TestCase Design: a practical guide to building the most powerful test cases

Tutorium – TestNet Summer School 2011

July 13th, 2011 | Wolfgang Platz

Agenda | proposal

Block I – 9:00 – 10:45 … Problem definition & methodological basis
- An intro to the participants and TRICENTIS®
- Starting situation – presentation of the problem
- Methodological basis of the TestCase and test data design

<Break>

Block II – 11:00 – 13:00 … TestCase and test data design
- Example: fictitious task from the banking sector
- Summary
About TRICENTIS®

Easy Testmanagement & -automation

Methodology/Conceptual
- Test Planning
- Test Case Finding/Design
- Test Case Automation
- Test Execution/-management

Services
- Consulting
- Trainings
- Project Support
- Test as a Managed Service

Solutions
- TOSCA Commander™: Test Management
- TOSCA Adapter: Test Automation
- TOSCA TestCase Design: Test Data Generation

Founded in: 1997
Headquarters: Vienna
Mitarbeiter: 130+
Customers: 220+
Operating in: Austria, Germany, Switzerland, Netherlands, UK, Australia/NZ, USA

Wolfgang Platz

Education
- Graduated with honors in Technical Physics at the University of Technology in Vienna
- Studies in Business management, Law and Economics
- Certified management consultant

Career
- 1997 TRICENTIS Technology & Consulting GmbH (Founder, CEO)
- 1995 - 1997 Cap Gemini Ernst & Young (Consultant)
- 1992 - 1995 VKI Verein für Konsumenten-Information (Developer)

Relevant project experience
- Conception, coordination and support with implementing the dynamic synthetic test data concept TOSCA@data in the banking sector
- Test concept, TestCase-Design, test management at about 30 different customers in the following sectors: insurance, banking, telcos, stock exchange, commerce, energy, industry, etc. (conception, moderation and management) of international range and importance
- Technical & business-based concept, architecture and implementation of the TOSCA Explorer (predecessor product of the newly developed TOSCA Testsuite™)
- Technical & business-based concept, architecture and implementation of the product definition machine of one of the world’s biggest insurance companies (project manager, Chief Architect)
**Efforts in test projects**

- Total effort: 100%
- Test Case Specification: 0%
- Manual Test (Execution): 50%
- Automated Test (Setup, maintenance and execution): 50%

Source: Customer projects TRICENTIS®

---

**Increased test-coverage**

- Test coverage: > 90%
- Test Case Design: 0%
- Automated Test: 100%

Source: Customer projects TRICENTIS®
Number of test cases and coverage

**Number of test cases**

- **Conventional approaches**: Exponential increase even at low coverage ranges!
- **TOSCA Test Case Design**: Linear increase up to near 100% coverage

Requirements for the test

"We expect a reliable statement on whether our new software (version) can go into production."

What are the requirements for this statement?

- Simple and comprehensive
- Traceable and verifiable at all times
- Reliable
Questions in the test process

Which basic questions should the test process answer?

Test preparation
- How is the test preparation progressing?

Test execution
- How much have we already managed to test?
- What level of quality have we achieved?
- What is the status of test execution?
- How many errors are there?

Test coverage

Test coverage – the key indicator

Test coverage [%] states the extent to which the entire functionality under test can be or was tested.

Absolute test coverage
- The quotient of the functions tested and all the testing functions
- All the functions under test are assessed in the same way.

Relative test coverage
- Risk assessment of all the functions under test
- Test coverage = sum of risk weights of the functions tested

Task 1: We must know the value of the functions under test!
Risk weighting level by level

Risk weight = frequency x expected damage

Weighting is done in levels from the top down to the basic function.

The contribution to the entire risk is determined for each function.

Task 2: We need test cases for the functions.
Measurable statements | Conclusion

Example of functional structure for a DWH/BI-application at Austrian Bank, extract from TOSCA Testsuite™

Why doing TestCaseDesign?

How can you test efficiently to minimize testing efforts but maximize testing results?
How can you test efficiently to minimize testing efforts but maximize testing results?

- highest possible test coverage
- little effort in error analysis
- minimize redundant test case errors

Example | Annual rate of a vehicle insurance

- N.A.
- <18
- 18..23
- >59

- +20% (above 18)
- -10% (18..23)
- -5% (above 59)
How do we combine age and gender?

Select Case age
Case Is < 18
break;
Case 18 To 23
factor_age=1.2
Case 24 To 59
factor_age=1.0
Case Is > 59
factor_age=0.9
End Select

Select Case gender
Case "male"
factor_gender=1.0
Case "female"
factor_gender=0.95
End Select

How do we combine age and gender?
What about additional attributes?

age
- <18
- 18..23
- 24..59
- >59

gender
- m
- w

location
- city
- land

Test Case #1: <18 m city
Test Case #2: 18..23 m city
Test Case #3: 24..59 m city
Test Case #4: >59 m city

Test Case #1: <18 m land
Test Case #2: 18..23 m land
Test Case #3: 24..59 m land
Test Case #4: >59 m land

What about additional attributes?
Example | Extension of the topic

N.A. | +20% | >59
<18 | 18..23 | -10%
>59 | >64 | -5%

Extension of the topic!

