

THE "DO NOTS" OF SOFTWARE TESTING

2018

1

WHAT ARE WE TALKING ABOUT?

The "do nots" serve as a reminder that innovation should be at the forefront of our industry

Even best-laid plans and ideas that make it in to our QA playbook should be re-visited to ensure they:

- are still applicable today
- align with the industry
- are collaboratively agreed to

2

DO NOT BE “THE ENABLER”

- We tend to make up the work at the end
- We save the team schedules
- Which is a perceived value to the team! However...
- We become the “why didn’t you catch that?” team
- Give it up!

3

DO NOT “AUTOMATE EVERYTHING”

- We all know that not all tests should or can be automated, right?
- No, really, right?
- When you are held to a meaningless outcome like “everything” your creative license is essentially removed

4

DO NOT HAVE “QA-ONLY” SPRINTS OR CYCLES

- QA Hardening (or similar)
- Basically, one or more sprints before release that's dedicated to testing by QA only

5

DO NOT “OWN ALL TESTING”

- Wait, what?
- Don't own “Regression” testing either...
- More on that later

6

DO NOT “HIDE INFORMATION”

- Explicit vs. Implicit information
- Explicit = information readily available
 - Stories
 - Acceptance Criteria
 - Requirements
 - Standup Info

7

DO NOT “HIDE INFORMATION”

- Implicit = information that is not centralized or shared with the team
 - One-on-one conversations
 - Subjective “look and feel” experiences
 - User information – CSMs, Support, ACTUAL users, etc.

8

DO NOT BE “THE ENABLER”

- Be a Musketeer! All for One and One for All!
 - If one part of the team fails, the team fails – same with success
- Use Risk-Based and Context-Driven Approaches
 - Know what gives when the schedule won't
- Prioritize your tests

9

DO NOT “AUTOMATE EVERYTHING”

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

- Use the A of A
- Automated of Automatable
- Selection process that succinctly shows the percentage of automated *valuable* tests given the criteria YOU define

10

DO NOT “AUTOMATE EVERYTHING”

How does it work in real-life?

- Refine
 - Define whether it is an Intake, Smoke, User Advocacy – in that order
- Automate
- Tie it to Shippable
- Speaking of, what's Shippable?
- The red bow or the cherry on top!

11

DO NOT HAVE “QA-ONLY” SPRINTS OR CYCLES

Shippable!

- The ABC rule
 - Always be Coupled! (with Dev)
 - More than two days is a flag and we discuss in the standup
 - The 8/10 Sprint

12

DO NOT “OWN ALL TESTING”

- Know your Devs and the rest of your team!
- Who else is (or should be) testing? Find out what they're doing and encourage consistency
- Use DoD to your Advantage and Swarm!
- Definition of Done! It's not just for Product anymore!
 - Each Pillar on the team commits to their own checks and high level activities
 - It's central and therefore, removes the mis-perception that QA owns all testing and quality

13

DO NOT “HIDE INFORMATION”

Once you've uncovered Implicit information, make it Explicit!

- Rules of Thumb – stretch goals...
- If it's written, it's explicit
 - If it's written, it's refined
- If it's explicit, it is understood
 - If it's a test, it should pass

14

DO NOT “HIDE INFORMATION”

Refinement!
Present State

- Misunderstood and/or Ambiguous Acceptance Criteria
 - One-sided Refinements
 - Usually with Dev and Product
- Anchored Sizing
 - One person or discipline anchors the team and the team tends to defer

15

DO NOT “HIDE INFORMATION”

New State

- “Meet and Potatoes” (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
- The outcome is that everyone has what they need to start work and a story is not considered refined/groomed until everyone gets there

16

DO BE A QUALITY ENGINEER!

By removing the Do Nots, you can shift to a more modern team member,

The Quality Engineer

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

17

SUMMARY

The Enabler

Automate everything

QA-only sprints/cycles

Own all testing

Hide information

Be a Musketeer. It's okay to let things fail sometimes

Use the AofA and define the criteria

Focus on Shippable sprints and the 8/10 sprint

Find out who else is testing and make it consistent and use the DoD

Refine and refine again and maybe even again!
Make implicit info explicit

18

LET'S TALK!

- LinkedIn: [MelissaTondi](#)
- Twitter: [@melissatondi](#)
- Email: melissa@rainforestqa.com