GeekBrains

Анализ требований. Декомпозиция

Цели урока

1. Узнать о требованиях и их атрибутах
2. Научиться находить баги в требованиях
3. Декомпозировать требования и создать майнд-карту

План урока

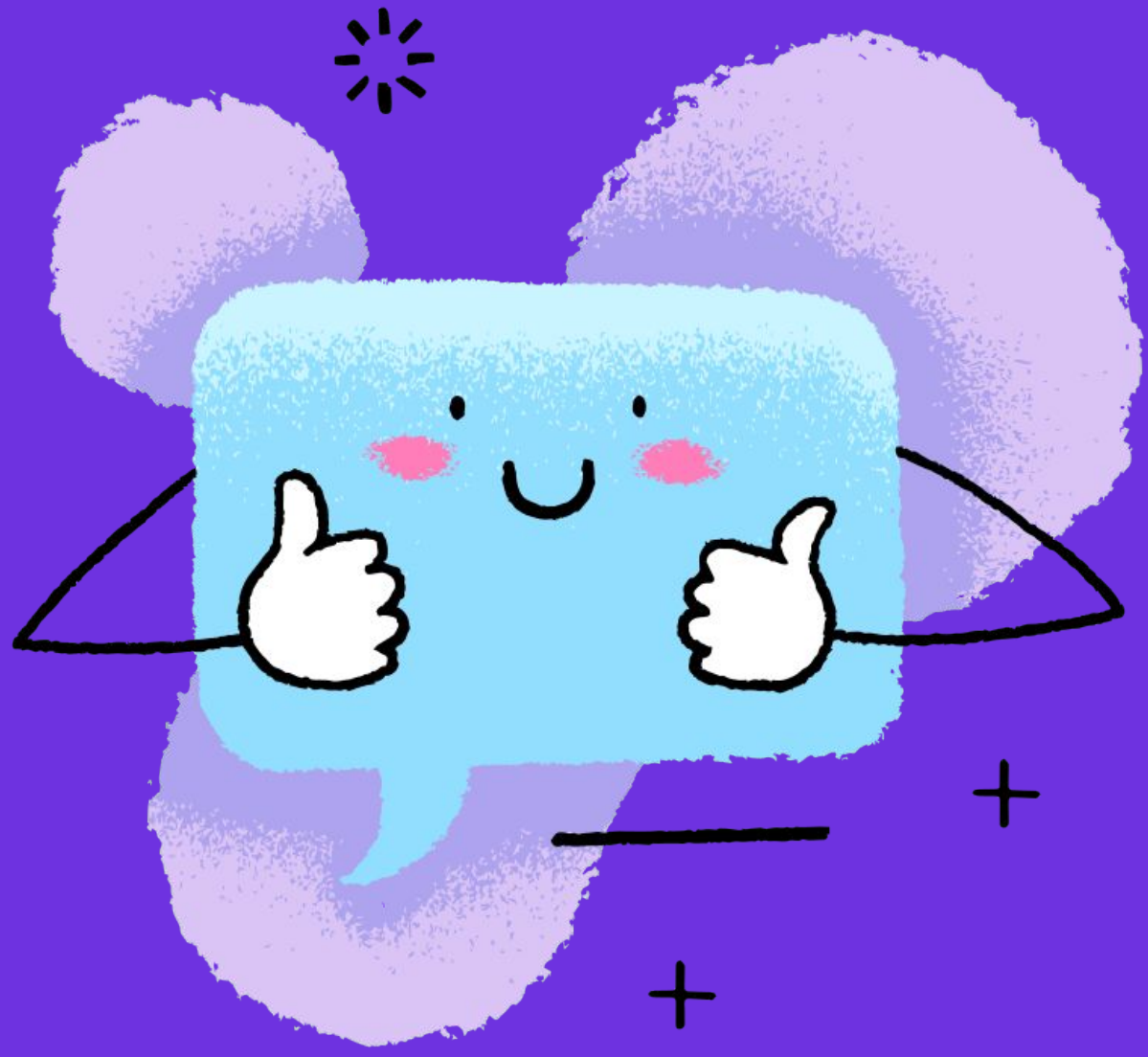
1. Виды требований
2. Атрибуты требований
3. Дефекты в требованиях
4. Пользовательские истории и сценарии
5. Системы управления требованиями
6. Декомпозиция требований

