

Фреймворк для авто-тестирования веб-сервисов своими силами

Воробьев Андрей
Test Automation Engineer
EPAM, Минск

О себе



Воробьев Андрей

➔ Test Automation Engineer EPAM Systems

➔ Автор и ведущий тренингов для начинающих по авто-тестированию веб-сервисов

➔ andrei_varabyeu@epam.com

Профессиональный жаргон

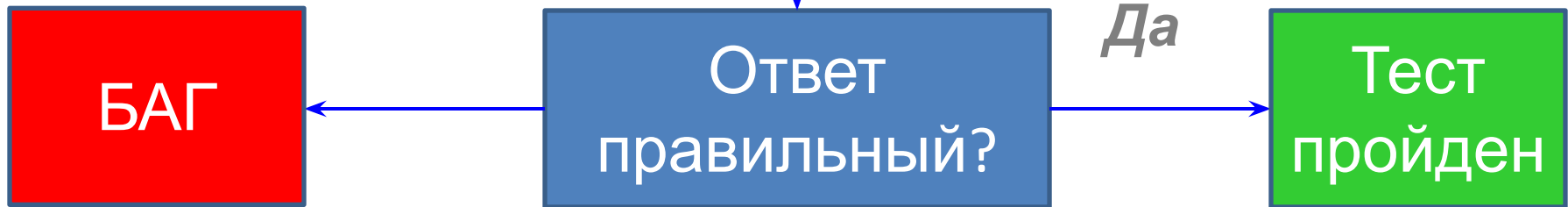


Авто-тестирование веб-сервиса



Запрос

Ответ



Авто-тестирование веб-сервиса



Покупать продукт

или

разрабатывать самим?

Как решить?



*Для того, чтобы принять
верное решение, достаточно
ответить только на один
вопрос:*

***“Насколько тривиальные задачи
требуется решить”***

Сложно или просто?



Сложно или просто?

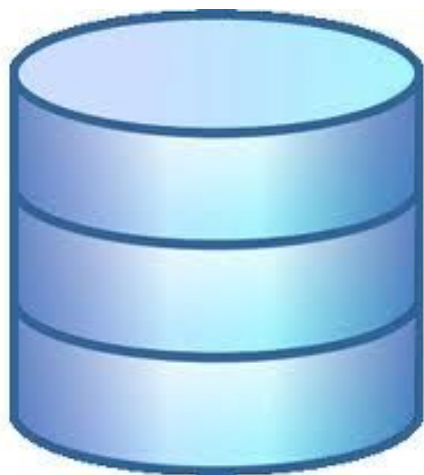
Просто:

- *Простые входные данные (числа, строки)*
- *Примитивные типы валидаций (=, >, <, contains)*
- *Простые выборки в базу. Малый объем выборок*

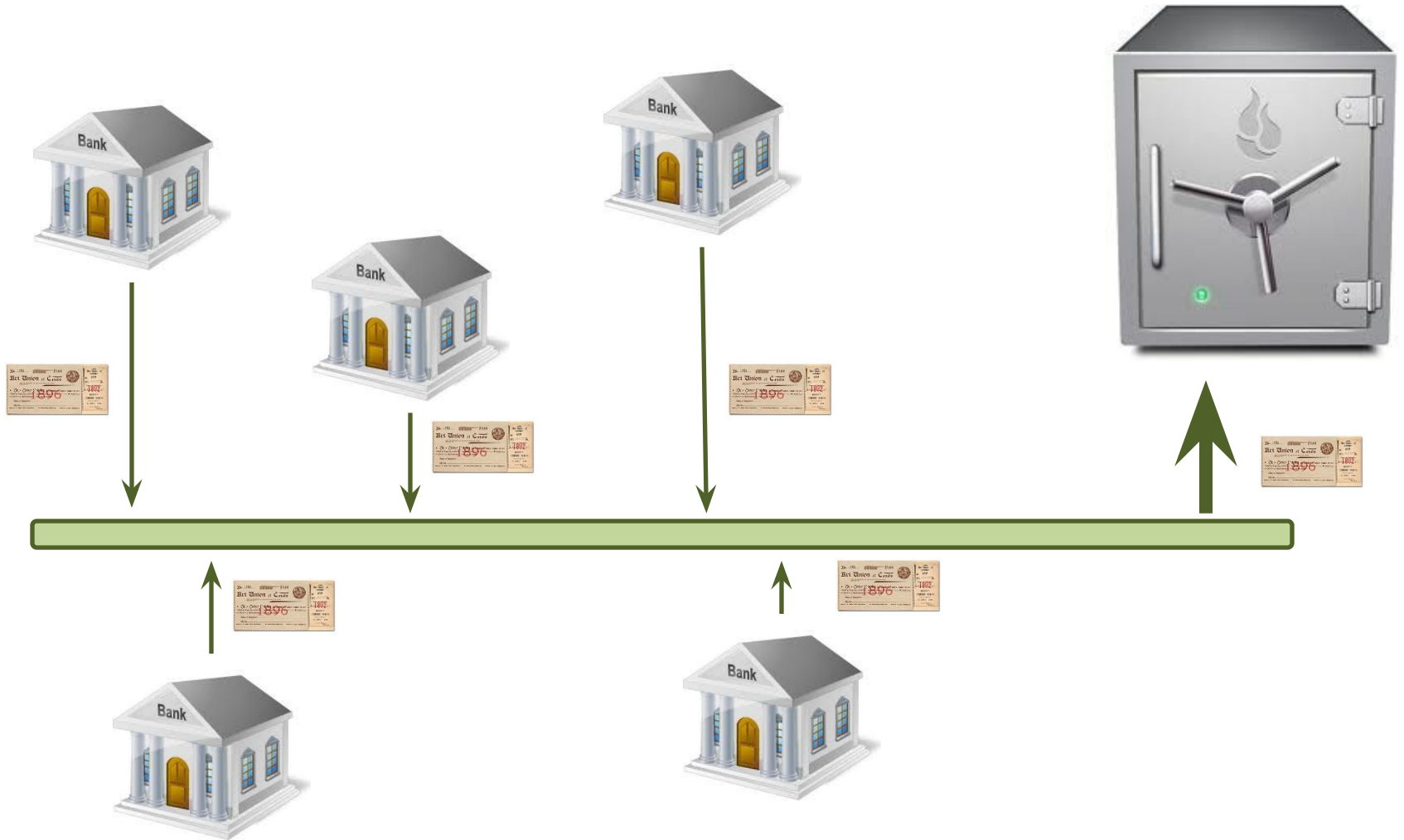
Сложно:

