PAPRYQARZ

WE TEST WITH TASTE

Funkcjonalne testy w Jmeter i Continuous Integration

Arkadiusz Bany

07.09.2016

Geneza

- Operator telekomunikacyjny
- Baza sprzedażowa dla nowego systemu
- REST API / kilka endpoint'ów / duże request'y i response'y
- Blackbox testing i odrobina whitebox testing

Wybór narzędzia – wymagania kluczowe

- Wsparcie dla XML
- Wsparcie dla SQL
- Wsparcie dla języków skryptowych
- Szybkie prototypowanie testów
- Open-source

Wybór narzędzia - wymagania dodatkowe

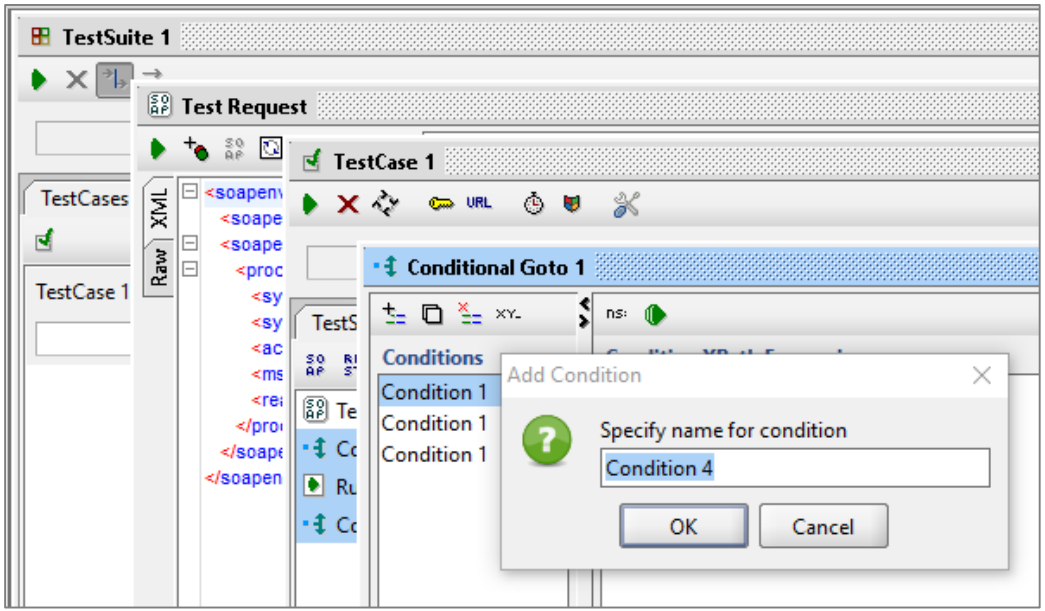
- Wsparcie dla testów wydajnościowych
- Wsparcie dla Continous Integration
- Popularność i łatwa konfiguracja
- Raporty testów

SoapUI – „plusy dodatnie”

- Wsparcie dla XML, SQL i Groovy
- Szybkie prototypowanie testów (idealny dla testów ad hoc)
- Bardzo popularny
- Wsparcie dla testów wydajnościowych (LoadUI)
- Darmowy (chociaż nie do końca...)

SoapUI – „plusy ujemne”

- Okrojne API w darmowej wersji
- Niewygodny interfejs (dużo okienek i klikania)



Java / Python + xUnit + biblioteka XML – plusy

- Największe możliwości programistyczne
- Dużo dodatkowych narzędzi i bibliotek
- Continuous Integration

Java + JUnit + biblioteka XML – minusy

- Brak możliwości szybkiego prototypowania testów
- Czasochłonna konifguracja (Java jest ciężka...)
- Ograniczone możliwości użycia dla testów wydajnościowych
- Duże wymagania względem zespołu testerskiego

Jmeter – plusy i minusy

- Wsparcie dla XML
- Wsparcie dla SQL
- Wsparcie dla języków skryptowych
- Szybkie prototypowanie testów
- Open-source
- Wsparcie dla testsów wydajnościowych

Czyli wszystkie zalety SoapUI bez jego wad...

