

Основы тест-аналитики

Тестирование по сценариям



На этом уроке

1. Рассмотрим варианты пользовательских сценариев
2. Создадим тест-кейсы для системного или приемочного тестирования на основе сценария

Оглавление

[Что такое сценарии использования?](#)

[Пример 1. Пользовательский сценарий - список шагов](#)

[Пример 2. Диаграмма вариантов использования](#)

[Системное тестирование](#)

[Составление тест-кейсов на основе пользовательских сценариев](#)

[Тест-кейс 1](#)

[Тест-кейс 2](#)

[Тест-кейс 3](#)

[Глоссарий](#)

[Контрольные вопросы](#)

[Домашнее задание](#)

[Дополнительные материалы](#)

[Используемые источники](#)

Что такое сценарии использования?

Use Case (сценарий использования, пользовательский сценарий) — это сценарий, по которому пользователь взаимодействует с приложением, программой для выполнения действия или для достижения цели. Тестирование по сценариям обнаруживает баги, которые сложно найти в тестировании частей приложения отдельно друг от друга. Сценарное тестирование может проводиться как часть приемочного тестирования.

Бизнес-заказчик предоставляет сценарии использования наряду с другой продуктовой документацией: user story, пользовательскими и системными требованиями. Как правило, одной user story соответствует один или несколько пользовательских сценариев. Тест-кейсы составляются таким образом, чтобы покрыть все представленные сценарии использования.

Сценарии использования описываются в разных формах:

- список шагов;
- таблица с основными и альтернативными сценариями;
- диаграмма вариантов использования;
- диаграмма классов.

Пример 1. Пользовательский сценарий - список шагов

User story: Я как администратор хочу настраивать проектные роли через панель администратора.

Use case 1

1. Как администратор системы, перехожу в раздел Проектные роли
2. Создаю новую роль
3. Выбираю уровень доступа для каждого раздела (по умолчанию уровень доступа - запрещено)
4. Сохраняю роль
5. Роль доступна в списке проектных ролей

Use case 2

1. Как администратор системы, перехожу в раздел Проектные роли
2. Редактирую существующую роль
3. Меняю уровни доступа для любого раздела
4. Сохраняю роль
5. Роль доступна в списке ролей, изменения роли применены сразу

Use case 3

1. Как администратор системы перехожу в раздел Проектные роли
2. Удаляю существующую роль

3. Роль недоступна в списке ролей, удалена из всех проектов где использовалась, изменения применяются сразу.

Use case 4

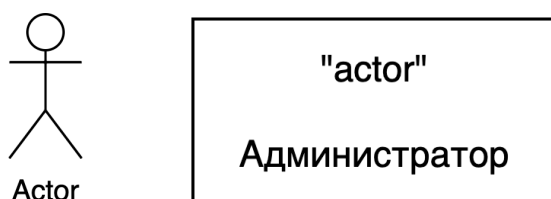
1. Как администратор системы перехожу в раздел Проектные роли
2. Создаю 2 роли:
 - а. Конфигурации - Запрет, Автотесты - Просмотр
 - б. Конфигурации - Просмотр, Автотесты - Запрет
3. 2 роли выданы 1 пользователю
4. Пользователь имеет: Конфигурации - Просмотр, Автотесты - Просмотр (т.е. права суммируются)

Пример 2. Диаграмма вариантов использования

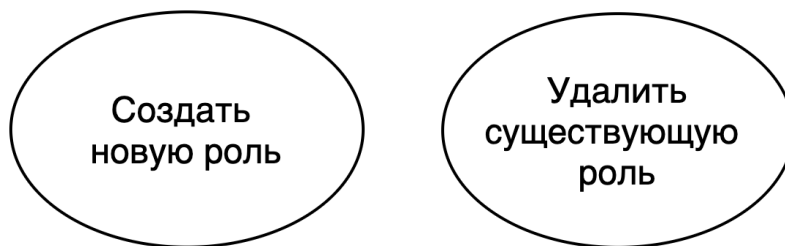
Диаграмма вариантов использования (сценариев поведения, прецедентов) - это концептуальное представление системы в процессе проектирования и разработки. Она состоит из **актеров**, **вариантов использования** и **отношений между ними**.

Проектируемая система представляется в виде множества актеров, взаимодействующих с системой с помощью вариантов использования. При этом актером (действующим лицом, актантом, актором) называется любой объект, субъект или система, взаимодействующая с моделируемой системой извне. Вариант использования – это спецификация сервисов (функций), которые система предоставляет актеру. Диаграмма составляется в соответствии с нотацией UML.

Как правило, актер изображается в виде человечка, однако возможны варианты: произвольная иконка или класс с текстовым стереотипом «actor».

