

What to Do When QA is Backed Against the Wall?

Hello

- Peter

- Software Engineer
- Quality Assurance
- Quality Engineer / Automation Specialist
- Engineering Manager
- QE Manager
- 8 Startups
- 5 Fortune 100 Companies

Hello

- Peter

- Co-organizer for DC Agile Software Testing Meetup (dcast.io)
- Network with QE/DEV leadership
 - Reality vs. “best practices”
 - Work closely with Consultants
 - Work closely with HR/Hiring
- Learning to improve my soft skills
- Tinkering with new technologies
- Crash/Reliability (Chaos) Testing
- AI/ML
- Advanced “Near Real Time” Test Reporting Systems that Scale
- GPU Powered Databases

Hello

- John Ruberto

John Ruberto takes an engineering approach to building quality into software. He teaches leadership principles applied to quality at the Software Leadership Academy, his blog, and as conference speaker.

John is currently working at Testlio, a crowd-sourced testing provider as QA project manager and solutions expert. He has previously held positions up to VP of Quality Engineering at First Data, Concur, Intuit, Alcatel, PhoenixBIOS, and Boeing. He holds a BSEE from Purdue, MSCS from Washington, and MBA from SJSU. You can find John at @JohnRuberto or blog.ruberto.com.

GUEST PANELIST



Andrew Csontos



John Ruberto



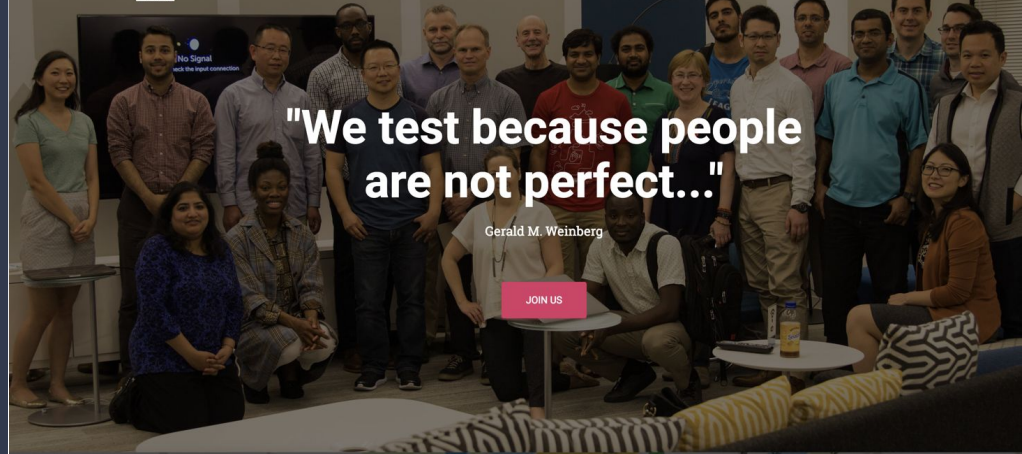
Edward Peterson

DCAST

DC Agile Software Testing Meetup



HOME JOIN DCAST HAPPENINGS RESOURCES SPONSORS



"We test because people
are not perfect..."

Gerald M. Weinberg

JOIN US

DC Agile Software Testing Group (DCAST)

The largest testing meetup in DC, we're a group of software quality and testing enthusiasts from all walks of life. DCAST was created to develop a community of those passionate about software testing driven to learn how to test products more effectively and efficiently. If you're interested in sharing how you have raised the quality bar of your agile processes or are looking to learn from others, this community is for you. Join us for more conversations on Slack to collaborate and find local support on all your testing needs.

DCAST

Round Table 3/27/2019

27
MAR

Past Meetup

QE Leadership Round table with special guest Andrew Csontos!



