TTCN-3 and the Test Process

Graham Bath

T-Systems
Testfactory
Munich, Germany

Testfactory: The challenge we face

- Our main focus
  - Big, complex projects
  - Functional, non-functional test types
  - Business-oriented

- Pressure on costs

- The trend to offshoring

- The complexity of our systems
The challenge we face: an example

Our task:
- Ensure the operational readiness for the T-Com application stack
- 4 releases/year with new requirements and products
- Perform integration tests at system level

Test team
- 150 persons
- 5 locations

Key facts
- 100 business processes
- 20 locations
- 40,000 users
- 40,000,000 data records

- If something goes wrong here – we’re in trouble!

Why TTCN-3?
- Possible enabler for offshoring
- Evaluate and leverage new technologies
  - practical use in our testing projects
  - potential uses for our customers
- But first we need an understanding of where TTCN-3 fits into the test process.
  - TTCN-3 in isolation
  - Big picture
What I am going to talk about

Specifically:
- Experiences in integrating TTCN-3 into our test process
- Focus offshore projects

Generically:
- Where could TTCN-3 fit into the big picture?
- What role could TTCN-3 play in improving the test process?

Integrating TTCN-3 into our test process

- We set up a pilot project „Proof of Concept“
  - Realism is a very significant factor
- Team nominated
  - Andreas Braun, T-Systems: a Test Manager in T-Com Project
  - Brahim Sakout Andaloussi, T-Systems/TU-Munich: Tester
  - Bernd Bruegge, TU-Munich: Professor of Applied SW-Engineering
- We had an open mind regarding TTCN-3
- This is what came out >>>>>>>>>>>
Establish a context for the Test Process

Test Strategy (generic)

Offshore-enabled Test Process (generic)

Test Process

Tool Set & Methodology

Creating the offshore-enabled Test Process

Offshore-enabled Test Process (generic)

Master Test Plan

Design for Test

Test Specification

Test Execution

Reporting

Business Domain
- German language
- Requirements engineers
- Modelling experts
- Test Designers
- T-Systems, Testfactory, Munich

Execution Domain
- Programmers TTCN-3
- Automation experts
- T-Systems, Testfactory, India
Creating the offshore-enabled Test Process

Offshore-enabled Test Process (generic)

Master Test Plan

Design for Test
Test Specification
Test Definition
Test Execution
Reporting

Business Domain
- Requirements engineers
- Modelling experts
- Test Designers
- T-Systems, Testfactory, Munich

Execution Domain
- Programmers TTCN-3
- Automation experts
- T-Systems, Testfactory, India

Supporting the Test Process with a Tool Set

Offshore-enabled Test Process (generic)

Master Test Plan

Design for Test
Test Specification
Test Definition
Test Execution
Reporting

Tool Set & Methodology

Business Domain
- Germany

Execution Domain
- India
Establish a tool set that combines UML and TTCN-3 notations to obtain a systematic method for implementing an offshore-enabled test process for automated testing.
Overall approach: Design for Test

- Start with "Design for Test" using UML.

- Analyse the Test Basis (e.g. business requirements) and capture as:
  - Use Case Diagrams
  - Sequence Diagrams
  - Class Diagrams

- Use a mix of English description and application-specific elements in German.

Overall approach: Test specification

- Take over the UML diagrams and extend for test purposes using UML-2 Testing Profile (U2TP).

- The result is a Test Model (Test Specification) containing information about the System under Test (SUT) and each test case.

- The Test Model is used for
  - Test Definition (automated testing)
  - Test Execution (manual testing)
Test definition takes place if an automatic test execution is required.

Take over the U2TP-Models for each test case and implement (code) in TTCN-3.

Predefined mapping rules to link U2TP Model to TTCN-3.

Overall approach: Test execution

- Manual strategy
  - The Test Model (U2TP) and UML-documents are the basis for test execution.
  - Execution on-shore or off-shore depends on the language used in the documentation.

