

Процесс тестирования. Составление чек-листов. Работа с TMS

План курса

- Занятие 1 - Введение, знакомство, цели
- Занятие 2 - Жизненный цикл разработки ПО, модели жизненного цикла, методологии разработки, качество ПО
- **Занятие 3 - Требования, атрибуты требований. Тестирование на основе чек-листов.**
- Занятие 4 - Чек-листы. Практика. Исследовательское тестирование.
- Занятие 5 - Виды тестирования. Тестирование на основе тест-кейсов.
- Занятие 6 - Тест-кейсы, практика.
- Занятие 7 - Тест-дизайн
- Занятие 8 - Ошибка, дефект, сбой/отказ. Правила составления баг-репортов.
- Занятие 9 - Особенности тестирования веб-приложений
- Занятие 10 - Инструменты тестирования веб-приложений

ПЛАН ЗАНЯТИЯ

- Процесс и этапы тестирования
- Виды требований
- Декомпозиция требований
- Подходы к тестированию
- Что такое чек-лист?
- Из чего состоят чек-листы
- Правила составления чек-листов
- Как осуществляются проверки по чек-листам
- Преимущества и недостатки чек-листов
- Исследовательское тестирование

1

ПОВТОРЕНИЕ ИЗУЧЕННОГО

Повторение

Вопрос 1:

Что из перечисленного не относится к Жизненному циклу ПО?

Планирование проекта

Внедрение

Разработка продукта

Согласование оплаты

Повторение

Вопрос 1:

Что из перечисленного не относится к Жизненному циклу ПО?

Планирование проекта

Внедрение

Разработка продукта



Согласование оплаты

Повторение

Вопрос 2:

Что из перечисленного
относится Моделям
жизненного цикла?

Каскадная модель

V-модель

Scrum

Спиральная модель

Повторение

Вопрос 2:

Что из перечисленного
относится Моделям
жизненного цикла?



Каскадная модель



V-модель



Спиральная модель

Scrum

Повторение

Вопрос 3:

Что из перечисленного не относится Моделям жизненного цикла?

Каскадная модель

V-модель

Спонтанная модель

Спиральная модель

Повторение

Вопрос 3:

Что из перечисленного не относится Моделям жизненного цикла?



Каскадная модель

V-модель

Спонтанная модель

Спиральная модель

Повторение



Вопрос 4:
Какая из представленных досок команды используется в Scrum, а какая в Kanban?

Projects / Test Project

TP Sprint 1

S

Type

5 days remaining

Complete sprint

GROUP BY

None

Insights

TO DO 2 ISSUES

Реализовать профиль пользователя

TP-1

Неактивна кнопка "подписаться" на карточке товара

TP-4

IN PROGRESS 1 ISSUE

Не отображается бейдж "продано" на карточке проданного товара

TP-3

DONE 1 ISSUE

Изменить цвет кнопки "Купить"

TP-2

Projects / kanban

KAN board

S

GROUP BY

None

TO DO 2 ISSUES

Реализация профиля пользователя

KAN-6

Поправить цвет шрифта

KAN-8

+ Create issue

IN PROGRESS 1 ISSUE

Изменить цвет кнопки "Продать" на экране карточки товара

KAN-7

DONE

See all Done issues

Повторение



Вопрос 4:

Для чего необходимы стандарты качества ПО?

2

ПРОЦЕСС И ЭТАПЫ ТЕСТИРОВАНИЯ

Этапы тестирования

Планирование тестирования

- объем работ
- стратегия тестирования
- критерии завершения тестирования

Мониторинг и контроль тестирования

- определение метрик тестирования
- какие метрики важны во время тестирования

Анализ тестирования

- анализ спецификации
- информация о реализации системы
- отчет анализа рисков

Проектирование тестов (тест-дизайн)

- проектирование тест-кейсов и чек-листов
- определение тестовых данных

Выполнение тестов

- выполнение составленных тестов

Завершение тестирования

- создаются итоговые отчеты о результате тестирования
- все необходимые отчеты о баг-репортах исправлены, проверены и закрыты
- все тесты выполнены

