# Rix Groenboom & Jaap Mulder
# Strategie voor het testen van webservices

**Samenvatting:**
Tijdens deze presentatie gaan we nader in op het testen van via webservices geïmplementeerde applicaties. Het gebruik van de juiste test methode maakt het mogelijke om betrouwbare en schaalbare webservices te ontwikkelen.

Door de stijgende populariteit van Service Oriented Architecture neemt de noodzaak voor het onafhankelijk kunnen testen van webservices toe wat een complexe testomgeving verlangt. Aspecten van governance, functionaliteit, interoperability en security zijn van groot belang. Bestaande testmethoden schieten al snel te kort door de fundamenteel andere werkwijze van webservies.

De strategie voor het testen van webservices bestaat uit de volgende stappen:

- Het valideren van de WSDL file die het interface van de webservices beschrijft
- Het testen van de afzonderlijke services op robuustheid
- Het valideren van de functionaliteit van de services
- Het testen van de beveiligingsaspecten van de aangeboden services
- De combineren van individuele webservices tot complexere operaties en het modelleren van user-requirements
- De bestuderen van het gebruik van de service bij toenemende aantal gebruikers / transacties.

Deze strategie vormt de basis voor het ontwikkelen en testen van een webservice. Het past goed in moderne ontwikkelmethodieken als Test Driven Development (TDD).

Deze methode is succesvol toegepast van vele ontwikkelproject van SOA. Tijdens deze presentatie worden voorbeelden gegeven van het ontwikkelen van de infrastructuur bij de centrale overheid.

**Biografie:**
Jaap Mulder is heading the recently opened Parasoft Netherlands office, with mission providing sales, training and implementation support to the Dutch marketplace. Jaap has over 15 years experience training, pre- and post sales roles within in the software development and testing industry. He experience goes from embedded systems development, software development lifecycle tools to service oriented architecture. Jaap Mulder holds a BSc in Computing Science.

Rix Groenboom is Support Manager for Parasoft, a leading editor of Automated Error Prevention solutions to support European users. His primary responsibility is working with Fortune 2000 customers in the field of error prevention and correction. He has written over 30 technical articles and presented on Open Source and quality issues at many IT industry conferences. His main area of expertise is in the use of formal languages for the specification, design and validation of software applications. He holds a MSc and PhD in Computing Science from the University of Groningen (Netherlands) and has also performed part of his studies at the Institut de Recherche en Informatique de Toulouse (IRIT in France).

PARASOFT
We make software work.

# Strategie voor het testen van webservices

Jaap Mulder & Rix Groenboom
{jmulder,rixg}@parasoft.com

PARASOFT
We make software work.

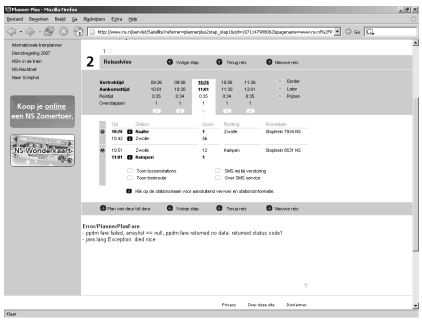
## Addressing Web Services Quality (1)

- Quality Concerns
  - Security
  - Interoperability
  - Reliability
  - Availability
  - Performance
  - Evolving Standards
- Reliable Web Services require
  - Client implementation is error-free
  - Server implementation is error-free
  - Client and server interact correctly
  - Business processes execute successfully



PARASOFT
We make software work.

## Example this week

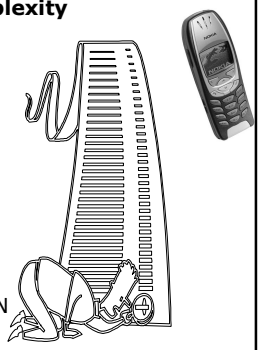




PARASOFT
We make software work.

