

Тест-дизайн и тест-аналитика
Урок 5

Диаграммы и таблицы переходов состояний





Оглавление

Диаграмма переходов состояний	3
Диаграмма переходов состояний	3
Составление тестов	5
Таблицы переходов состояний	7
Составление тестов	9
Пример «Публикация поста»	11
Контрольные вопросы	18
Глоссарий	19
Дополнительные материалы	19



На этом уроке

1. Познакомимся с диаграммой переходов состояний.
2. Научимся составлять таблицы переходов состояний.
3. Ознакомимся с примером “Публикация поста”.

Диаграмма переходов состояний

Диаграмма переходов состояний — это техника тест-дизайна для систем, которые изменяют характеристики в зависимости от внешних стимулов. Иными словами, система принимает то или иное состояние и находится в нём до тех пор, пока не получит стимул для изменения.

Хрестоматийный пример переходов состояний приведён в книге Л. Копланда «A Practitioner's Guide to Software Test Design». Рассмотрим пользовательский сценарий «Покупка билета на самолёт».

Вариант 1

Пользователь: оформляет заказ на билет (предоставляет данные)

Система: создаёт бронь

Система: запускает таймер, до истечения которого нужно оплатить билет

Пользователь: оплачивает билет

Система: выпускает билет

Контролер: принимает билет

Система: погашает билет

Вариант 2

Пользователь: оформляет заказ на билет (предоставляет данные)

Система: создаёт бронь

Система: запускает таймер, до истечения которого нужно оплатить билет

Пользователь: не оплатил билет до окончания таймера

Система: отменяет бронь

Вариант 3

Пользователь: оформляет заказ на билет (предоставляет данные)

Система: создаёт бронь

Система: запускает таймер, до истечения которого нужно оплатить билет

Пользователь: отменил заказ до окончания таймера



Система: отменяет бронь

Вариант 4

Пользователь: оформляет заказ на билет (предоставляет данные)

Система: создаёт бронь

Система: запускает таймер, до истечения которого нужно оплатить билет

Пользователь: оплачивает билет

Пользователь: возвращает билет

Система: оформляет возврат

Помимо агентов (системы, пользователя и контролера) в сценарии есть объект — билет, над которым выполняются различные действия. Также мы видим ряд условий: был ли оплачен заказ, истёк ли таймер. В этой ситуации четырёх позитивных тестов (по одному на каждый вариант использования) недостаточно, так как ошибки скрываются там, где условия нарушаются. Для более наглядного представления сценариев и выявления негативных кейсов используются диаграммы и таблицы переходов состояний.

Элементы диаграммы переходов состояний:

- **Точка входа** — пользователь ещё не начал взаимодействовать с системой.
- **Круг с названием состояний** — состояние, в которое система пришла из-за действий пользователя.
- **Стрелки** обозначают переход от одного состояния к другому.
- **События** — то, что приходит в систему извне и запускает изменение состояния (например, пользователь оплатил заказ). Пишется над стрелкой.
- **Действие** — то, что происходит внутри системы из-за смены состояния (например, запуск таймера). Пишется над стрелкой после **события**. Отделяется от события косой чертой — /.
- **Точка выхода** — взаимодействие с системой окончено, изменение состояния невозможно.

В случае с билетом состояния, события и действия выглядят так:

Состояния:

- Забронирован,
- Оплачен,
- Выпущен,
- Использован,

События:

- Указать персональные данные,
- Оплатить,
- Использовать,

Действия:

- Стартовать таймер,
- Выпустить,
- Возврат денег.



- Отменен по неоплате,
- Отмена,
- Отменен покупателем.
- Истёк таймер.

