Increasing the tester’s creativity – a scientific approach

Adam Roman
4it.team
Jagiellonian University
adangroman.pl@gmail.com
www.4it.team

#EuroSTARCConf
Agenda

1. Motivation
2. TQED Model
3. Example
4. The Model Works – Experimental Model Validation
I. Motivation
Have you noticed that...

test management, test organization, test process, managing the test team, test tools, test planning, writing the test plan, writing the bug reports, DevOps, agile, IoT, cloud, Scrum, V-model...
Have you noticed that...

test management, test organization, test process, managing the test team, test tools, test planning, writing the test plan, writing the bug reports, DevOps, agile, IoT, cloud, Scrum, V-model...

but what about TESTING? isn’t it the most important tester’s task?

not so many resources on how to test, how to design good tests/test ideas...

... because it is difficult (!) – each project is different, there are different „testing problems” and so on...
It still remains a scandal to philosophy and to human reason in general that the existence of things outside us (...) must be accepted merely on faith, and that if anyone thinks good to doubt their existence, we are unable to counter his doubts by any satisfactory proof.

*Immanuel Kant, Critique of Pure Reason*
The scandal in software testing

It still remains a scandal to testing and to tester’s reason that in our community people talk so much about things around testing, like test organization, management and automation, but they barely talk about testing itself.

Adam Roman
The proposal

A general model for boosting the tester’s creativity in finding good test ideas
II. The TQED Model
What is a model?

A model is a representation of an idea, an object or even a process or a system that is used to describe and explain phenomena that cannot be experienced directly.

Models are a mentally visual way of linking theory with experiment, and they guide research by being simplified representations of an imagined reality that enable predictions to be developed and tested by experiment.
What is a model for us?

A model is a representation of software that is used to describe and explain bugs that cannot be experienced directly.

Models are a mentally visual way of linking development with execution, and they guide testing by being simplified representations of an imagined reality that enable bug ideas to be developed and tested by testing (= to predict real bugs in software with high probability).
A good model...

- models the problem domain correctly
- is simple
- is understandable (speaks the right language)
- is flexible (can be used in many ways)
- is generic (can be used in many situations)
- allows us to make predictions

How to verify the model’s appropriateness?
By using the **scientific method**!
That is, by verifying the hypotheses by experimentation
How to describe software?
An inspiration from physics

so: data is like matter, events – like thermodynamics describing transformations of the physical objects, quantity – like the amount of substance and time – is just a good, old time known from physics ☺
<table>
<thead>
<tr>
<th>DATA + DATA</th>
<th>DATA + EVENT</th>
<th>DATA + EVENT</th>
<th>DATA + QUANTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVENT</td>
<td>EVENT</td>
<td>EVENT</td>
<td>QUANTITY</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>TIME</td>
<td>QUANTITY</td>
</tr>
<tr>
<td>QUANTITY</td>
<td>QUANTITY</td>
<td>QUANTITY</td>
<td>QUANTITY</td>
</tr>
</tbody>
</table>
Model dimensions

• straightforward, intuitive, easy to understand

• for example:
  • **Data**: input data, browser’s cookie, file name, configuration element, document, database record etc.
  • **Event**: creating a variable, invoking a method, pressing a button, renaming the file, fulfilling some rule/property etc.
  • **Quantity**: empty set, below minimal value, average value, over the maximum, on a domain boundary, just below the boundary etc.
III. Example
Simple example: a ticket machine

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Real entity mapped to this dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>D</strong></td>
<td>normal/reduced tickets to buy, ticket cards in the machine, coins/banknotes of different denominations in the machine, inserted coins/banknotes of different denominations, total price; valid and invalid coins/banknotes</td>
</tr>
<tr>
<td><strong>E</strong></td>
<td>pressing a button, giving a change, inserting a coin/banknote, printing a ticket, power on/off, cancel the operation</td>
</tr>
<tr>
<td><strong>T</strong></td>
<td>long/short/typical period of time between actions/process</td>
</tr>
<tr>
<td><strong>Q</strong></td>
<td>low or high value, boundary or close to the boundary, maximal possible, minimal possible</td>
</tr>
</tbody>
</table>
Example 1: creating the test ideas

