

Принципы разработки тестов



Какие бывают тесты?

Задача о треугольнике

Программа производит чтение трех целых чисел, которые интерпретируются как длины сторон треугольника.

Далее программа печатает сообщение о том, является ли треугольник неравносторонним, равнобедренным или равносторонним.

a

b

c

Проверить



Какие бывают тесты.

Задача о треугольнике

Техники тест дизайна

(метод черного ящика)

Наиболее распространенные техники тест дизайна:

- Эквивалентное разделение (*Equivalence Partitioning - EP*).
- Анализ граничных значений (*Boundary Value Analysis - BVA*).
- Предугадывание ошибки (*Error Guessing - EG*).
- Исчерпывающее тестирование (*Exhaustive Testing - ET*)

Классы эквивалентности

Обычно для системы (за исключением самых простых приложений) невозможно протестировать все логически возможные входные комбинации. Поэтому одна из важнейших задач для разработчиков - выбрать для тестирования те комбинации, где с наибольшей вероятностью можно найти большинство ошибок.

Тестирование, основанное на анализе классов эквивалентности (синонимы: разбиение на области эквивалентности, анализ области) - это анализ тестирования методом "черного ящика".

Цель анализа - сократить до минимума общее число циклов тестирования и обнаружить при этом максимальное возможное количество ошибок.

Этот метод позволяет разделить ряд входных и выходных данных на конечное число классов эквивалентности и выбрать характерное тестовое значение для каждого класса.

Результаты тестирования характерного значения класса считаются "эквивалентными" другим значениям того же класса. Если при тестировании характерного значения ошибки не были найдены, предполагается, что все другие "эквивалентные" значения также не будут содержать ошибки.

Классы эквивалентности. Пример

Есть простой скрипт или программа, которая предлагает ввести возраст человека в текстовое поле вместимостью в 2 символа и что ниже находится кнопка для отправки данных.

В результате ввода значения программа должна рассчитать стоимость мед страховки в зависимости от той цифры, которую мы введем. В соответствии с функциональной спецификацией все возраста разделили на четыре категории.

Первая	это <u>Дети</u>	(включает от 0 до 12 лет),
Вторая категория	это <u>Подростки</u>	(включает от 13 до 17 лет),
Третья категория	это <u>Взрослые</u>	(включает от 18 до 59 лет),
Четвертая категория	это <u>Пенсионеры</u>	(включает от 60 до 99 лет).

В зависимости от категории к которой относится человек, будет вычисляться стоимость мед страховки: Дети — 10 у.е. в месяц, Подростки — 20 у.е. в месяц, Взрослые — 30 у.е. в месяц, Пенсионеры — 40 у.е. в месяц

В данном случае все входные данные мы можем разделить на 4 эквивалентных класса.

Вопрос: *Сколько же тест кейсов нужно для проверки всех эквивалентных классов и какие именно?*

Для проверки всех четырех эквивалентных классов нам понадобится провести тесты с одним представителем из каждого Эквивалентного класса.

Покрыть все эквивалентные классы можно всего четырьмя тестами.

В первом случае берем любое число от 0 до 12 (к примеру 5)

Во втором случае берем любое число от 13 до 17 (к примеру 15)

В третьем случае берем любое число от 18 до 59 (к примеру 36)

В четвертом случае берем любое число от 60 до 99 (к примеру 78)

Классы эквивалентности

Класс эквивалентности (*equivalence class*)
– набор данных, обрабатываемый
одинаковым образом и
приводящий к одинаковому
результату.

Признаки эквивалентности

Несколько тестов эквивалентны, если:

- Они направлены на поиск одной и той же ошибки
- Если один из тестов обнаруживает ошибку, другие её тоже, скорее всего, обнаружат
- Если один из тестов **НЕ обнаруживает** ошибку, другие её тоже, скорее всего, **НЕ обнаружат**
- Тесты используют одни и те же наборы входных данных
- Для выполнения тестов мы совершаем одни и те же операции
- Тесты генерируют одинаковые выходные данные или приводят приложение в одно и то же состояние (например открытие валидных файлов одного типа и схожего объема, но с разным содержанием)
- Все тесты приводят к срабатыванию одного и того же блока обработки ошибок (*«error handling block»*)
- Ни один из тестов не приводит к срабатыванию определенного блока обработки ошибок (*«error handling block»*) (например, блок обработки ошибок открытия файла)

Граничные условия

- Граничные условия (или просто – границы) – это те места, в которых один класс эквивалентности переходит в другой.