Что такое требование

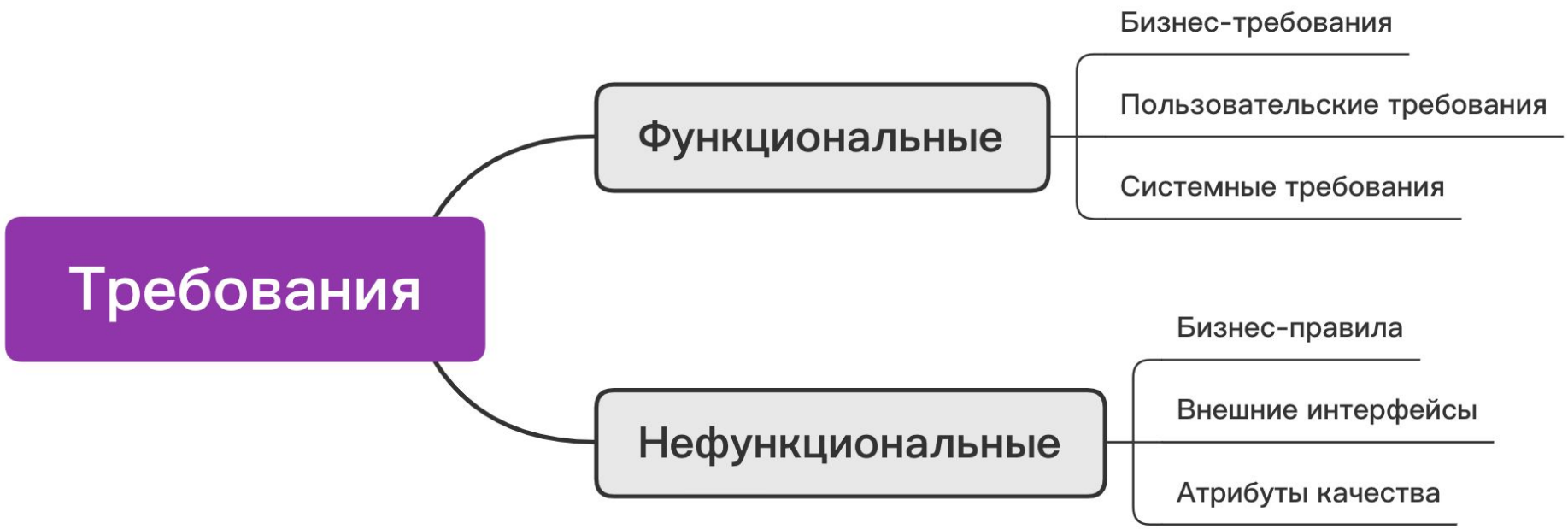
Требование — описание функции и условий, которые выполняет приложение в процессе решения задачи. Это отправная точка для процесса разработки.

Тестировщик участвует в проверке и анализе требований, находит неясности и противоречия, предоставляет отзыв о функциях и удобстве использования будущего приложения.

