



Занятие 7. Автоматизация тестирования

Гадияк Никита Андреевич
Старший инженер по тестированию ПО

Программа занятия

Автоматизация тестирования:

1. Преследуемые цели
2. Стратегия
3. Авто-тесты и инструменты

Занятие 7. Часть 1: цели

- **Что значит «автоматизация»?**
- Зачем автоматизировать?
- Когда не автоматизировать?
- Что можно автоматизировать?



Test automation is **any use** of tools to **support** testing

Rapid Software Testing

James Bach

Michael Bolton

Это – тоже автоматизация:

- `grep ERROR *.log`
- `diff actual.txt expected.txt`
- Excel – генерация тестовых данных
- NDepend – отбор тестов для выполнения
- BTS – поддержка процесса
- Vulnerability Scanners



Но обычно имеется в виду

автоматизация выполнения тестов, то есть:

1. **автоматизация воздействий**
на тестируемое приложение

+

2. **автоматизация проверок реакций**
на эти воздействия

Почему именно это?

- Это «**большая**» автоматизация, она может включать в себя «**мелкие**»
 - данные генерируем «**экселем**» через COM
 - проверки выполняем «**диффом**», сравнивая с заранее заготовленным образцом



Почему именно это?

- Это **«полная»** автоматизация, без промежуточного участия человека
 - ожидание чуда, «синдром Емели»
 - увы, чудо очень хрупкое



Почему именно это?

- Это **«сложная»** автоматизация, требующая внимания, усилий, вложений, навыков
 - требуются стратегические решения
 - не тестирование, а программирование

Человек и машина

Анализ требований	человек
Определение набора тестов	человек
Определение тестовых данных	машина/человек
Первый прогон тестов	человек
Оценка результатов первого прогона	человек
Отладка тестов	человек
Создание баг-репортов	человек
Документирование тестов	человек
Отслеживание покрытия	человек
Определение графика выполнения	человек
Выполнение тестов	машина
Оценка результатов выполнения тестов	машина/человек
Сопровождение тестов	человек

Занятие 7. Часть 1: цели

- Что значит «автоматизация»?
- **Зачем автоматизировать?**
- Когда не автоматизировать?
- Что можно автоматизировать?

Зачем автоматизировать?

- Должны быть выгоды по сравнению с «не автоматизировать»
 - основные ожидания выгод связаны с временем
 - увеличение тестового покрытия
- Какие альтернативы?
 - выполнять вручную
 - не выполнять совсем

Ручное и автоматизированное*

	Ручное	Автоматизированное
Первое выполнение	10 мин	4 часа / 15 мин / 2 мин**
Повторное выполнение	10 мин	1 мин
При наличии сбоя	15 мин	20 мин
Изменение и выполнение	10 мин	30 мин / 10 мин / 2 мин**

* цифры условные, но отражают порядок

** пояснение через два слайда

Выгоды: время

- Получение результата **раньше**
- Прогоны **чаще** (в идеале – непрерывно)
- Высвобождается время – можно выполнить другие тесты – увеличивается покрытие

Выгоды: покрытие

- Увеличение количества тестовых данных
- Усложнение тестовых сценариев и увеличение их количества
- Увеличение количества окружений
- Возможность сосредоточиться на проектировании тестов (да-да!)

Как считать «тесты»?

- Сценарий – **1**
- Сценарий с данными – **10**
- Сценарий с данными в окружении X – **100**

Выгоды: нематериальные

- Тестировщикам это нравится больше, чем рутинная работа (что неудивительно 😊)
- Тестировщики учатся программировать

Занятие 7. Часть 1: цели

- Что значит «автоматизация»?
- Зачем автоматизировать?
- **Когда не автоматизировать?**
- Что можно автоматизировать?