Например, одна группа тестов вызывает сообщение «вы ввели слишком маленькое число», а другая вызывает сообщение «вы ввели слишком большое число». Граница будет лежать где-то в районе чисел «самых больших из слишком маленьких» и «самых маленьких из слишком больших».

! Граничные условия очень важны, и их обязательно следует проверять в тестах, т.к. именно в этом месте чаще всего и обнаруживаются ошибки.

Граничные условия

В нашем примере программа позволяет вводить максимум 2 символа и текстовое поле принимает только числа от 0 до 99. Поэтому граничные значения для этого поля -1,0 и 99, 100

Но нам нужно также проверить и граничные значения эквивалентных классов, для большей уверенности в качестве нашей программы.

Какие же *граничные значения* у эквивалентных классов в рассматриваемом примере?

В данном случае это 0, 12, 13, 17, 18, 59, 60, 99.

Следовательно чтобы покрыть эквивалентные классы и граничные значения для примера необходимо протестировать программу с такими входными данными:

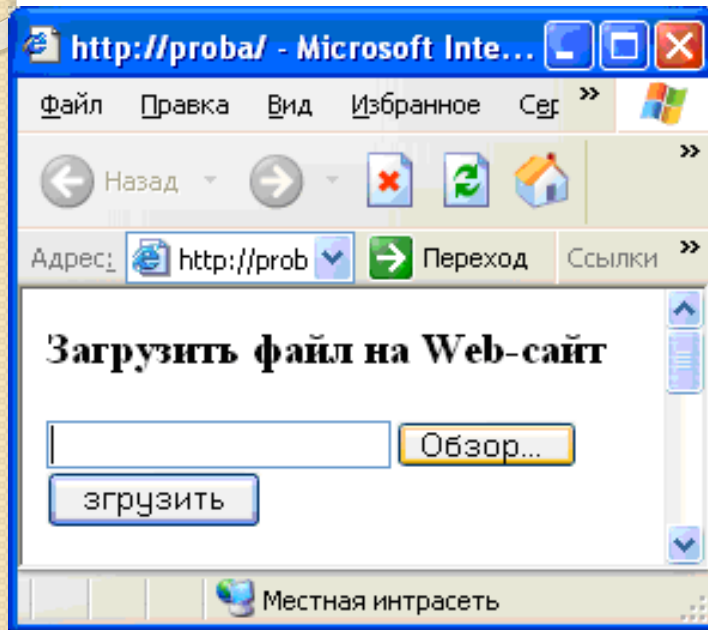
-1, 0, 5, 12, 13, 15, 17, 18, 36, 59, 60, 78, 99, 100

Негативные тесты:

- Ввести числа -1, 100
- Ввести нецелое число 5,1
- Ввести букву d, F,
- Ввести спецсимвол: ~`!"@'#\$;%=-_.....

Пример для обсуждения

Пример:



Открытие файла. Чтобы добавить файл в свою фотогалерею на сайте, пользователь должен кликнуть по кнопке «Открыть», выбрать файл и кликнуть по кнопке ОК.

Вопрос: *Какие случаи нам надо будет проверить?*

Выводы

- Классы эквивалентности не всегда очевидны.
- Как правило, негативных тестов получается больше, чем позитивных.
- Принадлежность теста к позитивным или негативным зависит от требований.

Рекомендации по написанию тестов

- Начинайте с простых очевидных тестов.
- Затем переходите к более сложным тестам.
- Помните о граничных условиях.
 - Если остаётся время, занимайтесь исследовательским тестированием.