Виды
и атрибуты
требований

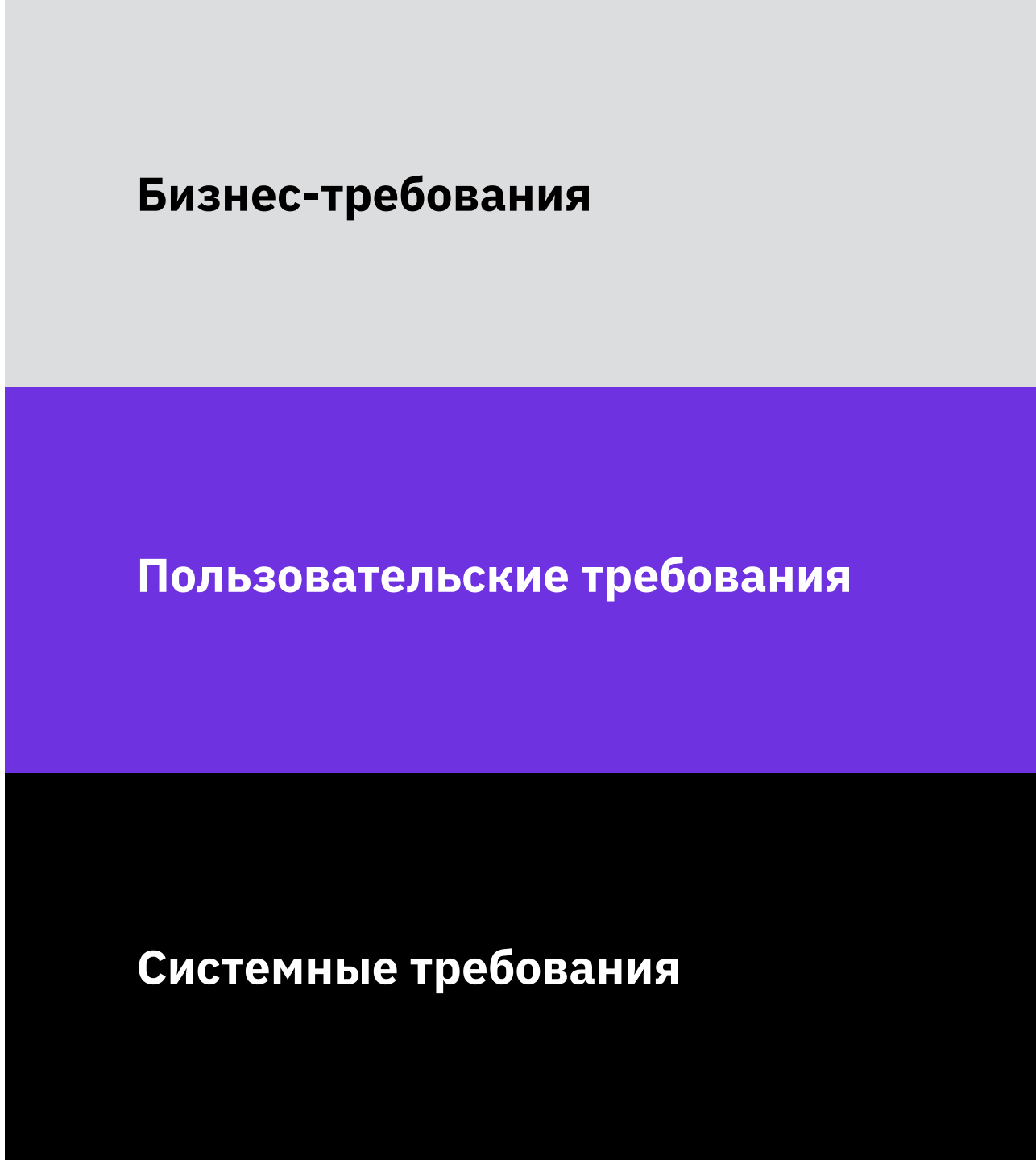


Виды требований



Функциональные требования

Что должна делать программа?



Нефункциональные требования

Как должна работать программа?

Нефункциональные показатели: безопасность, быстродействие и т. д.

Бизнес-правила

Внешние интерфейсы

Атрибуты требований

- Полнота
- Однозначность
- Непротиворечивость
- Необходимость
- Осуществимость
- Тестируемость

Полнота

Учитываются все возможные пользовательские действия, входные параметры, сообщения об ошибках

- При регистрации пользователь указывает дату рождения
- Если дата рождения указана, то возраст прописывается в анкете

А если дата не указана? Поле «Возраст» — пустое? Или не отображается вообще?

Однозначность

Требования не допускают двусмысленных формулировок

Страница быстро загружается

Что значит «быстро»? 5 секунд — это быстро или медленно? А если у пользователя нестабильное интернет-соединение?

Непротиворечивость

Поведение одного и того же компонента описывается разными аналитиками в разных разделах требований

Это поведение будет различаться

- **Отчёт о продажах формируется за 5 секунд**
- **...**
- **Формирование отчёта о продажах занимает около 15 минут**

Какое требование считать верным?

Необходимость

В документации содержится всё необходимое, но без лишней детализации

В техническом задании описывается инструментарий, основной сценарий и альтернативы, а также типы ошибок

В пользовательской документации — то, как пользоваться системой, не доходя до крайностей (как включать компьютер)

Осуществимость

Требования осуществимы с учётом внутреннего устройства программы и технологий, которые используются разработчиками

Отчёт о продажах за квартал агрегирует данные из 5 таблиц и отображается на экране за 1 секунду

Возможно, это требование невыполнимо, т. к. на выполнение запроса, агрегацию данных и отрисовку на экране физически может потребоваться больше времени

