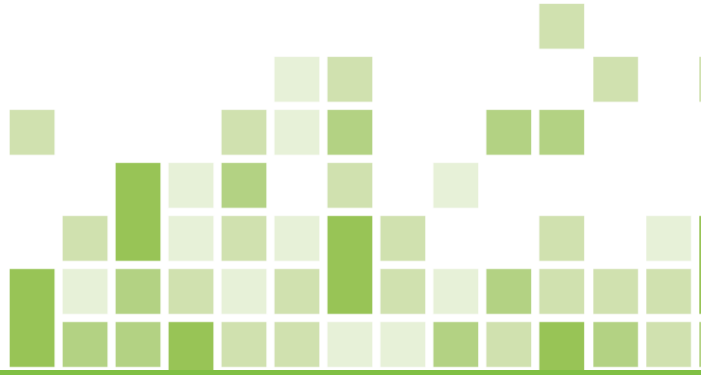


Тестирование программного обеспечения

Лекция №6.

Наборы тестовых сценариев

Копнов Максим Валериевич

Томск
2021

Вспоминаем прошлое занятие

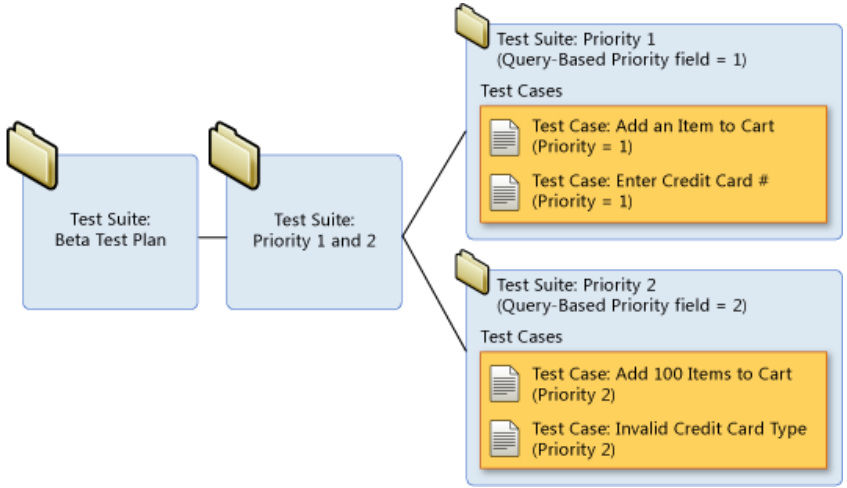
- **Тестовые сценарии**
  - Определение
  - Цели
  - Атрибуты
  - Жизненный цикл
  - Свойства качественных сценариев
  - Правила написания
  - Преимущества и недостатки

О чём сегодня будем говорить

- **Наборы тестовых сценариев**
  - Определение
  - Классификация наборов тестовых сценариев
  - Принципы построения наборов тестовых сценариев
- **Дефекты**
  - Определение
  - Отчёт о дефектах
  - Отчёт о дефектах – цели
  - Отчёт о дефектах – жизненный цикл

Наборы тестовых сценариев

Набор тест-кейсов — совокупность тест-кейсов, выбранных с некоторой общей целью или по некоторому общему признаку.



Свободные наборы — сценарии выполняются в произвольном порядке

Последовательные наборы — сценарии выполняются в установленном порядке

Классификация наборов тестовых сценариев

Преимущества свободных наборов:

- Тест-кейсы можно выполнять в любом удобном порядке, а также создавать «наборы внутри наборов».
- Если какой-то тест-кейс завершился ошибкой, это не повлияет на возможность выполнения других тест-кейсов.

Преимущества последовательных наборов:

- Каждый следующий в наборе тест-кейс в качестве входного состояния приложения получает результат работы предыдущего тест-кейса, что позволяет сильно сократить количество шагов в отдельных тест-кейсах.
- Длинные последовательности действий лучше имитируют работу реальных пользователей, чем отдельные «точечные» воздействия на приложение.