- *Сложные входные данные (части запросов, которые нужно группировать друг с другом)*
- *Сложная валидация (по шаблону, по данным которые тоже нужно откуда-нибудь получить)*
- *Поддержка транзакций БД, кэширования*

Сложно или просто?



Сложно или просто?



Сложно или просто?

<person name='Андрей Воробьев'>

<relative>Луна Воробьева</relative>

<relative>Сергей Воробьев</relative>

</person>

Сложно или просто?

<person name='Андрей Воробьев'>

<relative name="Ли́на Воробьева">

<relative>Валентина Криводубская</relative>

<relative>Дмитрий Криводубский</relative>

</relative >

<relative name="Сергей Воробьев">

<relative>Ядвига Воробьева</relative>

<relative>Николай Воробьев</relative>

</relative >

</person>

Сложно или просто?

```
<person name='Андрей Воробьев'>
  <relative name="Луна Воробьева">
    <relative name="Валентина Криводубская">
      <relative>Анна Криводубская</relative>
      <relative>Петр Криводубский</relative>
    </relative>
    <relative name="Владимир Криводубский">
      <relative>...</relative>
      <relative>...</relative>
    </relative>
  </relative>
  <relative name="Сергей Воробьев">
    <relative name="Ядвига Воробьева">
      <relative>...</relative>
      <relative>...</relative>
    </relative>
    <relative name="Николай Воробьева">
      <relative>...</relative>
      <relative>...</relative>
    </relative>
  </relative >
</person>
```

N == ?

Просто



- *SOAP, Rest, HTTP, JMS, DB*
- *XPath Assertions*
- *XQuery Assertions*
- *Logging*
- *Scripting Support (Groovy, JavaScript)*
- *WS Security*
- *.... and more*

СЛОЖНО

- *Java/C#/...*
- *WS implementation*
- **Unit testing Frameworks*



«Сделай сам»

**Фреймворк для тестирования
Веб Сервиса – это всего лишь:**



- *Клиент веб-сервиса (для отправки запросов и получения ответов от сервера)*
- *Реализация работы с БД, другими источниками входных/выходных данных*
- *Набор валидаций, которые нужны **VAM** (XPath, XQuery, DB, равно, содержит, больше\меньше, соответствует заданному формату)*

Task list 😊

подготовить входные данные
(получить из файлов, БД, etc)

сформировать и отправить
запрос/получить ответ

выполнить валидацию ответа

... и так много раз в течение одного
теста



Подготовка запроса

```
<soapenv:Envelope  
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
<soapenv:Body>  
  <familyTree>  
    <name>Андрей Воробьев</name>  
  </familyTree >  
</soapenv:Body>  
</soapenv:Envelope>
```

Подготовка запроса

soapUI

- *Java Property files*
- *Data Source (excel)*
- *DB via 'Property Transfer'*
- *использование groovy*

Java-based framework

- *Java Property files*
- *DB (+ORM, caching, transactions)*
- **ЛЮБЫЕ** *другие источники данных (Windows Active Directory, http, ftp, etc)*