- Automated strategy
  - Use TTCN-3 automation tools
  - Off-shore enabled!
What we found

- The design stage prior to using TTCN-3 is critical.
- For the type of tests we do at the Testfactory, it is not enough to use TTCN-3 in isolation.
- TTCN-3 appears to have its strengths at lower test levels (component tests, component integration tests).
- UML2-TP is a good way to design high level tests,
- The manual implementation of TTCN-3 on the basis of UML2-TP works well, but could be more efficient with tool support (the non-commercial tools which we used were not good enough for our needs.)

Consequences on the test process

- A test process can be defined in which TTCN-3 is an integral element.
- A mix of manual and automated testing strategies is recommended.
- Benefits of automation with TTCN-3 appear to be more at lower level tests. These tests can, however, be provided to developers by the testing organisation as a basis for daily builds.
- UML2-TP can provide a good basis for manual testing off-shore. The break even curve for implementing automatic tests with TTCN-3 has shifted back!
Changes to the cost-benefit of automation

Accumulated cost

Test Executions

Break-Even

New Break-even is later

Manual Tests

Manual Tests UML2-TP with offshore

Automated Tests (TTCN-3)

Specifically:
- Experiences in integrating TTCN-3 into our test process
- Focus offshore projects

Generically:
- Where could TTCN-3 fit into the big picture?
- What role could TTCN-3 play in improving the test process?
Where does TTCN-3 „fit“

- We need a model of a test process
- The TPI-Model is the one we use
  - The TPI-Model is a registered trademark of Sogeti Netherland B.V.
  - The model is based on the practical experiences of testers
Structure of the TPI-Model.

The TPI-Matrix: All Key Areas - all Levels

<table>
<thead>
<tr>
<th>Key Area</th>
<th>Increasing maturity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Strategy</td>
<td>A</td>
</tr>
<tr>
<td>Life-Cycle Model</td>
<td>A</td>
</tr>
<tr>
<td>Momentum of involvement</td>
<td>A</td>
</tr>
<tr>
<td>Estimating and Planning</td>
<td>A</td>
</tr>
<tr>
<td>Test specification techniques</td>
<td>A</td>
</tr>
<tr>
<td>Static test techniques</td>
<td>A</td>
</tr>
<tr>
<td>Metrics</td>
<td>A</td>
</tr>
<tr>
<td>Test Tools</td>
<td>A</td>
</tr>
<tr>
<td>Test Environment</td>
<td>A</td>
</tr>
<tr>
<td>Office Environment</td>
<td>A</td>
</tr>
<tr>
<td>Commitment and motivation</td>
<td>A</td>
</tr>
<tr>
<td>Test functions and training</td>
<td>A</td>
</tr>
<tr>
<td>Scope of Methodology</td>
<td>A</td>
</tr>
<tr>
<td>Communication</td>
<td>A</td>
</tr>
<tr>
<td>Reporting</td>
<td>A</td>
</tr>
<tr>
<td>Defect Management</td>
<td>A</td>
</tr>
<tr>
<td>Technology Management</td>
<td>A</td>
</tr>
<tr>
<td>Test Process Management</td>
<td>A</td>
</tr>
<tr>
<td>Evaluation</td>
<td>A</td>
</tr>
<tr>
<td>Low-Level Testing</td>
<td>A</td>
</tr>
</tbody>
</table>

Increasing maturity from A to D indicates an increase in maturity.
## Example

### Required test maturity for the organisation

<table>
<thead>
<tr>
<th>Kerngebiet</th>
<th>Controlled</th>
<th>Efficient</th>
<th>Optimising</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Teststrategie</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>2. Phasenmodell</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>3. Zeitpunkt der Beteiligung</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>4. Schätzung und Planung</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>5. Testfahndungsverfahren</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>6. Statistische Testmethodik</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>7. Metriken</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>8. Testtools</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>9. Testumgebung</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>10. Testumgebungstäfel</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>11. Engagement und Motivation</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>12. Testfahrung und Ausbildung</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>13. Geheime der Methodik</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>14. Kommunikation</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>15. Berichtswesen</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>16. Fehlermanagement</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>17. Testprozessmanagement</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>18. Testprozessmanagement</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>19. Pflege</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>20. Low-Level Testing</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>21. Integration</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