**Data:**
- denomination

**Quantity:**
- boundary
  - value

**Data:**
- total price
Example 1: creating the test ideas

**Data:** denomination

**D+Q**

**Quantity:** boundary value

**Data:** total price

**PARTIAL IDEAS**

use coins with minimal denomination (5c)
Example 1: creating the test ideas

**Data**: denomination

**Data**: total price

**Quantity**: boundary value

**PARTIAL IDEAS**
- Use coins with minimal denomination (5c)
- Buy for the maximal possible price
Example 1: creating the test ideas

**Data:** denomination

**Quanity:** boundary value

**Data:** total price

**PARTIAL IDEAS**

use coins with minimal denomination (5c)

buy for the maximal possible price

**RESULTING TEST IDEA:**

Insert only the coins with the minimal denomination (5c) for the highest possible price ($38) – will the machine accept such a large number of coins?
Example 2: an emerged design

Event: pressing a button

Event: Cancel

Quantity: a lot of

Time: a long process

E+Q

Buying a lot of tickets
Example 2: an emerged design

Event: pressing a button
Quantity: a lot of
Time: a long process

Event: Cancel
Quantity: a lot of
Time: repeating the whole process again...

E+Q
E+E+Q
E+Q
Example 2: an emerged design

Event: pressing a button

Event: Cancel

Quanity: a lot of

Time: a long process

how to define ‘Cancel’ operation? Cancel only the last ticket or cancel everything???

repeating the whole process again...

This issue is about the SYSTEM DESIGN (UX, usability)!
IV. The Model Works – Experimental Model Validation
Research questions

• How can we know that the model works?
• Let’s do the controlled experiment!
• Research hypothesis: using the TQED model increases the tester’s creativity and allows us to design more effective test ideas (= with higher probability of detecting the failures)

• Problem: how to define creativeness?...
Experimental setup

- TQED group
- Control group

**Test ideas**: Each test idea was evaluated by a so-called **CREATIVITY FACTOR** – an average expert’s evaluation on scale 1-4
  - 1 = trivial, 4 = very creative

**Test ideas evaluation (experts do not know to which group a given test idea belongs)**

**Comparison**
- Evaluation of TQED tester’s creativity
- Evaluation of control group tester’s creativity
### Characterization of the groups

<table>
<thead>
<tr>
<th>Factors (characteristics)</th>
<th>Mean for TQED group</th>
<th>Mean for control group</th>
</tr>
</thead>
<tbody>
<tr>
<td># of test ideas for Experiment A</td>
<td>15.52</td>
<td>14.37</td>
</tr>
<tr>
<td># of test ideas for Experiment B</td>
<td>12.47</td>
<td>12.87</td>
</tr>
<tr>
<td>Average mark from the studies (2.0-5.0)</td>
<td>3.74</td>
<td>3.91</td>
</tr>
<tr>
<td>Professional IT experience (months)</td>
<td>2.85</td>
<td>4.84</td>
</tr>
<tr>
<td>Testing knowledge (1-5)</td>
<td>2.58</td>
<td>2.93</td>
</tr>
<tr>
<td>Programming skills (1-5)</td>
<td>3.82</td>
<td>4.00</td>
</tr>
<tr>
<td>Analytical skills (1-5)</td>
<td>3.41</td>
<td>3.25</td>
</tr>
</tbody>
</table>

All the differences between groups for all the factors were statistically insignificant (p-value > 0.2), which means that both groups are comparable regarding these factors.
Two experiments

<table>
<thead>
<tr>
<th>Experiment →</th>
<th>Parking gate</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>type of system →</td>
<td>physical system</td>
<td>software system</td>
</tr>
<tr>
<td>focus on →</td>
<td>behavioral aspect</td>
<td>business logic</td>
</tr>
<tr>
<td>interaction →</td>
<td>user-centered</td>
<td>'back-end' oriented</td>
</tr>
</tbody>
</table>
Parking gate (simplified example)