Чек-лист

Check list — контрольный список

- Упрощенная форма тест-кейса
- Главный принцип чек-листов заключается в том, что каждый тестировщик по-своему проходит их, расширяя тестовый набор своей экспертизой

Проверка	Результат	Комментарии
Операции с файлами	ok	
Создание файла	ok	
Открытие файла	ok	
Сохранение документа	ok	
Печать	ok	
Редактирование файлов	bugs	
Отмена	ok	
Копирование	ok	
Вырезание	ok	
Вставка	ok	
Удаление	ok	
Поиск	fail	bug #123
Поиск с заменой	fail	bug #126
Вставка даты	ok	
Форматирование	ok	
Перенос строки	ok	
Изменение шрифта	ok	
Справка	ok	

Чек-лист

Преимущества использования чек листов

- ❖ Использование чек-листов способствует структурированию информации у сотрудника.
- ❖ При правильной записи необходимых действий у сотрудника появляется однозначное понимание задач. Это способствует повышению скорости обучения новых сотрудников.
- ❖ Чек-листы помогают избежать неопределенности и ошибок связанных с человеческим фактором. Увеличивается покрытие тестами программного продукта.
- ❖ Повышается степень взаимозаменяемости сотрудников.
- ❖ Экономия рабочего времени. Написав чек-лист единожды его можно использовать повторно, учитывая актуальность информации.

Чек-лист

Правила написания

Чек-листы устроены предельно просто. Любой из них содержит перечень блоков, секций, страниц, других элементов, которые следует протестировать, например:

Сайт "example.edu"
Регистрация и Личный профиль
Регистрация на сайте
Редактирование профиля
Форма обратной связи
Валидация полей
Отправка письма/уведомления
Доставка письма/уведомления
Поиск
Поиск по названиям
Переход по ссылкам
Работа поиска по разнообразным параметрам
Комментарии
Добавление комментария
Отображение комментария

Чек-лист

Правила написания

- ✓ Пункты могут содержать как линейную структуру, так и древовидную, с разделами/подразделами или без них.
- ✓ Они должны быть максимально краткие и в то же время понятные тестировщику, который еще не знаком с приложением.
- ✓ Пункты должны быть однозначными, чтобы их нельзя было понять как-то иначе.
- ✓ Все пункты должны быть оформлены на одном языке: русском или английском.

Чек-лист

Правила написания

- ✓ Как правило, каждый чек-лист имеет несколько столбцов. Каждый столбец предназначен для тестирования на отдельной платформе. Следует всегда указывать название устройства, браузера и его версии.
- ✓ Тестировать приложение может несколько человек одновременно. В этом случае каждый тестировщик берет по одной-две платформы и тестирует только на них. При этом следует напротив каждой платформы указать тестировщика, который выполняет указанный объем работ.

Иванов А. А.	Петров Б. Б.	Фёдоров В.В.	Васечкин Г. Г.
Google Chrome 64.0.3282.140	Mozilla Firefox 58.0.2	IE 11.0	Opera 51.0.2830.34

Чек-лист

Правила написания

- ✓ При прохождении чек-листов тестировщик отмечает статус напротив каждого тестируемого пункта. Возможны следующие варианты статусов:
 - «**Passed**» – проверка пройдена успешно, багов не найдено;
 - «**Failed**» – найден один или более багов;
 - «**Blocked**» – невозможно проверить, т.к. один из багов блокирует текущую проверку;
 - «**In Progress**» – текущий пункт, над которым работает тестировщик;
 - «**Not run**» – еще не проверено;
 - «**Skipped**» – проверяться не будет по какой-либо причине. Например, текущий функционал еще не реализован.

Чек-лист

Правила написания

Для большей наглядности, как правило, каждый из статусов имеет свой цвет.

После завершения тестирования не должно быть ячеек, отмеченных как «**Not run**».

Все заведенные по чек-листу баг-репорты должны быть добавлены в примечания к ячейке со статусом «**Failed**».

Passed
Failed
Blocked
Not run
Skipped

	Иванов А. А.	Петров Б. Б.	Фёдоров В.В.	Васечкин Г. Г.
Сайт "example.edu"	Google Chrome 64.0.3282.140	Mozilla Firefox 58.0.2	IE 11.0	Opera 51.0.2830.34
Регистрация и Личный профиль				
Регистрация на сайте	Passed	Passed	Passed	Passed
Редактирование профиля	Failed +	http://bt-w.qatestlab.com/view.php?id=...		
Форма обратной связи				
Валидация полей	Failed	Failed	Passed	Passed
Отправка письма/уведомления	Blocked	Blocked	Passed	Passed
Доставка письма/уведомления	Blocked	Blocked	Skipped	Skipped

Чек-лист

Правила написания

Для большей наглядности, как правило, каждый из статусов имеет свой цвет.

