ABRACADABRA, PULLING TEST ENVIRONMENTS OUT OF THE HAT.

Presented by Sergio Beltran
• INTRODUCTION
• TOOLS PIPELINES
• METRICS
• OBSTACLES VS TRICKS
ABOUT ME

User Conference on Advanced Automated Testing
MAGIC
Is the power of apparently influencing events by using mysterious or supernatural forces.

DevOps Movement

10 years.
4 Millions of Years.

40% 50% 10%
Normalization

Automation

Changes

Technology

Mentality

Collaboration

THINK BIG

ACT SMALL

LEARN FAST!
# OUR JOURNEY: TOOLS

<table>
<thead>
<tr>
<th>SCHEMA-CHANGES</th>
<th>BUILT</th>
<th>DEPLOY</th>
<th>TEST</th>
<th>PLATFORMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitLab</td>
<td>Jenkins</td>
<td>XL DEPLOY</td>
<td>cucumber</td>
<td>SauceLabs</td>
</tr>
<tr>
<td>JIRA</td>
<td>maven</td>
<td></td>
<td>behave</td>
<td></td>
</tr>
<tr>
<td>Confluence</td>
<td>JUnit</td>
<td></td>
<td>examples</td>
<td></td>
</tr>
<tr>
<td>Nexus</td>
<td>kiowan</td>
<td></td>
<td>selenium</td>
<td></td>
</tr>
<tr>
<td>gradle</td>
<td>npm</td>
<td></td>
<td>appium</td>
<td></td>
</tr>
<tr>
<td>node</td>
<td></td>
<td></td>
<td>ZePhyr</td>
<td></td>
</tr>
</tbody>
</table>

**Release Management, Deployment, and Testing Tools**
TESTING ENVIRONMENTS:

- **Test code/data**
- **Test infra code**
- **Test descriptor**
- **Type (Core, Functional)**
- **List of functionalities**
- **Data test**

**ORCHESTRATOR**

- **GitLab**
  - Test code/data
  - Test infra code
  - Test descriptor
- **Sw map**
- **GitLab**
  - Code
  - Functionality
  - Dependencies

**Test plan**

**Test technologies**

- **Java**
- **cucumber**
- **Se**
- **TestNG**
- **Apache JMeter**

**Built Agents**

- **Container registry**
- **Mock Registry**
- **VM Images**
- **Config Server**
- **Data Volume**

**User Conference on Advanced Automated Testing**
MOBILE TESTING:

ORCHESTRATOR

• Test code/data
• Test infra code.
• Test descriptor

• Code.
• Functionality
• Dependencies

• Type (Core, Functional)
• List of functionalities.
• Data test

SW Map

DEVICE FARMS

TEST plan

User Conference on
Advanced Automated Testing
PROMOTE ENVIRONMENTS (risk management):

ORCHESTRATOR

QA GATES for PRE/PRO
- 85% unit coverage, 80% functional coverage automated.
- 0 blocker issues in static code analysis.
- No critical issues in performance (soak, time response, degradation from previous).
- No critical security vulnerabilities.
- All dependencies have to be installed/planned.
- No mocks allowed.

RELEASES
- Train releases for INT environments.
- Rolling upgrades

ENVIROMENTS
- All environments must have the same configuration (firewalls, proxy rules, permissions..).
Comparative Metrics:

**Automated Tests**

- 9m: Before = 26, After = 122
- 6m: Before = 48, After = 85
- 3m: Before = 48, After = 118

**Media Test Execution**

- Core: Before = 51, After = 401
- Functionality: Before = 12, After = 114

**Defects Detected**

- Before = 288, After = 521

**Defects/Environment**

- Integration: Before = 172, After = 443
- Pre: Before = 88, After = 62
- PRO: Before = 28, After = 36
Comparative Metrics:

- More automated tests.
- More executions.
- More defects found.
- No blocked time by environment.
- Independent environments.
- Less administration.

INFRA TEST COST

Before ≤ 5%  After

QA RATE BY OTHER ACTORS

Before: 3
After: 7.1
1. Dependencies between products.

- **PLANIFICATION**
  - A → B → C

- **DEPLOYMENT PROBLEMS**
  - A → D

- **SW MAP**
  - Functionalities

- **SCALING**
  - Scrum of scrums

- **INTERFACES**
  - Annotations/parser
  -Mocks repo/ Reviews

- **DEPLOYMENT**
  - Policies / No pulling
2. Increase quality in each sprint.

**QUALITY**

CONTINUOUS IMPROVEMENT

- Include automation in DoD
- Specialist reviews
- Requirements traceability matrix
- Coverage/blocks
- No permissive
THANK YOU!