зашифрованном виде
2. Экспорт осуществляется в форматы PDF, PNG и т.д.
3. Приведённые ссылки неоднозначны (например: «см. выше» вместо «см. раздел 123.45.b»)



2

Атомарность, единичность (atomicity)

Требование является атомарным, если его нельзя разбить на отдельные требования без потери завершённости и оно описывает одну и только одну ситуацию.

Примеры

1. Кнопка “Restart” не должна отображаться при остановленном сервисе, окно “Log” должно вмещать не менее 20-ти записей о последних действиях пользователя
2. Если пользователь подтверждает заказ и редактирует заказ или откладывает заказ, должен выдаваться запрос на оплату
3. Когда пользователь входит в систему, ему должно отображаться приветствие; когда пользователь вошёл в систему, должно отображаться имя пользователя; когда пользователь выходит из системы, должно отображаться прощание



3

Непротиворечивость, последовательность (consistency)

Требование не должно содержать внутренних противоречий и противоречий другим требованиям и документам.

Примеры

1. После успешного входа в систему пользователя, не имеющего права входить в систему...
2. “712.a Кнопка “Close” всегда должна быть красной” и “36452.x Кнопка “Close” всегда должна быть синей” - противоречия в разных требованиях
3. “В случае, если разрешение окна составляет менее 800x600...” — разрешение есть у экрана, у окна есть размер



4

Недвусмысленность (unambiguousness, clearness)

Требование должно быть описано без использования жаргона, неочевидных аббревиатур и расплывчатых формулировок, должно допускать только однозначное объективное понимание и быть атомарным в плане невозможности различной трактовки сочетания отдельных фраз.

Примеры

1. «Доступ к ФС осуществляется посредством системы прозрачного шифрования» и «ФС предоставляет возможность фиксировать сообщения в их текущем состоянии с хранением истории всех изменений» - что такое ФС?
2. «Система конвертирует входной файл из формата PDF в выходной файл формата PNG» - некоторые вещи опущены, так как автор считает их очевидными

Слова-паразиты

адекватно (adequate), быть способным (be able to), легко (easy), обеспечивать (provide for), как минимум (as a minimum), быть способным (be capable of), эффективно (effectively), своевременно (timely), применимо (as applicable), если возможно (if possible), будет определено позже (to be determined, TBD), по мере необходимости (as appropriate), если это целесообразно (if practical), но не ограничиваясь (but not limited to), быть способно (capability of), иметь возможность (capability to), нормально (normal), минимизировать (minimize), максимизировать (maximize), оптимизировать (optimize), быстро (rapid), удобно (user-friendly), просто (simple), часто (often), обычно (usual), большой (large), гибкий (flexible), устойчивый (robust), по последнему слову техники (state-of-the-art), улучшенный (improved), результативно (efficient).



5

Выполнимость (feasibility)

Требование должно быть технологически выполнимым и реализуемым в рамках бюджета и сроков разработки проекта.

Примеры

1. Анализ договоров должен выполняться с применением искусственного интеллекта, который будет выносить однозначное корректное заключение о степени выгоды от заключения договора - технически невыполнимо на данный момент
2. Система поиска должна заранее предусматривать все возможные варианты поисковых запросов и кэшировать их результаты - нельзя реализовать
3. «озолочение» (*gold plating*) — требования, которые крайне долго и/или дорого реализуются и при этом практически бесполезны для конечных пользователей

Святослав Куликов “Тестирование ПО. Базовый курс.”





6

Обязательность, нужность (obligatoriness) и актуальность (up-to-date).

Если требование необязательное, оно должно быть просто исключено из набора требований. Если требование нужное, но «не очень важное», для указания этого факта используется указание приоритета. Также исключены (или переработаны) должны быть неактуальные требования

Примеры

1. Требование было добавлено «на всякий случай», хотя реальной потребности в нём не было и нет.
2. Требованию выставлены неверные значения приоритета по критериям важности и/или срочности.
3. Требование устарело, но не было переработано или удалено.



7

Прослеживаемость (traceability)

Вертикальная позволяет соотносить между собой требования на различных уровнях требований, горизонтальная позволяет соотносить требование с тест-планом, тест-кейсами, архитектурными решениями и т.д.

Примеры

1. Требования не пронумерованы, не структурированы, не имеют оглавления, не имеют работающих перекрёстных ссылок.
2. При разработке требований не были использованы инструменты и техники управления требованиями.
3. Набор требований неполный, носит обрывочный характер с явными «пробелами».



8

Модифицируемость (modifiability)

Это свойство характеризует простоту внесения изменений в отдельные требования и в набор требований.

