

$\oplus \times 4$

Retro

$\triangle + \times 4$
Action

⊕ Знать то, что информация о тестировании лучше всего сообщать...

⊕ Знать в понимании и статистике результатов тестирования от заказчика

Понимать "критерии" - не тесты

⊕ В целом много интересной и полезной информации
+ отзывы
+ практика
а пр.
но только в будущем

⊕ При разработке стоит внимание не только на тесты, но и на другие аспекты разработки

Возвращаемся в парадигму
Unit, Integration, System

⊕ Знать о том, что такое JUnit, JTest, JMock, JUnit4, JUnit5, JTest, JMock, JUnit4, JUnit5, JTest, JMock

Возврат к абстрактности
сервис, через который бизнес работает

JWT
оформить уверенность, что это правильно применяется в коде

+ Определение Unit и unit tests
2) Сравнение с другими бизнес-процессами с процессами

+ Основание необходимости тестов
4) Тест как документация

⊕ Не забыть момент до разработки, что необходимо разработать с gitHub, docker, Jenkins, CI/CD

⊕ Сделать мало времени на практическое задание

⊕ Assume
1) Как, чтобы не было

Проблема с push от GIT
(много раз было забыто, что нужно было + коммиты)

Хорошо было бы сделать раздаточный материал

⊕
Оформление задания?

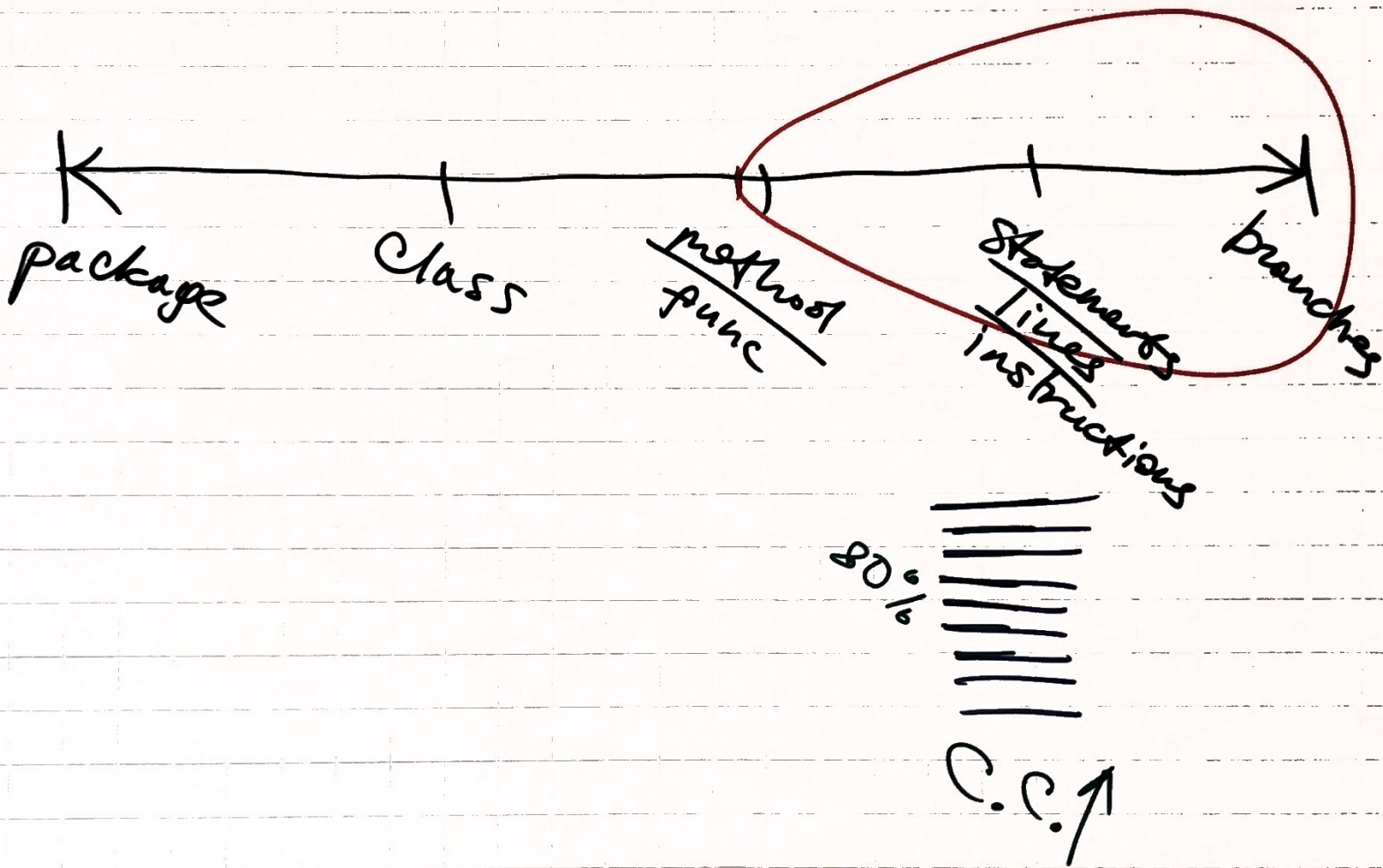
$\oplus \times 4$
TA
Action

⊕ Будущее развитие, повышение важности написания Unit Tests в сочетании с другими научными знаниями

Должно быть про unit tests в курсе по тестированию

TA
1) Наименование тестов (unit)
2) Разделение базиса внутри теста

Test Coverage



Application