Jmeter Tips & Tricks

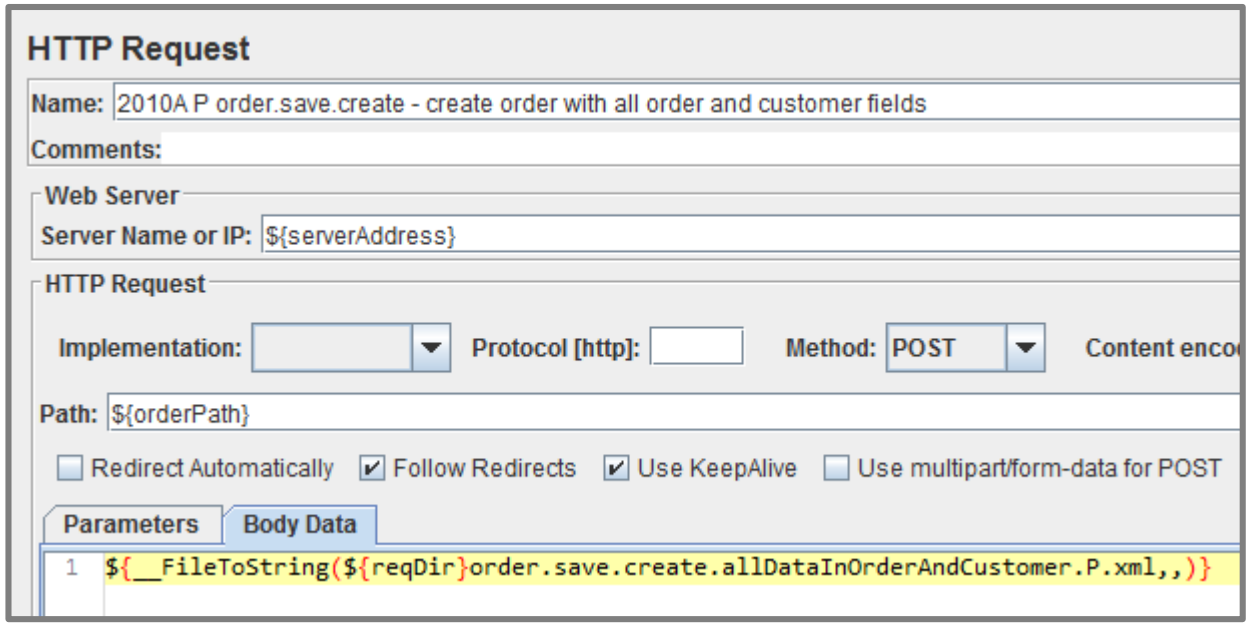
Czy ma Pan jakiś Problem?

Request'y XML w kodzie to same problemy...

- Bardzo wolne ładowanie projektu (JMX to zwykły XML!)
- Czy pisanie testów ma polegać na Ctrl+C i Ctrl+V?
- Koszmar refaktoringu

Ratunek – *FileToString* i zewnętrzne XML-e

```
${__FileToString(request.xml,,)}
```



A co z parametryzowanymi request'ami?