- A driver comes at the lowered gate, pushes the button
- Receives a ticket, the gate starts raising, enter time is logged
- When gate at top position, the car drives in
- Sensor carJustEntered changes to true, gate is lowered
- When a driver leaves the parking, comes at the exit gate
- System calculates the fee according to enter time (hh:mm) and exit time (hh:mm); first hour is for free; 2 EUR for each consecutive one

Task: provide the „test ideas” for this system
The main result

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Mean creativity factor for TQED tester</th>
<th>Mean creativity factor for control group tester</th>
<th>p-value for t-test (statistical significance)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment A (Parking gate)</td>
<td>31,28</td>
<td>25,31</td>
<td>0,07</td>
</tr>
<tr>
<td>Experiment B (Library)</td>
<td>22,13</td>
<td>17,29</td>
<td>0,1</td>
</tr>
</tbody>
</table>

The result of Experiment A is statistically significant
The result of Experiment B is statistically significant as well (however, on the boundary...)

Significance level: 0,1
The main result – cont’d

Number of unique test ideas

<table>
<thead>
<tr>
<th>TQED group</th>
<th>control group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parking Gate</td>
<td>28</td>
</tr>
<tr>
<td>Library</td>
<td>23</td>
</tr>
</tbody>
</table>

Mean value of the creativity factor

<table>
<thead>
<tr>
<th>TQED group</th>
<th>control group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parking Gate</td>
<td>2.23</td>
</tr>
<tr>
<td>Library</td>
<td>1.80</td>
</tr>
</tbody>
</table>
Sample test ideas in the TQED group

- take a ticket, don’t drive in, return the ticket at the exit gate
- drive in with a very long vehicle
- parking time so long that it can’t fit on the machine screen
- simulation of fire/smoke (will the gate open automatically?)
- contactless payment by using a card few times
- payment with a card with a different currency
- entering the ticket while other car exits
- etc.
Main results - conclusions

• Using TQED seems to increase the tester’s creativity
  • the results are statistically significant with significance level $\alpha = 0.1$

• Creativity was understood as, among others, the ability to design tests with high probability of failure detection

• So, we can argue that using TQED = creating better tests
Threats to validity

• small groups of participants ($N_{TQED} = 17$, $N_{control} = 16$, $N_{experts} = 13$)
• participants were students, not the experienced testers
• TQED group had to study the TQED model for 1h at home, before the experiment, but this couldn’t be verified
• creativity evaluated by humans (but maybe it’s a good thing…?)
• the notion of creativity is not precise – in particular, it is not equal to the notion of „high probability of the failure detection“! (although this characteristic was given in the „definition“ of creativity for the experts)
More details...

• ... about the TQED model and its applications
• ... but also, in general, about thinking in testing

Adam Roman

Thinking-Driven Testing. The Most Reasonable Approach to Quality Control

Springer, 2018
Final remarks

• We presented the TQED model
• It is universal, simple, effective and useful (scientifically verified), but...
• this is definitely not any kind of rocket science!
  • the model just combines some well-known pieces and puts them together
• the model cannot replace thinking
  • it is just an 'intellectual framework', a tool that can boost a tester’s creativity, but it won’t do it without the tester’s analytical skills!
• models don’t think – it’s humans who does it
• using the scientific method we confirmed that the model works
Final remarks

• We presented the TQED model.
• It is universal, simple, effective and useful (scientifically verified), but...
• this is definitely not any kind of rocket science!
  • the model just combines some well-known pieces and puts them together
• the model cannot replace thinking
  • it is just an ‘intellectual framework’, a tool that can boost a tester’s creativity, but it won’t do it without the tester’s analytical skills!
• models don’t think – it’s humans who do it
• using a scientific method we confirmed that the model works
Thank you for your attention!

Any questions?

(„science is interesting if you don’t agree” – Richard Dawkins)
Thank You!

Adam Roman
4it.team
Jagiellonian University

adamroman.pl@gmail.com
www.4it.team

#EuroSTARConf