Нет выгод

- **Результат раньше не востребован** (процесс)
- **Увеличение покрытия не востребовано** (цена ошибки невысока)

Не попадаем в «зеленую зону»

Первое выполнение	4 часа / 15 мин / 2 мин**
Повторное выполнение	1 мин
При наличии сбоя	20 мин
Изменение и выполнение	30 мин / 10 мин / 2 мин**

- Часто возникают сбои
 - либо меняются требования
 - либо в программе много ошибок
 - либо тесты написаны плохо
- Нет специалистов нужной квалификации
- Все тесты уникальные, нет «серийности»

Не автоматизируется

- Технологий доступа нет или они сложны
 - тач-скрин, управление взглядом
- Проверки плохо формализуются
 - удобство использования, недетерминизм, эстетическая привлекательность, верстка

Занятие 7. Часть 1: цели

- Что значит «автоматизация»?
- Зачем автоматизировать?
- Когда не автоматизировать?
- **Что можно автоматизировать?**

Аспекты качества: ISO 9126

- Функциональность
- Надёжность
- Эффективность

- Практичность
- Сопровождаемость
- Мобильность

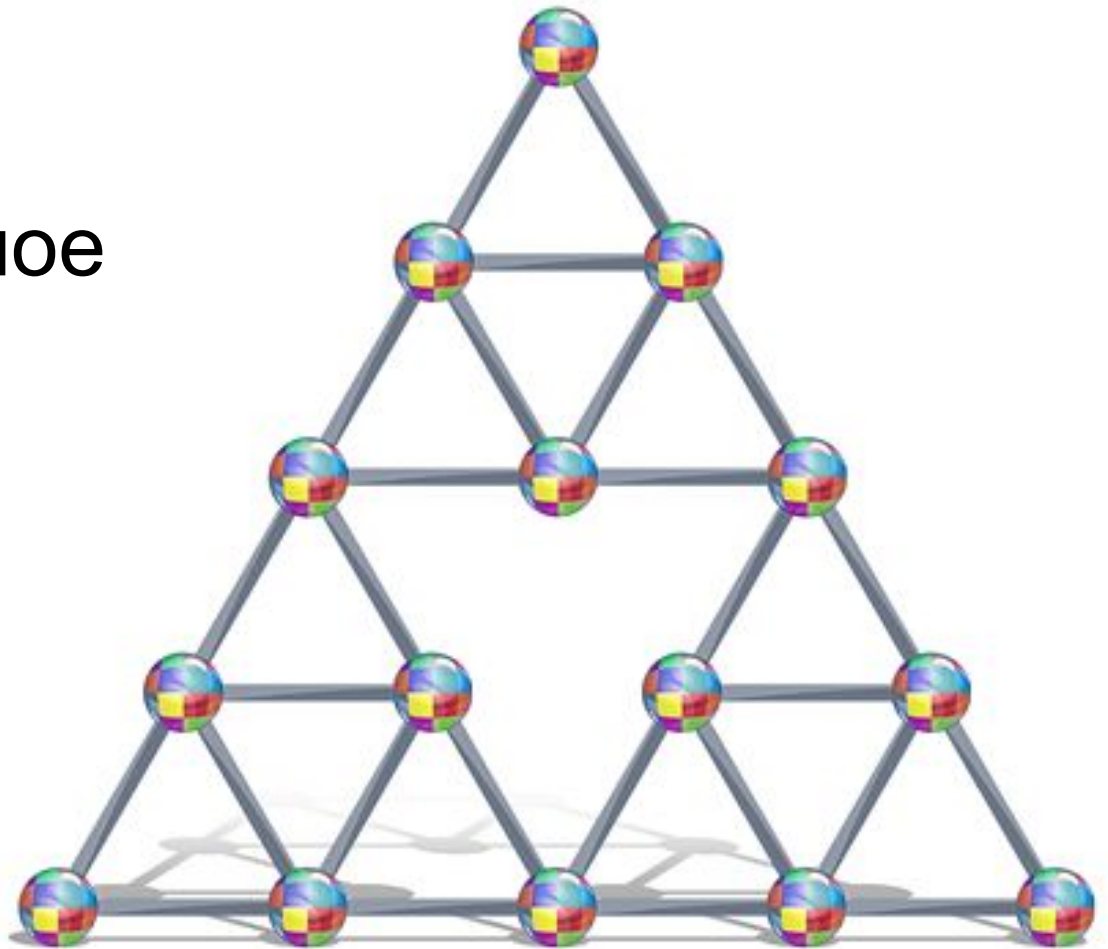


Интерфейсы

- Пользовательский интерфейс (UI)
- Программный интерфейс (API)
- Сетевой протокол
- Состояние окружения
- События
- Таймер

