The Modern QE/QA Role:
Supporting DevOps the Smart Way
What are we talking about?

The Quality Engineer role
Construction Project Analogy – QA vs. QE
Re-tuning Automation
Shifting traditional QA/QE “right” tasks left
  What a Modern Testing Role Looks Like
  The Outdated vs. Modern Regression Testing Approach
  Scripted vs. Unscripted Testing
Re-defining Refining
Shippable
The Modernized Definition of Done Example
The Quality Engineer Role – The “Mis”-Buster

The Misperceptions of Quality Assurance

There is one team that owns quality
It is not a skill (i.e., “anyone who is a user can test”)
If an issue is found Live or by a user, it’s QA’s fault
The Quality Engineer Role – The “Mis”-Buster

The Misperceptions of Quality Assurance

Mistake Finders – although sometimes we do find these
The “catch-all” for failed processes
The only testing that happens within the SDLC
The Quality Engineer Role – Defined

“Define, Design, Build, Execute, Measure, Report”

Define – success, outcome and measurements
Design – a comprehensive strategy
Build – the solution
Execute – the solution
Measure – the results
Report – the outcome
The Quality Engineer Role – Defined

“Influence the Building of the Software before the Software is Built”

Balances Technical Acumen with User Advocacy – with Equal Emphasis on Both

Context-Driven: Given the information we have, determine if it’s enough and if not, we find more
The QA Role Comparison
Construction Project Roles

Which one is QA?

Architect
Skilled Tradespeople
General Contractor
Project Manager
Building Inspector
Customer
The Misperceptions of QA as Inspector

It instills a false sense of non-ownership with Development
It assumes that they cannot be trusted to check their own work
It creates an “Us vs. them” mentality – and ultimately…
A crutch
A New Take – QE as General Contractor

QE as General Contractor...

QE is familiar with multiple components of the software development cycle
QE works with the owner/user throughout the course of the project
QE can vet out requirements and specific needs of the user/owner
A New Take – QE as General Contractor

Like the General Contractor…

QE can advocate that the desired level of quality has been met, including design and user experience.

QE offers timely and valuable feedback to the tradespeople that add to the overall success of the project.
Re-tuning Automation

“Test automation makes humans more efficient, not less essential”

What it’s been:

- Monolithic
- Focused on everything
- A Numbers Game
Re-tuning Automation

What it should be:

- Tiered approach, or
- “Multiple runs for multiple dones”
- Providing the most valuable information ASAP

Remember the AofA

Automated of Automatable
The Modern Approach

Continuous Improvement

Ask …

“Why do we do this?”

“What happens if we stop?”
The Modern QE

Strike balance between traditional Specialist roles and moving more toward Generalists

Test Automation – Start Small
  Run, Troubleshoot, Edit

Accessibility

Functional Security

Performance
Outdated Regression
Mis-Perceptions of Regression Testing…

QA Owns it Solely
We build in days—or sometimes weeks—to account for it
Usually at the end of a sprint or
While preparing for a big release
Modern Regression

Shifts Left within the sprint
Within a couple of days from Dev To Development!
Information gathered is shared real-time when the code is “fresh”
Scripted vs. Unscripted Testing

Start with Exploring!

Scripted Tests (automated and non-automated) are written before and during coding of the story.

Development refers to them throughout.

Explicit vs. Implicit information.
Re-define Refine

Time to Re-Define!

Or, is it Re-Refine?

Old School Refinement looks like:

The entire project team + lurkers
In a room
Cramming as many stories as possible in an hour
Re-define Refine

New School Refinement looks like:

Small working groups
Shorter times
Everyone represented – yes, even DevOps
Tiered approach
Shippable

The New “Definition of Done”

Because the Old Definition of Done was ”belonged” to Product

Each practice in the Agile team is represented
Shares their Playbook
And removes the guesswork
Shippable

Remove the silos of Dev, QA, PM and DevOps “done”

Put the red bow on it

Tech debt is addressed

Automation is green and in the pipeline

Sev 1/2 bugs are closed

All P1/2 test cases are passed
Shippable

Just because you can, doesn’t mean you should!
Just because you could, might not mean you did!
All that to say,
Get to Shippable!
Summary

The Quality Engineer role
Construction Project Analogy – QA vs. QE
Re-tuning Automation
Shifting traditional QA/QE “right” tasks left
Re-defining Refining
Shippable
The Modernized Definition of Done Example
Let’s Talk!