Отправка запроса/Получение ответа

soapUI

- **ВСЕ** уже сделано за вас

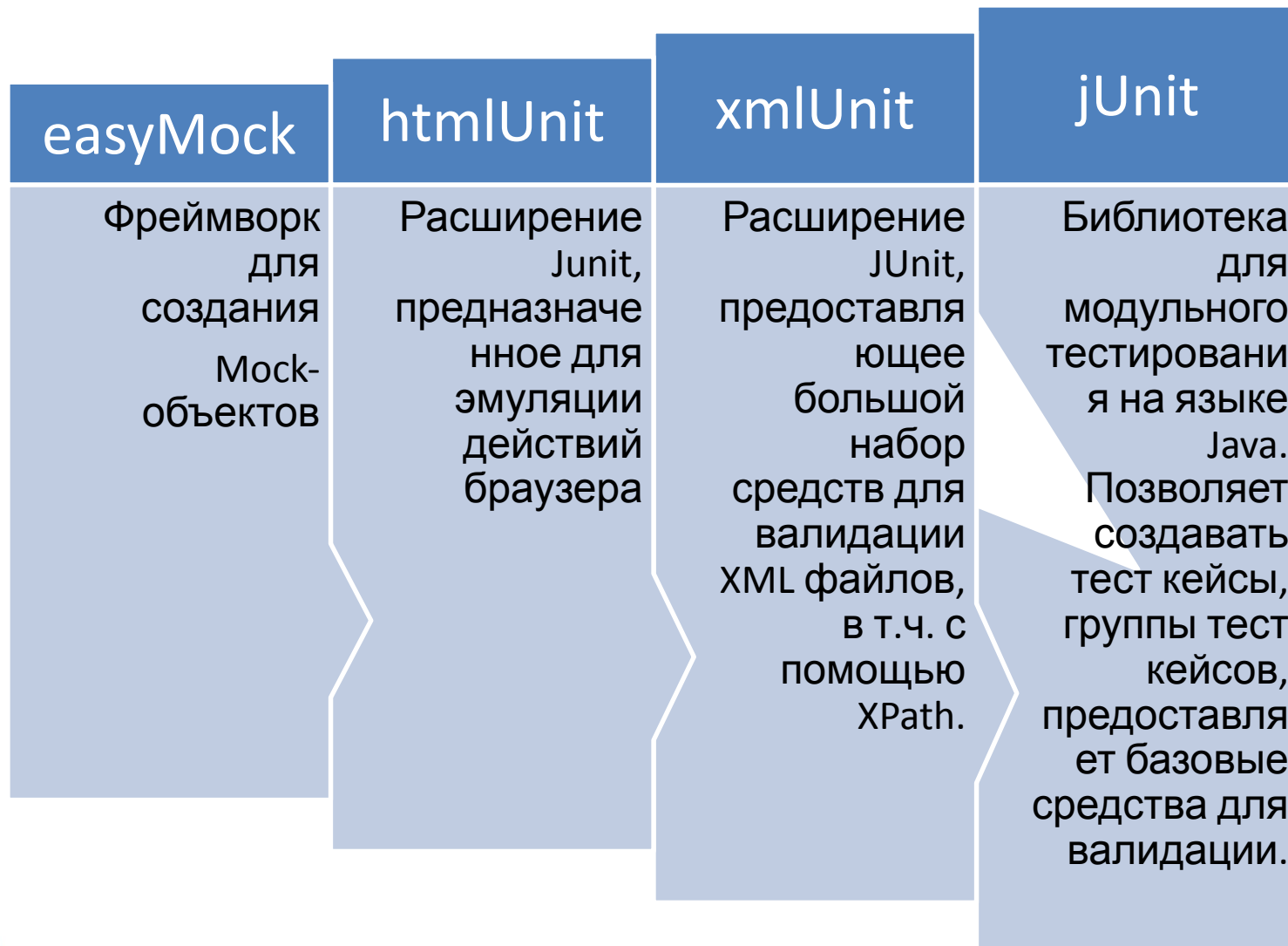
Java-based framework

- JAX-WS
- Spring-WS
- ...
- **ЛЮБЫЕ** другие источники имплементации, которые **удобно** использовать **ВАМ**

Валидация ответа

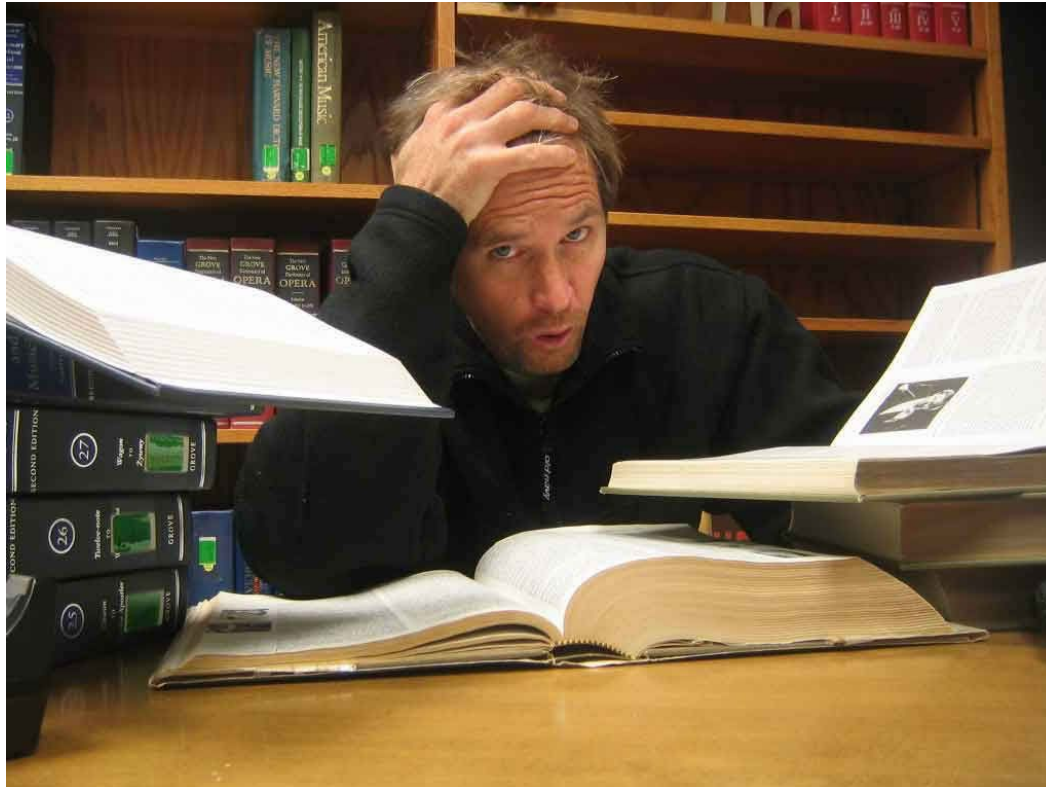
```
<soapenv:Envelope  
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
<soapenv:Body>  
  <person name='Андрей Воробьев'>  
    <relative>Луна Воробьева</relative>  
    <relative>Сергей Воробьев</relative>  
  </person>  
</soapenv:Body>  
</soapenv:Envelope>
```