## Problems: Size and Complexity

- 3 MLOC of SW
- 50 lines = 25 cm
- 100 = 50 cm
- 200 = 1 m
- 1,000 = 5 m
- 10 kloc = 50 m
- 100 kloc = 500 m
- 1 Mloc = 5 km
- 3 Mloc = 15 km
- 8 Mloc = MARATHON





PARASOFT
We make software work.

## Addressing Web Services Quality (2)

- Web Services testing differs from traditional Web Application testing
  - Different standards and skill sets (WSDL, SOAP, etc.)
  - The need for specialized tools and technologies that support Web Services standards and fit Web Services architectural patterns (SOA)
  - Security testing: tools and practices suitable for Web Services

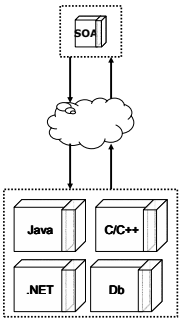


PARASOFT
We make software work.

## Addressing Web Services Quality (3)

- Web services error prevention requires a multi-layered approach:
  - Testing the messaging layer
  - Testing the application layer

PARASOFT
We make software work.




**Message Layer**
- Verify Service Description
- Verify Policies
- Test Web Services Infrastructure
- Unit test Service Layer
- Business Process Test
- Scenario Test
- Functional Security Test / Penetration Test
- Regression Test
- Verify Scalability and Performance

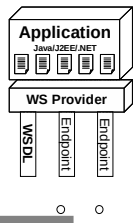
**Implementation Layer**
- Code Analysis
  - Security - Reliability
  - Performance - Maintainability
- Automated Unit/Regression Testing
- Component Unit/Regression Testing



PARASOFT
We make software work.

## Implementation layer testing

- Static analysis and best coding practices
  - Find errors before running dynamic tests
  - Adhere to standards (e.g. security, reliability, testability, maintainability)
- Unit testing
  - Make sure smallest pieces of code behave correctly
    - Check for unexpected inputs
    - Increase test coverage
    - Black box test cases to validate component

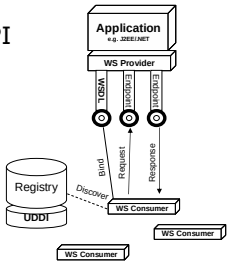





PARASOFT
We make software work.

## Message layer testing

- Message, description, discovery
  - WSDL, XML/SOAP, UDDI
- Transports/Messaging API
  - HTTP
  - JMS
  - EJB
  - TIBCO Rendezvouz
  - IBM MQ
  - SMTP
  - RMI

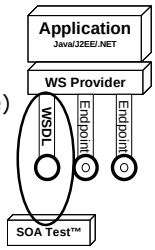





PARASOFT
We make software work.

## WSDL Testing

- Validate the WSDL (Web Services Description Language document)
  - W3C Schema validity
  - WS-I 1.1 Interoperability
  - Semantic correctness
  - Organisational rules (governance)
  - Regression






PARASOFT
We make software work.

## Message layer testing – Tier Isolation

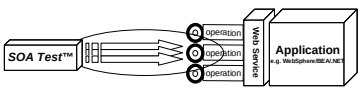

- Stub out the consumer (client) to test the service provider (server)
- Stub out the service provider (server) to test the consumer (client)
- Stub out both ends to test a proxy or intermediary



PARASOFT
We make software work.

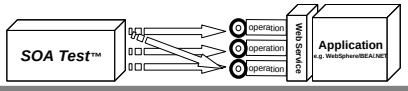
## Message layer testing – Unit Testing

- Has analogy to unit tests for programming languages:
  - Test each operation in isolation to ensure the validity of the XML payloads and that it returns the expected response per request
    - Positive conditions (positive tests)
    - Error and faulty conditions (negative tests)
    - Standards compliancy



## Message layer testing – Scenario Based Testing

- Functional test to ensure business process scenarios
  - Combine individual Web Service functions to realistic business use cases
- Create tests which are representing
  - Expected usage patterns of the end users (positive tests)
  - Unexpected usage patterns (negative tests)