Пример последовательного набора

Сценарий «Распечатать табличку»

- Запустить текстовый редактор.
- Создать новый документ.
- Набрать в документе текст.
- Отформатировать текст должным образом.
- Отправить документ на печать.
- Сохранить документ.
- Закрыть текстовый редактор.

Преимущества последовательного набора

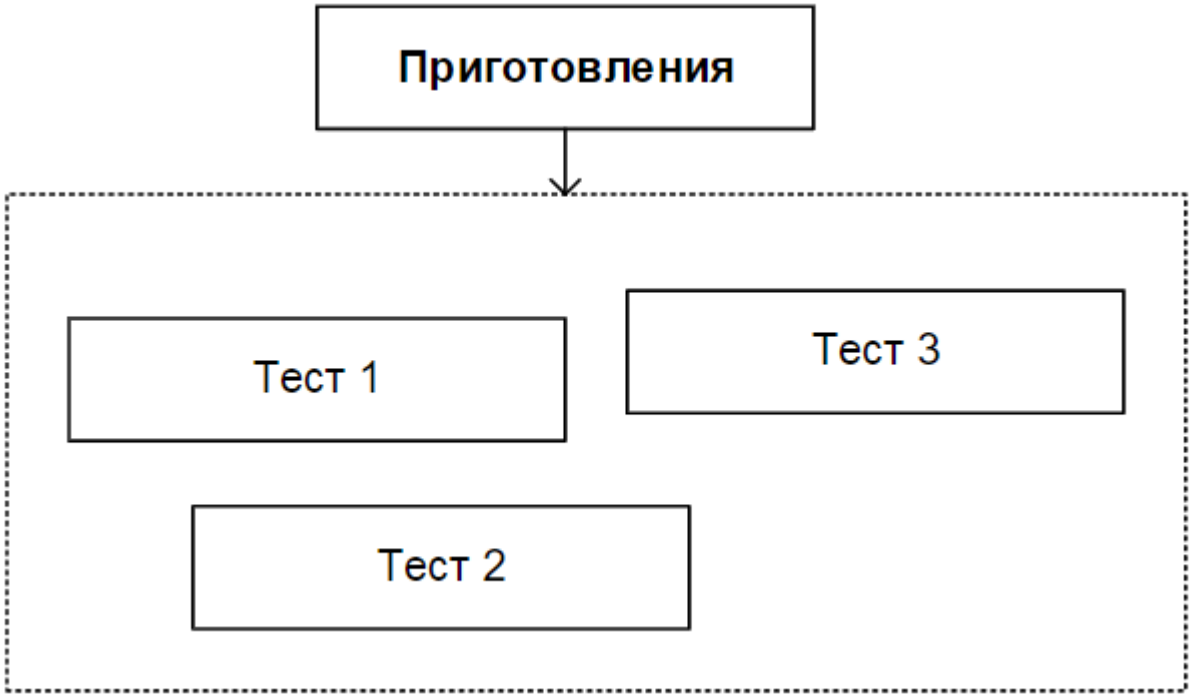
- Сценарии показывают реальные и понятные примеры использования продукта.
- Сценарии понятны конечным пользователям и хорошо подходят для обсуждения и совместного улучшения.
- Сценарии и их части легче оценивать с точки зрения важности, чем отдельные пункты требований.
- Сценарии показывают недоработки в требованиях.
- При нехватке времени или прочих форс-мажорах сценарии можно даже не прописывать подробно, а просто именовать.

Классификация наборов тестовых сценариев

- Набор изолированных свободных тест-кейсов: действия из раздела «приготовления» нужно повторить перед каждым тест-кейсом, а сами тест-кейсы можно выполнять в любом порядке.