Уровни тестирования

- Модульное
- Интеграционное
- Системное



Стратегия и тактика



Инструменты



Критерии выбора

- Поддержка интерфейса
- Переносимость
- Язык программирования и IDE
- Расширяемость
- Производительность
- Лицензии, цены, выход новых версий
- Техсаппорт

Не изобретайте велосипед



Автоматизированные тесты



Базовые принципы дизайна

1. Тесты должны быть простыми, короткими, лёгкими в сопровождении
2. Тесты должны быть изолированы друг от друга
3. Тесты должны работать с интерфейсом
4. Тест должен доносить намерение

Тесты должны быть изолированы друг от друга

Зачем?

- неважен порядок выполнения
- можно произвольно группировать

Как?

- каждый тест должен «убирать за собой»
 - фиктивные объекты
- каждый тест должен заботиться о себе сам
 - дельта-проверки

Тесты должны работать с интерфейсом

Зачем?

- независимость от реализации, и как следствие упрощение сопровождения

Как?

- делать адаптеры со стабильным интерфейсом

Тесты должны быть короткими

Зачем?

- простота отладки
- наглядность результатов
- простота локализации багов

Как?

- не копировать ручные «один в один», проектировать тесты специально

Тест должен доносить намерение

Зачем?

- облегчает понимание
- не требуется документировать тесты

Как?

- понятные имена
- вспомогательный код

Характеристики качества ТЕСТОВ

- Сопровождаемость
 - удобно читать
 - удобно вносить изменения
- Мобильность
 - тестирование в различных конфигурациях
- Оптимизация – не нужна
 - сопровождаемость важнее!
- Надёжность – средняя
 - сопровождаемость важнее!

Разделение аспектов

- Данные отделены от кода
- Конфигурация отделена от кода
- Код разделён на пакеты

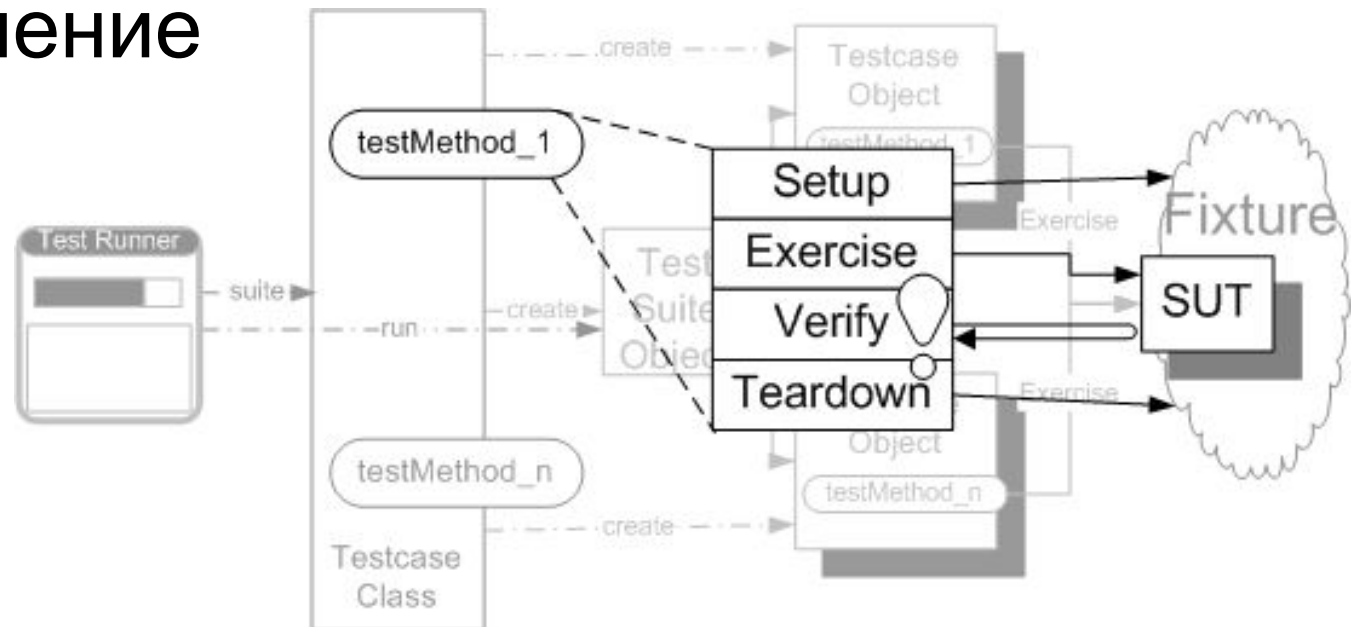
- Код разделён на слои
- Инициализация выделена
- Воздействия и проверки выделены