ВОПРОСЫ

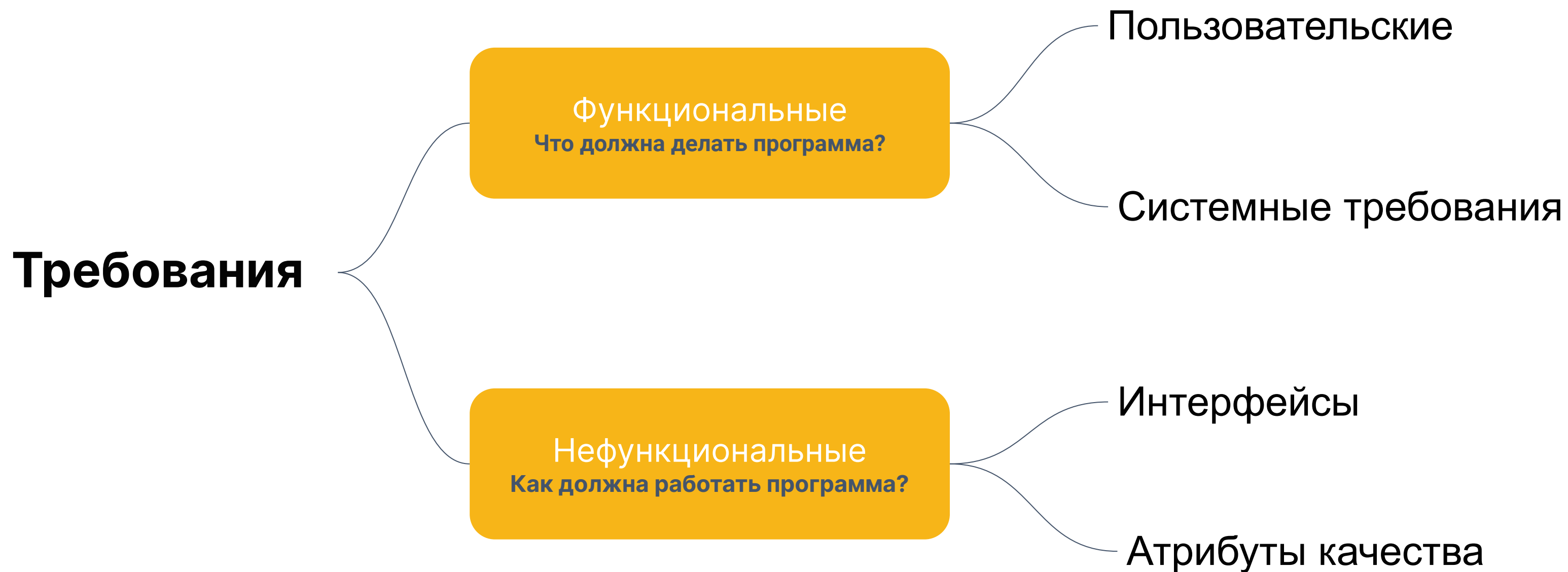
3

ВИДЫ ТРЕБОВАНИЙ

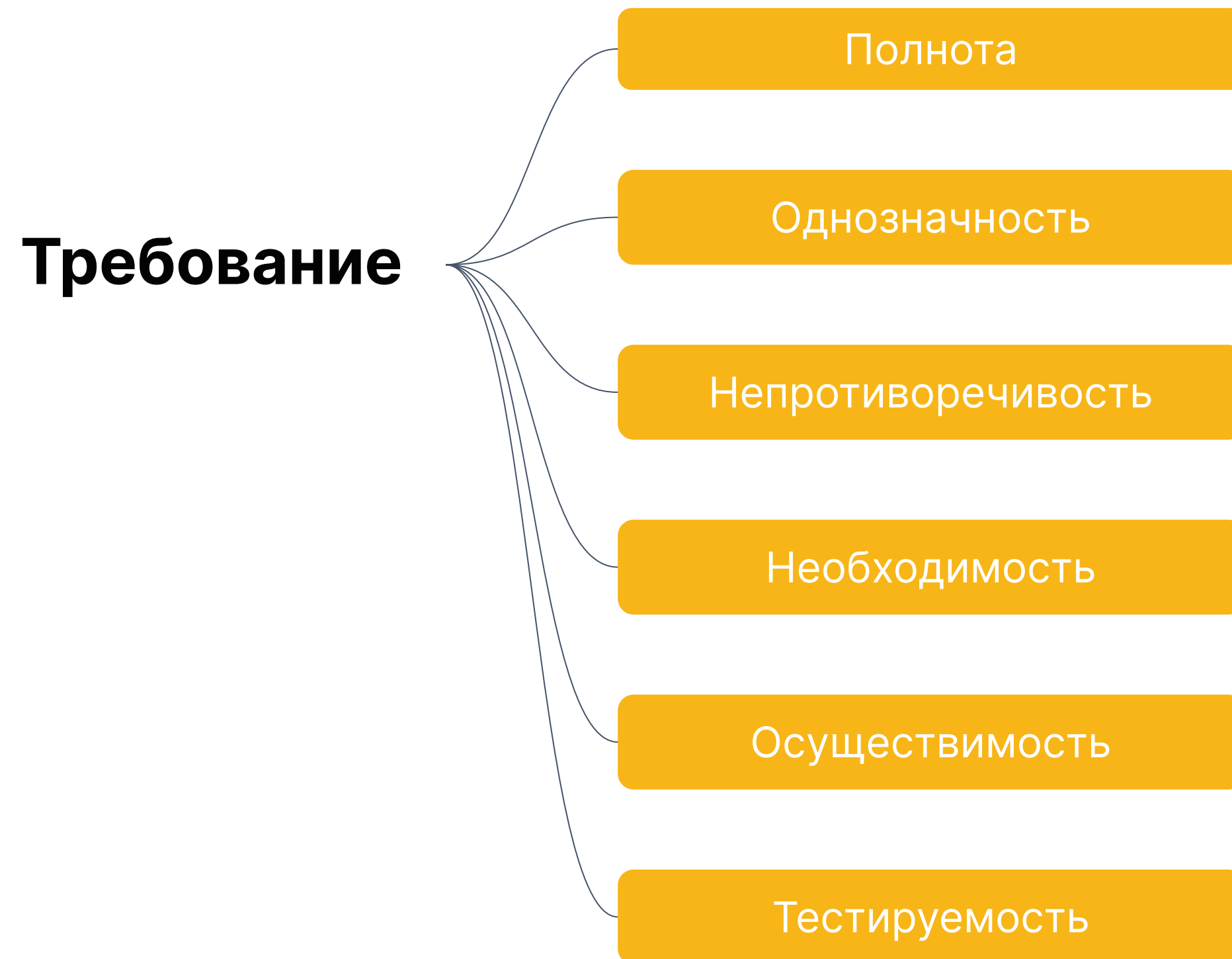
Требование

описание основных функций и условий, которые должны быть реализованы в программном обеспечении. Это отправная точка для процесса разработки.