## Message layer testing – Load and Performance Testing

- Identify bottlenecks
- Predict scalability
- Verify SLAs
  - Use as part of an iterative development process in a continuous automated manner, not only before deployment!
- Use realistic scenarios
  - Expected usage patterns of the end users
  - Unexpected usage patterns
  - Live messages with dynamic values



## Message layer testing – Security Testing

- Functional Security Testing
  - Scenarios tests to check implementation of security policy
  - Positive conditions, negative conditions and standards compliance
    - XML signature, encryption, WS Security Username Tokens, SAML
- Penetration Testing
  - Mitigate threats by simulating attacks and checking for potential vulnerabilities to SQL Injections, XPath Injections, XML Bombs, etc.



## Problems: XML Bomb

bomb.xml



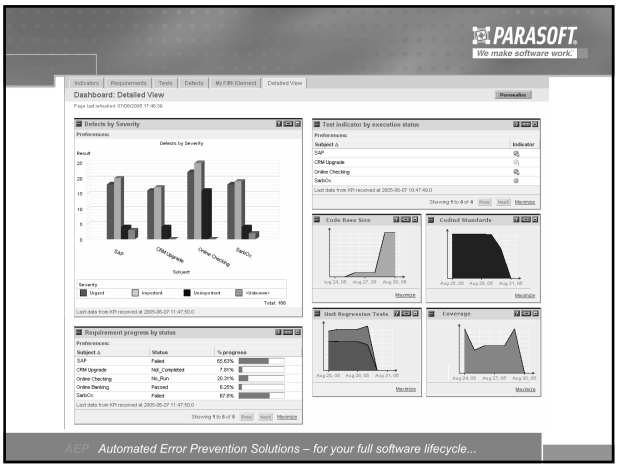
## Problems: XML Bomb

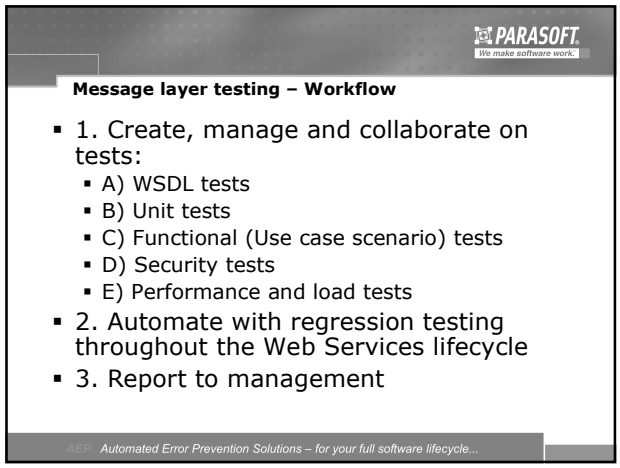
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SOAP-ENV:Envelope [
  <!ELEMENT SOAP-ENV:Envelope ANY>
  <!ATTLIST SOAP-ENV:Envelope entityReference CDATA #IMPLIED>
  <!ENTITY x0 "Bomb!">
  <!ENTITY x1 "&x0;&x0;">
  <!ENTITY x2 "&x1;&x1;">
  ...
  <!ENTITY x20 "&x19;&x19;">
  <!ENTITY x21 "&x20;&x20;">
  <!ENTITY x22 "&x21;&x21;">
]>
```

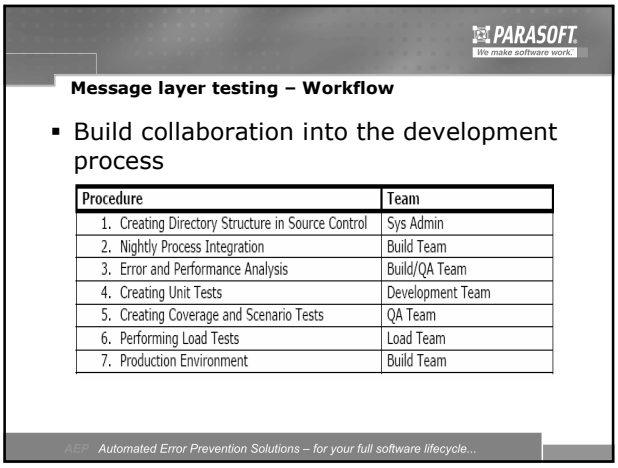


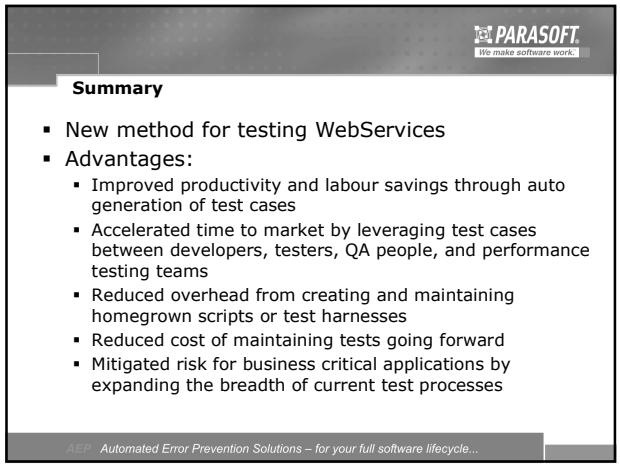
## Message layer testing – Regression Testing

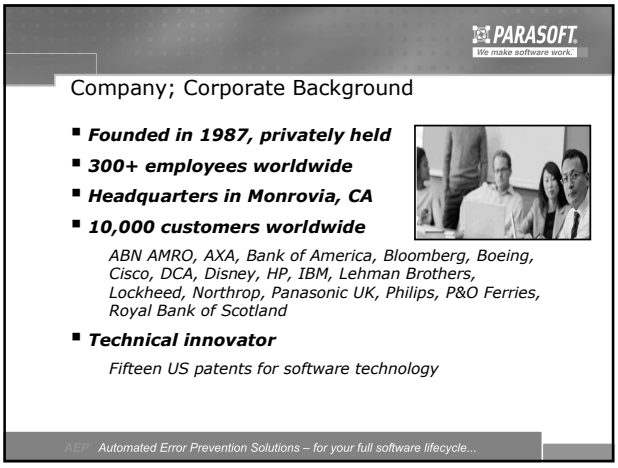
- Regression Testing: test execution is automated to run tests on a regular basis
- Spend time on creating tests, not running them!
- Regression tests created during development are reused in:
  - Maintenance projects
  - Troubleshooting services in production