Select Case age
Case Is < 18 'no driver’s license
Case 18 To 23 'young driver’s license
Case 24 To 59 'normal
Case 60 To 64 'senior discount; male <> female
Select Case gender
Case "male"
factor_age = 1.0
Case "female"
factor_age = 0.9
End Select
Case Is > 64 'senior discount; all
factor_age = 0.9
End Select

TestCase #1
18..23
m
city
land

TestCase #2
24..59
m
city

TestCase #3
60..64
w
city

TestCase #4
60..64
m
land

TestCase #5
>64
w

TestCase #6
>64
m
land
Boundaries

age

<18

18..23

24..59

>59

gender

m

w

Boundary values?

Equivalence partitioning

Should we care about the boundary values?

<1

1..99

>99
Example | withdrawing money at an ATM

if (amount ≤ 400.00) then Accept
if (amount < 400.01) then Accept

Orthogonal combinatorics | weaknesses

Orthogonality is of highest importance to the attributes (especially with business applications)

- Most attributes are orthogonal.
- If orthogonality does not apply, it usually only refers to some equivalence classes of attributes, but not to all of its instances.

However if applied consequently to too many attributes, it shows substantial weaknesses:

- The verification of results is not provided - different combinations could possibly show identical results.
- There is a higher probability of errors with increasing complexity - this decreases the probability that TestCases are executed completely.
- Error analysis increases in complexity.
Cardgame

- basic monthly fee
- bonus points
- mobile phone

Ortho
Ortho

- number of testcases
  - principle of cardinality = 4
  - all combinations: $4 \cdot 4 \cdot 3 = 48$
- ambiguous hotspot (focus)
- invalid instances?
- boundary values?
Straight Through criteria...

1. Risk
   Maximum risk means maximum coverage!

2. Ease
   Minimal functionality means quick implementation!

3. The possibility of combination
   It must ideally apply to many possible instances of other attributes!

Linear Expansion
Linear Expansion

Keep it simple!
- Straight through
- Linear Expansion
- More TestCases, but easier to manage

Number of TestCases:
- \(1 + \sum (\text{Cardinalities} - 1)\)
- Cardinalities are also **summed**, not multiplied
  (thus **NO** explosion of the solution space)

---

Linear Expansion

<table>
<thead>
<tr>
<th>age</th>
<th>Instance Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;17</td>
<td>0%</td>
</tr>
<tr>
<td>=17</td>
<td>0</td>
</tr>
<tr>
<td>=18</td>
<td>0</td>
</tr>
<tr>
<td>19..23</td>
<td>0</td>
</tr>
<tr>
<td>24..59</td>
<td>0</td>
</tr>
<tr>
<td>&gt;59</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>gender</th>
<th>Instance Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>0%</td>
</tr>
<tr>
<td>w</td>
<td>0</td>
</tr>
<tr>
<td>age</td>
<td>Instance Coverage</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------</td>
</tr>
<tr>
<td>&lt;17</td>
<td>0</td>
</tr>
<tr>
<td>=17</td>
<td>0</td>
</tr>
<tr>
<td>=18</td>
<td>0</td>
</tr>
<tr>
<td>19..23</td>
<td>0</td>
</tr>
<tr>
<td>24..59</td>
<td>1</td>
</tr>
<tr>
<td>&gt;59</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>gender</th>
<th>Instance Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>1</td>
</tr>
<tr>
<td>w</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>age</th>
<th>Instance Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;17</td>
<td>0</td>
</tr>
<tr>
<td>=17</td>
<td>0</td>
</tr>
<tr>
<td>=18</td>
<td>0</td>
</tr>
<tr>
<td>19..23</td>
<td>0</td>
</tr>
<tr>
<td>24..59</td>
<td>1</td>
</tr>
<tr>
<td>&gt;59</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>gender</th>
<th>Instance Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>1</td>
</tr>
<tr>
<td>w</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>age</th>
<th>Instance Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;17</td>
<td>0</td>
</tr>
<tr>
<td>=17</td>
<td>0</td>
</tr>
<tr>
<td>=18</td>
<td>0</td>
</tr>
<tr>
<td>19..23</td>
<td>0</td>
</tr>
<tr>
<td>24..59</td>
<td>1</td>
</tr>
<tr>
<td>&gt;59</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>gender</th>
<th>Instance Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>1</td>
</tr>
<tr>
<td>w</td>
<td>0</td>
</tr>
<tr>
<td>Block I – 9:00 – 10:45</td>
<td>Problem definition &amp; methodological basis</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>An intro to the participants and TRICENTIS®</td>
</tr>
<tr>
<td></td>
<td>Starting situation – presentation of the problem</td>
</tr>
<tr>
<td></td>
<td>Methodological basis of the TestCase and test data design</td>
</tr>
</tbody>
</table>

*Break*

<table>
<thead>
<tr>
<th>Block II – 11:00 – 13:00</th>
<th>TestCase and test data design</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Example: fictitious task from the banking sector</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
</tr>
</tbody>
</table>

- TestCase-Design in practice (exercises in small groups, chapter 1.2 of accompanying document)
- Documentation of a sample solution in TOSCA Testsuite™
Workshops, which include business analysts (BA), are the most efficient types of TestCase-Design.

- Minimum 1 day and maximum 2 days en suite
- Maximum 1 workshop per week for the participants

Composition of the workshops

- Moderator
  - Methodological skills ++
  - Abstraction capability ++
  - Tool expertise ++

- BA 1
- Testing expert

Would you like to learn more about methodical testcase design?
Would you like to elaborate and extend your knowledge even more about what you learned today and put it to work in your own environment?

Please contact:

Ernst Jan Smit
Sales Manager Benelux
+31 6 11 533103
e.smit@tricentis.com
Thank you for your attention!