Виды требований

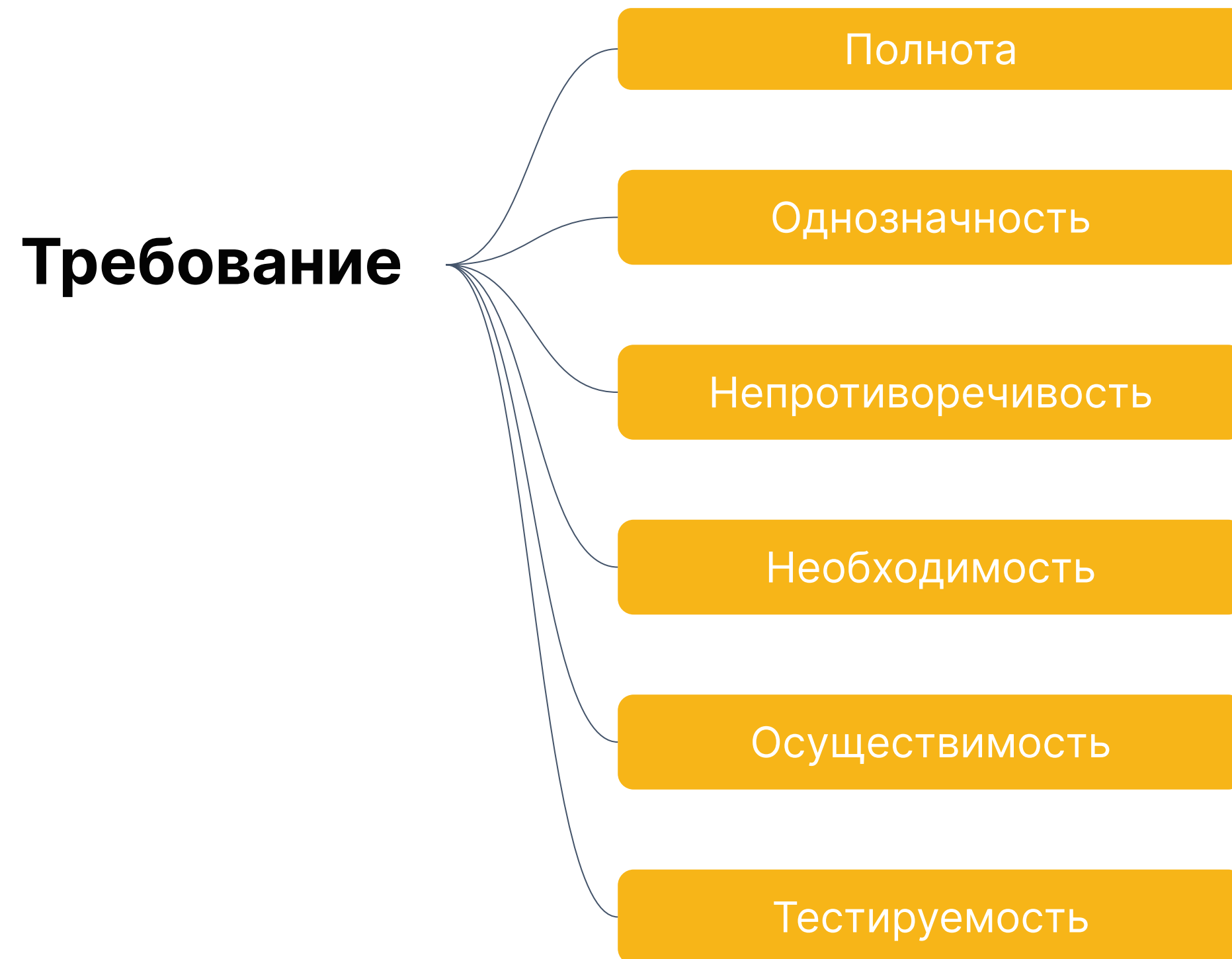


Атрибуты требований



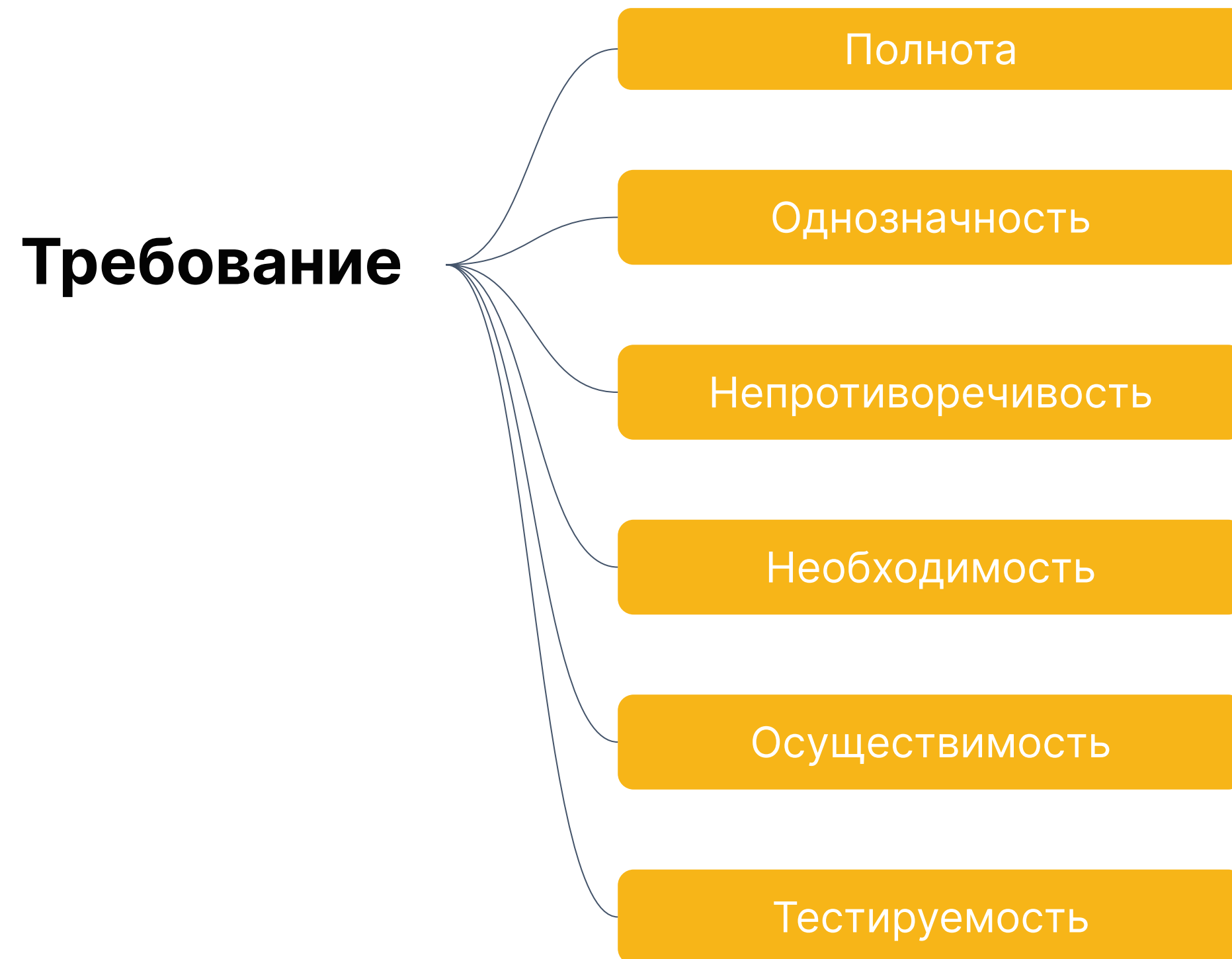
Описаны все возможные пользовательские
действие

Атрибуты требований



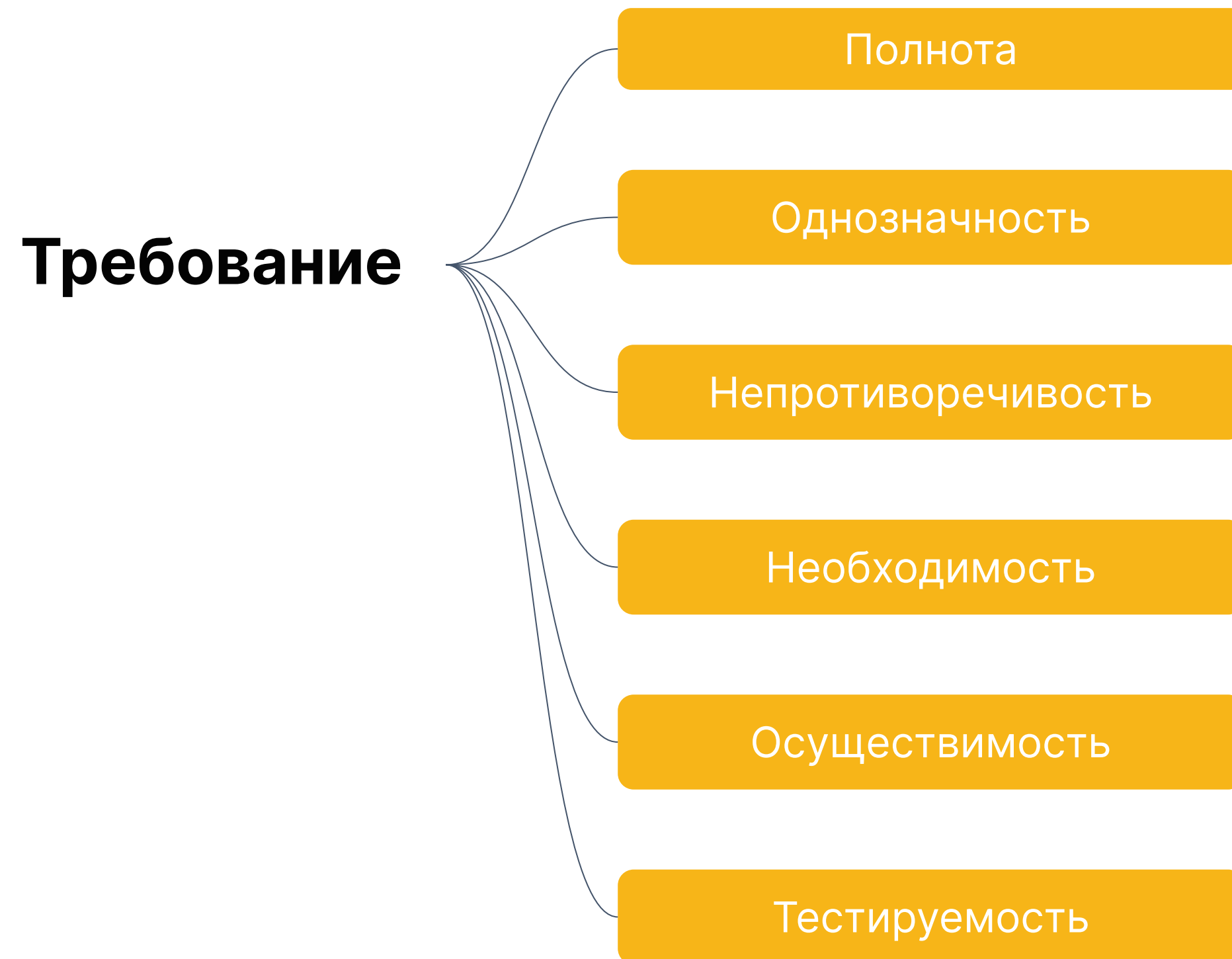
Отсутствуют все двусмысленные формулировки