Примеры

1. Требования неатомарны и непрослеживаемы, а потому их изменение с высокой вероятностью порождает противоречивость.
2. Требования изначально противоречивы. В такой ситуации внесение изменений (не связанных с устранением противоречивости) только усугубляет ситуацию, увеличивая противоречивость и снижая прослеживаемость.
3. Требования представлены в неудобной для обработки форме (например, не использованы инструменты управления требованиями, и в итоге команде приходится работать с десятками огромных текстовых документов).

Святослав Куликов “Тестирование ПО. Базовый курс.”





9

Проранжированность по важности, стабильности, срочности (ranked for importance, stability, priority)

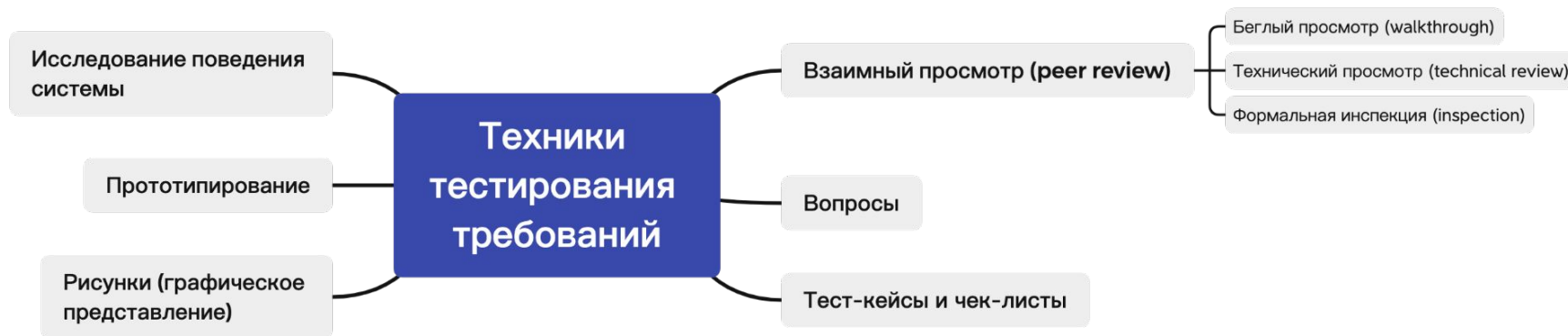
Важность характеризует зависимость успеха проекта от успеха реализации требования. Стабильность характеризует вероятность того, что в обозримом будущем в требование не будет внесено никаких изменений.

Срочность определяет распределение во времени усилий проектной команды по реализации того или иного требования.

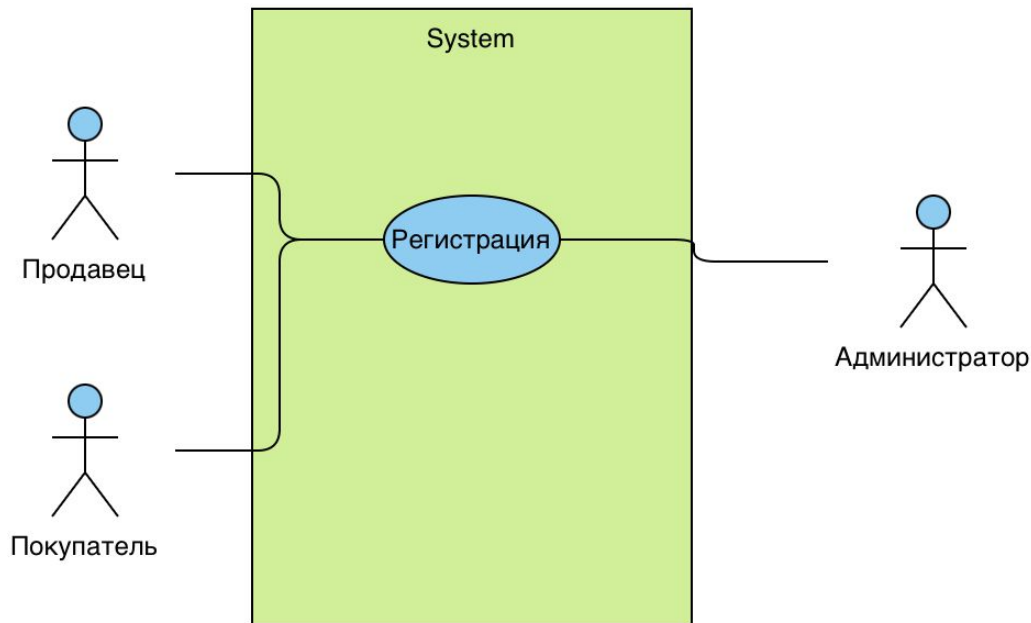
Корректность (correctness) и **проверяемость** (verifiability)

Эти свойства вытекают из соблюдения всех вышеперечисленных. В дополнение можно отметить, что проверяемость подразумевает возможность создания объективного тест-кейса (тест-кейсов), однозначно показывающего, что требование реализовано верно и поведение приложения в точности соответствует требованию.

