7 steps for digital app test automation success

October 2018
Speakers

Guy Arieli
CTO

Ruth Zamir
Director of Marketing
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>About Experitest + Intro</td>
<td>5 min</td>
</tr>
<tr>
<td>02</td>
<td>7 steps for digital app test automation success</td>
<td>35 min</td>
</tr>
<tr>
<td>03</td>
<td>Summary &amp; Q&amp;A</td>
<td>10 min</td>
</tr>
</tbody>
</table>
SeeTest Continuous Testing Platform for web and mobile applications

Digital Lab
Web & Mobile

Test Automation

Test Creation

Test Execution

Analytics & Reporting

Manual testing

Non-functional testing

Developer tools

Codeless automation (e.g. Worksoft, Tricentis etc)
SeeTest Continuous Testing Platform for web and mobile applications

Digital Lab
Web & Mobile

- Test Automation
- Test Creation
- Test Execution
- Analytics & Reporting

- Manual testing
- Non-functional testing
- Developer tools

Codeless automation (e.g. Worksoft, Tricentis etc)

- Galen Framework
- eclipse
- XCTest
- espresso
- TeamCity
- Jenkins
- Bamboo
- Creative Center
- Android Studio

- SeeTest
- Continuous Testing Platform for web and mobile applications

- 5
SaaS or on-premise deployment available

Fully secure flexible deployment options to suite your needs

SaaS
Private / shared lab at Experitest datacenters

On-premise
Private lab at Customer datacenter
SaaS datacenters worldwide with real devices and browsers

All datacenters ISO 27001 and SOC 2 certified
Analyst Recognition
Forrester Wave™: Omnichannel Functional Test Automation Tools, Q3 2018

- **Top Ranked for market presence**
- **Best current offering for digital quality**
  - “a complete testing environment for both web and mobile, a robust mobile device cloud, and extensive cross-browser testing”
  - One of the few players to offer a unified platform for both mobile and web
- **Visual testing**
~75% of global services decision makers are investigating or implementing digital transformation

The Forrester Wave™: Omnichannel Functional Test Automation Tools, Q3 2018
Continuous testing

Achieving Speed

What Percentage of your tests are automated?

- Over 90%
- 50%-90%
- 10%-50%
- less than 10%
- Don't know
- Don't measure

3% 20% 40% 22% 8% 7%
52% biggest challenge in web and mobile app testing is not enough time to test

61% find it difficult to automate because our applications change too much with every release

* World quality report 2018-2019, Capgemini
It’s a scale issue
# Agenda

<table>
<thead>
<tr>
<th></th>
<th>Content</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Intro</td>
<td>5 min</td>
</tr>
<tr>
<td>02</td>
<td>7 steps for digital app test automation success</td>
<td>35 min</td>
</tr>
<tr>
<td>03</td>
<td>Summary &amp; Q&amp;A</td>
<td>10 min</td>
</tr>
</tbody>
</table>
What is causing low automation rates?

Going back to automation basics

➔ Slow test creation
➔ Test flakiness
➔ High maintenance efforts
➔ Time consuming analysis
1 Object Identification
Why object identification?
Workaround example: Complex object property combinations

XPath=//*[@class='android.widget.Switch' and ./preceding-sibling::*/[@text='Wi-Fi']] 

Challenge:
Properties change across versions ➔ Broken identifiers ➔ Failed tests
Best practice for object identification:
DevOps team Collaboration

✓ Make creation of unique identifiers part of the design process
  - Create a well defined convention: what needs to be identified, how to identify it
  - Make identifier requests from automation engineers a high priority task

✓ Teach automation engineers to add identifiers to the application code
  - (and grant them authority)
Break Test Chains
What are test chains?

Tests that are dependent on the success of previous tests.

Example:
• Register user
• Log in
• Perform action
About test chains

Why are test chains created?

Test “overhead” may be much longer than test itself

• Avoiding repetition:
  – one test may be another test’s configuration
  – Common steps at the beginning

• Lack of test data management

• No attention to test suite design

So What’s the problem?