Атрибуты требований



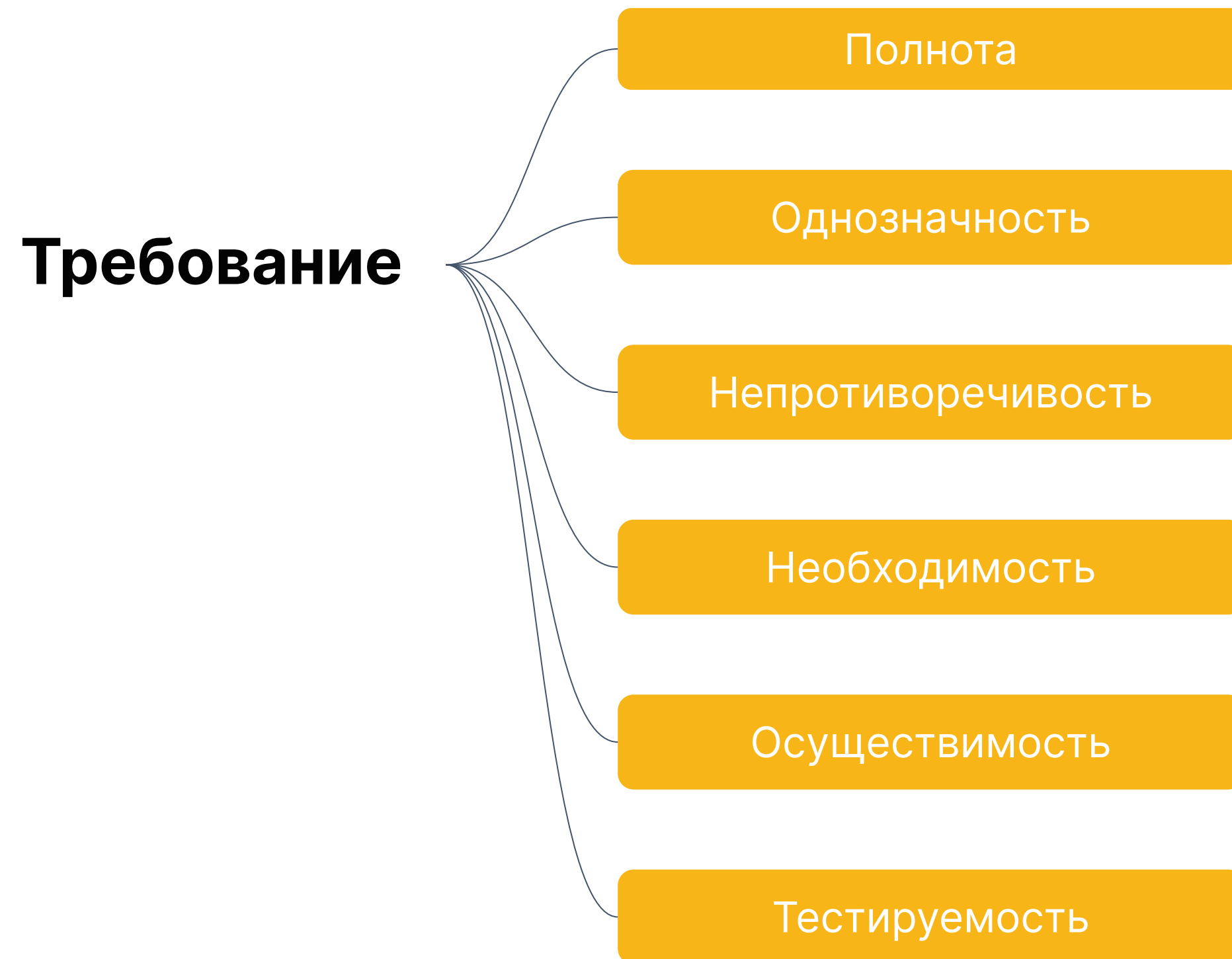
Нет противоречий между формулировками

Атрибуты требований



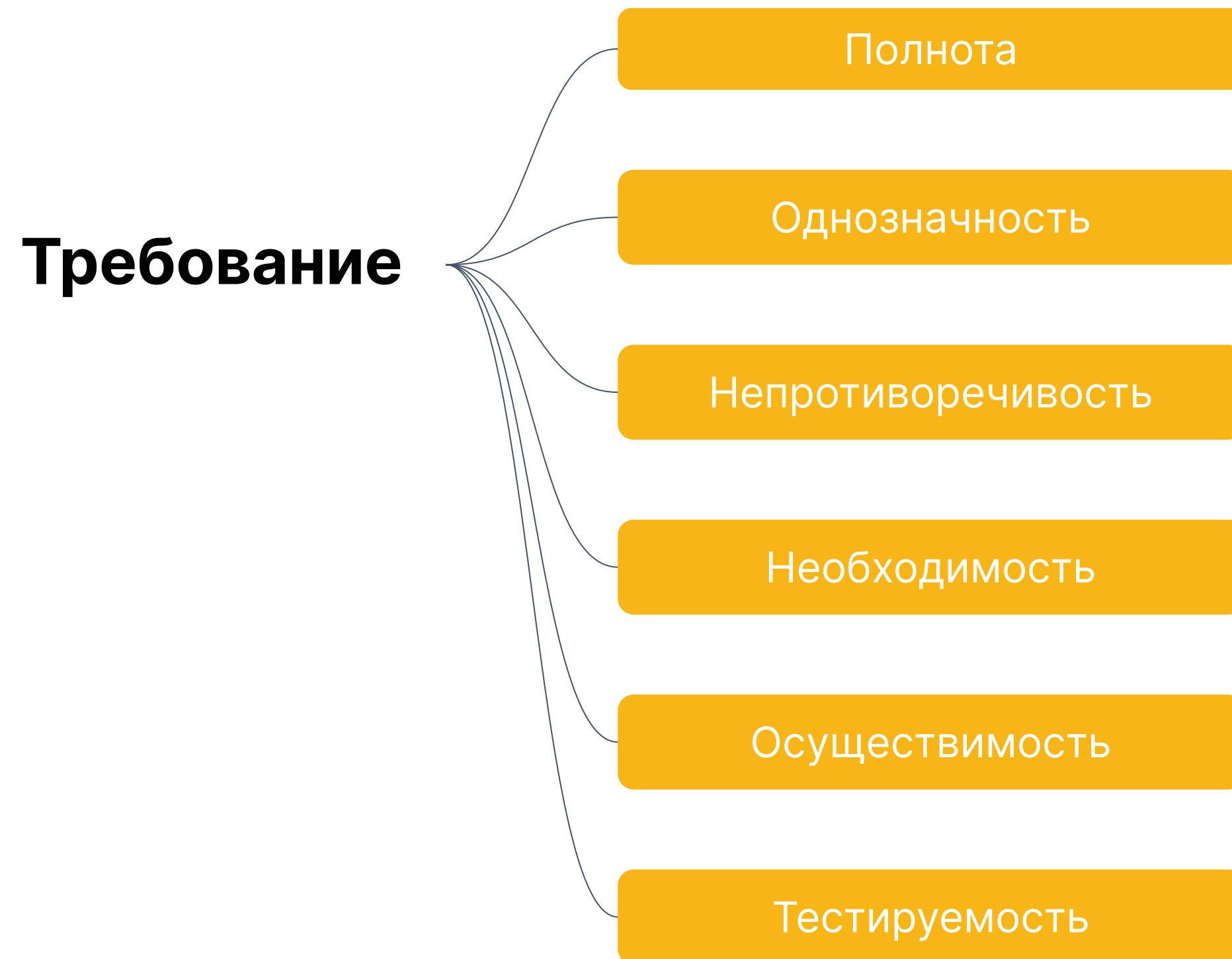
Отсутствует излишняя детализация

Атрибуты требований



Описанное требование можно реализовать

Атрибуты требований



Реализованную функциональность можно проверить

Пример

Имеется следующее требование. Как вы считаете, какие тут имеются ошибки?

