

# Программа

## Введение

1. Примеры наиболее дорогих ошибок
2. Причины возникновения:
  - Космические лучи :)
  - Ошибки в ПО
3. Виды тестирования (различные классификации)
  - По цели
  - По свойствам
  - По исполнителю
  - По уровню
  - По интерфейсу
4. Особенности тестирования протоколов
  - RPC: WSDL/SOAP + REST/JSON
  - Messaging
5. Политика версионирования при тестировании:
  - Альфа-версии
  - Бета-версии

## Тестирование API. WSDL

### Теория

1. XSD-схемы - основы и примеры описания типов
2. WSDL-сервис - основы и пример описания методов
3. Пример генерированного кода для сервера и клиента

### Практика

1. Сетевой сниффинг SOAP с помощью wireshark
2. SoapUI:
  - Создание проекта на основе WSDL
  - Посылка запросов и получение ответов
  - Создание TestSuit
  - Проверки на основе XPath
  - Валидация WSDL/XSD

## Выбор вида тестирования

1. Пирамида тестирования:
  - Приоритеты различных видов тестирования
  - Соотношение видов тестов

## 2. Принципы семантического версионирования

- Стандарт версионирования
- Пример на разделяемом протоколе взаимодействия
- Пример на библиотечных решениях

# Тестирование API. REST

## Теория

1. Сложности тестирования messaging-протоколов и механизмов сериализации сообщений (protobuf / asn / ...)
2. Напоминание принципов протокола HTTP (GET/POST/...)
3. JSON-schema/OpenAPI/Swagger - основы и примеры описания REST API
4. Аналогия с XSD/WSDL

## Практика

1. Сетевой sniffing JSON с помощью tcpdump / wireshark
2. Postman:
  - Импорт описания API
  - Применение окружений
  - Создание запросов
  - Использование переменных и их переопределение
  - Тесты на JS для проверки:
    1. Кода возврата
    2. Полей ответа
    3. Соответствия схеме
  - Назначение и применение mock-серверов

# Тестирование интерфейса пользователя (Web)

## Теория

1. Архитектура Selenium:
  - WebDriver
  - API на Python, Java, ...
  - IDE как расширение браузеров
2. Принципы идентификации элементов web-страниц

## Практика

1. Selenium:
  - Создание виртуального окружения на Python
  - Запуск WebDriver
  - Поиск элементов на странице (css, id, атрибуты)

- Ввод текстовых данных
- Автоматизированная генерация сценария в IDE

## Нагрузочное тестирование

### Теория

1. Фокусы нагрузочного тестирования:
  - Производительность
  - Стабильность
  - Отказоустойчивость
  - Масштабируемость
  - Стресс-тестирование
2. Профили нагрузки:
  - SLA
  - Пределы производительности
3. Параметры:
  - Время обработки
  - Частота запросов
  - Размер данных
4. Откуда брать профили нагрузки:
  - БД
  - Журналы
  - Прогноз
5. Инструменты:
  - Web-консоль
  - JMeter
  - Gatling
  - K6
6. Место в CI:
  - Базовый workflow сборки проекта
  - Интеграционное тестирование с записью видео/результатов

### Практика

1. JMeter:
  - Поддерживаемые протоколы
  - Ручное создание HTTP-запросов
  - Запись сценариев через Proxy
  - Thread group и его параметры
  - Вынесение общих параметров
  - Просмотр результатов в графическом и табличном видах

# Тестирование интерфейса пользователя (Desktop)

## Теория

1. Примеры технологий разработки интерфейса и соответствие инструментов тестирования со знанием идентификаторов элементов интерфейса:
  - Java/Swing или SWT: ?
  - Qt: Squish
  - JS: Selenium
  - Windows forms: ?
2. Применение компьютерного зрения: Sikuli
3. Применение машинного обучения: Testolang

## Практика

1. Sikuli
2. Testolang

# Fuzzing-тестирование

## Теория

1. [Введение в fuzzing](#)

## Практика

1. [libfuzzing "Hello world"](#)
2. OSS-Fuzz
  - [OSS-Fuzz документация](#)
  - [Список проектов под OSS-Fuzz](#)
3. [RESTler документация и репозиторий](#)
4. [Schemathesis документация и репозиторий](#)

From:

<http://se.moevm.info/> - **se.moevm.info**

Permanent link:

<http://se.moevm.info/doku.php/courses:testing:lectures?rev=1646290417>

Last update: **2022/12/10 09:08**