Диаграмма переходов состояний:

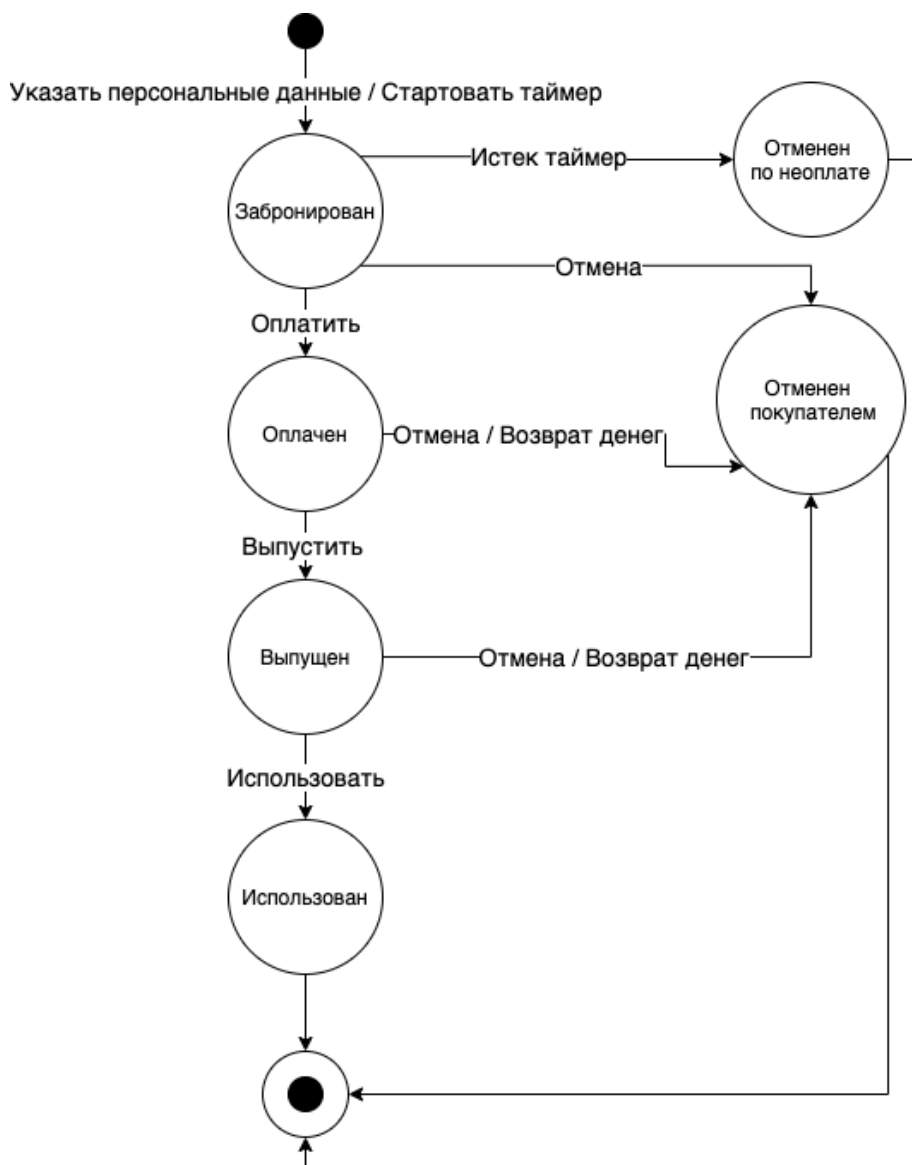


Диаграмма наглядно иллюстрирует бизнес-правила и варианты взаимодействия пользователя с системой. Если есть сложная логика, диаграммы обычно создаются аналитиками как часть документации к ПО.