Пользователь берет в руки смартфон от компании iPhone с версией 16.3. Открывает приложение, авторизуется путем ввода номера телефона, затем вводит проверочный код, который получит по средствам СМС от оператора Kablenet. Ждет пока откроется Главный экран на котором он должен увидеть ленту с выдачей ресторанов. В ленте всего две колонки (как в альбомной, так и в портретной ориентации), где отображаются доступные для принятия заказа рестораны. Карточка ресторана содержит изображение ресторана, описание и количество. В альбомной ориентации отображается три колонки с ресторанами. Также в обеих ориентациях на карточки имеется кнопка “Заказать”. При нажатии на эту кнопку пользователь попадает на карточку ресторана.

ВОПРОСЫ

4

ДЕКОМПОЗИЦИЯ ТРЕБОВАНИЙ

Декомпозиция

разделение большого и сложного на небольшие простые части.

Декомпозиция требований

При работе с задачами специалисты разбивают абстрактную большую задачу на маленькие задачи, которые можно легко оценить, удобнее с ними работать.

Простой пример декомпозиции — сбор чемодана. Это большая задача, которую сложно решить. Но можно разбить её на задачи поменьше:

- составить список вещей,
- разбить на категории,
- достать чемоданы,
- укладывать вещи по категориям.

Уже на этом уровне декомпозиции задача перестаёт казаться невыполнимой — можно понять, сколько времени и ресурсов понадобится на каждый этап.



Виды декомпозиций требований

1. Разбиение по позитивным и негативным сценариям.

Любая функциональность имеет как прямой (позитивный) сценарий использования, который приводит к ожидаемому результату, но и косвенный (негативный) сценарий, например:

- переданы не те данные,
- выполнены не все обязательные условия,
- нет необходимых прав доступа и т.д.



Виды декомпозиций требований

2. Разбиение по условиям бизнес процесса.

Требование разбивается на логические варианты работы функционала.

Например, можно выделить следующий набор правил для совершения покупки в интернет-магазине:

- Показывается минимальная сумма, если сумма покупки меньше, то клиенту отображается соответствующая подсказка.
- Если сумма покупки больше какой-то суммы, то пользователю предлагаются дополнительные варианты оплаты (бонусами).
- Если выставленный счет не оплачен в течение 1 часа, то заказ автоматически отменяется.



Виды декомпозиций требований

3. Разбиение по видам операций.

Существуют стандартные операции, которые часто встречаются в различных функциях:

- **C**reate - создание нового продукта в интернет магазине
- **R**ead - просмотр описания продукта
- **U**ppdate - редактирование/обновление описания продукта
- **D**eleate - удаление продукта из магазина



Виды декомпозиций требований

4. Декомпозиция по типам платформы.

Например, нам необходимо разработать в интернет-магазине функцию привязки банковской карты. В этом случае можно декомпозировать требование на задачи по реализации функции покупки:

1. Для разных платформ: ПК, планшеты, смартфоны.
2. Для разных ОС: Windows, iOS, Android.
3. Для работы в различных браузерах.



Виды декомпозиций требований

5. Разбиение по типам данных и параметрам.

Мы можем разбить большое требование на ряд мелких пользовательских историй (user story), в рамках каждой из которых нужно реализовать работу только с каким-то одним типом данных.

1. Ввод номера карты в поле для ввода карты
2. Ввод букв в поле для ввода номера карты
3. Ввод спецсимволов в поле для ввода номера карты



Виды декомпозиций требований

6. Разбиение по ролям.

Многие бизнес процессы и функциональности часто подразумевают участие\работу с ними нескольких ролей и групп пользователей.

- неверифицированный пользователь
- верифицированный пользователь
- администратор



Виды декомпозиций требований

7. Декомпозиция по сценариям тестирования или тест-кейсам.

Разбиение больших пользовательских историй, задавая вопрос, как та или иная часть функциональности будет проверена. Мы определяем какие сценарии необходимо проверить, какие тесты выполнить, чтобы узнать, работает ли эта функция.

1. Товар есть в наличии и он доступен покупки
2. Товар есть в наличии, но он уже зарезервирован другим покупателем
3. Товар есть в наличии, он появился после отмены заказа первым покупателем
4. Товара нет в наличии



Работа с требованиями

Валидация - определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, ЕГО требованиям к системе.

Верификация - процесс оценки того, насколько система (программа, устройство) по итогам некоторого этапа ее разработки соответствует условиям, заданным в начале этапа.



Источник изображения: <https://securelist.ru/validaciya-i-verifikaciya/27213/>

ВОПРОСЫ

5

ПОДХОДЫ К ТЕСТИРОВАНИЮ

Подходы к тестированию

На основе тест-кейсов

На основе чек-листов

Исследовательское тестирование



6

ЧЕК-ЛИСТ

Чек-лист

это список, содержащий ряд проверок во время тестирования программного продукта.

Как вы составляете список продуктов, которые необходимо купить?



Чек-лист в жизни

Овощи и фрукты:

- картофель
- морковь
- зелень
- клубника

Колбасный отдел:

- колбаса
- сосиски

Молочный отдел:

- молоко
- творог
- кефир

Соки и воды:

- вода
- яблочный сок



Чек-лист на проекте

модуль

подмодуль

упорядоченный список

Страница "Настройки"		
Профиль		
1	Раздел "Профиль"	failed
2	Аватарка пользователя	passed
3	Фамилия и Имя пользователя	passed
4	Отображение города, страны проживания	passed
5	Должность сотрудника	failed
6	Адрес проживания сотрудника	passed
7	Номер телефона	passed
8	Поле "Имя"	passed
9	Поле "Фамилия"	failed
10	Поле "Дата рождения"	failed
11	Поле "Ваш адрес проживания"	passed
12	Поле "Ваше подразделение"	failed
13	Поле "Выберите пол"	failed
14	Поле "Персональный код"	failed
15	Прикрепление аватарки пользователя	passed
16	Раздел "Требования к паролю"	failed
17	Старый пароль	passed
18	Новый пароль	failed
19	Подтверждение пароля	failed
20	Раздел "У вас есть домашние животные?"	skipped
21	Блок "Какие у вас есть сертификаты об образовании?"	skipped
22	Кнопка "Сохранить изменения"	failed

результат
проверки

Чек-листы

Кто исполняет и занимается актуализацией чек-листов?

Команда или один тестировщик проходят и отмечают пункты списка чек-листа.

Что происходит после того, как команда отметила все пункты чек-листа?

Делается вывод о готовности продукта к релизу.

Где обычно составляются чек-листы?

Excel, Checkvist, TestRail, TestIT, Qase.io...

На сколько детальным составлять чек-лист?

Все зависит от требований, уровня знания продукта сотрудниками и сложности продукта.

Правила составления

Один пункт – одна операция.

✓ Покупка товара анонимом

✗ Покупка товара анонимом и отмена заказа после покупки

Пункты начинаются с существительного.

✓ Авторизация в B2B пользователя через Facebook

✗ Авторизовался в B2B пользователя через Facebook

Составление чек-листа по уровням детализации.

Проверки должны быть разбиты по модулям и подмодулям, с целью структуризации

Статусы при прохождении по чек-листам

Passed – багов не обнаружено, все работает согласно документации;

Failed – найден баг, имеется расхождение с документацией;

Blocked – невозможно проверить, т.к. один из багов блокирует текущую проверку (например мы не можем проверить кейсы из раздела “профиль”, т.к у нас не работает авторизация, которую в свою очередь мы поместили как “failed”);

Skipped – проверяться не будет по какой-либо причине (например, текущий функционал еще не реализован);

Untested – пункт еще не брали в тестирование.