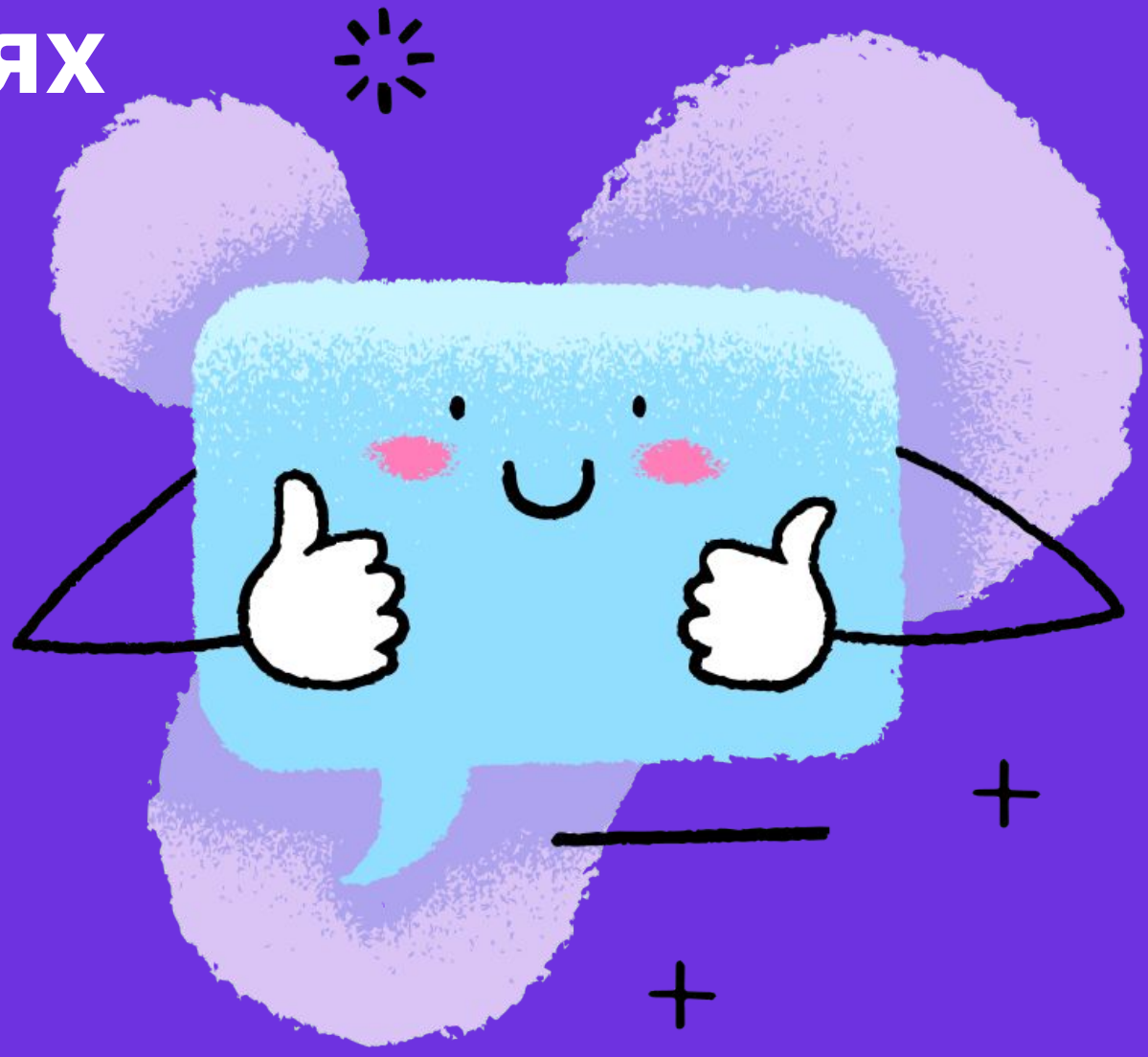
Тестируемость

Тестировщик может проверить функциональность, которую создал разработчик

Для регистрации пользователя надо указать уникальный email

Есть ли у тестировщика доступ к базе данных, где хранятся уже зарегистрированные адреса? Могут ли автотесты создавать уникальные адреса или проверять их уникальность? Если нет, то требование становится не тестируемым

Баги в требованиях



Дефекты в требованиях

«Хочу, чтобы группам выдавались проектные роли в настройках проекта или в панели администрирования»

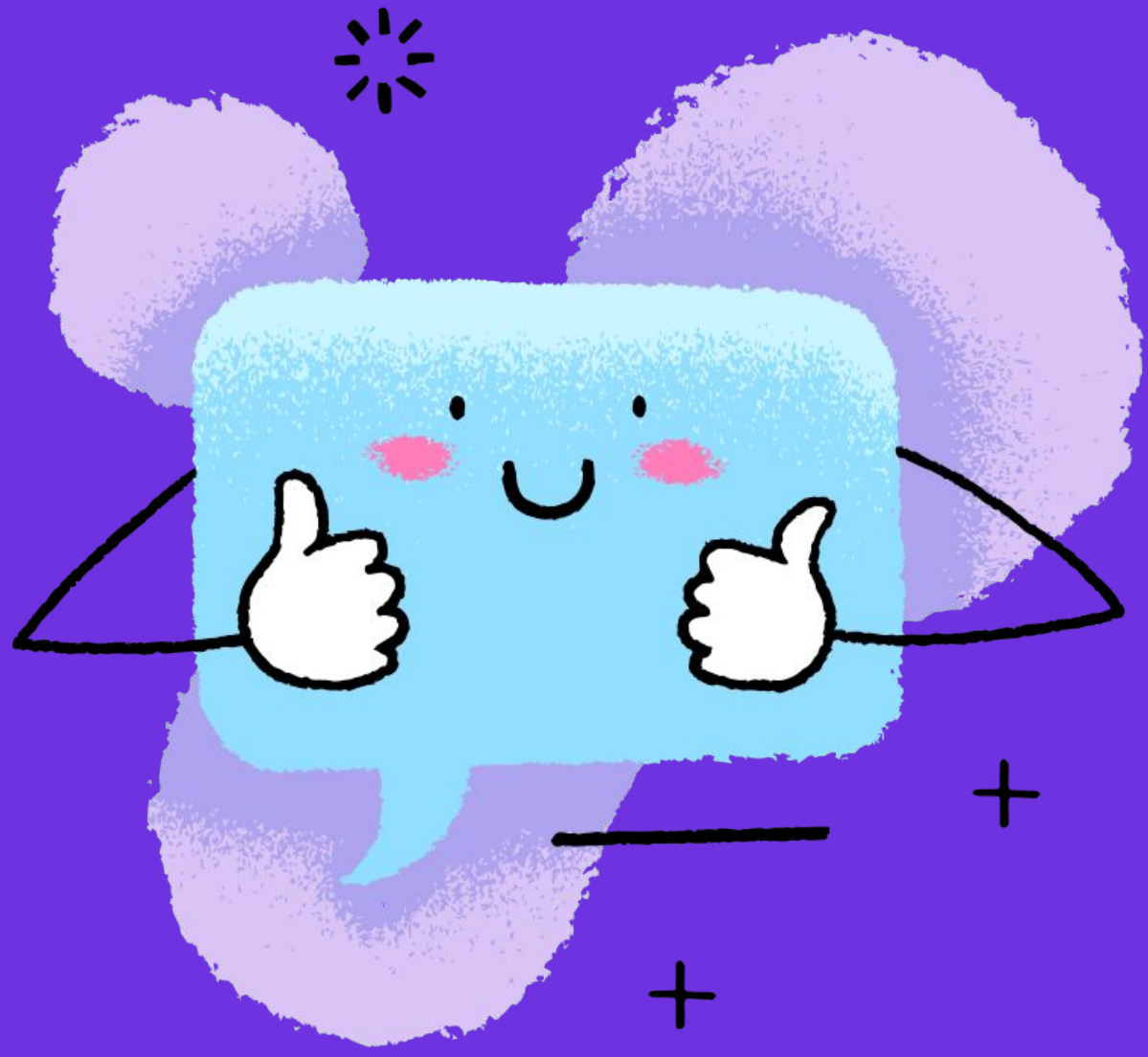
Где именно должны выдаваться проектные роли: в настройках проекта, в панели администрирования или в обоих местах?

