Finding Efficiencies in Testing

What Are we Talking about?

Why Me?

Why is there Never Enough Time?

What Testing Activities do we Emphasize?

Where do we Find Inefficiencies?

How do we Fix them?

What would you do with that Time Saved?

Open Q&A

Why Me?

- I’m devoted to the Quality (with a Capital Q) of the SDLC
- My experience is in the QA/QE and Agile spaces, but I also have expertise in management of most areas of the SDLC
- I spend my career focusing on three tenets: efficiency, innovation, and culture

Why is there Never Enough Time?

- Decreased Schedules
- Decisions Made in a Silo
- Risks not Vetted Appropriately or with enough Emphasis
- QE/QA/Test may not have an appropriate “Seat” at the Table
- The List Goes on…

What Testing Activities do We Emphasize?

- Test Planning/Management
  - What's your Ratio (of Planning/Management vs. Execution)?
- Planning Stories by Vetting out Requirements or Acceptance Criteria
- Testing of the Stories' Acceptance Criteria or Requirements
  - We tend to test these first because they are (or at least should be) apparent
- Finding and Writing up Bugs
- Reporting on our Testing Activities

Where do We Find Inefficiencies?

- Test Planning/Management
  - We tend to write very detailed test cases
    - How much detail is enough?
  - Test Case Reviews
    - With the team or for approval
  - Test Plans as a separate document

Where do We Find Inefficiencies?

- Duplicating Testing Efforts
  - Give credit where credit is due, but…
  - Know your Devs and the rest of your team!
  - Who else is (or should be) testing?
    - Find out what they're doing

Where do We Find Inefficiencies?

- Being an Enabler (AKA “the Martyr”) – with Schedule or Scope Creep
  - Give it up!
  - We tend to make up the work at the end – which makes us awesome, but it also lets bad/impactful behaviors continue to be acceptable
  - Remember, in order for Agile scrum methods to work efficiently, ALL team members work needs to be accounted for and it should be evenly distributed across each day of the sprint

Where do We Find Inefficiencies?

- QE-Only Sprints or Cycles
- QE Hardening (or similar)
- Basically, any time of more than 2 days within a sprint where QE and Dev are not working together

Where do We Find Inefficiencies?

- Not taking advantage of the Agile Ceremonies
- The Usual Suspects:
  - Planning
  - Standups
  - Retros
  - Others?

How do We Fix Them?

- Test Planning
  - Rule of Thumb “ABT” (Always be Testing!)
  - 10/80/10 (the only ratio I like)
  - Don’t overthink it – determine how much detail is enough across the team

How do We Fix Them?

- Duplicating Testing Efforts
  - Unit Testing
    - Use the DoD (Definition of Done) to your advantage – talk about what Dev is doing on their own
  - Acceptance Criteria
    - Use Demos! Have Dev show that the AC is met before checking in code
    - Better yet – automate the AC and include that in QE's DoD

How do We Fix Them?

- Being the Martyr
  - Use Risk-Based and Context-Driven Approaches
    - Know what gives when the schedule won't
  - Be a Musketeer! All for One and One for All!
    - If one part of the team fails, the team fails – same with success
  - Try an 8/10 sprint:
    - The first two days are spent with Dev coding and QE writing tests in conjunction with Dev
    - The next six, we track that stories flow consistently
    - The last two are spent either swarming the rest of the testing (because stories weren't consistent) or on bug fixes and sprint hardening

How do We Fix Them?

- QE-Only Sprints or Cycles
  - Ask why? And listen!
- The ABC rule
  - Always be Coupled! (with Dev)
- Use the DoD to your Advantage and Swarm!

How do We Fix Them?

- Not taking advantage of the Agile Ceremonies
- Demos – from Dev to QE and PO (and anyone else on the team)
    - Before Check-in/merge – this shifts the approval of stories wayyyyyyy left (where it belongs)

How do We Fix Them?

- Refinements/Grooming - The “Meet” and Potatoes
  - “Meet” (not necessarily formally) with everyone that has responsibility for 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 about 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.
  - The outcome is that everyone has what they need to start work and a story is not considered refined/groomed until everyone gets there

Finding Efficiencies in Software Testing

Inefficiencies	Fix Them
Test Planning	The ABT Rule 10/80/10
Duplicating Testing Efforts	Know your Devs!
The Enabler	RBT and CD – be a Musketeer!
QE-Only Sprints	Ask Why?
Agile Ceremonies	Demos, Grooming/Refinement

Q&A

- Now what?

Let’s Talk!

- LinkedIn: Melissa Tondi
- Twitter: @melissatondi
- Email: melissa.tondi@gmail.com