После завершения тестирования не должно быть ячеек, отмеченных как «**Not run**».

Все заведенные по чек-листу баг-репорты должны быть добавлены в примечания к ячейке со статусом «**Failed**».

Passed
Failed
Blocked
Not run
Skipped

	Иванов А. А.	Петров Б. Б.	Фёдоров В.В.	Васечкин Г. Г.
Сайт "example.edu"	Google Chrome 64.0.3282.140	Mozilla Firefox 58.0.2	IE 11.0	Opera 51.0.2830.34
Регистрация и Личный профиль				
Регистрация на сайте	Passed	Passed	Passed	Passed
Редактирование профиля	Failed +	http://bt-w.qatestlab.com/view.php?id=...		
Форма обратной связи				
Валидация полей	Failed	Failed	Passed	Passed
Отправка письма/уведомления	Blocked	Blocked	Passed	Passed
Доставка письма/уведомления	Blocked	Blocked	Skipped	Skipped

Чек-лист

Правила составления чек листов

1. Один пункт — одна операция

Пункты чек-листа – это однозначные атомарные и полные операции. Например, добавление товара в корзину сайта и оплата заказа – две разные задачи. В списке проверок подобные операции оформлены отдельными пунктами: добавлен товар в корзину, оплата отправлена.

2. Пункты начинаются с существительного.

Цель чек-листа – учесть все действия для наиболее полного покрытия тестами ПО, поэтому составляя пункты следует придерживаться унифицированной формы. Для понятного и однозначного представления пункты лучше начинать с существительного – «Проверка», «Добавление», «Отправка» или глагола неопределенной формы – «Проверить», «Добавить», «Отправить».”

3. Составление чек-листа по уровням детализации.

Для удобства прохождения чек-листа лучше всего составлять тесты в том виде, который будет последовательным исходя из логики использования функционала. В рамках раздела «Регистрация и Личный профиль»: регистрация на сайте, редактирование профиля. Раздел «Форма обратной связи»: валидация полей, отправка письма, доставка письма.

Чек-лист

- Разбивайте приложение на модули (модуль авторизации, модуль настроек и т.д.)
- Используйте «косметику»
- Используйте техники ускорения написания (copy-paste)

Чек-лист

	Иванов А. А.	Петров Б. Б.	Фёдоров В.В.	Васечкин Г. Г.
Сайт "example.edu "	Google Chrome 64.0.3282.140	Mozilla Firefox 58.0.2	IE 11.0	Opera 51.0.2830.34
Регистрация и Личный профиль				
Регистрация на сайте	Passed	Passed	Passed	Passed
Редактирование профиля	Failed	Failed	Failed	Failed
Форма обратной связи				
Валидация полей	Failed	Failed	Passed	Passed
Отправка письма/уведомления	Blocked	Blocked	Passed	Passed
Доставка письма/уведомления	Blocked	Blocked	Skipped	Skipped
Поиск				
Поиск по названиям	Passed	Passed	Passed	Passed
Переход по ссылкам	Passed	Passed	Passed	Passed

Чек-лист нужен чтобы:

1. Не забыть требуемые тесты.
2. Делить задачи по уровню квалификации.
3. Сохранять отчетности и результаты тестирования.

Чек-лист содержит:

1.Список проверок (с требуемой степенью детализации).

2.Окружение проверки:

- ✓ сборка, на которой проводилось тестирование;
- ✓ тестовое окружение (если применимо);
- ✓ информация о тестирующем.

3.Результат проверки.

Контрольные вопросы

1. Описать техники тест дизайна?
2. Что такое «Классы эквивалентности»?
3. Что такое «Граничные условия»?
4. Описать артефакт чек-лист.
5. Перечислить преимущества чек-листа.
6. Основные правила составления чек-листа.