Баг на требования
Пример

Название. Требование №1 — неоднозначное

Описание. В требовании непонятно, как именно выдаются проектные роли: в настройках проекта, в панели администрирования или в обоих местах?

User story
Use case



User story
пользовательские истории

Пользовательские истории — способ описания требований к системе в виде одного или нескольких предложений.

User story описывает:

- человека, использующего систему (заказчик);
- то, что содержится в этой системе (примечание);
- то, для чего она нужна пользователю (цель).

Я как тестировщик хочу присваивать автотестам лейблы для создания метрик в отчете.

Преимущества user story

- Истории представляют маленькие кусочки бизнес-ценности, которые реализуются в период от нескольких дней до нескольких недель
- Позволяют разработчикам и клиентам обсуждать требования на протяжении всей «жизни» проекта
- Нуждаются в небольшом обслуживании
- Рассматриваются только в момент использования
- Поддерживают близкий контакт с клиентом
- Позволяют разбить проект на небольшие этапы
- Подходят для проектов, где требования изменчивы или плохо поняты
- Облегчают оценку заданий

Недостатки user story

- Без конкретных приёмочных испытаний открыты для различных интерпретаций
- Требуют близкого контакта с клиентом на протяжении всего проекта
- Плохо масштабируются на больших проектах — историй становится слишком много, сложно выделить самые приоритетные
- К каждой пользовательской истории в какой-то момент прикрепляется одно или более приёмочное тестирование

Use cases
пользовательские сценарии

Пользовательский сценарий описывает взаимодействия участников, как правило, пользователя и системы. Количество участников — от 2 и больше. Пользователь — человек или другая система