Базовые шаблоны проектирования

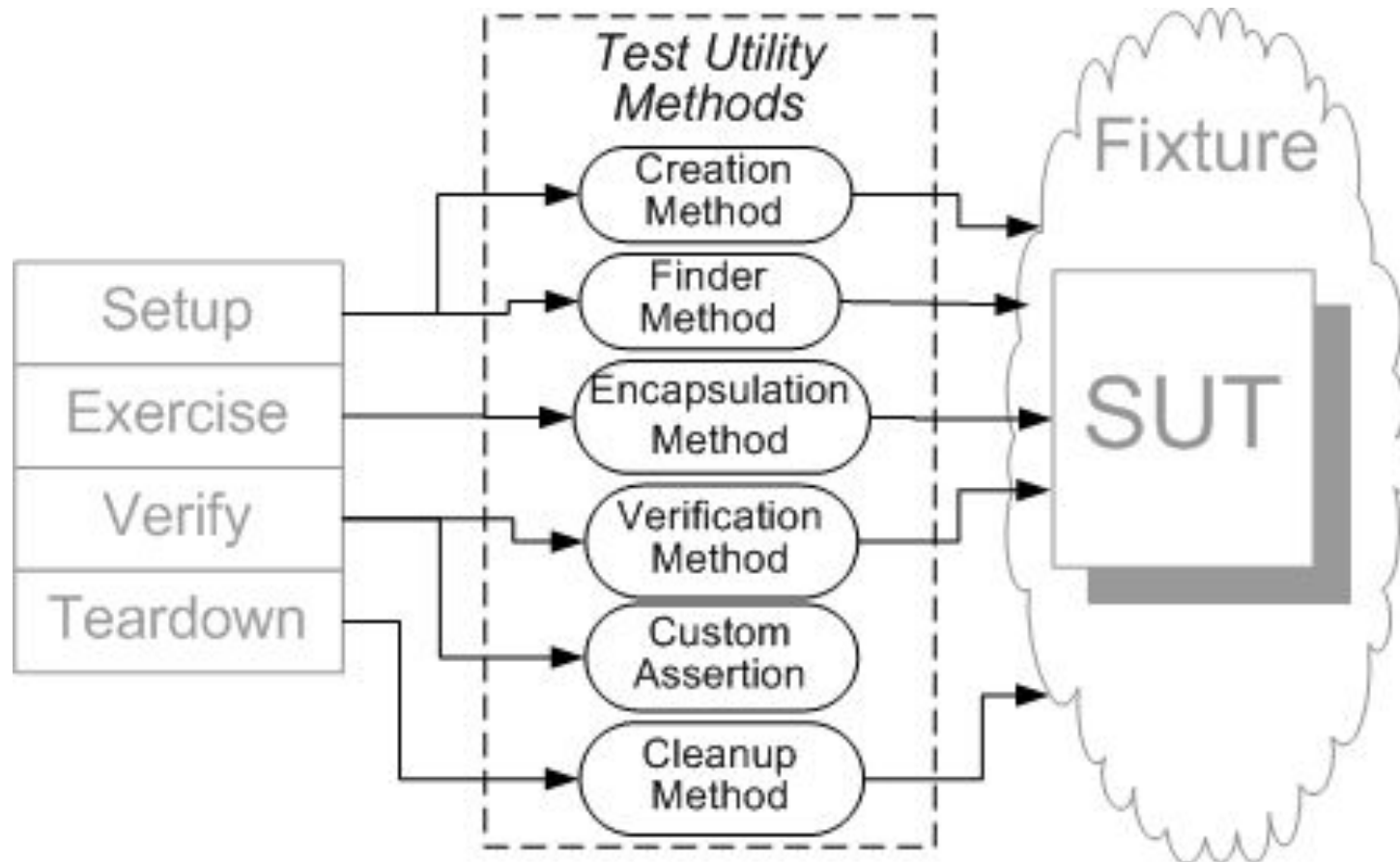
1. Четыре части – ИВПЗ
2. Переиспользуемые фрагменты
3. Адаптеры (изоляция)
4. Слои (тоже изоляция)

