Orchestrating Your Testing Process

Coordinating the manual & automated tests in your project

Joel Montveliskiy

PNSQC 2020
About me

QA / Testing
(last 23 years)

- Tester
- QA Manager
- Blogger / Podcaster
- Speaker
- Consultant
- Solution Architect

PractiTest

- Chief Solution Architect

Other stuff

- State of Testing™
- OnlineTestConference

Joel Montvelisky
Follow: @joelmonte
Manual vs Automated Testing
Manual vs Automated Testing
What are the challenges of managing manual & automated tests together?

Why is this interesting **Right Now?**
EVOLUTION OF THE BICYCLE
FROM 1817 TO 2000

1817
DRAISINE

1830
TRICYCLE VELOCIPEDE

1870
HIGH-WHEEL BICYCLE

1885
SAFETY BICYCLE

1960
RACING BIKE

2000
MOUNTAIN BIKE

Orchestrating Your Testing Process
Changes in Automation Adoption and Tools

- More tools and also more diverse
- Less expensive
- Easier to use
- Automated tests are becoming more robust
Changes in Dev & Testing Processes

- Agile - shorter cycle requiring constant stability
- CI - “Clever” idea of constantly gilding and testing the system
- DevOps - laser-focused testing, reducing the cost of *some* bugs in production, shifting to whole team testing
Changes in Players

- Less testers within each team
- Testers are more technical
- Developers are taking a more active role in testing
- Other teams taking part in testing
More & Better Automated Testing

Tools

Players

Process
What are the differences between manual & automated tests?

Actually, quite a lot...
Purpose of the Tests

**Manual**
- Evaluate the product to ensure desired quality, value, fit, etc
- Find bugs and areas to improve

**Automated**
- Maintain the product’s functional and non-functional stability
- Detect changes that may point to issues
## Nature of the Tests

<table>
<thead>
<tr>
<th>Manual</th>
<th>Automated</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Relatively Inexpensive to write Expensive to run</td>
<td>• Relatively Expensive to write Inexpensive to run</td>
</tr>
<tr>
<td>• Long run duration</td>
<td>• Short run duration</td>
</tr>
<tr>
<td>• Results depend directly on the tester</td>
<td>• Results are independent from the tester</td>
</tr>
</tbody>
</table>
## Runs and results

<table>
<thead>
<tr>
<th>Manual</th>
<th>Automated</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Small number of runs &amp; repetitions</td>
<td>● Large number of runs and repetitions</td>
</tr>
<tr>
<td>● Tests are distributed among team</td>
<td>● Tests are run from frameworks</td>
</tr>
<tr>
<td>● Interest resides in direct results - bugs</td>
<td>● Interest resides in trends &amp; benchmarks</td>
</tr>
</tbody>
</table>
One common misconception about automated testing projects
In theory

Test Planning → Manual Test → Automated Test → Runs Forever → Magically Reports
In practice

Existing project → Legacy Manual Tests → Manual Test Cleanup → Isolated Manual Test Reports

Automation POC 1 → Automation Framework 1 → Isolated Outputs → Isolated Automated Test Reports A

Automation POC 2 → Automation Framework A → Automation Framework B → Isolated Automated Test Reports B
Challenges of Managing Manual & Automated Testing Together
Very different entities

- Automated tests are usually deeper and more specific
- Manual Tests are usually more extensive and less defined

- There are different types of manual tests:
  e.g. Scripted vs. Exploratory / Regression vs. New Funct.
- There are also different types of automation:
  e.g. GUI vs API vs Performance vs. Unite vs. Integration...
Different timelines for results

- Automated tests can complete full cycles in minutes / hours
- Manual results can take days or weeks to be completed
No one really knows which scripts exist or where they are running

- Teams create the tests they need.
- There are no guidelines or coordination.
- No one really knows where is running or when it runs.
- Very hard to reuse tests or integrate results to create unified visibility
We need a different approach...

First of all
Think about the overall desired result.

Only then
Plan the contribution, scope & priority of the parts.
We need a different approach...
Orchestrating Your Testing Process
Test Orchestration I - Master Planning

1. Define the purpose and scope of the overall testing process

2. Agree on what will be tested Manually and what Automated

3. Break down into tasks

4. Agree on priorities and timelines
Test Orchestration II - Ongoing coordination

5. Define a communication and coordination process across teams

6. Deploy a single reporting and visibility framework

7. Integrate between the different tools & frameworks (manual + automated)
Test Orchestration III - Broadcasting Value

8. Deliver results both with stand-alone value and as integrated Quality Dashboard

9. Generate business valuable information for the main stakeholders

10. Distribute information via multiple channels for better diffusion
Test Orchestration IV - Important Points

- Centralized process should not interfere with the ability to run trials and innovate
- Invest in cleaning up and refactoring tests as part of the daily tasks
- Make the integrated testing framework an enabling priority and not a side cost
Finally, Keep in mind...
Keep in mind

- The balance is shifting from Integrated Platforms (one-stop shops) to Best of Breed.

- Constant visibility and transparency ensure healthy processes.

- Align from the “get-go” the testing efforts with the Company’s Business Value
Thank you!

joel@practitest.com
@joelmonte
http://qablog.practitest.com