soapUI – это тоже Java код



Архитектура

или то, над чем стоит поломать голову...



Архитектура. Основные принципы

Модульность



Каждый из компонентов не зависит от другого

Архитектура. Основные принципы

Расширяемость

Легко добавлять:

новый функционал

новые типы валидаций



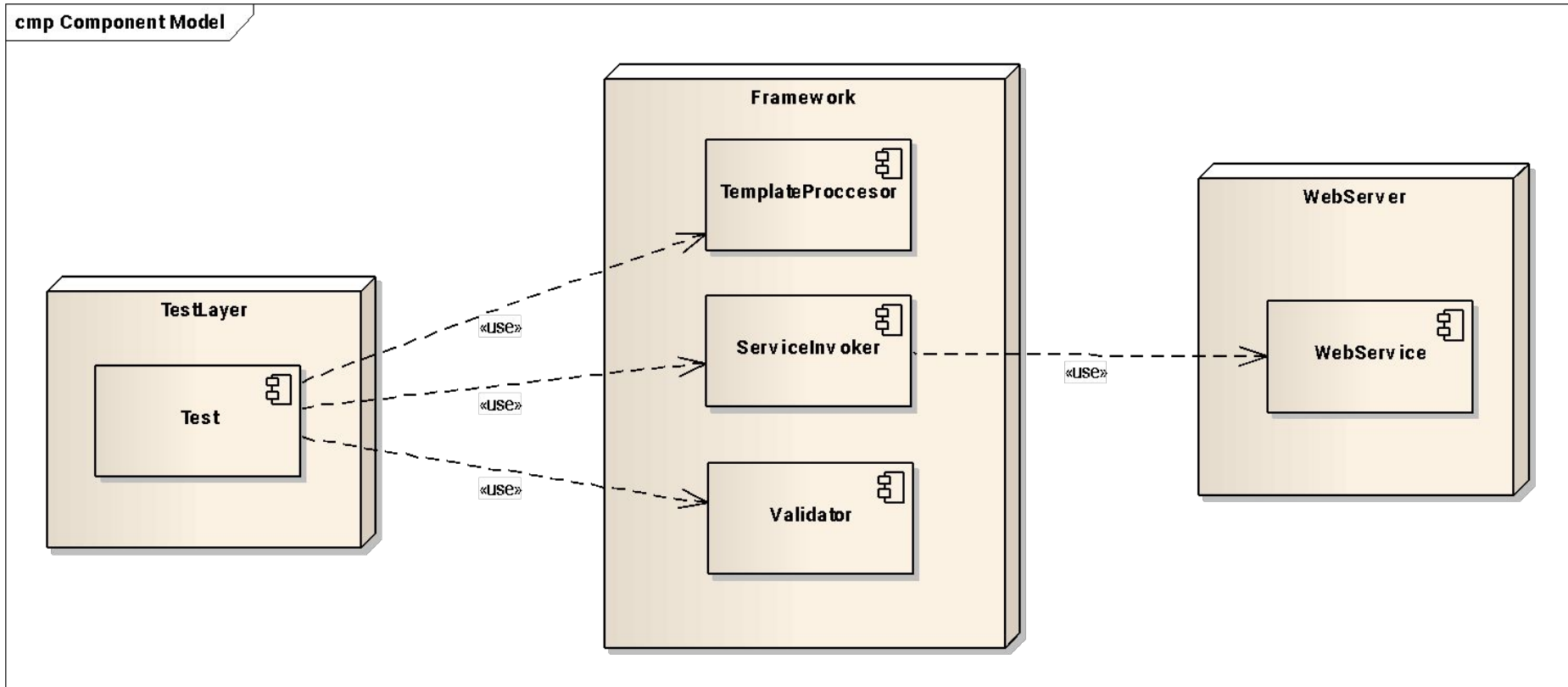
Архитектура. Основные принципы

Читаемость

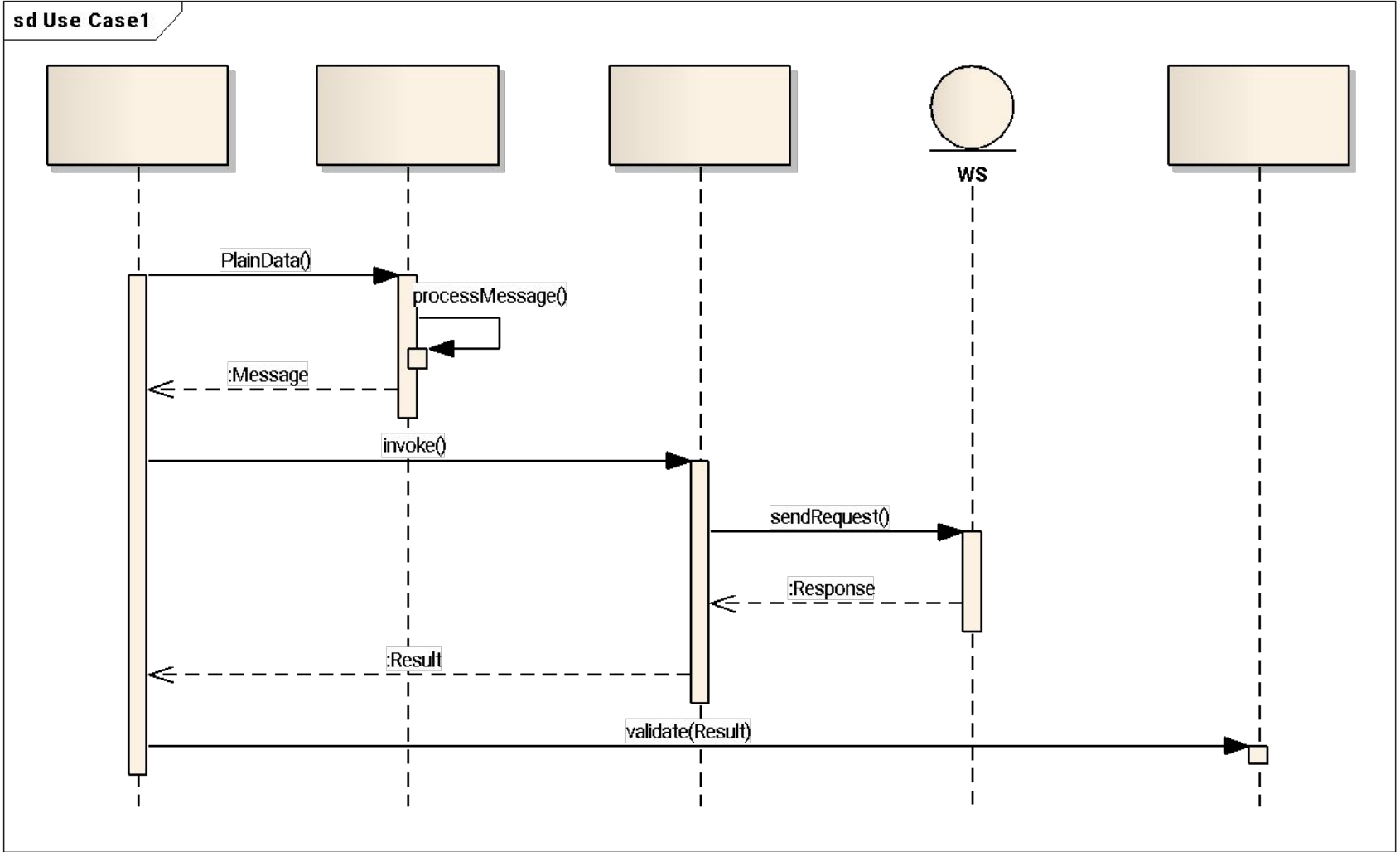
*Достаточно посмотреть
на исходный код теста,
чтобы понять что он делает*



