Testing Inside Your Timebox:
Death To The Hardening Sprint

An Agile2014 Orlando Presentation

Presented by John Hughes and Tanusree McCabe
Blackstone Technology Group
John Hughes
Agile Coach and Strategist
PMI-ACP, CSM, PMP

Agile Coach and Strategist with Blackstone Technology Group's Federal Practice.

Developed and delivered solutions in the Federal IT space for over 16 years, focusing on Agile for the past five years.

I love to spread the Agile gospel and teach teams how to continuously improve their delivery.

I believe we should strive to "Be Agile" not just "Do Agile"
Tanusree (Tanu) McCabe
Project Manager
PMP

Project Manager with Blackstone Technology Group’s Federal Practice

...has 10 years experience leading enterprise technical solutions delivery

...has led the delivery of automated solutions facilitating Agile development across the federal government

...gets better at Agile by applying Agile to a myriad of projects
Overview

Our Goal: For you to understand the value of completing all required testing inside your timebox without extending your iteration, requiring a hardening sprint or test iteration, or accepting low-quality; and ways to do so

Not just about testers, but the whole organization

Not just about Scrum, but all Agile practices

Poll – who makes up our audience?

- Who works on a Federal program or other highly regulated sector?
- Who is a tester? A developer? Manager/PMO? ScrumMaster or Coach?
- Anyone already completing all their required tests within their iteration timebox?
Poll – who makes up our audience?

- Who works on a Federal program or other highly regulated sector?
- Who is a tester? A developer? Manager/PMO? ScrumMaster or Coach?
- Anyone already completing all their required tests within their iteration timebox?
Why Test Inside Our Timebox?
### Traditional Waterfall

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Design</th>
<th>Development</th>
<th>Integration and Test</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Unit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Integration</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Func/Comp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Smoke/Expl</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Regression</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Ind/V&amp;V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Usability/S08</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Perf/Load</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Security</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- UAT</td>
<td></td>
</tr>
</tbody>
</table>

### Typical Agile

<table>
<thead>
<tr>
<th>Sprint 1</th>
<th>Sprint 2</th>
<th>Sprint 3</th>
<th>Sprint n</th>
<th>Hardening Sprint</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular Sprint Testing:</td>
<td></td>
<td></td>
<td></td>
<td>Hardening Sprint Tests:</td>
<td></td>
</tr>
<tr>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Integration</td>
<td></td>
</tr>
<tr>
<td>- Unit</td>
<td></td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Integration</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Typical Agile

<table>
<thead>
<tr>
<th>Sprint 1</th>
<th>Sprint 2</th>
<th>Sprint 3</th>
<th>Sprint n</th>
<th>Hardening Sprint</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Regular Sprint Testing:</strong></td>
<td>- Unit</td>
<td>- Unit</td>
<td>- Unit</td>
<td>- Unit</td>
<td>Production Release</td>
</tr>
<tr>
<td>- Integration</td>
<td>- Integration</td>
<td>- Integration</td>
<td>- Integration</td>
<td>- Integration</td>
<td></td>
</tr>
<tr>
<td>- Func/Comp</td>
<td>- Func/Comp</td>
<td>- Func/Comp</td>
<td>- Func/Comp</td>
<td>- Func/Comp</td>
<td></td>
</tr>
<tr>
<td>- Smoke / Expl</td>
<td>- Smoke / Expl</td>
<td>- Smoke / Expl</td>
<td>- Smoke / Expl</td>
<td>- Smoke / Expl</td>
<td></td>
</tr>
<tr>
<td>- Acceptance</td>
<td>- Acceptance</td>
<td>- Acceptance</td>
<td>- Acceptance</td>
<td>- Acceptance</td>
<td></td>
</tr>
</tbody>
</table>

**Hardening Sprint Tests:**
- Regression
- Ind/V&V
- Usability / 508
- Perf/Load
- Security
- UAT

Optimized Agile

Production Release
Production Release
Production Release
Production Release
Production Release
### Optimized Agile

