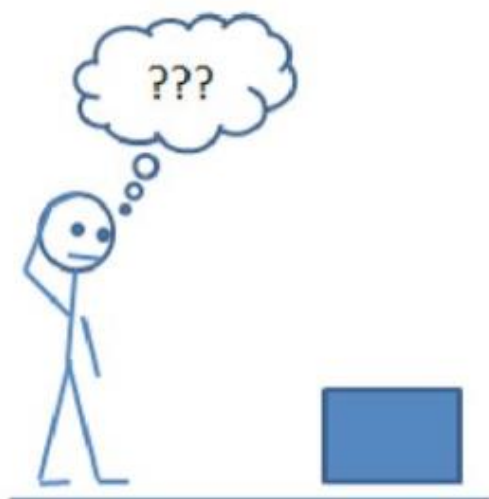

ТЕСТ-ДИЗАЙН. ГРАНИЧНЫЕ ЗНАЧЕНИЯ

Тестирование ПО

Что такое тест дизайн?

Тест-дизайн – это этап процесса тестирования ПО, на котором проектируются и создаются тестовые случаи (тест-кейсы), в соответствии с определёнными ранее критериями качества и целями тестирования.



Гипотезы (что в ящике?):

Тесты (как проверить?):

- Ничего
- Мяч
- Кот

- Поднять и взвесить ящик
- Послушать
- Потрясти ящик
- Открыть ящик

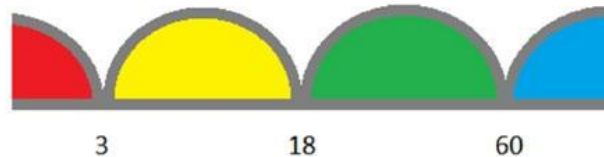


Цели тест-дизайна

- Придумать тесты, которые обнаружат наиболее серьезные ошибки продукта.
- Минимизировать количество тестов, необходимых для нахождения большинства серьезных ошибок.

Техника анализа граничных значений

Техника анализа граничных значений является продолжением метода эквивалентного разбиения, но может быть применима, только если классы состоят из упорядоченных числовых значений. Максимальное и минимальное значение класса являются его границами



Примерный алгоритм использования техники анализа граничных значений:

- Во-первых, нужно выделить классы эквивалентности. Опять же, это очень важный шаг и от правильности разбиения на классы эквивалентности зависит эффективность тестов граничных значений.
- Далее нужно определить граничные значения этих классов.
- Нам нужно понять, к какому классу будет относиться каждая граница.
- Для каждой границы нам нужно провести тесты по проверке значения до границы, на границе, и сразу после границы.

Неправильно

```
If (applicantAge >= 0 && applicantAge <=16)
    hireStatus="NO";
If (applicantAge >= 16 && applicantAge <=18)
    hireStatus="PART";
If (applicantAge >= 18 && applicantAge <=55)
    hireStatus="FULL";
If (applicantAge >= 55 && applicantAge <=99)
    hireStatus="NO";
```

Правильно

```
If (applicantAge >= 0 && applicantAge <=15)
    hireStatus="NO";
If (applicantAge >= 16 && applicantAge <=17)
    hireStatus="PART";
If (applicantAge >= 18 && applicantAge <=54)
    hireStatus="FULL";
If (applicantAge >= 55 && applicantAge <=99)
    hireStatus="NO";
```

Плюсы и минусы техники:

- К плюсам можно отнести то, что она выявляет комбинации условий, которые могли быть не проверены при тестировании. Помогает определить несоответствия в требованиях, может быть применена во всех ситуациях и на любом уровне, где поведение программного обеспечения зависит от комбинации условий.
- К минусам можно отнести то, что, при неправильном использовании техники, мы рискуем потерять баги.



—

**Спасибо за
внимание! :)**