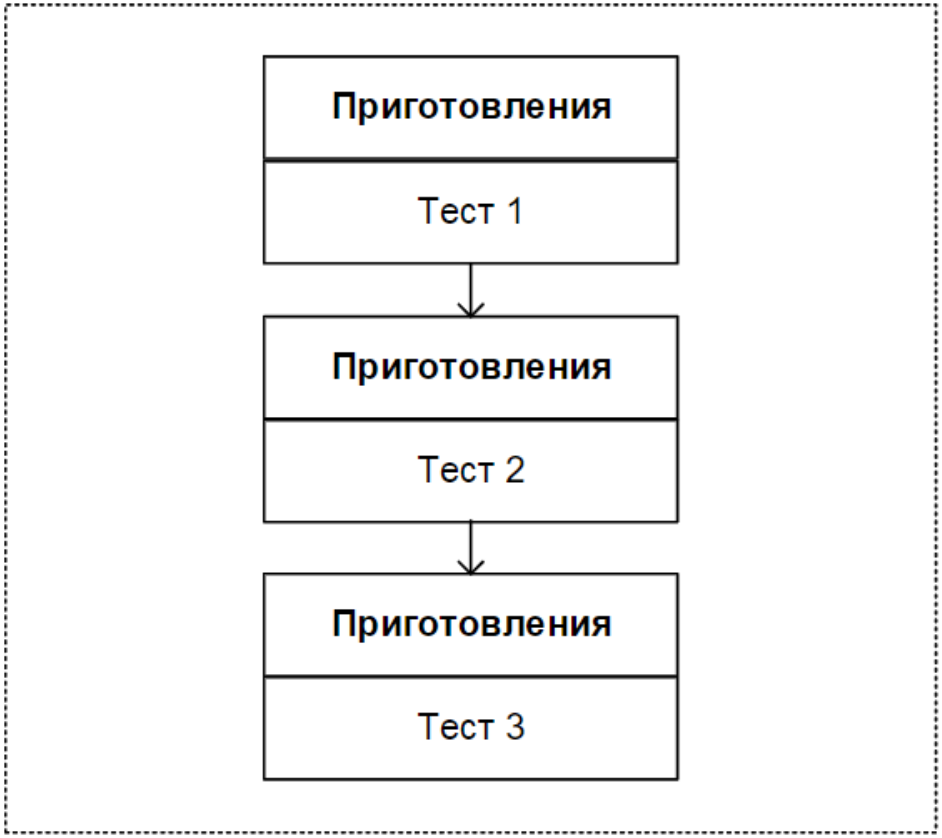
Классификация наборов тестовых сценариев

- Набор обобщённых свободных тест-кейсов: действия из раздела «приготовления» нужно выполнить один раз (а потом просто выполнять тест-кейсы), а сами тест-кейсы можно выполнять в любом порядке.



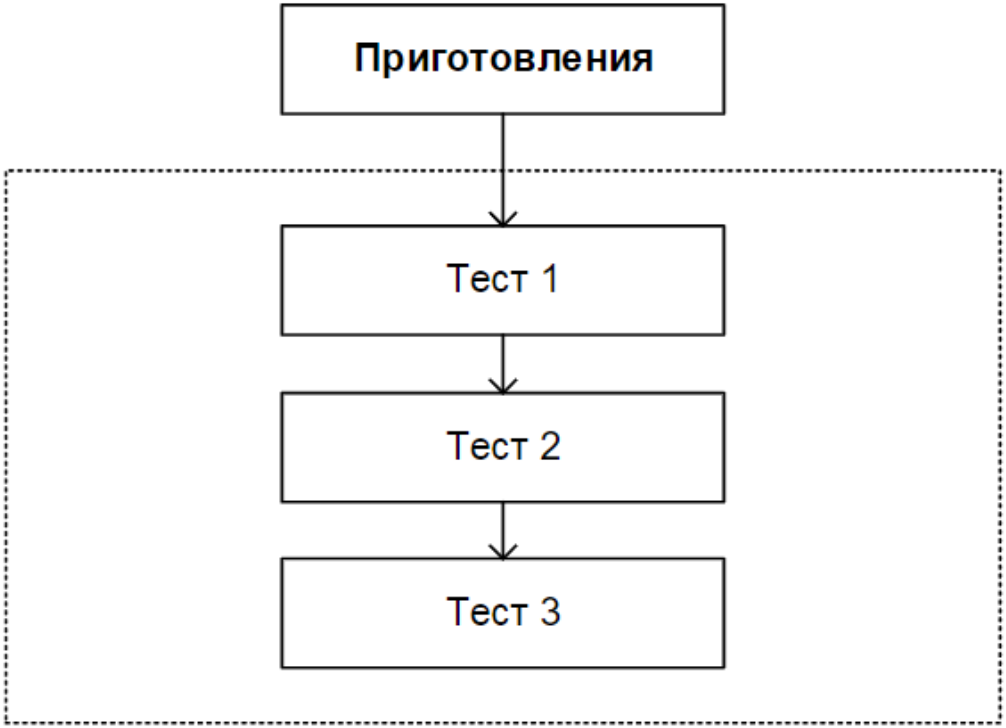
Классификация наборов тестовых сценариев