- plik *request1.xml:*

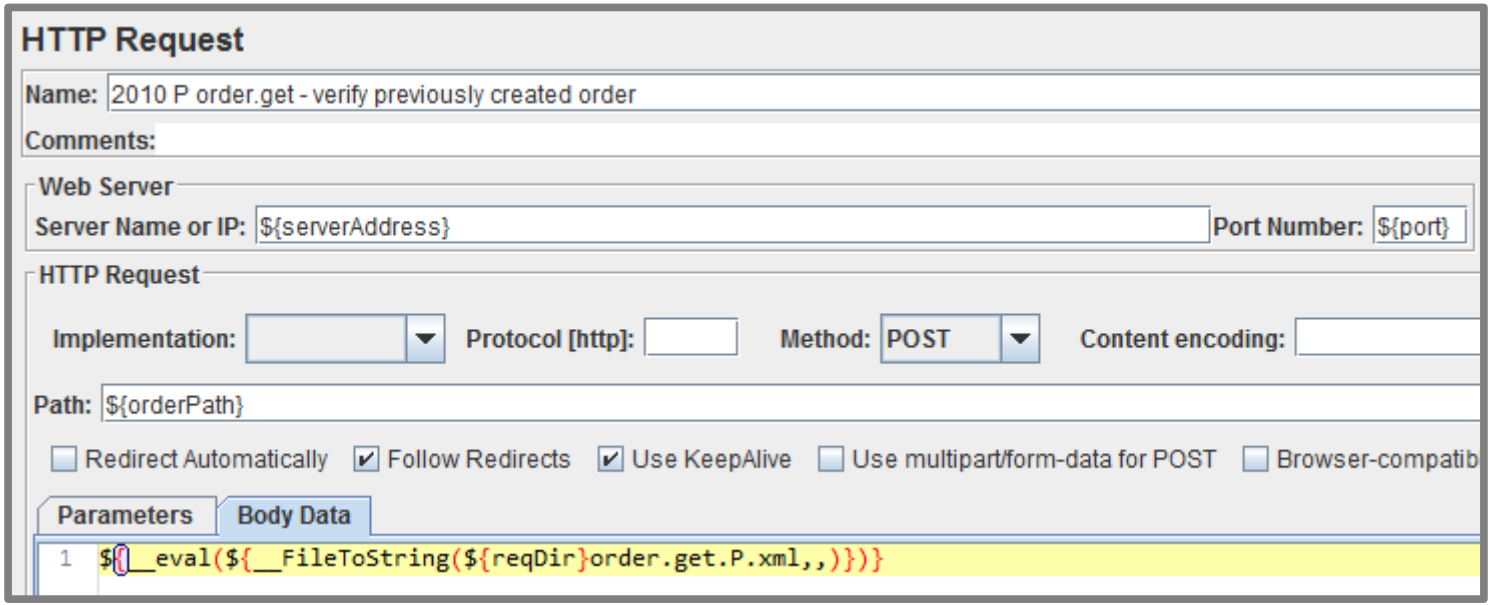
```
<xml>...
      <firstName>Arkadiusz</firstName>...
</xml>
```

- plik *request2.xml*:

```
<xml>...
      <firstName>${firstName}</firstName>...
</xml>
```

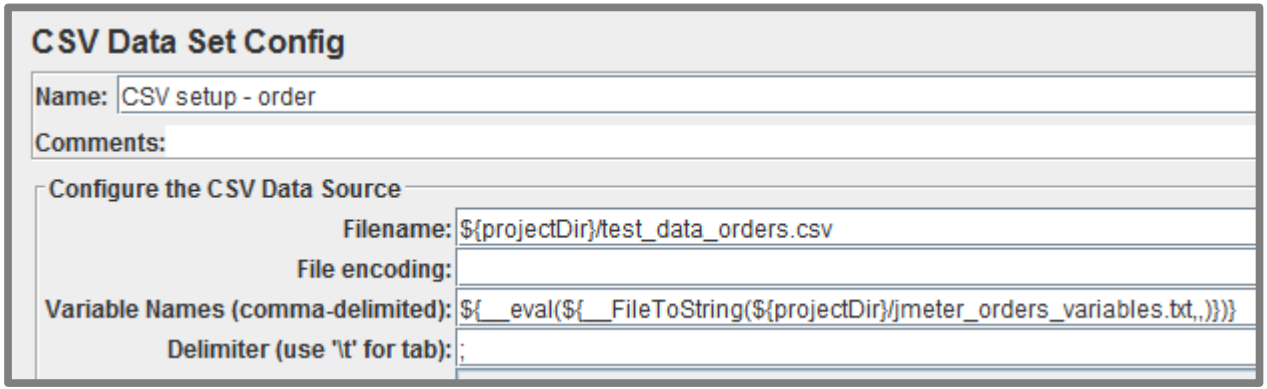
Samo nie zadziała... ale z *eval* tak!

- **`${__eval(${__FileToString(request2.xml,,)})}`**



Data-driven testing? Tak, ale trochę inaczej...

- Niekonwencjonalne użycie *CSV Data Set Config*
- Plik CSV (z dowolnym separatorem) – tylko jedna linia!
- Plik TXT ze zmiennymi (z przecinkiem) – tak samo jak wyżej! :)



Dane losowe – czyli *Random* i *RandomString*

- `${__Random(0,100)}`
- `${__RandomString(1, 0123456789,)}`
- `${__RandomString(10, abcdeABCDE,)}`

Przykład generowania „losowej” daty:

```
<xml>...
        <date> 201${__RandomString(1, 123456789,)}
        -0${__RandomString(1, 123456789,)}
        -0${__RandomString(1, 123456789,)}
        T00:00:00+01:00</date>...
</xml>
```