Плюсы и минусы

- ✚ Структурированная информация у всех участников команды по всему продукту.
- ✚ Экономия рабочего времени, т.к. составление чек-листов не требует большого количества времени.
- ✚ Исключение эффекта пестицида
- Список недостаточно детализирован и будет сложно новичкам ориентироваться в нем, отсюда большая вероятность пропуска багов.
- Со временем могут быть “потеряны” специфические знания про ту или иную особенность фичи.

ВОПРОСЫ

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Задание 1

Требование

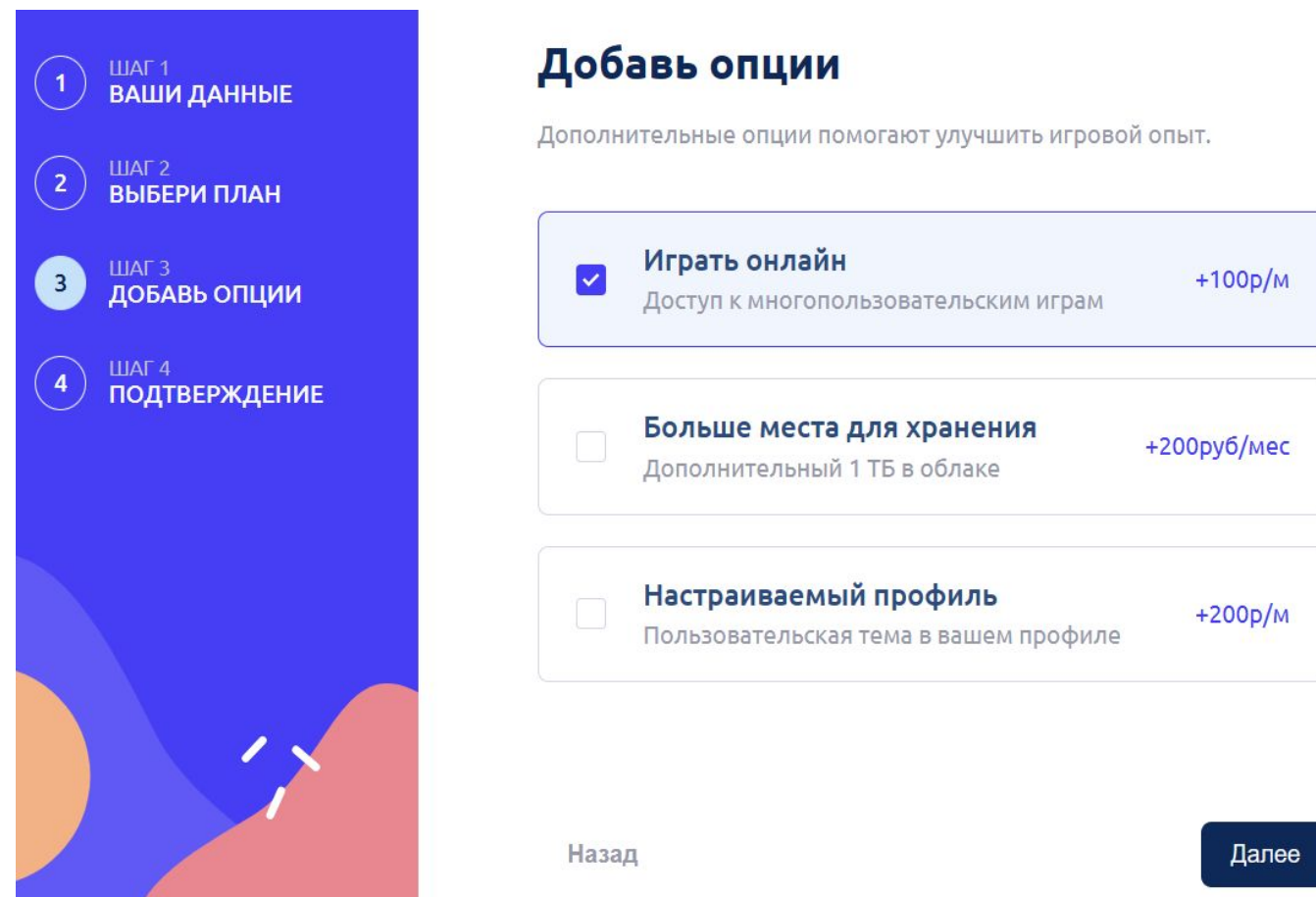
Шаг 3. Выбор дополнительных опций

В качестве дополнительных опций к каждому тарифу можно выбрать:

- Возможность играть онлайн - 100 руб/мес или 1000 руб/год;
- Получить 1 ТБ в облаке - 200 руб/мес или 2000 руб/год;
- Персонально настраивать профиль - 200 руб/мес или 2000 руб/год;

Пользователь может не выбрать ни одной опции, выбрать одну, две или все опции.

По дефолту при переходе на шаг 3 стоит выбор на опции играть онлайн.



The screenshot shows a mobile app interface for selecting additional options. On the left, a vertical sidebar contains four steps: 1. ВАШИ ДАННЫЕ, 2. ВЫБЕРИ ПЛАН, 3. ДОБАВЬ ОПЦИИ (highlighted with a red circle), and 4. ПОДТВЕРЖДЕНИЕ. The main area is titled 'Добавь опции' and includes a subtitle 'Дополнительные опции помогают улучшить игровой опыт.' Below this, there are three option cards. The first card, 'Играть онлайн' (+100р/м), is selected with a checked checkbox. The second card, 'Больше места для хранения' (+200руб/мес), and the third card, 'Настраиваемый профиль' (+200р/м), are not selected. At the bottom, there are 'Назад' and 'Далее' buttons.

Опция	Цена
<input checked="" type="checkbox"/> Играть онлайн Доступ к многопользовательским играм	+100р/м
<input type="checkbox"/> Больше места для хранения Дополнительный 1 ТБ в облаке	+200руб/мес
<input type="checkbox"/> Настраиваемый профиль Пользовательская тема в вашем профиле	+200р/м

Задание

Необходимо составить чек-лист на требование на раздел “Шаг 3”

Реализация:

[Выбор плана \(qa-faculty.github.io\)](https://qa-faculty.github.io)

Требование:

[Требования к форме выбора тарифного плана - Google Документы](#)

Задание 2



Персональные данные

Пожалуйста, введите ваше имя, адрес электронной почты и телефон.

Имя

Александр Пушкин

Email

poet@mail.ru

Телефон

+7 999 888 77 66

Далее

Задание

Необходимо составить чек-лист на требование на раздел “Шаг 1”

Реализация:

[Выбор плана \(qa-faculty.github.io\)](https://qa-faculty.github.io)

Требование:

[Требования к форме выбора тарифного плана - Google Документы](#)