- Набор изолированных последовательных тест-кейсов: действия из раздела «приготовления» нужно повторить перед каждым тест-кейсом, а сами тест-кейсы нужно выполнять в строго определённом порядке.



Классификация наборов тестовых сценариев

- Набор обобщённых последовательных тест-кейсов: действия из раздела «приготовления» нужно выполнить один раз (а потом просто выполнять тест-кейсы), а сами тест-кейсы нужно выполнять в строго определённом порядке.



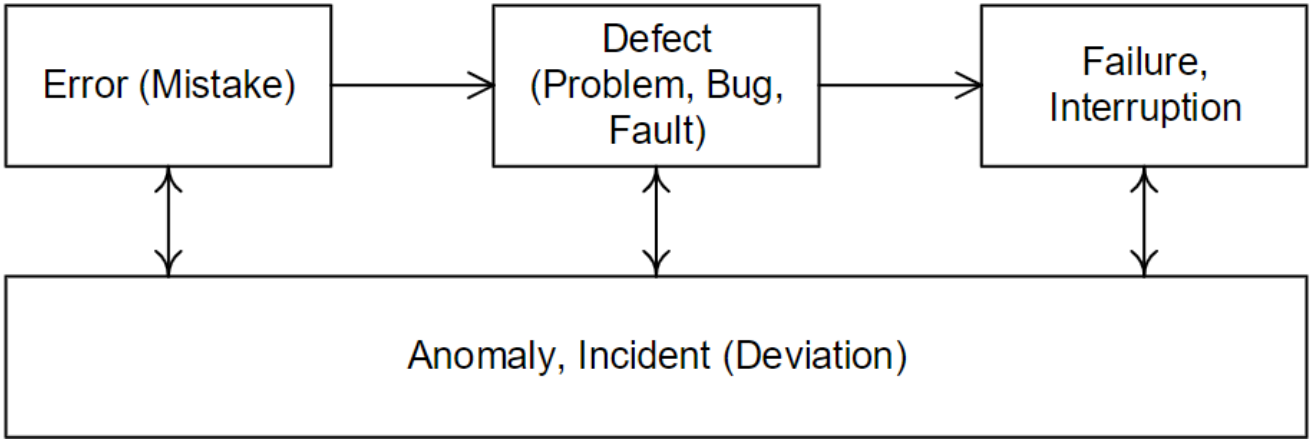
Принципы построения наборов тестовых сценариев

- На основе чек-листов. Каждый пункт чек-листа может превратиться в несколько тест-кейсов.
- На основе разбиения приложения на модули и подмодули.
- По принципу проверки самых важных, менее важных и всех остальных функций приложения.
- По принципу группировки тест-кейсов для проверки некоего уровня требований или типа требований, группы требований или отдельного требования.