## Example: Results

### Target achieved

- **A**
- **B**
- **C**
- **D**

### Need for improvement

- **A**
- **B**
- **C**
- **D**

### Main problem areas

- **A**
- **B**
- **C**
- **D**
Influence of TTCN-3 on Test Process Improvement

Using TTCN-3 will help achieve these testing maturity levels.

<table>
<thead>
<tr>
<th>Key Area</th>
<th>Increasing maturity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Strategy</td>
<td>A B C D</td>
</tr>
<tr>
<td>Life-Cycle Model A</td>
<td>B C D</td>
</tr>
<tr>
<td>Process of Involvement A</td>
<td>B C D</td>
</tr>
<tr>
<td>Estimating and Planning A</td>
<td>B C D</td>
</tr>
<tr>
<td>Test Specification Techniques A</td>
<td>B C D</td>
</tr>
<tr>
<td>Static Test Techniques A</td>
<td>B C D</td>
</tr>
<tr>
<td>Metrics A</td>
<td>B C D</td>
</tr>
<tr>
<td>Test Tools A</td>
<td>B C D</td>
</tr>
<tr>
<td>Environment A</td>
<td>B C D</td>
</tr>
<tr>
<td>Commitment and motivation A</td>
<td>B C D</td>
</tr>
<tr>
<td>Test Functions and Training A</td>
<td>B C D</td>
</tr>
<tr>
<td>Scope of Methodology A</td>
<td>B C D</td>
</tr>
<tr>
<td>Communication A</td>
<td>B C D</td>
</tr>
<tr>
<td>Reporting A</td>
<td>B C D</td>
</tr>
<tr>
<td>Defect Management A</td>
<td>B C D</td>
</tr>
<tr>
<td>Test Management A</td>
<td>B C D</td>
</tr>
<tr>
<td>Test Process Management A</td>
<td>B C D</td>
</tr>
<tr>
<td>Evaluation A</td>
<td>B C D</td>
</tr>
<tr>
<td>Low-Level Testing A</td>
<td>B C D</td>
</tr>
</tbody>
</table>

Influence of TTCN-3 on improving Test Strategy

- Mature test strategies include specific strategies for a number of testing issues, including regression testing.

- The use of TTCN-3 can be considered to be an „enabling“ technology for modern, mature regression testing strategies.

- In particular, TTCN-3 enables daily builds to be implemented for development staff and can strengthen the co-operation between tester and developer.
Influence of TTCN-3 on improving Test Specifications

- More mature test processes require that test specifications are made reusable within the test team by means of a uniform working method.

- TTCN-3 implements a "uniform working method" by which several testers can share test specifications.

Influence of TTCN-3 on better Test Tool Use

- More mature test processes require that test tools can be reused for a future test projects.

- Furthermore, the use of a test tool must not result in inefficiency or undesired limitations being placed on the test process.

- TTCN-3 fulfills both of these criteria.
Influence of TTCN-3 on better Low-Level Testing

- More mature test processes require that low-level tests have a recognizable planning, design and execution activities.

- TTCN-3 is an „enabler“ which permits these test process objectives (which are frequently absent!) to be achieved.

Summary of main points

- TTCN-3 has to be seen in the context of an overall test process.

- T-Systems, Testfactory has evaluated how to integrate TTCN-3 into an offshore-enabled test process for high-level tests (further pilot studies are planned).

- TTCN-3 can be instrumental in improving the test process.

- Specific improvements can be identified within the TPI-model.