Архитектура. Компоненты



Архитектура. Диаграмма последовательности



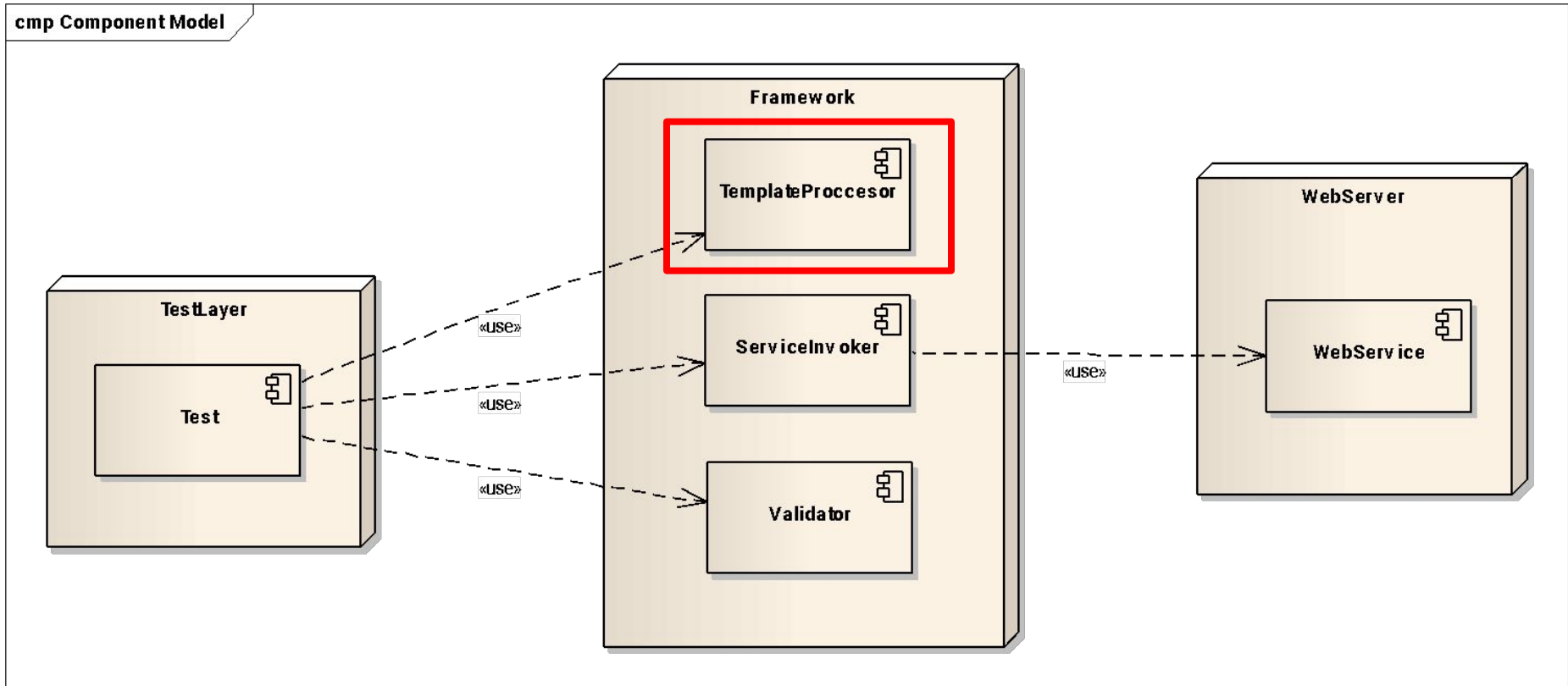
Компоненты. Тест



TestNG

- *базовые типы валидации*
- *группировка в сьюты*
- *отчеты о прохождении тестов*
- *возможность выполнять тесты в несколько потоков*
- *зависимости между тестами*

Компоненты



Компоненты. Процессор шаблонов

Данные



ГОТОВЫЙ
запрос

Шаблон

Компоненты. Процессор шаблонов