7

КАК ПРОВОДИТЬ ИССЛЕДОВАТЕЛЬ- СКОЕ ТЕСТИРОВАНИЕ

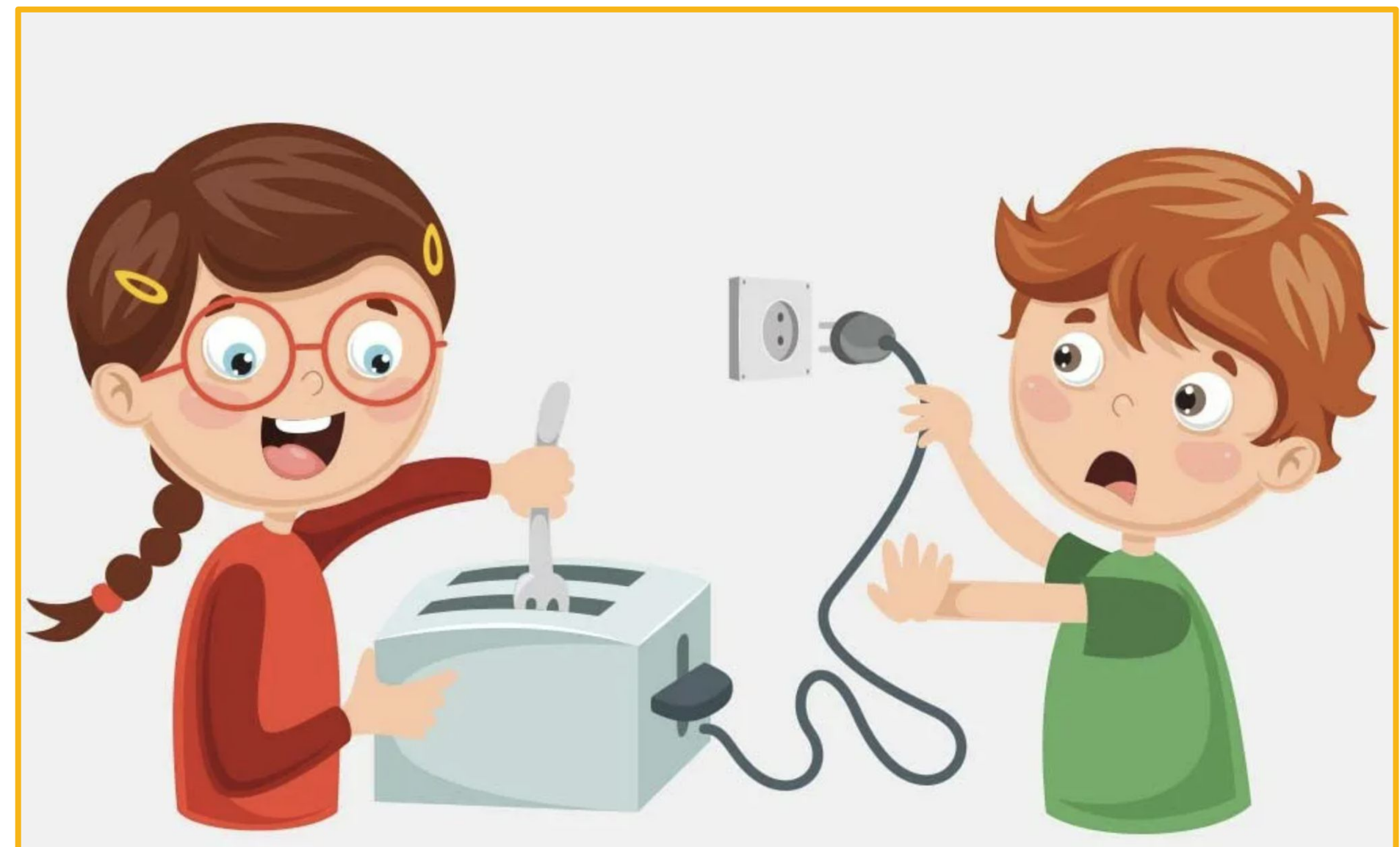
Исследовательское тестирование

это метод ручного тестирования, который базируется на взаимодействии с приложением без детальной подготовки, основанное на знаниях и опыте тестировщика, из-за чего его квалификация может серьезно повлиять на результат.

Исследовательское тестирование

По формальности документирования выделяют три вида тестирования:

1. Ad-hoc тестирование,
2. Исследовательское тестирование,
3. Сценарное тестирование.



ad-hoc тестирование

Под ad-hoc тестированием понимают тестирование без использования спецификаций, планов и разработанных тест-кейсов — здесь преимущественно чистая импровизация. В таком виде тестирования полностью отсутствует предварительная подготовка, происходит наиболее достоверная имитация случайного пользователя.



Исследовательское тестирование

Исследовательское тестирование — более формальная версия ad-hoc: тестирование, не требует написания тест-кейсов без необходимости, но подразумевает, что каждый последующий шаг(тест) выбирается на основании результата предыдущего шага(теста). А по Сэму Канеру, «Testing Computer Software», «исследовательское тестирование» — вдумчивый подход к ad-hoc тестированию.



Сценарное тестирование

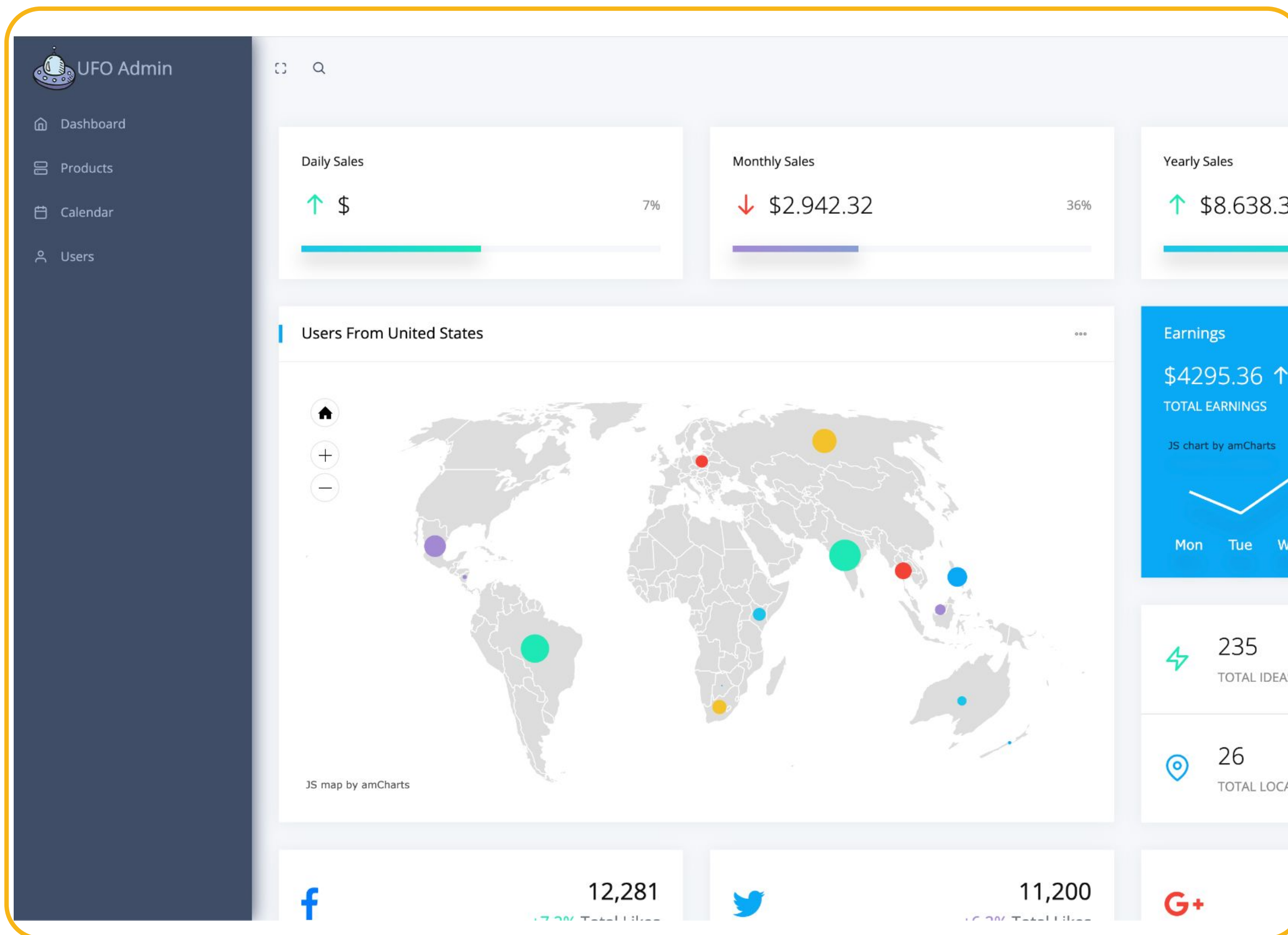
Сценарное тестирование является классическим тестированием по предварительно написанным и задокументированным сценариям. Оно требует наибольшей степени формальности и детализации документирования, что затрачивает гораздо больше сил тестировщика и его времени, однако наиболее подходит для ведения отчетности, статистики и т.п.