#### Sprint 1

**Complete testing for production release:**
- Acceptance
- Unit
- Integration
- Func/Comp
- Smoke / Expl
- Ind/V&V
- Regression
- UAT
- Usability / 508
- Perf/Load
- Security

- Acceptance
- Unit
- Integration
- Func/Comp
- Smoke / Expl
- Ind/V&V
- Regression
- UAT
- Usability / 508
- Perf/Load
- Security

- Acceptance
- Unit
- Integration
- Func/Comp
- Smoke / Expl
- Ind/V&V
- Regression
- UAT
- Usability / 508
- Perf/Load
- Security

- Acceptance
- Unit
- Integration
- Func/Comp
- Smoke / Expl
- Ind/V&V
- Regression
- UAT
- Usability / 508
- Perf/Load
- Security
### Traditional Waterfall

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Design</th>
<th>Development</th>
<th>Integration and Test</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Unit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Integration</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Func/Comp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Smoke / Expl</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Acceptance</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Unit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Integration</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Func/Comp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Smoke / Expl</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Acceptance</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Unit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Integration</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Func/Comp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Smoke / Expl</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Acceptance</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Typical Agile

<table>
<thead>
<tr>
<th>Sprint 1</th>
<th>Sprint 2</th>
<th>Sprint 3</th>
<th>Sprint n</th>
<th>Hardening Sprint</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular Sprint Testing:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
</tr>
<tr>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
</tr>
<tr>
<td>- Func/Comp</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
</tr>
<tr>
<td>- Smoke / Expl</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
</tr>
<tr>
<td>- Acceptance</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Optimized Agile

<table>
<thead>
<tr>
<th>Sprint 1</th>
<th>Sprint 1</th>
<th>Sprint 1</th>
<th>Sprint 1</th>
<th>Sprint 1</th>
<th>Sprint 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex testing for production release:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Acceptance</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
</tr>
<tr>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
</tr>
<tr>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
</tr>
<tr>
<td>- Func/Comp</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
</tr>
<tr>
<td>- Smoke / Expl</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
</tr>
<tr>
<td>- Acceptance</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
<td>- Integration</td>
<td>- Unit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Reduced feedback loop
- More frequent releases
- Quicker value to users
- Higher Quality
- Reduced Defects
Challenges and Solutions

• Testing within your timebox isn’t a simple exercise
  • A lot of testing, very little time
  • Highly regulated industries add additional complexity
• We’ve all experienced challenges
• There are solutions
What Are the Tests?

“All required tests” include a set such as these:

- Acceptance
- Unit
- Integration
- Functional/Component
- Smoke/Exploratory
- Regression
- Independent/IV&V
- UAT
- Usability
- Accessibility/Section 508
- Performance/Load/Capacity
- Security
First Exercise: Audience Challenges...

Each table please consider what it would take to get rid of Hardening Sprints and complete all your testing inside your timebox.

Discuss the challenges with doing this. Come up with one that your table feels is the biggest, or most likely challenge.

We will collect your challenge at the end of the exercise for upcoming discussion.
Putting it all Together

Agile Testing Quadrants

- **Q2**: Functional Tests, Examples, Story Tests, Prototypes, Simulations
- **Q3**: Exploratory Testing, Scenarios, Usability Testing, UAT (User Acceptance Testing), Alpha/Beta
- **Q1**: Unit Tests, Component Tests

*Agile Testing: A Practical Guide for Testers and Agile Teams – Crispin and Gregory*
Second Exercise: Audience Solutioning

Each table discuss the potential solutions you could implement to address the challenge your table came up with earlier

We will collect these results afterwords for discussion
Testing Continuum

*Graphic based upon image found in Lean-Agile Acceptance Test-Driven Development – Ken Pugh*
Solutions
Based on actual experiences