Техники тестирования требований



Варианты использования (use cases)



Варианты использования (use cases)

Основные сценарии	Номер	Шаги
Акторы/пользователи	1	Ввод логина Ввод пароля
	2	Проверить логин и пароль
	3	Зарегистрировать пользователя
Расширения	1a	Логин не прошел валидацию Система выводит сообщение об ошибке
		...

Варианты использования (use cases)

Название use case	Регистрация
Описание use case	Пользователь регистрируется в системе, чтобы получить доступ к работе в ней
Акторы	Продавцы, Покупатели, Администратор
Предусловия	Система должна быть подсоединена к сети
Постусловия	После успешного регистрации пользователю высылается письмо о подтверждении на email



Пользовательские истории (user story)

“As a [persona], I [want to], [so that].”

«Как [тип клиента], [хочу то-то], [чтобы делать что-то]».

- Как Макс, я хочу пригласить друзей, чтобы мы вместе могли пользоваться этим замечательным сервисом.
- Как Саша, я хочу организовать свою работу, чтобы лучше контролировать ситуацию.
- Как менеджер, я хочу видеть, как продвигается работа у моих коллег, чтобы можно было составлять более точные отчеты о наших успехах и неудачах.



<https://www.atlassian.com/ru/agile/project-management/user-stories>



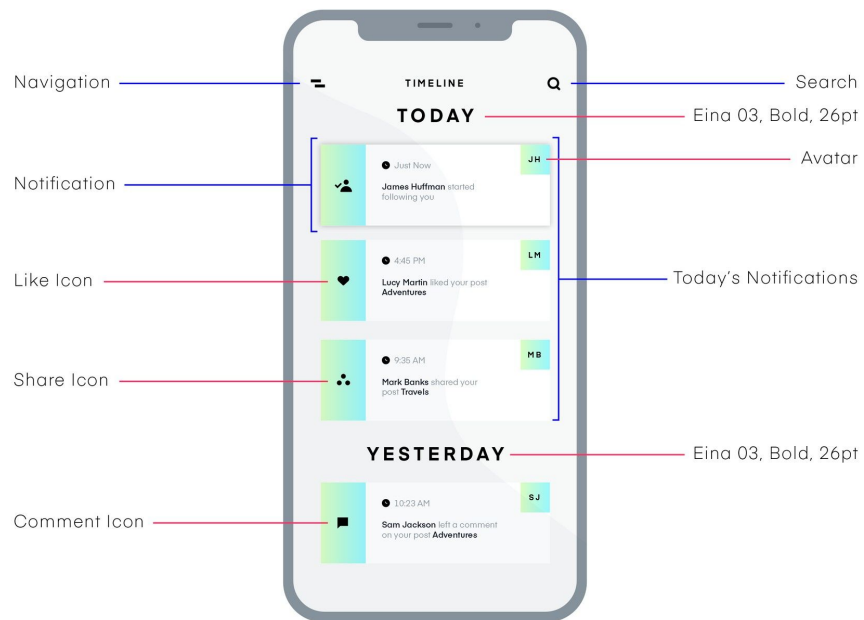
Пользовательские истории (user story)

INVEST — слово (модель), использующееся для запоминания критериев качественной истории:

- I – independent. История должна быть независима от других историй.
- N – negotiable. История обсуждаема, в ней отражается суть, а не детали или конкретные шаги по реализации.
- V – valuable. История представляет ценность для клиентов и бизнеса.
- E – estimable. Пригодность для оценки: историю можно оценить по сложности и трудозатратам.
- S – small. Хорошая история маленькая, ее можно реализовать за одну итерацию.
- T – testable. История имеет критерии приемки и ее можно протестировать.

<https://testengineer.ru/chto-takoe-user-story-i-kak-ee-pisat/>

Mockup (макет)







<https://www.invisionapp.com/defined/mockup>

Спецификация

Спецификация требований программного обеспечения (англ. Software Requirements Specification, SRS) является полным описанием поведения системы, которая будет создана. Она включает ряд сценариев использования, которые описывают все виды взаимодействия пользователей с программным обеспечением.

<https://analytics.infozone.pro/requirements-analysis/template-specification-requirements/>



“ Тестовый [оракул] предсказатель ”
(test oracle) – источник для
определения ожидаемого
результата для сравнения с
фактическим результатом
тестируемой системы.