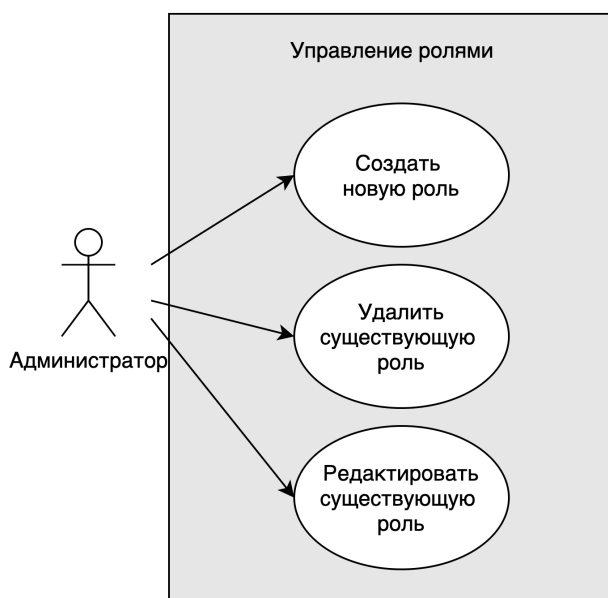


Вариант использования обозначается эллипсом, внутри которого содержится его описание, обозначающее выполнение какой-либо операции или действия.



Связи между актерами и вариантами отображаются с использованием различных видов отношений. Подробнее о них можно узнать в [статье](#).

Для процесса “Управление проектными ролями” в Test IT диаграмма вариантов использования может выглядеть так:



Пользовательские сценарии лежат в основе системного тестирования.

Системное тестирование

В отличие от модульного и интеграционного тестирования, системное тестирование рассматривает программу как совокупность большого количества модулей и подсистем. Как правило, системное тестирование осуществляется через пользовательский интерфейс. На уровне системы сложно и малоэффективно отслеживать правильность работы отдельных функций. Основная задача системного тестирования - в выявлении дефектов, вызванных работой системы в целом: неверное использование ресурсов, не предусмотренные комбинации пользовательских данных, несовместимость с окружением, нетипичные сценарии использования, неудобство использования.

Системное тестирование производится методом черного или серого ящика. Системные тесты выявляют ошибки:

- Неполное или неверное решение функциональных задач.
- Неэффективное использование ресурсов (утечка памяти, возврат ресурсов).
- Низкая производительность.
- Отсутствие защиты от искажения данных и некорректных действий.
- Ошибки инсталляции и конфигурации на разных платформах.
- Ошибки документации

Для минимизации рисков, связанных с особенностями поведения системы в той или иной среде, во время тестирования используют окружение, максимально приближенное к боевому.

Системные тесты удобно писать на основе сценария использования, т.к. он содержит все шаги, которые выполняет пользователь. Остается обогатить сценарий тестовыми данными, и тест-кейс готов.

Составление тест-кейсов на основе пользовательских сценариев

Рассмотрим составление тест-кейса из пользовательского сценария - описания шагов. Это самый просто способ, т.к. шаги пользовательского сценария станут основой тест-кейса.

Алгоритм написания тест-кейса по пользовательскому сценарию:

1. Определить, какие параметры влияют на выполнение каждого шага
2. С помощью техник тест-дизайна выявить значения параметров
3. Скомпоновать параметры и представить в виде одного или нескольких тест-кейсов.

Рассмотрим создание тест-кейса по сценарию

Use case 1

1. Как администратор системы, перехожу в раздел Проектные роли
2. Создаю новую роль
3. Выбираю уровень доступа для каждого раздела (по умолчанию уровень доступа - запрещено)
4. Сохраняю роль
5. Роль доступна в списке проектных ролей

Выделим параметры и значения для каждого шага ([документация по ролям](#)). Значения для негативных проверок выделим красным.

| № шага | Параметры | Значения |
|--------|---------------------------------|--|
| 1 | Роль пользователя | <ul style="list-style-type: none"> Администратор Иная |
| 2 | Имя роли | <ul style="list-style-type: none"> Пустое от 1 до 30 символов 1000 символов разные вариации по структуре |
| 3 | Уровни доступа роли | <ul style="list-style-type: none"> Все “Запрещено” (по умолчанию) Все отличное от “Запрещено” Комбинация “Запрещено” с иными доступами изменить права, которые нельзя изменить |
| 4 | Роль пользователя | <ul style="list-style-type: none"> Администратор Иная |
| 5 | Откуда открывается список ролей | <ul style="list-style-type: none"> Панель администрирования Настройки проекта |

Из таблицы получаем тест-кейсы:

1. Успешное создание роли администратором
2. Создание роли пользователем без прав администратора
3. Создание роли с пустым названием
4. Создание роли с модификацией неизменяемых прав

Сценарии 3 и 4 можно объединить, т.к. проверки небольшие.