Continuous Testing and a Test-Forward Approach

Reduce Find, Fix, Retest cycle

Set Internal Dates

Schedule tests in advance

Reduce Velocity

Add all tests to the Definition of Done

Add all test tasks to Sprint Backlog

Attain Organizational Buy-in

Strive for Embedded and fully Integrated Teams

Propose/Purchase Automation Tools from the start (or ASAP)

Script and Automate Builds, Deployments, and Tests

Practice DevOps and CI/CD

Certify Team members as 508 Trusted Testers (qualify other team members in specialties)

Train/Hire Necessary Resources

Create an inclusive Team Area

Shared Vision of Team Success

Generalizing Specialists

Set Working Agreements

Create Story Maps

Hold inclusive Design Discussions or Three Amigos sessions

Have Developer and Architect Demonstrations

Broad and Frequent Communication

Establish a Communications Plan
Theory is good... an actual approach is a more helpful example
An ideal solution for achieving success should...

<table>
<thead>
<tr>
<th>Be Automated</th>
<th>Facilitate Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automate builds</td>
<td>Automated tests create automated results</td>
</tr>
<tr>
<td>Automate deployments</td>
<td>Sends notifications to stakeholders</td>
</tr>
<tr>
<td>Automate tests</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Be Properly Resourced</th>
<th>Follow Efficient Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Have the right tools</td>
<td>Embeds all active roles in the team</td>
</tr>
<tr>
<td>Have the right staff with the</td>
<td>Engages in continuous testing</td>
</tr>
<tr>
<td>right skills on team</td>
<td>Employs a test forward approach</td>
</tr>
<tr>
<td>Have a team area</td>
<td></td>
</tr>
</tbody>
</table>
Team develops in the development environment. When done with a testable feature, the team automatically builds and deploys to the test environment.

While testing, developers can continue to develop the next testable feature.

Unit tests can be created and automated by the developers to run in development.

Following the build and deployment process, the test environment immediately runs a number of automated tests:

- Regression
- Integration
- 508

Additional manual tests can be run:
- Exploratory

The team records new functional test scripts for the recently created testable feature and automatically runs them until the test passes.

Developers are able to develop, build, and test cyclically until the defects are resolved.

Following a defect-free test cycle, automated deployment to the staging environment occurs.

Upon deployment to staging, many tests are automatically run:

- Acceptance
- Performance\Load
- Interface
- Security

Additional manual tests can be run:
- Usability
- IV\&V
- User Acceptance

Once a test cycle passes within parameters, the release is ready to be deployed to production.

Upon successful validation and approval from the product owner and organizational governance processes, the system automatically deploys the release candidate to production.

In production, automated smoke and scenario tests ensure the application was properly deployed.
An example of a platform solution that enables this...

- A tier that hosts all the tools and capabilities that enable processes.
- A tier that hosts the basic code and applications.
- A services tier that enables organizational governance and ensures compliance.
- An integration tier that enables interfacing with other applications.

Multi-tenant Platform (Processes, Tools & Standards)

Management Zone (Tools + Capabilities)
- Source code management
- Application Security
- Data registry
- Asset Management
- Monitoring
- Automated Build
- Data visualization
- Collaboration tools
- Issue Tracking
- Audit logging
- Identity Management
- Automated testing
- Server Automation
- Automated Deploy
- Proxy
- Enterprise Data Relay

Infrastructure & Platform as a Service

Baseline
- Shared code
- Shared apps
- Collaboration

Service Management (Processes + Governance)
- Compliance (Security, SOL, SELC, FDI)
- Onboarding
- Mobile Methodology
- Training
- Change Management
- Deployments
- Data standards
- Tier 1/2/3 support
- Configuration Management
- Release Management
- Developer Pages & Standards
- Operations & Maintenance
- SLA Management
- License Management
- Acquisitions Management
- Billing & Financial