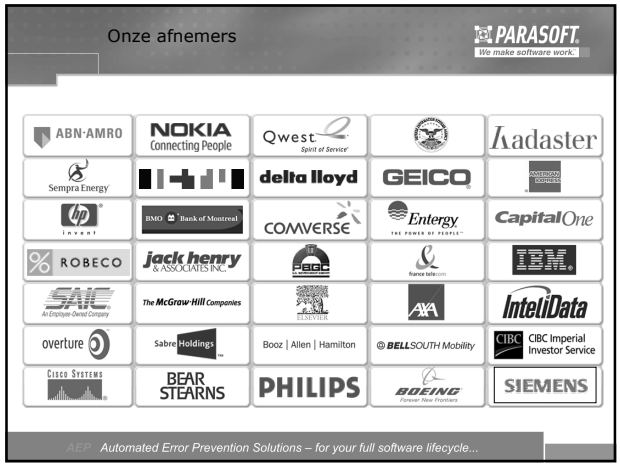

PARASOFT
We make software work.
Dashboard: Detailed View
Defects by Severity
Test Indicator by execution status
Code Base Size
Coding Standards
Unit Regression Tests
Coverage
Requirement progress by status
AEP Automated Error Prevention Solutions – for your full software lifecycle...

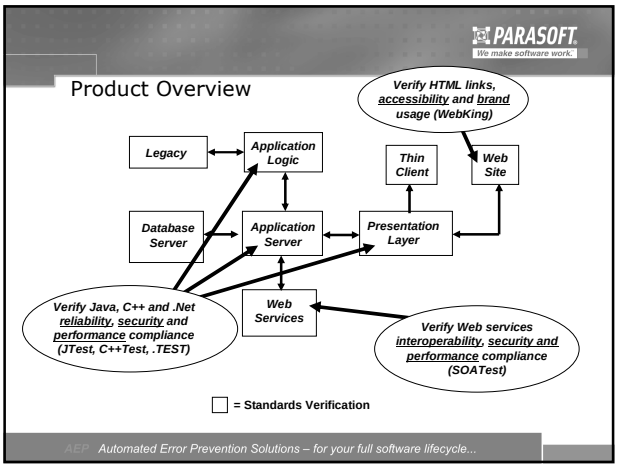

PARASOFT
We make software work.
Message layer testing – Workflow
1. Create, manage and collaborate on tests:
A) WSDL tests
B) Unit tests
C) Functional (Use case scenario) tests
D) Security tests
E) Performance and load tests
2. Automate with regression testing throughout the Web Services lifecycle
3. Report to management
AEP Automated Error Prevention Solutions – for your full software lifecycle...


PARASOFT
We make software work.
Message layer testing – Workflow
Build collaboration into the development process
Procedure | Team
1. Creating Directory Structure in Source Control | Sys Admin
2. Nightly Process Integration | Build Team
3. Error and Performance Analysis | Build/QA Team
4. Creating Unit Tests | Development Team
5. Creating Coverage and Scenario Tests | QA Team
6. Performing Load Tests | Load Team
7. Production Environment | Build Team
AEP Automated Error Prevention Solutions – for your full software lifecycle...


PARASOFT
We make software work.
Summary
New method for testing WebServices
Advantages:
Improved productivity and labour savings through auto generation of test cases
Accelerated time to market by leveraging test cases between developers, testers, QA people, and performance testing teams
Reduced overhead from creating and maintaining homegrown scripts or test harnesses