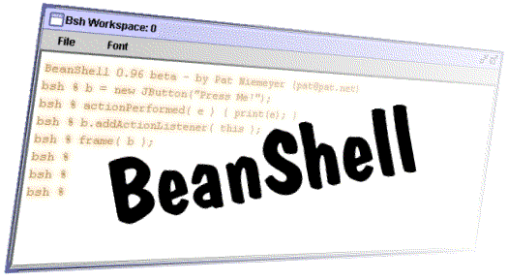
Inne metody

- *time* - zwraca aktulany czas
- *counter* - zwraca wartości inkrementowane o 1
- *intSum* - zwraca sumę dwóch liczb
- *property* - zwraca wartość z konfiguracji
- *BeanShell* - wykonuje kod BeanShell
- *javaScript* - wykonuje kod JavaScript
- *log* - zapis do logu

XPath to samo zło? Wręcz przeciwnie!

- Pierwsze przykazanie Selenium - *Nie stosuj XPath'a bo jest wolny!*
- Bzdu... nie jest to do końca prawda :)
- *XPath Extractor & XPath Assertion*
- Dobre dopasowanie to podstawa
- Testujemy to co napisaliśmy (np. Notepad++ & XML Tools)

BeanShell




```
1   import java.util.Date;
2   import java.text.DateFormat;
3   import java.text.SimpleDateFormat;
4   import java.util.Calendar;

6   DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
7   Calendar cal = Calendar.getInstance();
8   cal.add(Calendar.DATE, 1);
9   vars.put("tomorrow_date", dateFormat.format(cal.getTime()));
10  log.info("tomorrow_date " + vars.get("tomorrow_date"));
```

BeanShell i SQL

JDBC Request

Name: history status test 1

Comments:

Variable Name Bound to Pool

Variable Name: dbConnection

SQL Query

Query Type: Prepared Select Statement

```
1  SELECT id, modify_user
2  FROM order_history
3  WHERE order_id = ?
```

Parameter values: ${orderId}

Parameter types: INTEGER

Variable names:

Result variable name: queryResult

Query timeout (s):

BeanShell Assertion

Name: number of rows in db assertion

Comments:

Reset bsh.Interpreter before each call

Parameters (-> String Parameters and String []bsh.args)

Script file

Script (see below for variables that are defined)

```
1  rows = vars.getObject("queryResult");
2
3  if (rows.size() != 1) {
4      Failure = true;
5      FailureMessage = "wrong number of rows";
6  }
7
```

BeanShell – wady?

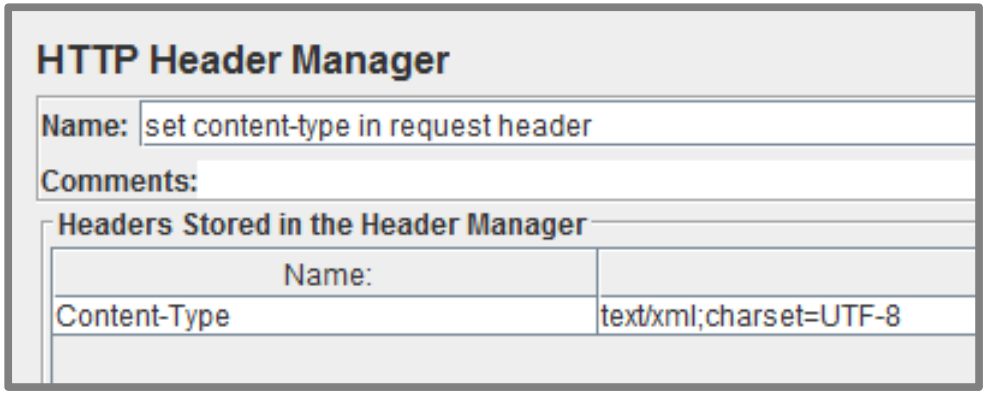
- Mało popularny
- Trudne debugowanie
- Groovy jest lepszy!

>

Czasami trzeba być Ą Ę...

Problemy z polskimi diakrytycznymi?