Integration Services
- API
- Transformation
- Orchestration

Mobile
Apps
Big Data/BI
Web and Enterprise Content
Testing Inside Your Timebox is Important

<table>
<thead>
<tr>
<th>Optimized Agile</th>
<th>Production Release</th>
<th>Production Release</th>
<th>Production Release</th>
<th>Production Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint 1</td>
<td>Sprint 1</td>
<td>Sprint 1</td>
<td>Sprint 1</td>
<td>Sprint 1</td>
</tr>
<tr>
<td>Complete testing for production release:</td>
<td>Complete testing for production release:</td>
<td>Complete testing for production release:</td>
<td>Complete testing for production release:</td>
<td></td>
</tr>
<tr>
<td>- Acceptance</td>
<td>- Acceptance</td>
<td>- Acceptance</td>
<td>- Acceptance</td>
<td>- Acceptance</td>
</tr>
<tr>
<td>- Unit</td>
<td>- Unit</td>
<td>- Unit</td>
<td>- Unit</td>
<td>- Unit</td>
</tr>
<tr>
<td>- Integration</td>
<td>- Integration</td>
<td>- Integration</td>
<td>- Integration</td>
<td>- Integration</td>
</tr>
<tr>
<td>- Func/Comp</td>
<td>- Func/Comp</td>
<td>- Func/Comp</td>
<td>- Func/Comp</td>
<td>- Func/Comp</td>
</tr>
<tr>
<td>- Smoke / Expl</td>
<td>- Smoke / Expl</td>
<td>- Smoke / Expl</td>
<td>- Smoke / Expl</td>
<td>- Smoke / Expl</td>
</tr>
<tr>
<td>- Ind/V&amp;V</td>
<td>- Ind/V&amp;V</td>
<td>- Ind/V&amp;V</td>
<td>- Ind/V&amp;V</td>
<td>- Ind/V&amp;V</td>
</tr>
<tr>
<td>- Regression</td>
<td>- Regression</td>
<td>- Regression</td>
<td>- Regression</td>
<td>- Regression</td>
</tr>
<tr>
<td>- UAT</td>
<td>- UAT</td>
<td>- UAT</td>
<td>- UAT</td>
<td>- UAT</td>
</tr>
<tr>
<td>- Usability / 508</td>
<td>- Usability / 508</td>
<td>- Usability / 508</td>
<td>- Usability / 508</td>
<td>- Usability / 508</td>
</tr>
<tr>
<td>- Perf/Load</td>
<td>- Perf/Load</td>
<td>- Perf/Load</td>
<td>- Perf/Load</td>
<td>- Perf/Load</td>
</tr>
</tbody>
</table>

- Reduced Feedback loop
- More frequent releases; quicker delivery of value to users
- Higher quality; reduced defects
- Reduced cost and overhead
Testing Inside Your Timebox is Achievable

- There are Challenges
- There are Solutions
- You've seen an example of the type of over-arching solution platform you can build
- Here are some near term takeaways which you can begin to implement immediately
## Immediate Takeaways

- Set a goal to have NO Hardening Sprint
- Put ALL testing into your Definition of Done
- Put ALL test tasks into your Sprint Backlog, including the tasks for Independent/Systems Assurance, Accessibility, Security, etc. testing
- Ensure your Agile teams are fully INTEGRATED, embedding non-dev team testers such as IV&V, OAST, etc.
- Implement a TEST-FORWARD strategy. Look to principles of ATDD to help.
- Strive to achieve CONTINUOUS Testing
- Put together an AUTOMATION strategy, choose tools to enable that strategy
- INCREASE COMMUNICATION to all parties, earlier, more often, more broadly
- Hold 3 AMIGOS for each user story
Thank You For Your Time...

John Hughes
Strategist
jhughes@bstonetech.com
Twitter: @jwhughesjr

Tanusree McCabe
Project Manager
tmccabe@bstonetech.com