

# О чем поговорим

---



## Когда тестов достаточно?

- Что такое покрытие, зачем нужно, как и в чем измерять?
- Почему 100% покрытие не гарантирует качество
- Почему без хорошего покрытия качество невозможно
- Что делать когда покрытие 100% а баги есть
- Мутационное тестирование

**Покрытие (coverage)** — метрика, применяемая для оценки полноты/качества тестов. Зависит от выбранной техники тестирования и может изменяться, например:

- В процентах покрытой функциональности (например, по спецификации)
- В количестве покрытых строк кода (в процентах от общего количества строк)
- В процентах покрытых ветвей в графе потока управления
- ... операторов, условий, путей, функций, ввод/вывод, значений параметров

100% покрытие  $\neq$  100% тестирования  $\neq$  100% качества

Тестирование:

- Проверить все возможные значения каждой входной переменной
- Проверить все возможные комбинации значений переменных
- Проверить каждую конфигурацию ПО и оборудования, а также конфигурации серверов
- Проверить взаимодействие с другими программами, работающими параллельно с нашей

# Почему нельзя протестировать всё?

---



Проверить все допустимые значения каждой переменной на входе:

Пример: функция, вычисляющая квадратный корень из 32-битного числа без знака.

Сколько значений нужно проверить?

$2^{32}$  значений  $\approx$  6 минут

# Как всё на самом деле

---



В жизни — сначала пишем тесты, затем уже считаем метрики, насколько наши тесты удовлетворяют тому или иному критерию покрытия.



# Покрытие операторов

---



```
1.  if (a > 10)
2.      a++;
3.  if (b == 'foo')
4.      b = 'bar'
```

Тестовый набор?



# Покрытие операторов



```
1.  if (a > 10)
2.      a++;
3.  if (b == 'foo')
4.      b = 'bar'
```

```
a = 11, b = 'foo'
```



# Покрывтие ветвей



```
1.  if (a > 10)
2.      a++;
3.  if (b == 'foo')
4.      b = 'bar'
```

Тестовый набор?





# Покрывтие ветвей



```
1.  if (a > 10)
2.      a++;
3.  if (b == 'foo')
4.      b = 'bar'
```

(a=11, b='foo'), (a=0, b='bad')



# Покрывтие путей



```
1.  if (a > 10)
2.      a++;
3.  if (b == 'foo')
4.      b = 'bar'
```

Тестовый набор?



# Покрывтие путей



```
1.  if (a > 10)
2.      a++;
3.  if (b == 'foo')
4.      b = 'bar'
```

(a=11, b='foo'), (a=0, b='bad'), (a=0, b='foo'), (a=11, b='bad')

# Code coverage. Unit testing

---



100% покрытие инструкций и ветвей — необходимый, но не достаточный критерий полноты тестирования.



# Code coverage. Unit testing

---



1.  $a/b$

Любые значения  $a$  и  $b$  обеспечивают 100% покрытия, но не обнаруживают ошибку Division by Zero