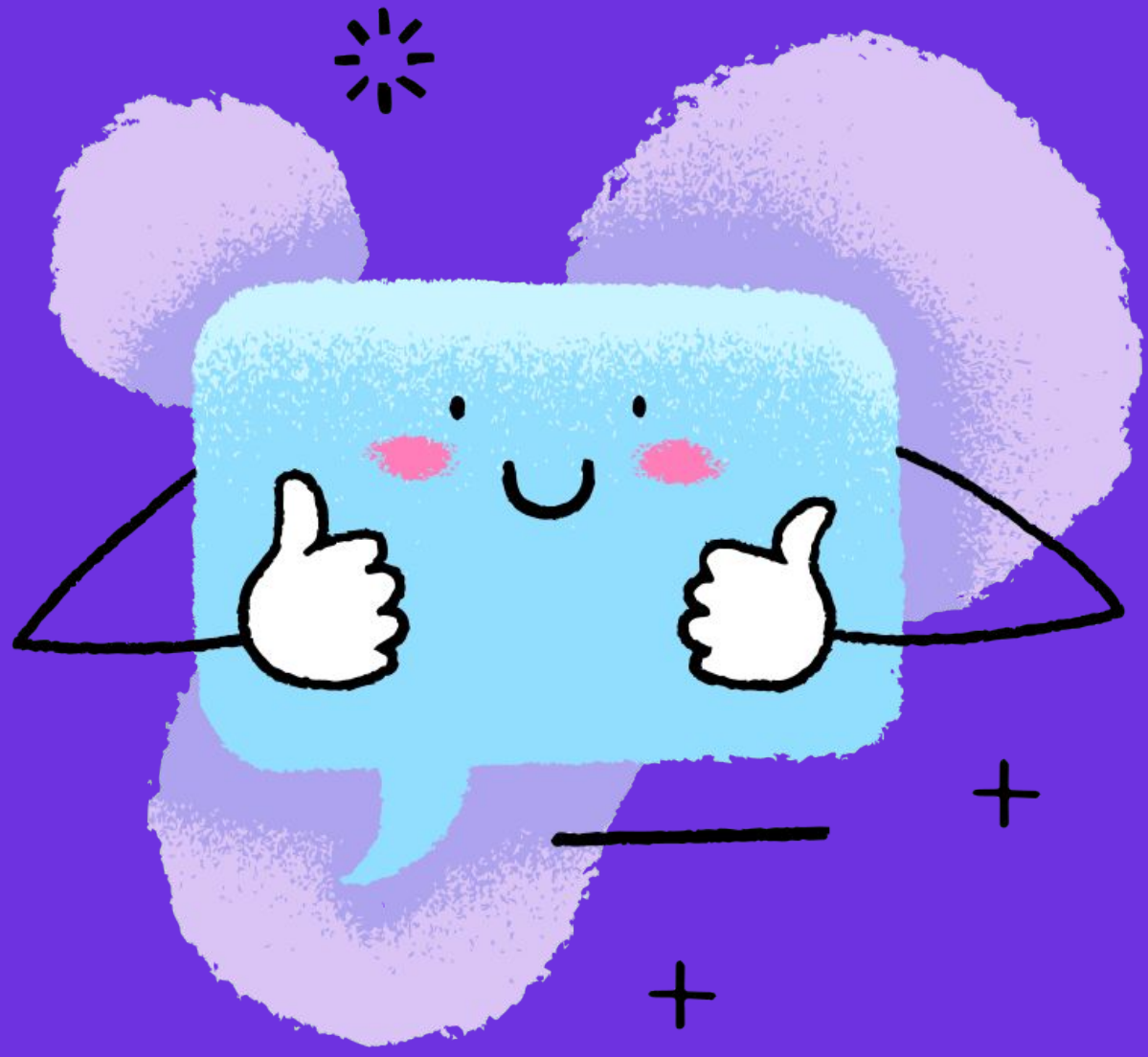
Use case в виде таблицы

Действующие лица	**Пользователь, система**
Цель	Изменить статус учётной записи пользователя на «Активный»
Предусловие	Учётная запись пользователя неактивна
Успешный сценарий 1. Администратор выбирает учётную запись пользователя и нажимает кнопку «Активировать» 2. Система переключает учётную запись в статус «Активный» 3. Система отправляет сообщение пользователю на email.	
Результат	Учётная запись пользователя перешла в статус «Активный»

Задачи пользовательских сценариев

1. Оценивать трудоёмкость проекта
2. Помогать планировать график работ
3. Выявлять пропущенные требования

RMS
Confluence



Системы управления требованиями

Система управления требованиями — requirements management systems, RMS — средство поддержки и автоматизации процесса работы с требованиями на протяжении всего «жизненного цикла» разработки программного продукта.

Задачи RMS

- Хранение требований в одном месте
- Единое управление требованиями
- Повышение производительности труда благодаря контролю над изменениями в требованиях и управлению ими
- Минимизация расходов и рисков благодаря оценке влияния происходящих изменений
- Демонстрация соответствия требований благодаря полному отслеживанию требований
- Сокращение объёма доработок и ускорение выхода на рынок благодаря совместной работе с заинтересованными лицами