ВОПРОСЫ

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Задание 2



Задание

1. Создать архитектуру чек-листа
2. Составляем чеклист, содержащий от 3-5 проверок.

Проект доступен по следующей ссылке:

<http://ufo.telran-edu.de:8888/admin>

login: admin password: 123

Техническое задание:

https://docs.google.com/document/d/1wEPABKq072R0G_YYq_uODLpmOxPT5d0xccSb01DjX8E/edit?usp=sharing

ОСТАВШИЕСЯ ВОПРОСЫ

Итоги

- Занятие 1 - Введение, знакомство, цели
- Занятие 2 - Жизненный цикл разработки ПО, модели жизненного цикла, методологии разработки, качество ПО
- **Занятие 3 - Требования, атрибуты требований. Тестирование на основе чек-листов.**
- Занятие 4 - Чек-листы. Практика. Исследовательское тестирование.
- Занятие 5 - Виды тестирования. Тестирование на основе тест-кейсов.
- Занятие 6 - Тест-кейсы, практика.
- Занятие 7 - Тест-дизайн
- Занятие 8 - Ошибка, дефект, сбой/отказ. Правила составления баг-репортов.
- Занятие 9 - Особенности тестирования веб-приложений
- Занятие 10 - Инструменты тестирования веб-приложений

Домашнее задание

Задание 1. Ознакомиться с техническим заданием <https://docs.google.com/document/d/1wEPABKq072R0G_YYq_uODLpmOxPT5d0xccSb01DjX8E/edit> (стр9) и составить чеклист с проверками на страницу Contacts: от 3-х до 5 проверок.

Домашнее задание

Требование

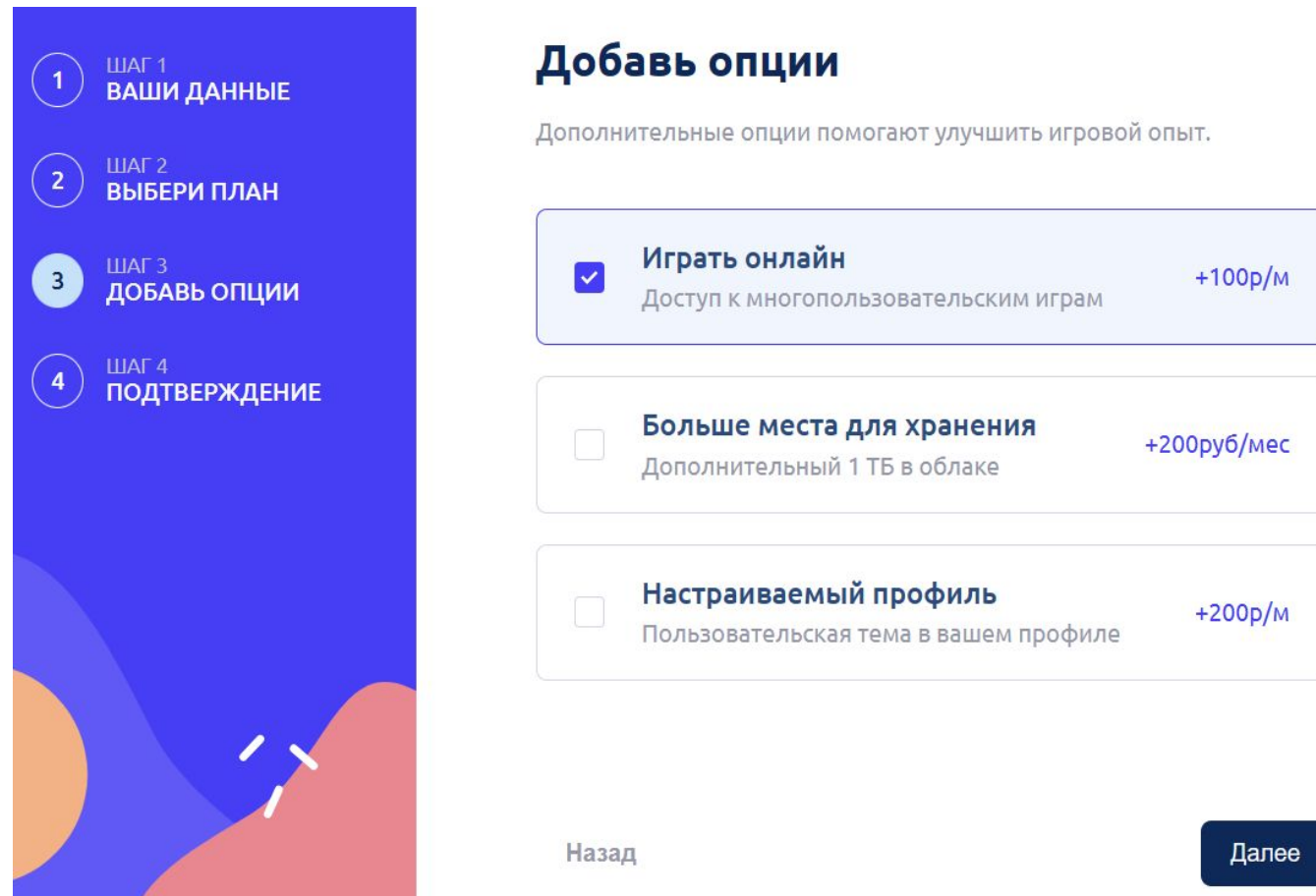
Шаг 3. Выбор дополнительных опций

В качестве дополнительных опций к каждому тарифу можно выбрать:

- Возможность играть онлайн - 100 руб/мес или 1000 руб/год;
- Получить 1 ТБ в облаке - 200 руб/мес или 2000 руб/год;
- Персонально настраивать профиль - 200 руб/мес или 2000 руб/год;

Пользователь может не выбрать ни одной опции, выбрать одну, две или все опции.

По дефолту при переходе на шаг 3 стоит выбор на опции играть онлайн.



Добавь опции

Дополнительные опции помогают улучшить игровой опыт.

<input checked="" type="checkbox"/>	Играть онлайн Доступ к многопользовательским играм	+100р/м
<input type="checkbox"/>	Больше места для хранения Дополнительный 1 ТБ в облаке	+200руб/мес
<input type="checkbox"/>	Настраиваемый профиль Пользовательская тема в вашем профиле	+200р/м

Назад Далее

Задание

Необходимо составить чек-лист **на требование** на раздел “Шаг 3”
5-6 проверок

Реализация:

[Выбор плана \(qa-faculty.github.io\)](https://qa-faculty.github.io)

Требование:

[Требования к форме выбора тарифного плана - Google Документы](#)

Полезные ссылки

- [Что такое хороший чек-лист и как его составить](#)
- [Чек-листы в помощь QA](#)
- [Что такое чек-лист в тестировании?](#)
- Чит листы по тестированию [Чит-лист по Web UI контролам от Игоря Любина — Testopedia \(software-testing.ru\)](#)

Вывод

Сегодня мы с вами узнали как правильно работать с требованиями, какие подходы к составлению тестовой документации существуют, а также научились составлять чек-листы.

КТО МОЛОДЦЫ?