Reduced cost of maintaining tests going forward
Mitigated risk for business critical applications by expanding the breadth of current test processes
AEP Automated Error Prevention Solutions – for your full software lifecycle...


PARASOFT
We make software work.
Company; Corporate Background
Founded in 1987, privately held
300+ employees worldwide
Headquarters in Monrovia, CA
10,000 customers worldwide
ABN AMRO, AXA, Bank of America, Bloomberg, Boeing, Cisco, DCA, Disney, HP, IBM, Lehman Brothers, Lockheed, Northrop, Panasonic UK, Philips, P&O Ferries, Royal Bank of Scotland
Technical innovator
Fifteen US patents for software technology
AEP Automated Error Prevention Solutions – for your full software lifecycle...


Onze afnemers
PARASOFT
We make software work.
ABN·AMRO
NOKIA
Connecting People
Qwest
Kadaster
Sempra Energy
delta lloyd
GEICO
hp
invent
BMO Bank of Montreal
COMVERSE
Entergy
CapitalOne
ROBECO
jack henry & ASSOCIATES INC.
IBM
SAIC
The McGraw-Hill Companies
AXA
InteliData
overture
Sabre Holdings
Booz | Allen | Hamilton
BELLSOUTH Mobility
CIBC Imperial Investor Service
Cisco Systems
BEAR STEARNS
PHILIPS
BOEING
SIEMENS
AEP Automated Error Prevention Solutions – for your full software lifecycle...


PARASOFT
We make software work.
Product Overview
Verify HTML links, accessibility and brand usage (WebKing)
Legacy
Application Logic
Thin Client
Web Site
Database Server
Application Server
Presentation Layer
Web Services
Verify Java, C++ and .Net reliability, security and performance compliance (JTest, C++Test, .TEST)
Verify Web services interoperability, security and performance compliance (SOATest)
= Standards Verification
AEP Automated Error Prevention Solutions – for your full software lifecycle...