Составление тестов

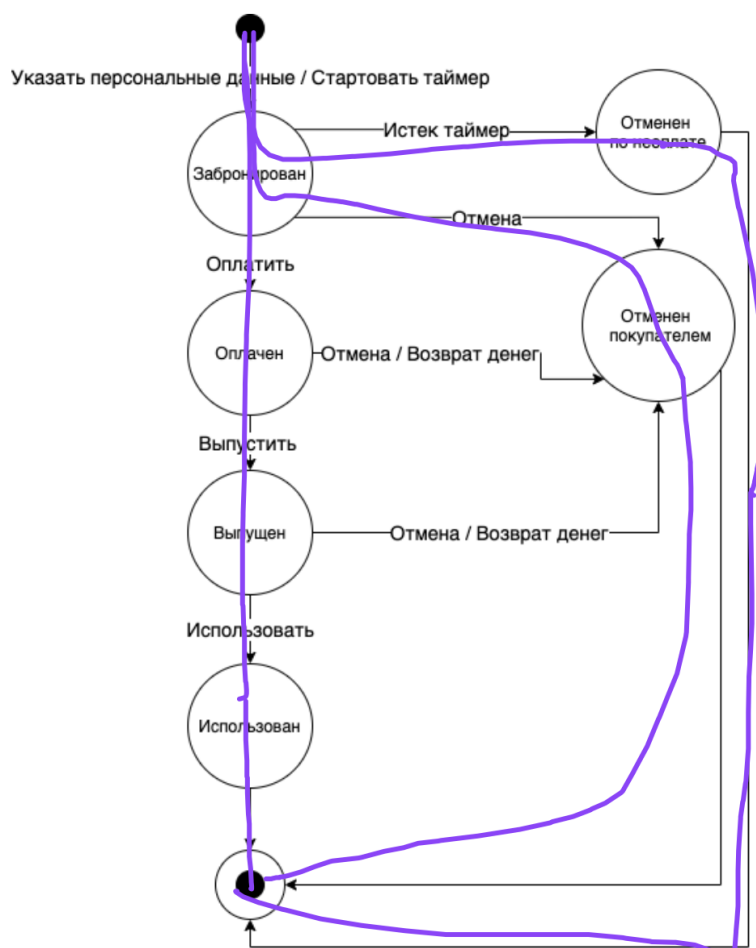
При составлении тест-кейсов по диаграммам переходов состояний, можно выделить 4 уровня покрытия:



1. **Каждое состояние затрагивает хотя бы один тест.** Тестовое покрытие неоптимально, так как тестируются не все действия.
2. **Каждое событие вызывается хотя бы в одном тесте.** Уровень покрытия также не оптимален.
3. **Каждый путь выполняется хотя бы в одном тесте.** Предпочтительный вариант, но бывает недостижимым. Если в диаграмме есть циклы, количество возможных путей становится бесконечным. Например, если есть два состояния (A и B), и есть переходы из состояния A в B и обратно из B в A, то возможны пути:

a. $A \rightarrow B$

b. $A \rightarrow B \rightarrow A$



c. $A \rightarrow B \rightarrow A \rightarrow B \rightarrow A \rightarrow B \rightarrow A \rightarrow B \rightarrow A \dots$

Важно тестировать циклы, так как в них могут накапливаться ошибки или происходить утечки памяти.



4. **Каждый переход используется хотя бы в одном тесте.** Уровень тестового покрытия оптимален, так как затронет все состояния, действия и события. Он может совпадать с покрытием всех возможных путей. В этом случае мы получим следующий набор тестовых сценариев:

- a. Использование оплаченного билета (*Забронирован — Оплачен — Выпущен — Использован*)
- b. Отмена брони по неоплате (*Забронирован — Отменен по неоплате*)
- c. Отмена брони покупателем (*Забронирован — Отменен покупателем*)
- d. Отмена покупателем после оплаты (*Забронирован — Оплачен — Отменен*)
- e. Отмена покупателем после выпуска (*Забронирован — Оплачен — Выпущен — Отменен*)

Недостаток составления тестов по диаграмме — отсутствие негативных сценариев. Что будет, если покупатель попытается использовать отменённый билет? Или оплатит его после того, как истечет таймер? Эти переходы не предусмотрены на схеме, так как при большом количестве стрелок она станет нечитаемой. Для негативных тестов удобнее использовать таблицу переходов.

Таблицы переходов состояний

Диаграммы переходов состояний — не единственный способ показать поведение системы. Они более понятны, но таблицы переходов состояний лучше систематизируют сведения о системе. Состоят из четырёх колонок:

- Текущее состояние,
- Событие,
- Действие,
- Новое состояние.

Алгоритм составления таблицы переходов состояний:

1. Сопоставить состояния и события по принципу «все ко всем» и заполнить столбцы «Текущее состояние», «Событие».
2. Там, где смена состояний вызывает действие, описать его в столбце «Действия».
3. В каждой строке указать новое состояние. Если событие не ведёт к смене состояния, то новое состояние совпадает с текущим.