Делим тест на четыре части

- Инициализация
- Воздействие
- Проверка результата
- Завершение



2. Выделяем части теста

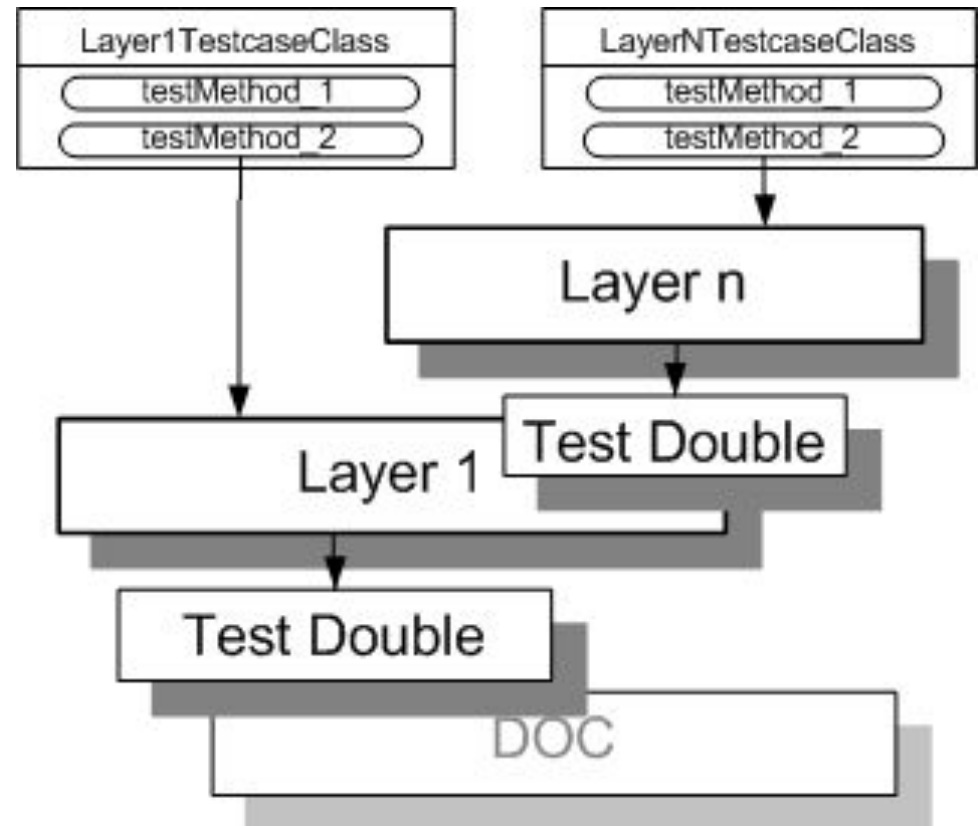


3. Выделяем адаптеры

- Адаптер реализации первого уровня
 - идентификация элементов реализации
 - непосредственное воздействие
 - непосредственная реакция
- Адаптер реализации второго уровня
 - навигация
 - комплексные воздействия
 - сохранение состояния

4. Выделяем слои

- Слои адаптеров
- Слои тестов



После выполнения тестов

- Возвращать систему в исходное состояние
- Чистка данных
- Сбор статистики



Компания хороших идей

Bellintegrator