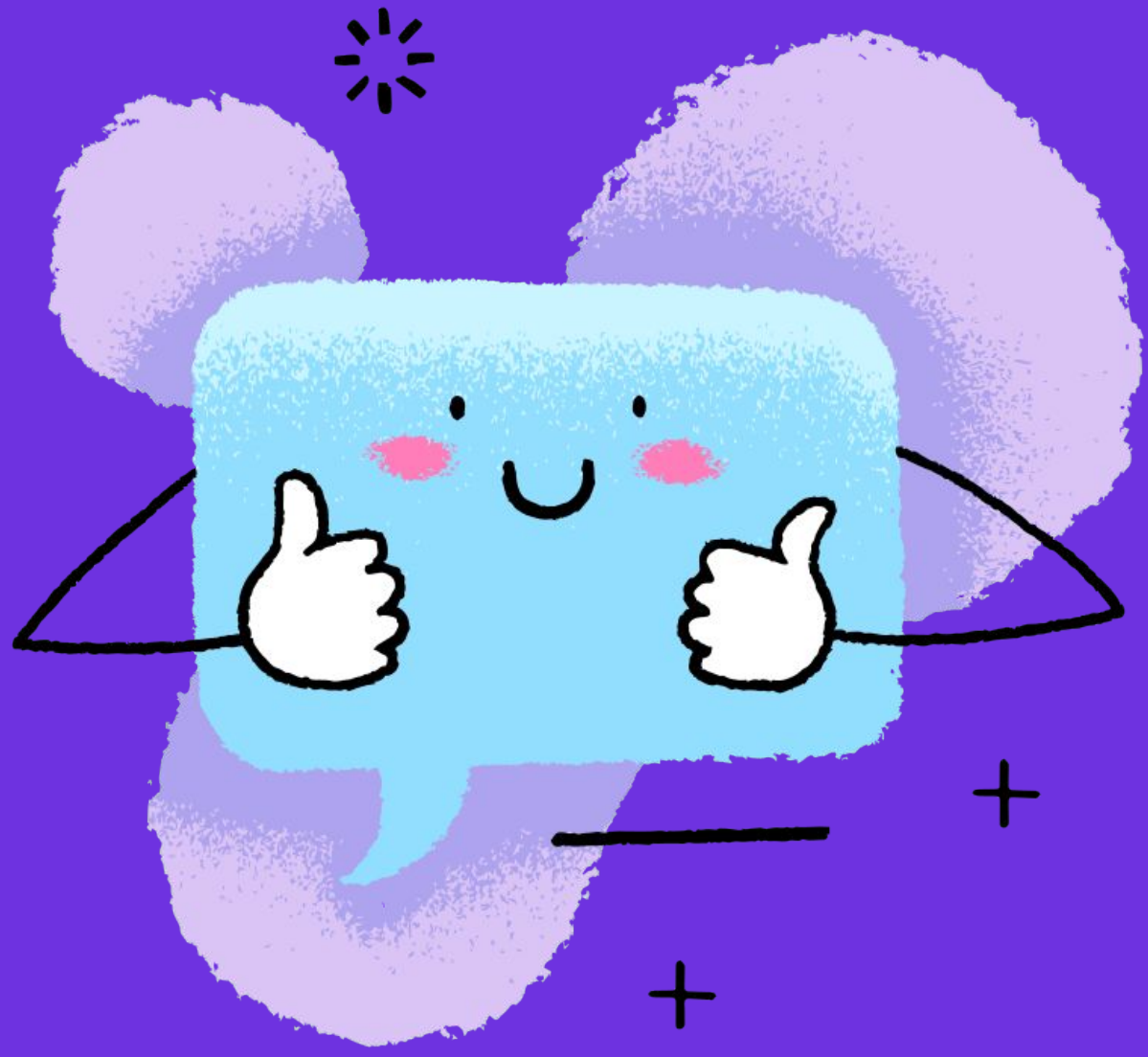
Confluence

Confluence — внутренняя вики-система для организаций, разработанная, чтобы создать единую базу знаний.

1. Создание и хранение проектной и технической документации
2. Создание и управление требованиями в более узком виде, чем RMS
3. Экспорт и импорт документации (документов)
4. Создание связи (ссылок и меток) между документами
5. Комментирование и обсуждение требований и документации
6. Отслеживание версионности и внесения изменений
7. Автоматические уведомления о внесении изменений
8. Управление доступом к проектам