Таблица переходов состояний для покупки билета:



Текущее состояние	Событие	Действие	Новое состояние
null	Указать персональные данные	Стартовать таймер	Забронирован
null	Оплатить	--	null
null	Использовать	--	null
null	Отмена	--	null
null	Истек таймер	--	null
Забронирован	Указать персональные данные	--	Забронирован
Забронирован	Оплатить	--	Оплачен
Забронирован	Использовать	--	Забронирован
Забронирован	Отмена	--	Отменен покупателем
Забронирован	Истек таймер	--	Отменен по неоплате
Оплачен	Указать персональные данные	--	Оплачен
Оплачен	Оплатить	--	Оплачен
Оплачен	--	Выпустить	Выпущен
Оплачен	Использовать	--	Оплачен
Оплачен	Отмена	Возврат денег	Отменен покупателем
Оплачен	Истек таймер	--	Оплачен
Выпущен	Указать персональные данные	--	Выпущен
Выпущен	Оплатить	--	Выпущен
Выпущен	Использовать	--	Использован
Выпущен	Отмена	Возврат денег	Отменен покупателем
Выпущен	Истек таймер	--	Выпущен
Использован	Указать персональные данные	--	Использован
Использован	Оплатить	--	Использован
Использован	Использовать	--	Использован



Использован	Отмена	--	Использован
Использован	Истек таймер	--	Использован
Отменен по неоплате	Указать персональные данные	--	Отменен по неоплате
Отменен по неоплате	Оплатить	--	Отменен по неоплате
Отменен по неоплате	Использовать	--	Отменен по неоплате
Отменен по неоплате	Отмена	--	Отменен по неоплате
Отменен по неоплате	Истек таймер	--	Отменен по неоплате
Отменен покупателем	Указать персональные данные	--	Отменен покупателем
Отменен покупателем	Оплатить	--	Отменен покупателем
Отменен покупателем	Использовать	--	Отменен покупателем
Отменен покупателем	Отмена	--	Отменен покупателем
Отменен покупателем	Истек таймер	--	Отменен покупателем

Составление тестов

Алгоритм составления тестов по таблице:

1. Выделить зелёным все возможные переходы.
2. Выявить невозможные переходы, которые несут риски, и выделить их красным. Это будут негативные проверки. Для выявления рисков тестировщики советуются с разработчиками или аналитиками.

При покупке билета существуют риски:

- покупатель использует неоплаченный, невыпущенный, ранее использованный или отменённый билет;
- покупатель оплатит билет после отмены брони.

Эти кейсы отмечены красным цветом.

Текущее состояние	Событие	Действие	Новое состояние
null	Указать персональные данные	Стартовать таймер	Забронирован
null	Оплатить	--	null



null	Использовать	--	null
null	Отмена	--	null
null	Истек таймер	--	null
Забронирован	Указать персональные данные	--	Забронирован
Забронирован	Оплатить	--	Оплачен
Забронирован	Использовать	--	Забронирован
Забронирован	Отмена	--	Отменен покупателем
Забронирован	Истек таймер	--	Отменен по неоплате
Оплачен	Указать персональные данные	--	Оплачен
Оплачен	Оплатить	--	Оплачен
Оплачен	--	Выпустить	Выпущен
Оплачен	Использовать	--	Оплачен
Оплачен	Отмена	Возврат денег	Отменен покупателем
Оплачен	Истек таймер	--	Оплачен
Выпущен	Указать персональные данные	--	Выпущен
Выпущен	Оплатить	--	Выпущен
Выпущен	Использовать	--	Использован
Выпущен	Отмена	Возврат денег	Отменен покупателем
Выпущен	Истек таймер	--	Выпущен
Использован	Указать персональные данные	--	Использован
Использован	Оплатить	--	Использован
Использован	Использовать	--	Использован
Использован	Отмена	--	Использован
Использован	Истек таймер	--	Использован
Отменен по неоплате	Указать персональные данные	--	Отменен по неоплате



Отменен по неоплате	Оплатить	--	Отменен по неоплате
Отменен по неоплате	Использовать	--	Отменен по неоплате
Отменен по неоплате	Отмена	--	Отменен по неоплате
Отменен по неоплате	Истек таймер	--	Отменен по неоплате
Отменен покупателем	Указать персональные данные	--	Отменен покупателем
Отменен покупателем	Оплатить	--	Отменен покупателем
Отменен покупателем	Использовать	--	Отменен покупателем
Отменен покупателем	Отмена	--	Отменен покупателем
Отменен покупателем	Истек таймер	--	Отменен покупателем

Диаграммы и переходы состояний не применимы, если система не изменяет состояние и не должна реагировать на внешние стимулы определённым образом.