`<request>`

`<amount>${amountValue}</amount>`

`<currency>${currencyValue}</currency>`

`</request>`

Шаблон

Компоненты. Процессор шаблонов

Данные

amountValue = 100

currencyValue = USD

Компоненты. Процессор шаблонов

<request>

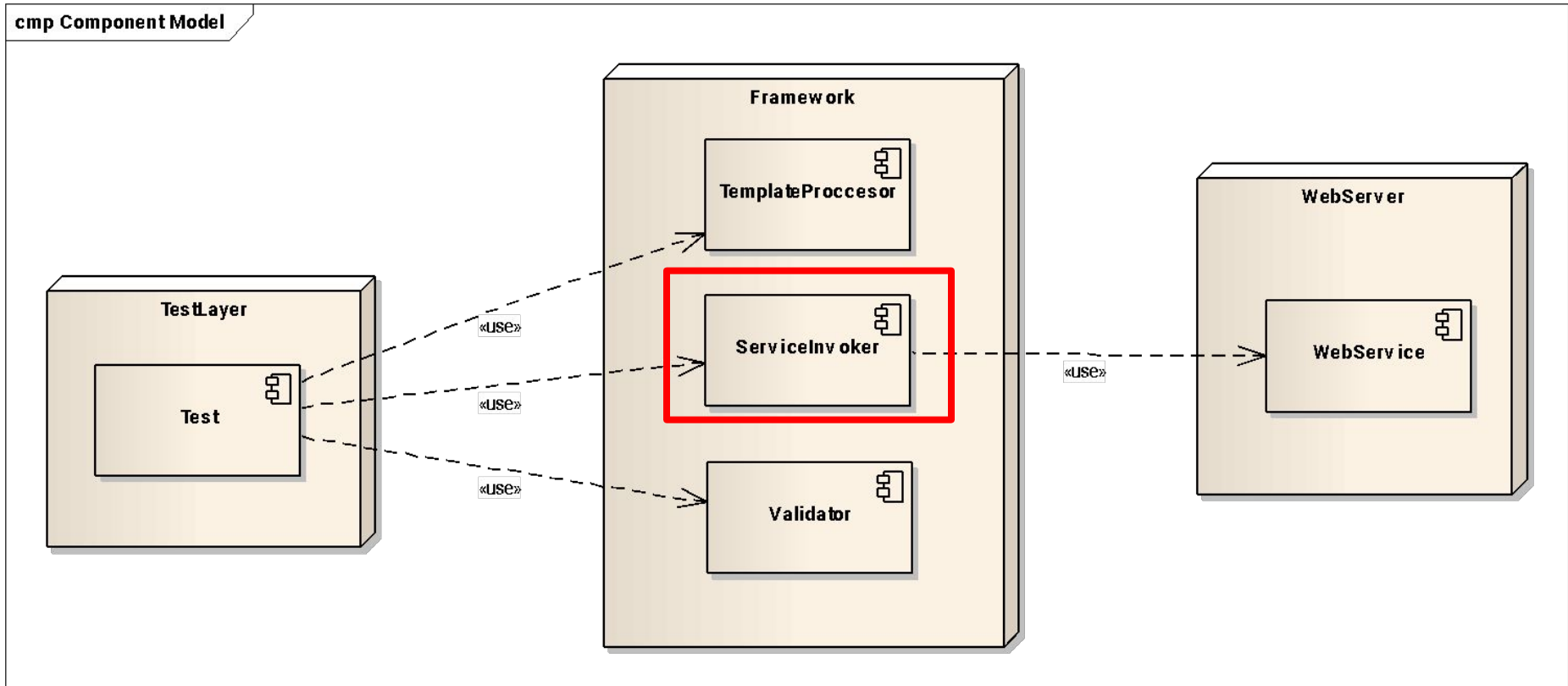
*<amount>***100***</amount>*

*<currency>***USD***</currency>*

</request>

ГОТОВЫЙ
запрос

Компоненты



Компоненты. XML - модель

Мы используем...