LinkedIn: Melissa Tondi
Twitter: @melissatondi
Email: melissa.tondi@gmail.com
How do we do it?

Definition of Done

The Playbook for how we “do” testing
Deep-Dive: DoD Example

Quality Engineering vs. Quality Assurance
Continuous Efficiency
Balances Technical Acumen with User Advocacy – with Equal Emphasis on Both

Context-Driven: Given the information we have, determine if it’s enough and if not, we find more by:
- Collaborating within Product Engineering
- Partnering with Professional Services, Support and other teams
- Meeting with customers
- Reaching out to the QE community
Deep-Dive: DoD QE – What it’s Not

Mistake Finders of others in the SDLC – although sometimes we do find these

The “catch-all” for failed processes

The only testing that happens within the SDLC

- Refining the Definition of Done for all (Dev, Product, QE, etc.)
- Demos of the AC from Dev to PO, QE, etc.
Deep-Dive: DoD QE– What We Do

Emphasized collaboration within the project team

Refine/Groom all work– discuss the acceptance criteria success so that each person with tasks within a feature or story/PBI can begin their work as an outcome. Work is not considered refined/groomed without it and therefore, work should not be committed to within a sprint if it’s not fully refined/groomed

Tighter communication between Development, QE, DevOps and Product within our sprints

Continuous refinement, increased documented information with the goal of continuous improvement
Assess activities for more efficiency – the goal is to increase testing

When we find things to remove from our plate, we replace them with more valuable testing

Categorize Tests: Scripted and Unscripted

Scripted (automated and traditional test cases)
Visible and Centralized in Test Lodge and Jenkins and Included in the Story
How we plan to test
Code should be written to pass the tests

Unscripted: Exploratory and AdHoc

Prioritize Tests: P1-P3. At a minimum, P1s and P2s are executed and passing as release success criteria
Deep-Dive: DoD QE – What We Do

Follow a Well-defined Test Automation Strategy

Centralized and critical suites of tests that map to critical functions available to everyone in Engineering

P1: Smoke Test = A subset of all defined/planned test cases that cover the main functionality of a component or system, to ascertain that the most crucial functions of a program work, but not bothering with finer details. Things we consider:

- Can it release without it working?
- Is it part of the MVP (Minimum Viable Product)?
- If it doesn’t work, is there a monetary or customer loss associated with it?
- Is it a Security Vulnerability?
- Does it run in five minutes or less?

P2 = User/Customer Flows

P3 = Dealer’s Choice and based on feedback from the product team
Deep-Dive: DoD QE– What We Do

User Advocacy Test Runs – Tied to our P2 Test Automation Suite

Working with PO and PS to ensure we are testing how our users are using the product

Performance Engineering

Current = 3 seconds or less and vetted out with the product team and written as bugs

Accessibility

Level AA (using the WCAG 2.0 and 2.1 guidelines)

Functional Security Testing

https://www.owasp.org/index.php/Top_10_2013-Top_10
Deep-Dive: DoD: Expectations of Dev

Intake Test = Unit and Integration

Run upon Dev Commits

Unit Testing Visibility

Dev’s DoD

Demo of AC at the Story Level – When Requested via Label in TFS

Before it’s Checked-in (agreement is that Dev will demo after merge)

To PO, QE and anyone else within the Product Engineering team that may need to see the AC

Automatable Code
Deep-Dive: DoD: Expectations of Dev

Stories are begun to be Dev Complete and In Test by sprint day 3 (EOD sprint day 2)

And consistently coming our way from days 3-8

There is a two-day “sprint hardening” at the end where we should complete all the refined stories

   No stories should come to QE after sprint day 8 unless discussed and a plan agreed on with the team

Devs are also focusing on Bug Fixes from the sprint
Deep-Dive: DoD: Expectations of the Team

All Stories and Customer-Reported Bugs are Refined

“Meet” (not necessarily formally) with everyone that has responsibilities on the story

Dev should give an overview of their plan and talk about regression needs and impact analysis

PO should be prepared to answer questions on the AC and edit to add more details while discussions are happening

QE should give an overview of what they will test, permutations, scenarios, ask questions about Dev’s approach and PO’s expectations, etc.