Пример «Публикация поста»

В тестируемой системе есть функциональность публикации поста.

Что у вас нового?

Сейчас ▾ Тематика ▾

📷 🎥 🎵 📝 Ещё ▾ ⚙️ **Опубликовать**

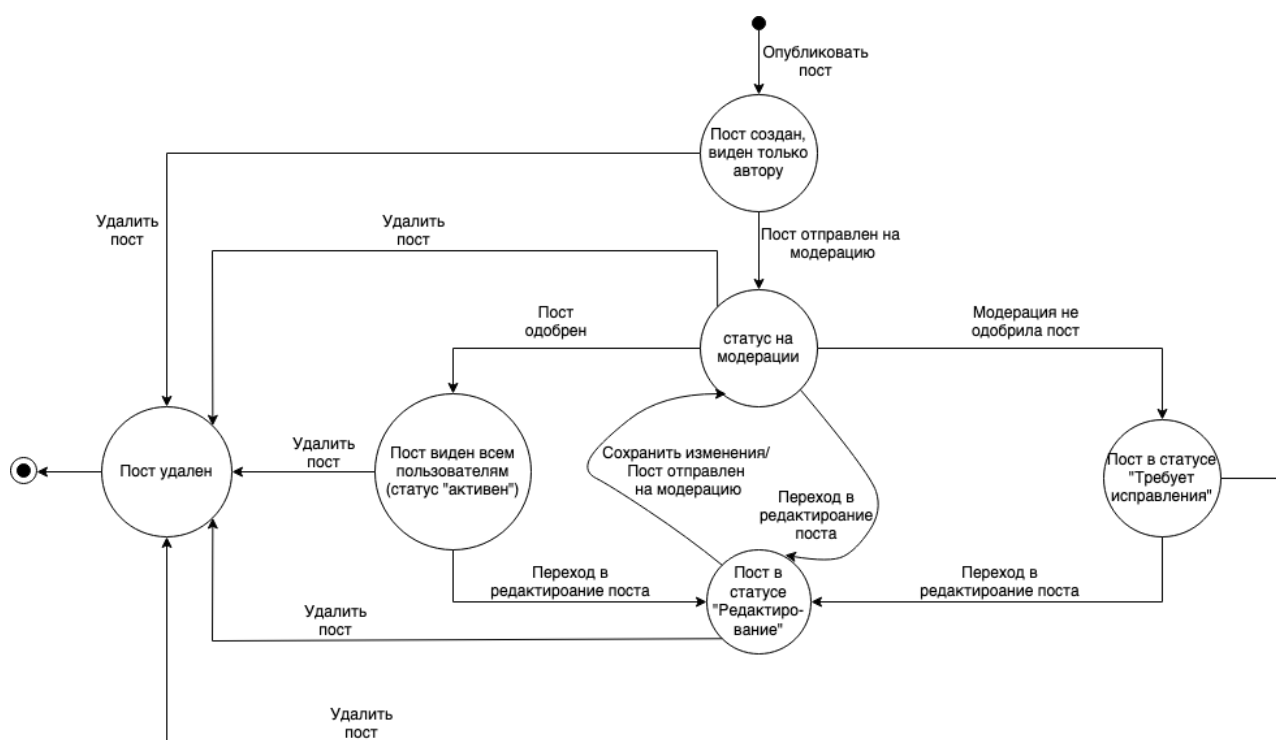
Поле создания поста выглядит так, как на скриншоте выше. Есть поле для ввода текста, значок добавления эмоджи, выбор времени публикации (сейчас или в определённое время), выбор тематики поста, выбор фона (можно выбрать картинку из предложенных или загрузить свою), добавление фото, видео, музыки, статьи, файла, карты, граффити, опроса (часть пунктов может быть скрыта в выпадающем списке «ещё»), настройка публикации и кнопка «Опубликовать».