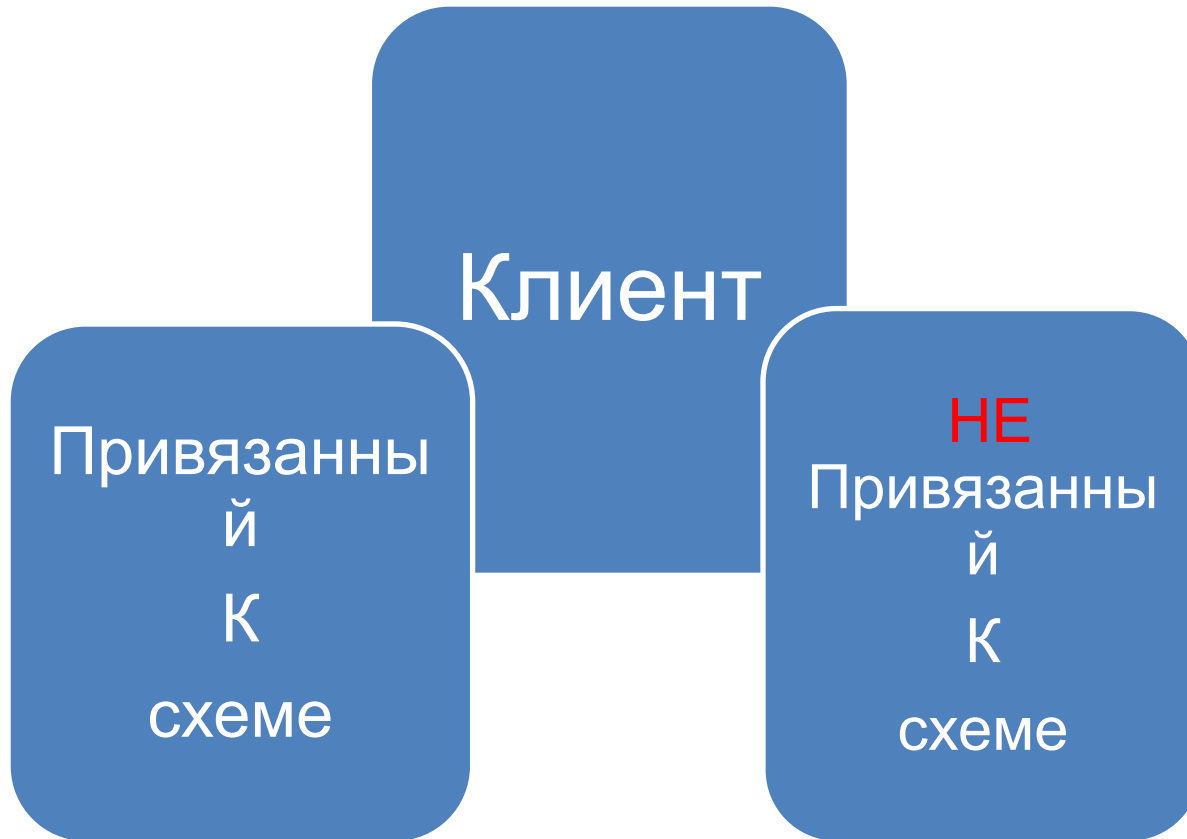
DOM

- XML в чистом виде
- любые (в т.ч. невалидные) данные
- парсинг, XPath
- слабая привязка к схеме

