Testing
Red Hat Enterprise Linux
the Microsoft way

Alexander Todorov

FOSDEM 2018
Kiwi TCMS
the leading open source test case management system

- Efficiently manage test cases, plans and runs
- Improve testing productivity & reporting
- Integrates with popular issue trackers
- External API interface
- GPL 2 licensed

Open source test case management system, with a lot of great features, such as bug tracker integration, fast search, powerful access control and external API.

Everyday testing
Use the dashboard to see pending work. Execute tests, mark results and report bugs. Collect automation results!

Test management
Define test plans and cases, track progress and assign work across multiple teams. Perform peer reviews.

Patternfly & jQuery

Reporting
See who's doing what and provide status report to stakeholders. Centralize your acceptance books!

Docker

Integration
Integrated with Bugzilla, JIRA and GitHub. The API interface provides full access so you can get creative!

Tested
What is pairwise testing
• Wheels
  – 19” or 21”
• Battery (if you remember)
  – 60 kWh, 75 kWh, 85 kWh or 100 kWh
• Engine
  – Single or Dual
• Performance mode
  – Yes or No
2 x 2 x 2 x 4 == 32 combinations
<table>
<thead>
<tr>
<th>Wheels</th>
<th>Battery</th>
<th>Engine</th>
<th>Performance mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>19 &quot;</td>
<td>60 kWh</td>
<td>Single</td>
<td>Yes</td>
</tr>
<tr>
<td>19 &quot;</td>
<td>75 kWh</td>
<td>Single</td>
<td>Yes</td>
</tr>
<tr>
<td>19 &quot;</td>
<td>85 kWh</td>
<td>Single</td>
<td>Yes</td>
</tr>
<tr>
<td>19 &quot;</td>
<td>100 kWh</td>
<td>Single</td>
<td>Yes</td>
</tr>
<tr>
<td>21 &quot;</td>
<td>60 kWh</td>
<td>Dual</td>
<td>No</td>
</tr>
<tr>
<td>21 &quot;</td>
<td>75 kWh</td>
<td>Dual</td>
<td>No</td>
</tr>
<tr>
<td>21 &quot;</td>
<td>85 kWh</td>
<td>Dual</td>
<td>No</td>
</tr>
<tr>
<td>21 &quot;</td>
<td>100 kWh</td>
<td>Dual</td>
<td>No</td>
</tr>
</tbody>
</table>
Pairwise Testing

Combinatorial Test Case Generation

Pairwise (a.k.a. all-pairs) testing is an effective test case generation technique that is based on the observation that most faults are caused by interactions of at most two factors. Pairwise-generated test suites cover all combinations of two therefore are much smaller than exhaustive ones yet still very effective in finding defects. Learn more

$L_4(2^3)$ Orthogonal Array

<table>
<thead>
<tr>
<th>Expt. No.</th>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

★ What's New
Effectiveness of Pairwise
Research Papers
Available Tools
About this Website

Maintained by Jacek Czerwonka, Last updated: October 2016
I've pairwise tested Red Hat Enterprise Linux install during the entire test campaign! across all product variants!
Installation testing 101
INSTALLATION SUMMARY

LOCALIZATION

DATE & TIME
Americas/New York timezone

LANGUAGE SUPPORT
English (United States)

SOFTWARE

INSTALLATION SOURCE
http://download.lab.b...tion/x86_64/os

SOFTWARE SELECTION
GNOME Desktop

SYSTEM

INSTALLATION DESTINATION
Automatic partitioning selected

NETWORK & HOSTNAME
Not connected

We won't touch your disks until you click ‘Begin Installation’.

Please complete items marked with this icon before continuing to the next step.
New Red Hat Enterprise Linux 7.1 Installation

DATA

/system 476 MiB

/ 4420 MiB

swap 1020 MiB

vda2

Mount Point:

Device(s):
Virtio Block Device (vda) and 1 other

Desired Capacity:

4420 MiB

Device Type:

RAID

File System:

xfs

Label:

Note: The settings you make on this screen will not be applied until you click on the main menu’s ‘Begin Installation’ button.
9 different product variants

I consider them platform independent
3 test groups: Tier #1, #2 and #3
6000 test case executions
“Insanity - doing the same thing over and over and expecting different results.”

Albert Einstein
1) Take all platform dependent tests (pairwise where possible)
2) Pairwise all tests with parameters
storage / iSCSI / No authentication / Network init script

storage / iSCSI / CHAP authentication / Network Manager

storage / iSCSI / Reverse CHAP authentication / Network

• **Authentication type:** None, CHAP, reverse CHAP (3)
• **Networking system:** NetworkManager or SysVinit (2)
• $3 \times 2 == 6$
• *Pairwise*: $3 \times 2 == 6$
• Across all variants: $9 \times 3 \times 2 == 54$
• *Pairwise across all variants*: $9 \times 3 == 27$
3) Randomize tests without parameters
Partitioning / swap on LVM

• No parameters!
• Pairwise can't reduce variant as parameter
  - $9 \times 1 = 9$
• Execute on random product variant each time!
Acceptance criteria
Less test case executions
Don't miss existing bugs *

how does pairwise compare to full test suite wrt defect finding abilities ?
Don't increase product risk *

how many critical defects would I miss if I don't execute the full test suite?
Experiment results
65 % less test case executions!

2119 test cases in pairwise test plan
76 % execution completion rate

previous releases are around 85%
3 x 30% bug discovery rate
30 % of bugs found by Tier #1

good job, test cases not included in experiment
30% of bugs found by Pairwise

same were detected by following regular test plan
30 % of bugs found by ET

we don't have test cases for them! Ouch!
Pairwise missed 4 critical bugs, 3 were regressions.
• #1396949 - After installation with ibft the default route is missing
  – gPXE, firmware dependent

• #1421039 - Anaconda fails to get kickstart from nfs on s390x
  – Corner case on s390x
  – IPv6 != IPv4
• #1400844 - Interface binding makes iscsi connection fail
  - Waived due to bad infrastructure setup
  - Waived again b/c ComputeNode doesn't support Specialized Storage

• #1420300 - Certmonger scriptlet outputs errors to update.log during anaconda upgrade
  - tested and not being re-tested
Lessons learned
WILL CODE FOR FOOD
Perform test review regularly

found hidden parameters in tests

found (sort of) duplicate test cases
Observed optimization patterns

combine or pipeline independent TCs

common set-up for multiple TCs across variants

... and pairwise, pairwise, pairwise
Risk of not detecting regressions

risk is significant in Snapshots phase

due to historical aggregation of results
Ask me anything!

@atodorov_

http://atodorov.org

atodorov@redhat.com