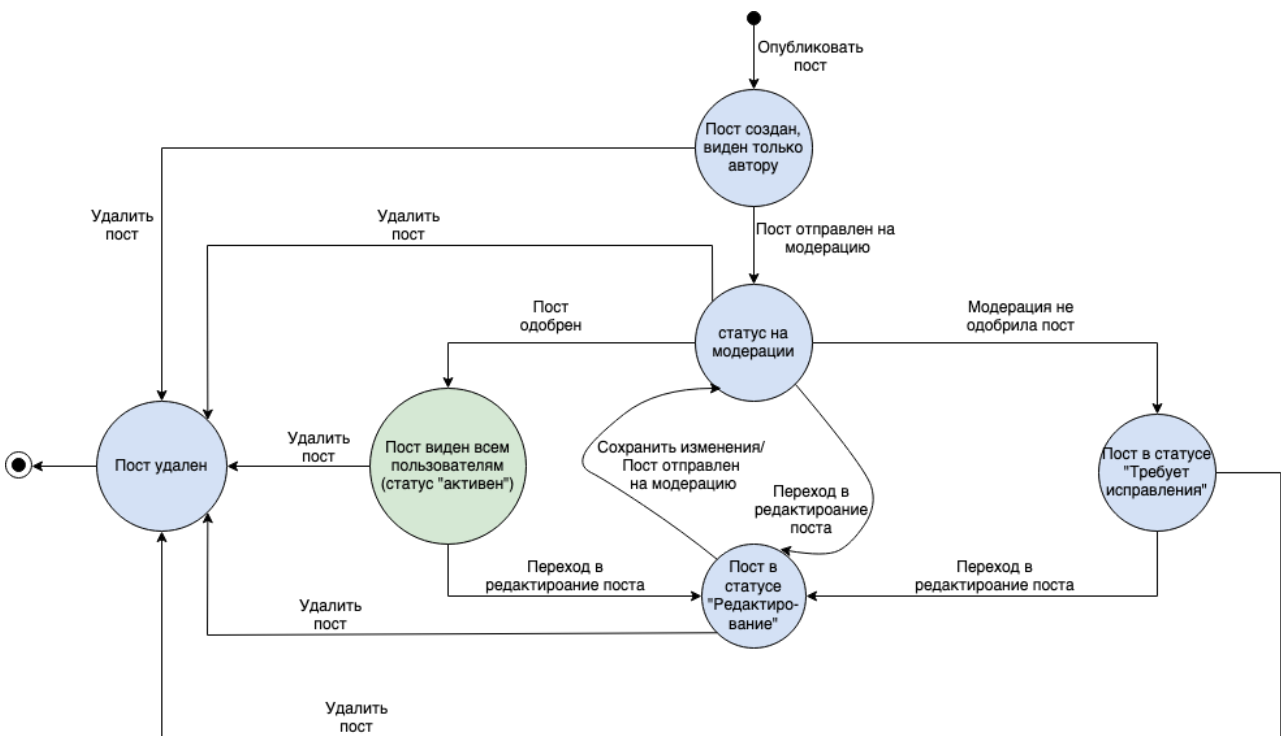
После ввода текста и нажатия на кнопку «Опубликовать», пост будет создан в системе. Затем он будет автоматически отправлен на модерацию. Пока пост на модерации, его не видят другие пользователи, но видит автор. В это время автор может его удалить или отредактировать, но после сохранения изменений пост повторно будет отправлен на модерацию.

Если модерация отклонила пост, он возвращается автору на исправление и повторную публикацию.

Если пост был одобрен модерацией, он становится видимым всем пользователям системы. Автор может удалить или отредактировать пост. После редактирования пост снова отправляется на модерацию.



Для наглядности выделим зелёным статусы поста, когда он виден всем, и голубым, когда он виден только автору. Получим такую диаграмму:



Теперь построим таблицу переходов состояний — для каждого состояния нужно рассмотреть каждое событие. Событие в паре с действием рассматриваем только тогда, когда действие относится именно к текущему состоянию. В остальных случаях указываем только событие. Там, где нет одного из вариантов, в соответствующем столбце будем ставить прочерк. Так как модерация — это какая-то интеграция в нашу систему, то ответ от модерации также считаем событием. Получим:

Текущее состояние	Событие	Действие	Новое состояние
null	Опубликовать пост	-	Пост создан, виден только автору
null	Пост одобрен	-	null
null	Модерация не одобрила пост	-	null
null	Переход в редактирование поста	-	null
null	Сохранить изменения	-	null
null	Удалить пост	-	null
Пост создан, виден только автору	Опубликовать пост	-	Пост создан, виден только автору



Пост создан, виден только автору	-	Пост отправлен на модерацию	Статус на модерации
Пост создан, виден только автору	Пост одобрен	-	Пост создан, виден только автору
Пост создан, виден только автору	Модерация не одобрила пост	-	Пост создан, виден только автору
Пост создан, виден только автору	Переход в редактирование поста	-	Пост создан, виден только автору
Пост создан, виден только автору	Сохранить изменения	-	Пост создан, виден только автору
Пост создан, виден только автору	Удалить пост	-	Пост удален
Статус «На модерации»	Опубликовать пост	-	Статус «На модерации»
Статус «На модерации»	Пост одобрен	-	Пост виден всем пользователям (статус «Активен»)
Статус «На модерации»	Модерация не одобрила пост	-	Пост в статусе «Требуется исправления»
Статус «На модерации»	Переход в редактирование поста		Пост в статусе «Редактирование»
Статус «На модерации»	Сохранить изменения	-	Статус «На модерации»
Статус «На модерации»	Удалить пост	-	Пост удален
Пост виден всем пользователям (статус «Активен»)	Опубликовать пост	-	Пост виден всем пользователям (статус «Активен»)
Пост виден всем пользователям (статус «Активен»)	Пост одобрен	-	Пост виден всем пользователям (статус «Активен»)
Пост виден всем пользователям (статус «Активен»)	Модерация не одобрила пост	-	Пост в статусе «Требуется исправления»
Пост виден всем пользователям (статус «Активен»)	Переход в редактирование поста	-	Пост в статусе «Редактирование»



Пост виден всем пользователям (статус «Активен»)	Сохранить изменения	-	Пост виден всем пользователям (статус «Активен»)
Пост виден всем пользователям (статус «Активен»)	Удалить пост	-	Пост удален
Пост в статусе «Требуется исправления»	Опубликовать пост	-	Пост в статусе «Требуется исправления»
Пост в статусе «Требуется исправления»	Пост одобрен	-	Пост в статусе «Требуется исправления»
Пост в статусе «Требуется исправления»	Модерация не одобрила пост	-	Пост в статусе «Требуется исправления»
Пост в статусе «Требуется исправления»	Переход в редактирование поста	-	Пост в статусе «Редактирование»
Пост в статусе «Требуется исправления»	Сохранить изменения	-	Пост в статусе «Требуется исправления»
Пост в статусе «Требуется исправления»	Удалить пост	-	Пост удален
Пост в статусе «Редактирование»	Опубликовать пост	-	Пост в статусе «Редактирование»
Пост в статусе «Редактирование»	Пост одобрен	-	Пост в статусе «Редактирование»
Пост в статусе «Редактирование»	Модерация не одобрила пост	-	Пост в статусе «Редактирование»
Пост в статусе «Редактирование»	Переход в редактирование поста	-	Пост в статусе «Редактирование»
Пост в статусе «Редактирование»	Сохранить изменения	Пост отправлен на модерацию	Статус «На модерации»
Пост в статусе «Редактирование»	Удалить пост	-	Пост удален
Пост удален	Опубликовать пост	-	Пост удален
Пост удален	Пост одобрен	-	Пост удален



Пост удален	Модерация не одобрила пост	-	Пост удален
Пост удален	Переход в редактирование поста	-	Пост удален
Пост удален	Сохранить изменения	-	Пост удален
Пост удален	Удалить пост	-	Пост удален

Теперь составим тесты по таблице. Все переходы с изменением состояния, которые отражены на диаграмме, — положительные тесты, их выделим зелёным.