JAXB

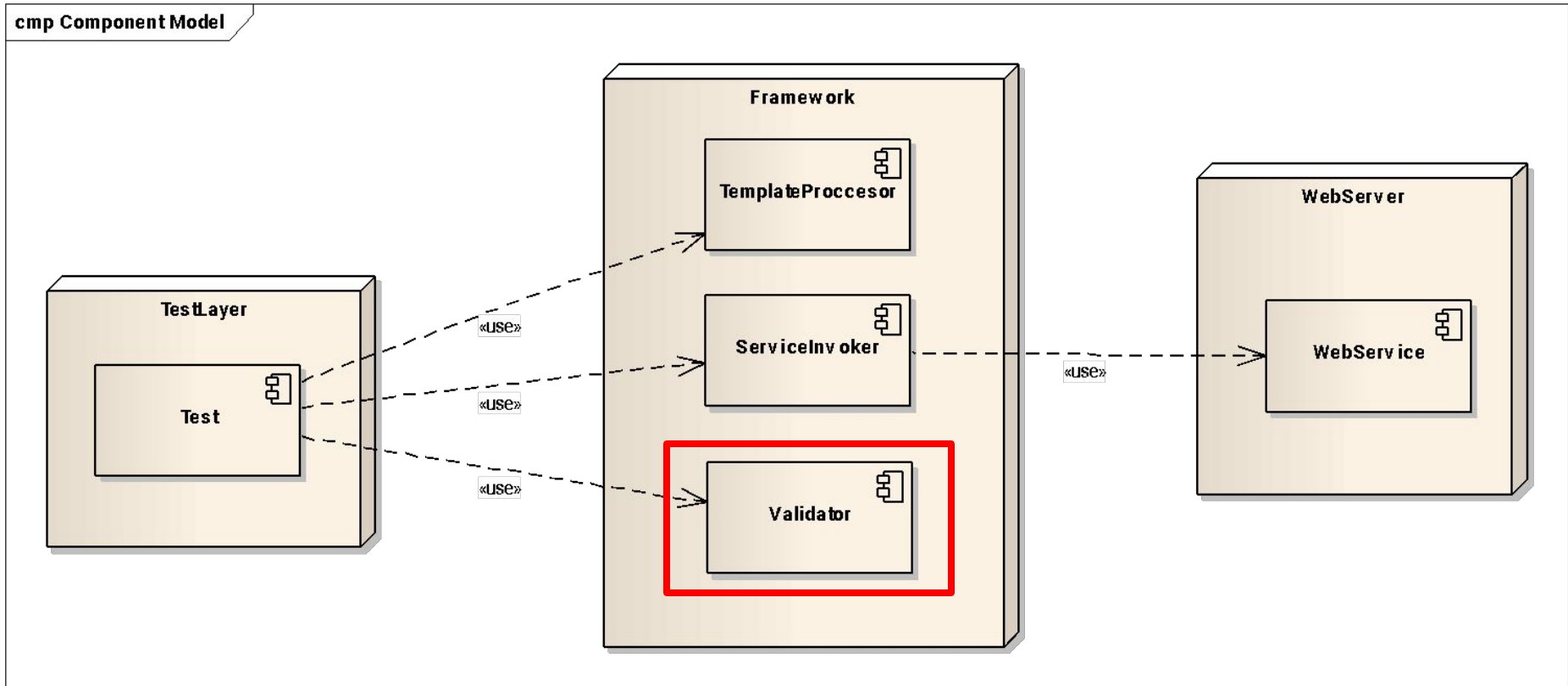
- нет парсинга
- нет необходимости писать XPath выражения
- невозможно вставить невалидные данные
- жесткая привязка к схеме

Компоненты. SOAP-клиент



- *исходный код может быть сгенерирован автоматически по WSDL-схеме*
- *исходный код нужно писать самому*

Компоненты



Компоненты. Валидация



Компоненты. Валидация. Ответ

DOM

- JUnit /TestNG + XPath
- XmlUnit + XPath
- XMLUnit + шаблон ответа

JAXB

- JUnit/TestNG
- XPath

Почему это так важно

Новые тесты

создаются

быстро



Почему это так важно

Старые тесты

поддерживать

просто

2 x 2

5 x 5

soapUI Pro умеет...

- *измерять степень покрытия тестами*
- *тестирование безопасности (SQL-инъекции, XPath-инъекции)*
- *производить нагрузочное тестирование*
- *интегрироваться в сборку сервиса и CI-цикл*
- *создавать Моск-сервисы*

soapUI vs. разработка



- *не нужно никаких подготовительных работ*
- *новые тесты создаются **медленно***
- *старые тесты поддерживаются **медленно***
- *квалификация авто-тестировщика ниже среднего*
- *базовый набор функционала**

** расширяется только groovy-скриптингом*

soapUI vs. разработка



- *перед написанием тестов необходимо разработать механизмы отправки, получения и валидации*
- *новые тесты создаются **быстро***
- *старые тесты поддерживать **просто***
- *квалификация авто-тестировщика выше среднего*

soapUI vs. разработка



Возможности ограничены

ТОЛЬКО ЯЗЫКОМ

программирования

soapUI vs. разработка



- **Любые** протоколы
- **Любые** источники входных и выходных данных (Active Directory, POP3, SMTP)
- **Простота** валидации сложных типов данных (даты, UUID)

Наш опыт



На разработку

Фреймворка и первого

теста путем

проб и ошибок

ушло около

**2 человеко-
месяцев**

Наш опыт



Сейчас,

имея опыт,

Это можно

сделать за

***2 человеко-
недели***

Ваши вопросы!



Спасибо за внимание!

Воробьев Андрей
Test Automation Engineer
ЕРАМ, Минск

Валидация ответа

soapUI

- *Fault/Success*
- *Contains*
- *XPath*
- *Xquery*
- *Schema Compliance*
- *Timeout*
- *WS-Security Status*
- *Groovy*
- ...

Java-based framework

- *JUnit*
- *TestNG*
- *XmlUnit*
- *HtmlUnit*
- *DbUnit*
- *...*Unit*

Компоненты. Процессор шаблонов

<FreeMarker>

 The **Apache Velocity Project**
<http://velocity.apache.org/>

Feature	FreeMarker	Velocity
Циклы	Да	Да
Условия	Да	Да
Макросы	Да	Да
Импорт других шаблонов	Да	Да
Процессинг импортированных шаблонов	Да	Да
Выполнение Java-кода	Нет	Частично

Компоненты. XML – модель. JAXB

