Shift Left Testing: are you ready?

Live Webinar, Sept 19
Guy Arieli
CTO,
Experitest
Agenda

01 What exactly is Shift Left?

02 How Shift Left affects application development & testing organizational structures

03 Case study - US bank is shifting left

04 New technologies and skills required - including demos
Digital transformation impacts business

Customer empowerment

Digital is a key survival strategy

Frequent app releases
(mobile: 2 weeks, web: daily, even hourly)
Paradigm shift in software/application development

Focus on speed, business agility and risk reduction

- Smaller development increments
- Communication
- Early validation

![Diagram showing delivery cycle profiles with feedback styles: Quarters 35%, Months 40%, Weeks 15%, Days 10%]

Source - Forrester
Why shift left?

Testing has not caught up with development - becoming less effective

Rethinking the role and method of testing in the organization

- Lack of ownership impacts quality
- Inability to evaluate developers
- No alignment of goals - code is not “automation friendly”
What does “shift left” mean?

Start testing as soon as you start developing

- **Shift in mindset** – you are the only quality gate
- **Shift in responsibility** – quality is the developer’s responsibility
- **Shift in timing** – test earlier
- **Shift in organization** – fewer official testers, more quality engineers
<table>
<thead>
<tr>
<th>Testing maturity model</th>
<th>Waterfall Testing</th>
<th>Implementing Agile</th>
<th>Shift left &amp; Continuous Testing</th>
</tr>
</thead>
</table>
| **General**            | Mostly manual testing – focused on regression  
Significant variation between releases  
⇒ low ROI on automation  
Technical debt  
Automation coverage <10% | Test automation as a goal.  
Joint teams, and more automation  
Developers not accountable for quality.  
Automation coverage ~50% | Developers accountable for quality.  
Quality as part of design and development:  
Manual testing evolved to exploratory business driven testing  
Automation coverage ~90% |
| **Time to feedback**   | Months            | Days              | minutes                         |
| **Implications**       | Testing is a time consuming bottleneck  
Difficult to distinguish between good developers and fast developer. | Efforts wasted on element identification and fixing broken tests. | Code designed for automation.  
Developers get instant feedback.  
Code fixed prior to commit, eliminating wide impact and costly rollback. |
| **Testing frequency and timing** | Every few months (post-development) | Daily (post commit) | Constantly (pre-commit) |
01  What exactly is Shift Left

02  How Shift Left affects organizational structure

03  Case study - how a US bank is shifting left

04  New technologies and skills required - including demos

Agenda

What exactly is Shift Left - goals and drivers
Shift Left is a spectrum

Everyone is different. Mobile plays a role

Google
On adwords – dedicated team of test engineers

Microsoft
High number of testers - software development engineers in test

Spotify
A combined model with matrix responsibilities

Facebook
Developers responsible for testing – there are no testers

Amazon
No testing - rely on developers and easy rollback

No testing org. no QA
## Testing maturity model

### impact on organizational structure

<table>
<thead>
<tr>
<th>Waterfall Testing</th>
<th>Implementing Agile</th>
<th>Shift left &amp; Continuous Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Who is responsible for quality?</strong></td>
<td>QA team</td>
<td>QA team</td>
</tr>
<tr>
<td><strong>Who performs testing?</strong></td>
<td>Testing analyst</td>
<td>Manual testers, automation engineers</td>
</tr>
<tr>
<td><strong>QA skills</strong></td>
<td>Domain expertise - understand business needs</td>
<td>Automation expertise Familiarity with coding concepts</td>
</tr>
<tr>
<td><strong>Tests type</strong></td>
<td>Blackbox testing End to end functional</td>
<td>Blackbox testing User stories</td>
</tr>
<tr>
<td><strong>Tools</strong></td>
<td>HP QTP</td>
<td>Selenium / Appium</td>
</tr>
<tr>
<td><strong>Organizational structure</strong></td>
<td>Separate testing organization.</td>
<td>Manual / Automation Under QA org</td>
</tr>
</tbody>
</table>
Agenda

What exactly is Shift Left - goals and drivers

01 What exactly is Shift Left - goals and drivers

02 How Shift Left affects organizational structures

03 Case study - how a US bank is shifting left

04 New technologies and skills required - including demos
Background - why shift left?

Wasteful, disconnected testing process

- Misalignment with business goals: Over-testing and under-testing
- Slow feedback
- High cost of defect handling
- Wait times

Solution – combine dev-test to a single team and a continuous process
What changed?

- Converted manual testing to automation engineers
- Dissolved QA organization
- Created small teams of up to 10 people with clear responsibilities
- Architectural changes to allow systems to be more usable and testable (mocking server)
The dev-test process

1. Epic
2. Story thinning
3. Prototype
4. Acceptance Criteria
5. Functional Spec
6. Showcase / Demo Sign-off
7. Acceptance testing
8. Automated testing
9. Build & Deploy
10. Code (incl. White box Test)

Key Roles:
- Product Owner
- Analyst
- Designer
- Tester
- Test Engineer
- Dev Engineer
- Dev & Test Engineer
- DevOps Engineer
Shift left organizational impact – case study

- Performance & Volume: 10%
- Environment and Data: 20%
- Automation Testers:
  - Regression automation
  - Services virtualization skills
- Manual Testers:
  - Manual System testing
  - Manual Integration Testing
  - Acceptance Testing
- Exploratory testers:
  - Domain Expertise
  - Business Rules testing
  - Manual Integration Testing
  - Acceptance Testing
- Software Engineers - Testers:
  - Automation roadmap
  - Continuous testing
  - User Story Thinning
  - In sprint progression automation
  - Services virtualization skills
  - Engineering mindset
Outcome

Release cycle down from 3 months to 6 weeks (targeting 2 weeks)

R&D total productivity up

Customer satisfaction up

It's an ongoing year-long process...
Agenda

What exactly is Shift Left - goals and drivers

01 What exactly is Shift Left - goals and drivers

02 How Shift Left affects application development and testing organizational structures

03 Case study - how a US bank is shifting left to shorten dev-test cycle and provide better digital experiences

04 New technologies and skills - including demos
Testers

Explore unexpected scenarios and provide the user perspective

What does this mean?
Need to do a transition
Developers

Quality owners

What does this mean?
- Wider responsibility – from just code to working product
- Be the feature owner

New skills & knowledge
- Automation principles – writing code that can be easily automated
- QA methodologies

Tools you can use
- SeeTest Application Development & Debugging
- Grid Execution Mobile & Web
Test engineers
Enable wide scale automation and rapid feedback

What does this mean?
- You should perceive yourself as a Developer
- Shift to code base automation
- Build the CI pipeline

New skills & knowledge
- Know common coding languages – be able to read code
- Learn the dev environments (IDE)
- Learn Git / Gradle / Maven ...

Tools you can use
- Appium Studio
- Appium Studio for Eclipse
- Grid Execution Mobile & Web
10 things you can do to be ready to shift left

- Build a basic java project with Intellij
- Learn how to use Git
- Open an account in GitHub
- Install android SDK and create a demo application, run on device and emulator
- Take a tutorial on Gradle and Maven
- Use Gradle or maven script to create selenium project in eclipse or intelij, build a basic google search test
- Record a test in espresso
- Install XCode on your computer, build an iOS demo app and record an XCUItest
- Download Appium studio and write a test
- Install Jenkins and create a script that downloads Git projects and runs them
Demo