Декомпозиция Mind map

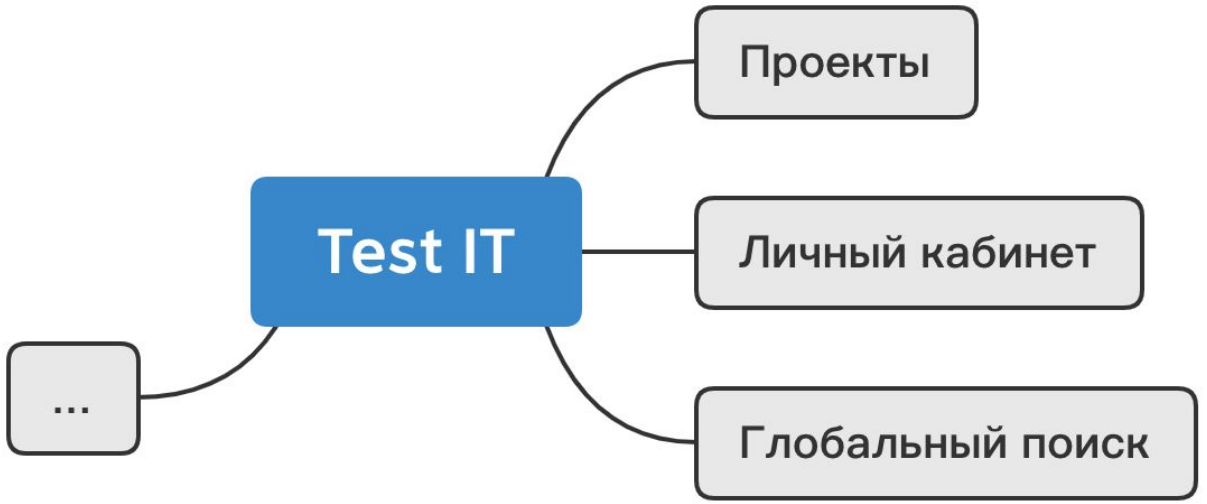
на основе требований к TMS Test IT



GeekBrains

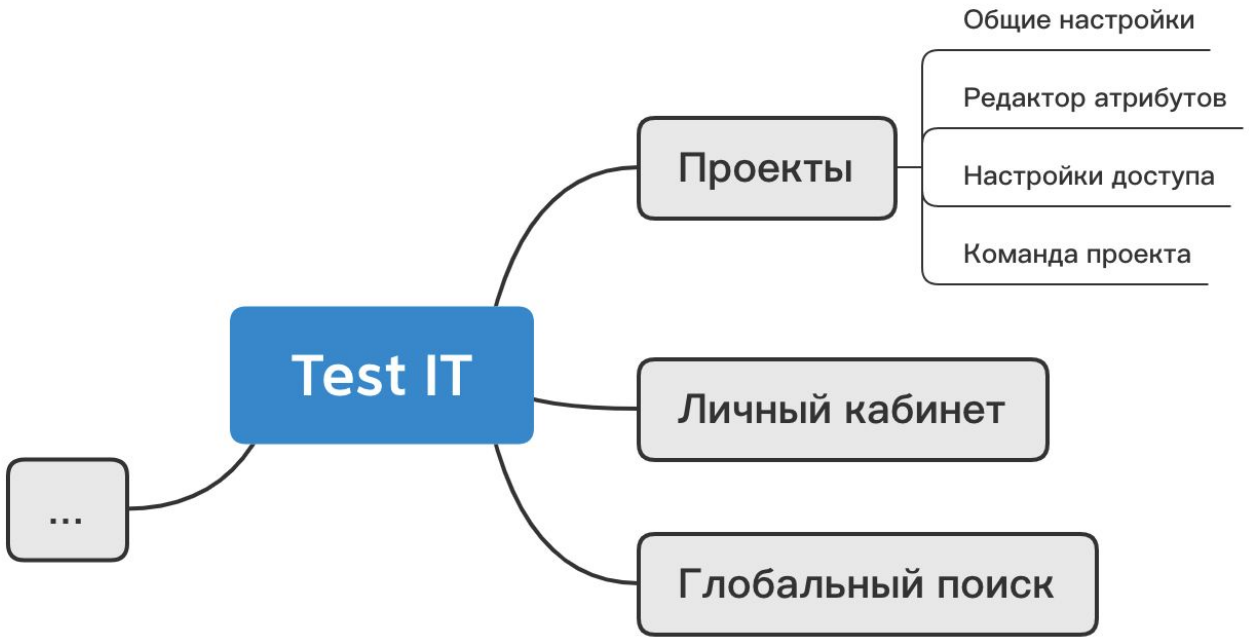
Уровни декомпозиции

1 уровень. Крупные блоки или компоненты



Уровни декомпозиции

2 уровень. Страницы сайта или экраны мобильного приложения

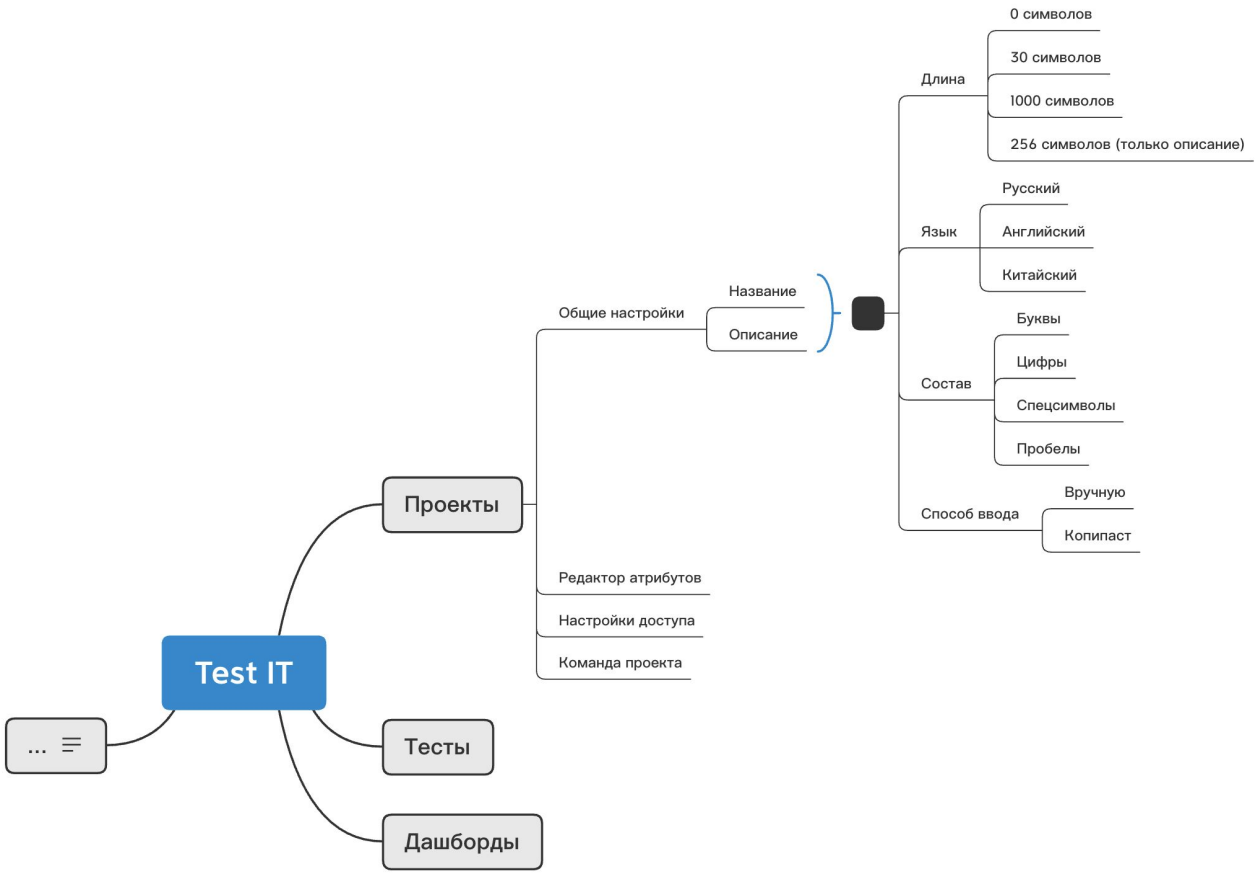


Уровни декомпозиции

3 уровень. Содержание экранов

- элементы, присутствующие на экране
- действия, которые может совершить пользователь
- параметры действий пользователя

Уровни декомпозиции



Спасибо!

Каждый день вы становитесь лучше :)