Принципы построения наборов тестовых сценариев

- По принципу частоты обнаружения тест-кейсами дефектов в приложении.
- По архитектурному принципу:
  - наборы для проверки пользовательского интерфейса и всего уровня представления,
  - для проверки уровня бизнес-логики,
  - для проверки уровня данных.
- По области внутренней работы приложения:
  - «тест-кейсы, затрагивающие работу с базой данных»
  - «тест-кейсы, затрагивающие работу с файловой системой»
  - «тест-кейсы, затрагивающие работу с сетью»
  - и т.д.
- По видам тестирования.

Дефекты



- **Ошибка (error, mistake)** — действие человека, приводящее к некорректным результатам.
- **Дефект (defect, bug, problem, fault)** — недостаток в компоненте или системе, способный привести к ситуации сбоя или отказа.
- **Сбой (interruption)** или **отказ (failure)** — отклонение поведения системы от ожидаемого.
- **Аномалия (anomaly)** или **инцидент (incident, deviation)** — любое отклонение наблюдаемого (фактического) состояния, поведения, значения, результата, свойства от ожиданий наблюдателя, сформированных на основе требований, спецификаций, иной документации или опыта и здравого смысла.

Отчёт о дефектах

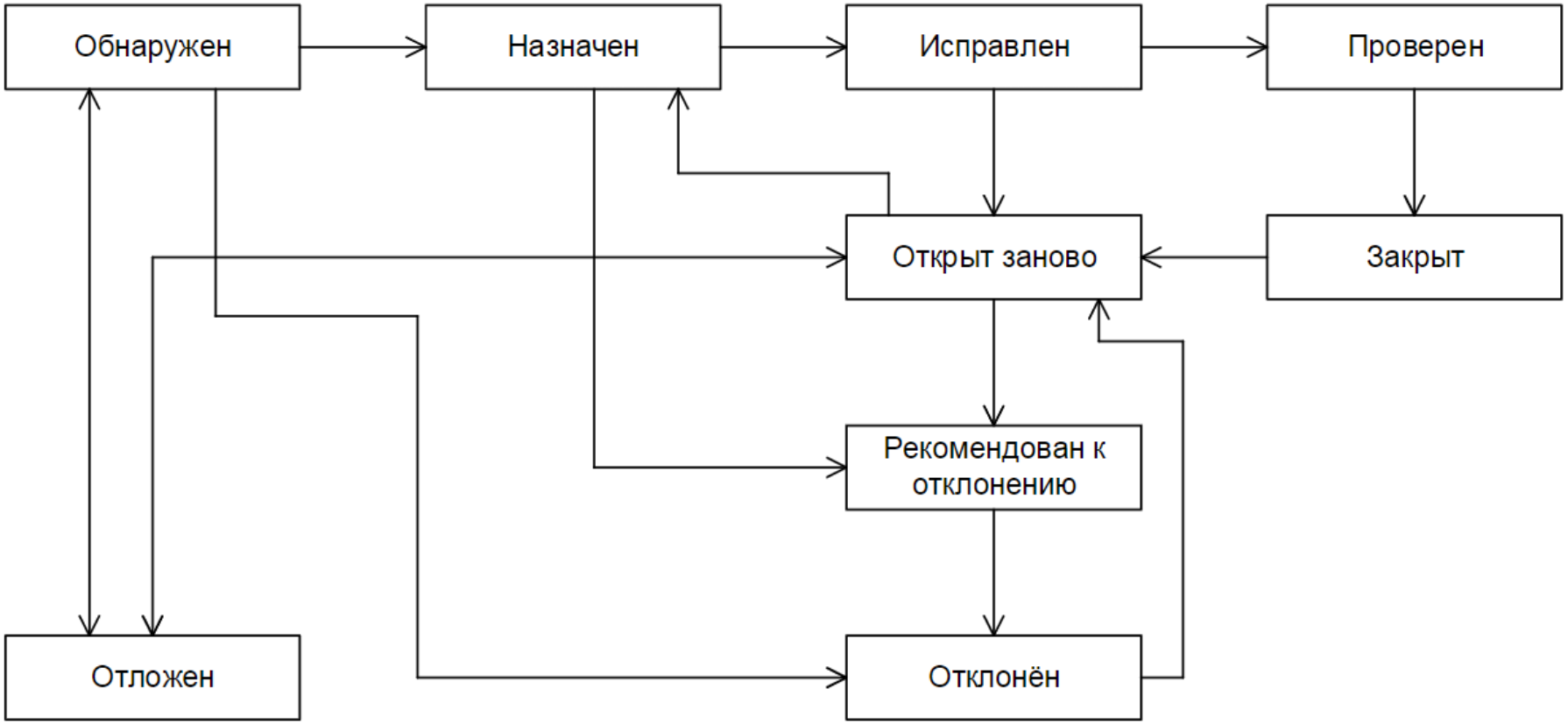
- **Дефект** — отклонение фактического результата от ожиданий наблюдателя, сформированных на основе требований, спецификаций, иной документации или опыта и здравого смысла.
- **Отчёт о дефекте** — документ, описывающий и приоритизирующий обнаруженный дефект, а также содействующий его устранению.