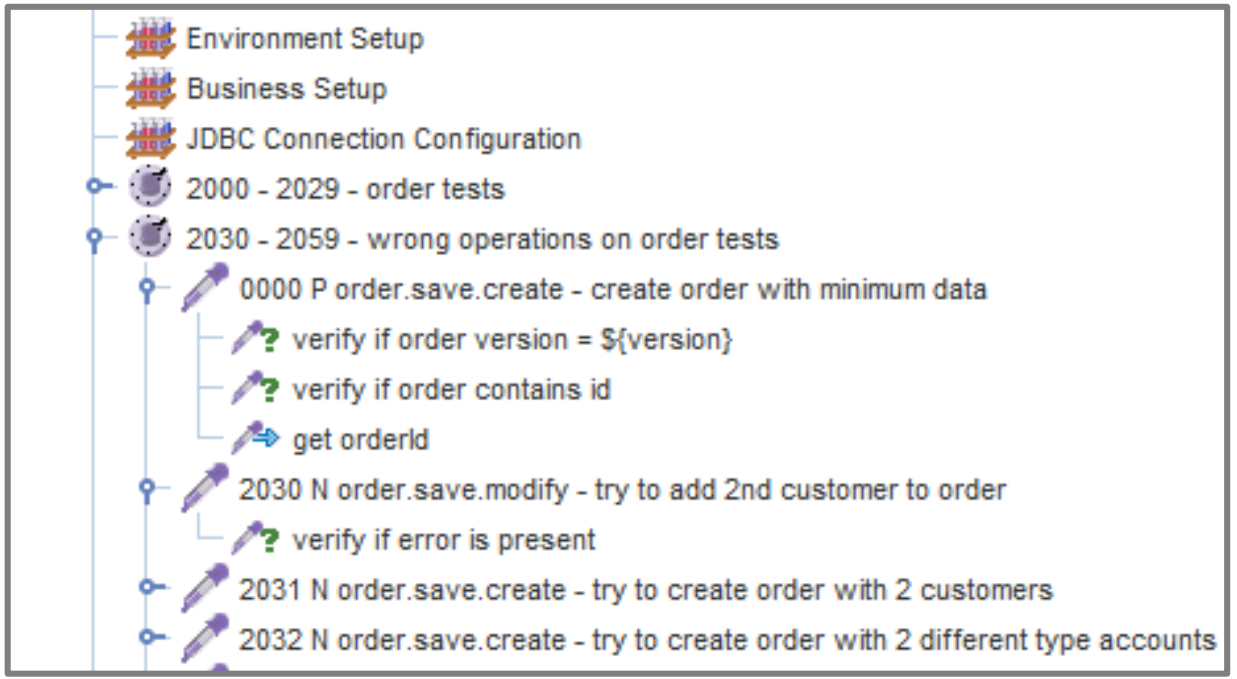
- *HTTP Header Manager*
- **`Content-Type: text/xml;charset=UTF-8`**