Текущее состояние	Событие	Действие	Новое состояние
null	Опубликовать пост	-	Пост создан, виден только автору
null	Пост одобрен	-	null
null	Модерация не одобрила пост	-	null
null	Переход в редактирование поста	-	null
null	Сохранить изменения	-	null
null	Удалить пост	-	null
Пост создан, виден только автору	Опубликовать пост	-	Пост создан, виден только автору
Пост создан, виден только автору	-	Пост отправлен на модерацию	Статус «На модерации»
Пост создан, виден только автору	Пост одобрен	-	Пост создан, виден только автору
Пост создан, виден только автору	Модерация не одобрила пост	-	Пост создан, виден только автору
Пост создан, виден только автору	Переход в редактирование поста	-	Пост создан, виден только автору
Пост создан, виден только автору	Сохранить изменения	-	Пост создан, виден только автору
Пост создан, виден только автору	Удалить пост	-	Пост удален
Статус «На модерации»	Опубликовать пост	-	Статус «На модерации»



Статус «На модерации»	Пост одобрен	-	Пост виден всем пользователям (статус «Активен»)
Статус «На модерации»	Модерация не одобрила пост	-	Пост в статусе «Требуется исправления»
Статус «На модерации»	Переход в редактирование поста		Пост в статусе «Редактирование»
Статус «На модерации»	Сохранить изменения	-	Статус «На модерации»
Статус «На модерации»	Удалить пост	-	Пост удален
Пост виден всем пользователям (статус «Активен»)	Опубликовать пост	-	Пост виден всем пользователям (статус «Активен»)
Пост виден всем пользователям (статус «Активен»)	Пост одобрен	-	Пост виден всем пользователям (статус «Активен»)
Пост виден всем пользователям (статус «Активен»)	Модерация не одобрила пост	-	Пост в статусе «Требуется исправления»
Пост виден всем пользователям (статус «Активен»)	Переход в редактирование поста	-	Пост в статусе «Редактирование»
Пост виден всем пользователям (статус «Активен»)	Сохранить изменения	-	Пост виден всем пользователям (статус «Активен»)
Пост виден всем пользователям (статус «Активен»)	Удалить пост	-	Пост удален
Пост в статусе «Требуется исправления»	Опубликовать пост	-	Пост в статусе «Требуется исправления»
Пост в статусе «Требуется исправления»	Пост одобрен	-	Пост виден всем пользователям (статус «Активен»)
Пост в статусе «Требуется исправления»	Модерация не одобрила пост	-	Пост в статусе «Требуется исправления»



Пост в статусе «Требуется исправления»	Переход в редактирование поста	-	Пост в статусе «Редактирование»
Пост в статусе «Требуется исправления»	Сохранить изменения	-	Пост в статусе «Требуется исправления»
Пост в статусе «Требуется исправления»	Удалить пост	-	Пост удален
Пост в статусе «Редактирование»	Опубликовать пост	-	Пост в статусе «Редактирование»
Пост в статусе «Редактирование»	Пост одобрен	-	Пост в статусе «Редактирование»
Пост в статусе «Редактирование»	Модерация не одобрила пост	-	Пост в статусе «Редактирование»
Пост в статусе «Редактирование»	Переход в редактирование поста	-	Пост в статусе «Редактирование»
Пост в статусе «Редактирование»	Сохранить изменения	Пост отправлен на модерацию	Статус «На модерации»
Пост в статусе «Редактирование»	Удалить пост	-	Пост удален
Пост удален	Опубликовать пост	-	Пост удален
Пост удален	Пост одобрен	-	Пост удален
Пост удален	Модерация не одобрила пост	-	Пост удален
Пост удален	Переход в редактирование поста	-	Пост удален
Пост удален	Сохранить изменения	-	Пост удален
Пост удален	Удалить пост	-	Пост удален

Контрольные вопросы

1. В чем заключается суть техники диаграмм переходов состояний?
2. Для чего нужна таблица переходов состояний?
3. Из каких столбцов состоит таблица переходов состояний?



Глоссарий

Диаграмма переходов состояний — техника тест-дизайна, основанная на методе «чёрного ящика». Демонстрирует поведение системы при получении управляющих воздействий извне.

Таблица переходов состояний - см. диаграмма переходов состояний.

Дополнительные материалы

1. Lee Copeland «A Practitioner's Guide to Software Test Design»
2. [ISTQB – What is State Transition Testing in Software Testing](#)
3. [What is State Transition Testing? Diagram, Technique, Example](#)
4. [State Transition Testing Technique And State Transition Diagram With Examples](#)
5. [State Transition Testing](#)