Отчёт о дефектах

Отчёт о дефектах – цели

- предоставить информацию о проблеме — уведомить проектную команду и иных заинтересованных лиц о наличии проблемы, описать суть проблемы;
- приоритизировать проблему — определить степень опасности проблемы для проекта и желаемые сроки её устранения;
- содействовать устранению проблемы — качественный отчёт о дефекте не только предоставляет все необходимые подробности для понимания сути случившегося, но также может содержать анализ причин возникновения проблемы и рекомендации по исправлению ситуации.

Отчёт о дефектах – жизненный цикл



Отчёт о дефектах – жизненный цикл

- Обнаружен — начальное состояние отчёта, в котором он находится сразу после создания.
- Назначен — в это состояние отчёт переходит с момента, когда кто-то из проектной команды назначается ответственным за исправление дефекта.
- Исправлен — в это состояние отчёт переводит ответственный за исправление дефекта член команды после выполнения соответствующих действий по исправлению.
- Проверен — в это состояние отчёт переводит тестировщик, удостоверившийся, что дефект на самом деле был устранён.

Отчёт о дефектах – жизненный цикл

- Обнаружен — начальное состояние отчёта, в котором он находится сразу после создания.
- Назначен — в это состояние отчёт переходит с момента, когда кто-то из проектной команды назначается ответственным за исправление дефекта.
- Исправлен — в это состояние отчёт переводит ответственный за исправление дефекта член команды после выполнения соответствующих действий по исправлению.
- Проверен — в это состояние отчёт переводит тестировщик, удостоверившийся, что дефект на самом деле был устранён.

Отчёт о дефектах – жизненный цикл

- Закрыт — состояние отчёта, означающее, что по данному дефекту не планируется никаких дальнейших действий.
- Открыт заново —дефект по-прежнему воспроизводится на билде, в котором он уже должен быть исправлен.
- Рекомендован к отклонению — в это состояние отчёт о дефекте может быть переведён с целью вынести на рассмотрение вопрос об отклонении отчёта по той или иной причине.

Отчёт о дефектах – жизненный цикл

- Отклонён — переводится в случаях:
  - «Не является дефектом» — приложение так и должно работать, описанное поведение не является аномальным.
  - «Дубликат» — данный дефект уже описан в другом отчёте.
  - «Не удалось воспроизвести» — разработчикам не удалось воспроизвести проблему на своём оборудовании.
  - «Не будет исправлено» — дефект есть, но по каким-то серьёзным причинам его решено не исправлять.
  - «Невозможно исправить» — непреодолимая причина дефекта находится вне области полномочий команды разработчиков.
- Отложен — в это состояние отчёт переводится в случае, если исправление дефекта в ближайшее время является нерациональным или не представляется возможным.

ТОМСКИЙ
ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

Тестирование программного обеспечения

Копнов Максим Валериевич
kopnovmv@tpu.ru