• A chain is only as strong as it weakest link

• Hard to debug tests at the end of the chain (execution time)

• “Hidden configurations” create lack of test visibility
Breaking Test Chains

- Use test data management to create the state
- Add test parallelization
- Use an API call to build an environment from scratch
  - Will work only if the environment setup time is very fast
- Use adaptive tests

Any test should be executable with no dependence on other tests
3 Create Small Tests
Why do we want small tests?

Make it easy to pinpoint an issue

- Makes test simpler to create, simpler to read
- Easier to identify what failed
- Easier to debug

*and you are really productive, creating a lot of tests...
Enabling small tests

✓ Every test should test the smallest functionality possible
✓ Use an API to do anything that is not the core test functionality
✓ Share setup between tests
  - The test will identify the state and if needed will perform setup

Extend the number of tests as functionality grows
4 Business driven tests
Test like users, not like testers

Business driven test script:

Step 1: Login
Step 2: Check balance
Step 3: Make payment
Step 4: Check new balance

UI driven test script:

Step 1: Find the username field
Step 2: Send the username,
Step 3: Find the password field
Step 4: Send the password,
Step 5: Click on the login
...
Benefits of business driven tests

- Easily readable by everyone (coders and non-coders)
- Entire DevOps team can write tests – including business analysts
- Abstraction enables cross-platform reuse for consistency
  - Android/iOS/Web
- Minimal code repetition
5 Dealing with environment availability
• Staging / QA environment may be available less than 20% of time
• Often hard to identify in test results
Environment availability

✓ Execute your UI tests only after all of the API tests passed successfully

✓ Run your tests against a mocked environment (my favorite)
Continuous execution
Testing the testing

Your automation program is a very complex application

• Once a test is create, start running it on the existing version to test it. Continue to do so to make sure your suite isn’t broken.

It takes around 1,000 executions to stabilize a test.
After that, if a test failed, I know the issue is with the application.
The Continuous Testing process

Commit app code (1)

Commit tests

Pull app source (2)

Pull test source (4)

Trigger build

Load app to cloud (6)

Trigger test execution (7)

Consolidate test results (8)

Version approved? (9) no yes

Debug / update test suite (10)

Continuous Integration

Build app (3)

Build test project (5)

Maven, Gradle
The Continuous Execution Process

1. Pull app source
2. Build app
3. Pull test source
4. Build test project
5. Load app to cloud
6. Trigger test execution
7. Consolidate test results
8. Test suite ok?
9. Debug / update test suite

Continuous Integration

Test suite ok?

Test Engineer

Commit tests

Pull test source

Build test project

Build app

Maven

Gradle

Load app to cloud

Trigger test execution

Consolidate test results

Debug / update test suite
Continuousy execute

✓ Once a test is create, start running it on the existing version to test it.

✓ Run your entire test project periodically even when there is no new code, to make sure your suite isn’t broken.

✓ Fix your test suite quickly, to be
Analyze quickly:

be ready to scale
The test automation target is moving

- How do I automate this test?
- How do I find the object and click it?
- How do I run all my tests quickly?

I have results from a million* of tests. Now what? Should I release the code?

* literally. This is an example from one of our customers.
Analyze quickly - be ready to scale

Continuous testing of thousands of tests creates mountains of data

Analysis time grows exponentially with amount of test
Actionable Analytics to drive decisions

- Clear sharable reports
- Consolidated test results for analysis
- Dashboards for management decisions
Demo
Agenda

01 Intro 5 min

02 7 steps for digital app test automation success 35 min

03 Summary & Q&A 10 min
7 Steps For successful test automation

- Collaborate on object identification
- Break test chains
- Create small tests
- Business driven tests
- Deal with environment availability
- Execute continuously
- Analyze quickly

AT SCALE
Get started absolutely free.
No credit card needed.

FIRST NAME
First name

LAST NAME
Last name

EMAIL ADDRESS
your@email.com

Country
Select country

PHONE
+1

PASSWORD
Enter your password

By signing up, you confirm that you agree to our Terms of Use.
Questions?

Thank You!