Тест-кейс 1

Название: Успешное создание роли Администратором.

Предусловие: Есть учетная запись пользователя с ролью Администратор.

| № | Шаг | Тестовые данные | Ожидаемый результат |
|---|---|--|----------------------------------|
| 1 | Залогиниться в системе | Пользователь - Администратор | Открыта стартовая страница |
| 2 | Перейти в панель администратора, вкладка “Редактор ролей” | | Открыт редактор ролей |
| 3 | Нажать кнопку “Создать роль” и ввести имя роли | <ul style="list-style-type: none"> от 1 до 30 символов 1000 символов разные вариации по структуре | Роль создана |
| 4 | Нажать на карандаш рядом с названием роли | | Открыт экран редактирования прав |

| | | | |
|---|--------------------------------------|---|---|
| 5 | Настроить права | <ul style="list-style-type: none"> • Все “Запрещено” (по умолчанию) • Все отличное от “Запрещено” • Комбинация “Запрещено” с иными доступами | Права настроены |
| 6 | Сохранить роль | Пользователь - Администратор | Роль сохранена |
| 7 | Проверить отображение созданной роли | <ul style="list-style-type: none"> • Панель администратора • Настройки проекта | Роль отображается так, как была создана |

Тест-кейс 2

Название: Создание роли пользователем без прав Администратора.

Предусловие: Есть 2 учетные записи с ролью Администратор.

| № | Шаг | Тестовые данные | Ожидаемый результат |
|---|--|---|----------------------------------|
| 1 | Залогиниться в системе | Пользователь - Администратор-1 | Открыта стартовая страница |
| 2 | Перейти в панель администратора, вкладка “Редактор ролей” | | Открыт редактор ролей |
| 3 | Нажать кнопку “Создать роль” и ввести имя роли | <ul style="list-style-type: none"> • от 1 до 30 символов | Роль создана |
| 4 | Нажать на карандаш рядом с названием роли | | Открыт экран редактирования прав |
| 5 | Настроить права | Изменить права, которые не модифицируются | Права нельзя изменить |
| 6 | Залогиниться в систему под другой учеткой и снять права с Администратора-1 | Администратор-2 | Права сняты |
| 7 | Сохранить роль | Администратор-1 | Ошибка при сохранении роли |
| 8 | Перейти в панель администратора | Администратор-1 | Панель недоступна |
| 9 | Проверить отображение созданной роли | Администратор-2 | Роль не отображается |

Тест-кейс 3

Название: Негативные проверки полей при создании роли

Предусловие: Есть учетная запись с ролью Администратор.

| № | Шаг | Тестовые данные | Ожидаемый результат |
|---|---|---|---|
| 1 | Залогиниться в системе | Пользователь - Администратор | Открыта стартовая страница |
| 2 | Перейти в панель администратора, вкладка "Редактор ролей" | | Открыт редактор ролей |
| 3 | Нажать кнопку "Создать роль" и ввести имя роли | <ul style="list-style-type: none">от 1 до 30 символов | Роль создана |
| 4 | Нажать на карандаш рядом с названием роли | | Открыт экран редактирования прав |
| 5 | Настроить права | Изменить права, которые нельзя менять | Права не меняются |
| 6 | Изменить название роли | <ul style="list-style-type: none">Оставить пустымОдни пробелы | Название не меняется |
| 7 | Проверить отображение созданной роли | <ul style="list-style-type: none">Панель администратораНастройки проекта | Роль отображается так, как была создана первоначально |

Глоссарий

Use Case (сценарий использования, пользовательский сценарий) — это сценарий, по которому пользователь взаимодействует с приложением, программой для выполнения действия или для достижения цели.

Диаграмма вариантов использования (сценариев поведения, прецедентов) - это концептуальное представление системы в процессе проектирования и разработки. Состоит из актеров, вариантов использования и отношений между ними.

Домашнее задание

User story: Я как администратор хочу настраивать проектные роли через панель администратора.

Use case 3

1. Как администратор системы перехожу в раздел Проектные роли
2. Удаляю существующую роль
3. Роль недоступна в списке ролей, удалена из всех проектов где использовалась, изменения применяются сразу.

Для данного пользовательского сценария составьте:

1. Таблицу “Шаг - Параметры - Значения”. Негативные значения выделите красным
2. Два тест-кейса: один позитивный и один негативный

Результат внесите в [документ](#) (предварительно создав личную копию)

Используемые источники

1. https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema12/tema12_2
2. <https://habr.com/ru/company/luxoft/blog/312188/>
3. https://ashamray.wordpress.com/2011/03/02/tips_for_writing_good_use_cases/