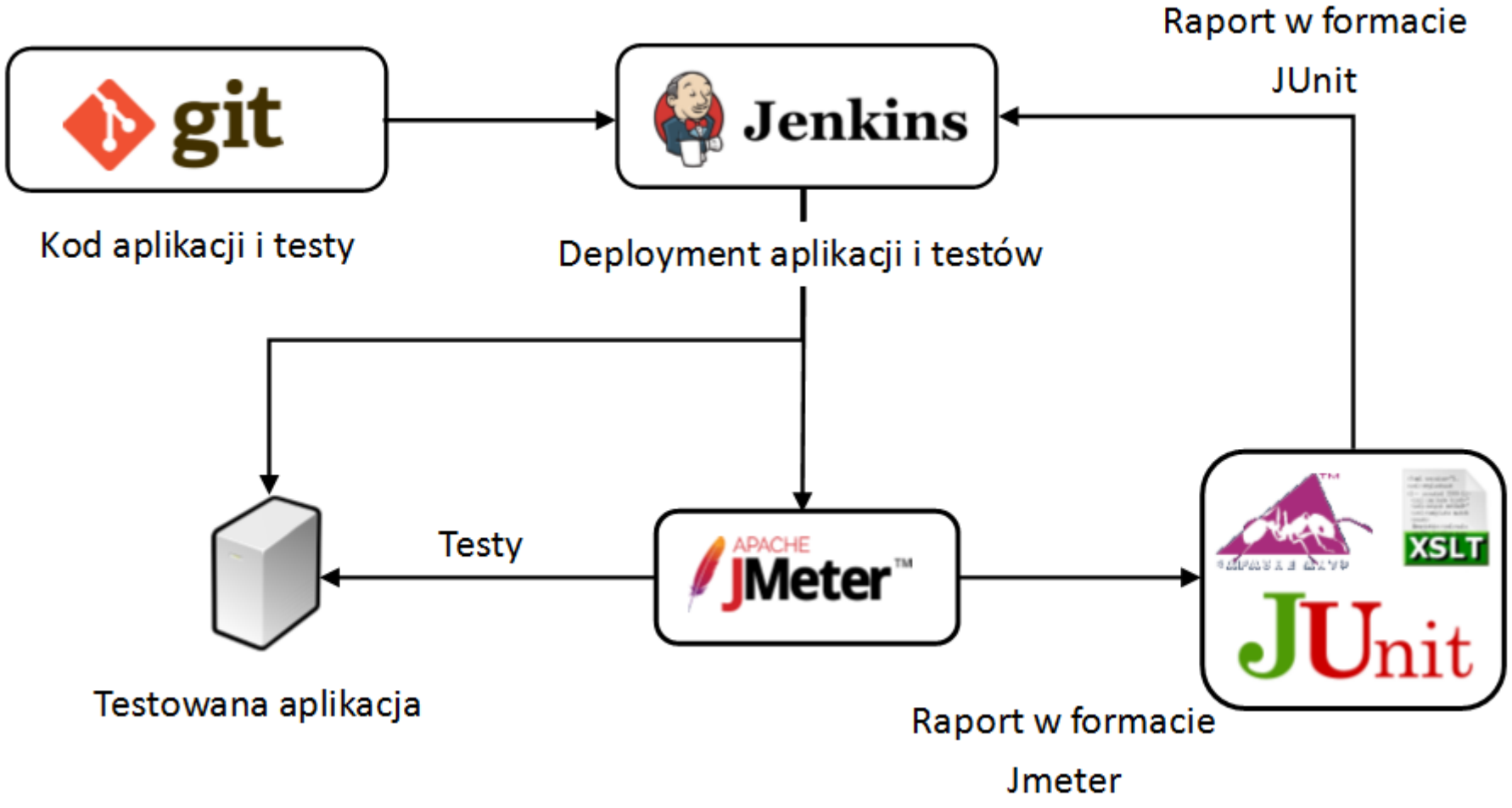
Porządek w testach musi być!

- *User Define Variables*
  - Dane konfiguracyjne (adresy, porty, ścieżki itp.)
  - Dane biznesowe (loginy, identyfikatory systemów itp.)
- Grupowanie testów przy użyciu *Simple Controler*
- Numerowanie testów oraz grup testów
- Stosowanie krótkich i treściwych nazwa dla testów
- Rozróżnianie scenariuszy pozytywnych od negatywnych (*P*/*N*)
- Komentarze

Przykład struktury testów



To co z tym Continuous Integration?


Raport w formacie
JUnit
git
Jenkins
Kod aplikacji i testy
Deployment aplikacji i testów
Testy
APACHE
JMeter™
XSLT
JUnit
Testowana aplikacja
Raport w formacie
Jmeter

Continuous Integration – co jeszcze?

- *Jenkins Performance* plugin niestety się nie przyda
- Walidacja raportu przy użyciu XSD dla właściwej wersji JUnit'a
- Problem z określeniem katalogu z zewnętrzymi plikami:

```
${__BeanShell(import org.apache.jmeter.services.FileServer;
FileServer.getFileServer().getBaseDir();)}
```

lub

Zdefiniowanie własnej zmiennej w pliku *jmeter.properties* ze ścieżką do katalogu testów.

Podsumowanie

- Jmeter to świetne narzędzie
- Idealny do małych i średnich projektów
- Bardzo duże projekty? Być może warto użyć innych narzędzi...

Pytania

Użyte grafiki

https://www.soapui.org/
http://junit.org/junit4/
https://en.wikipedia.org/wiki/Java_(programming_language)
https://commons.wikimedia.org/wiki/File:Gorilla-server.svg
https://en.wikipedia.org/wiki/Groovy_(programming_language)
http://jmeter.apache.org/
https://git-scm.com/downloads/logos
http://ant.apache.org/
https://www.pinterest.com/pin/296393219208704561/
https://wiki.jenkins-ci.org/display/JENKINS/Logo