Hosted by Peter Kim

From DC Agile Software Testing Group (DCAST)

Public group

Details

First things first...DCAST is extremely pleased to have on our panel - Andrew Csontos, Head of Engineering, at Social Tables. Andrew's wealth of experience, from successfully leading startups to Fortune 100 companies, delivering global solutions and advanced enterprise software will give our QA/QE Agile meetup a perspective and insight that's rare.

... but wait! There's more..

We are also excited to have John Ruberto on our panel! John has previously held positions up to VP of Quality Engineering at First Data, Concur, Intuit, Alcatel, PhoenixBIOS, and Boeing. John is currently working at Testlio as QA project manager and solutions expert.

... still more!

Edward Peterson, VP of Engineering at Xandr, will be joining our panel - sharing his insight from past roles as VP of Quality Engineering and championing quality across teams.

Phew ... now

Discussion ..

Imagine your leading the quality engineering team to success - ensuring you're confidently releasing your product with a reliable yet dynamic process. However, now you're realizing there are more and more issues that are being reported by your customers rather than the QA/DEV team.

This is the reality facing QA teams, this is not a surprise. However, how QE leadership reacts and manages these inflection points are a surprise. .. with different outcomes.

The Round Table

- QA/QE/SDETS from the DC AREA
 - VP of Engineering (Xandr/AT&T)
 - Senior Dir. of Engineering Capital One
 - Head of Engineering at Social Tables
 - Former VP of Quality Engineering Intuit/Clover
 - Dir. of Quality Engineering (Kinetica/NPR)
 - QA Managers (Appian)
 - Principal Solutions Architects / Tech Leads (Coveros)
 - SW Developers*
-
- Govt.
 - DoD
 - Startups
 - Fortune 50/500
 - Hiring/Consultants (TrustedQA/Coveros)

DEV doing QA?

- What is DEV doing to improve quality?
 - DEV is being tasked to do more testing beyond UT
 - “DEV must uplevel their work and be more responsible for testing.”
- What’s driving the shift from testing from QA to DEV (aside from being “Agile”)?
 - “Perceptions” that QA isn’t effective.
 - Customers are consistently finding feature failures in PROD.
 - Turn around time (release acceptance testing) is too slow.
- QA/QE are being “converted” to DEV .. but focusing on testing.

DEV doing QA?

“The mindset that I can write something and hand it off to someone to test” is broken. You have to have ownership on the software you write.

*DEV

What did DEV share with QA ?

- **DEV**

We just don't want the same thing to be found twice. We should not be doing root cause analysis on the same thing twice.

- **To QA:**

QA needs to explain that .. when it happens.

*DEV

.. advice to QA from VPE (1 of 3)

If you want to **grow** your career ...

Do learn automation and **do learn** to program. You don't need to be as “good” as a software engineer but you must understand the software language they're using. The people we hire understand the basics of code and then understand the why of the product.

“Why are we trying to build ... what we are trying to build?”

.. advice to QA from VPE (2 of 3)

Trying to understand how to automate software testing, you must understand 4 major topics:

- data structures
- algorithms
- design patterns
- debugger of the programming language your organization uses

.. interview stories shared with QA from VPE (3 of 3)

.. after numerous SDET interviewees

QA engineers showing how they can leverage a cookie-cutter approach to automate tests, but lacking in structuring/designing basic code.

Automation should not be limited to “sequential commands”.

If SDET's are writing software, shouldn't they be held to the same standards, scrutiny, and practices of software design engineers too?

DEV doing QA?

“The mindset I can write something and hand it off to someone to test” is broken. You have to have ownership on the software you write.

It's up to the team, no longer to scapegoat QA, because QA will never think of every test.

Leadership must create/support a culture where ... the entire team has to be responsible.

Facing challenges .. or fear?

This Q raised one of the biggest fears from QA.

This issue directly asks .. “How effective is the QA team?”

Q: Why are defects missed by QA .. yet our customers are reporting them in PROD ?

Q: Why are defects missed by QA .. yet our customers are reporting them in PROD ?

QA:

We hear about companies that do continuous testing, continuous integration, automated testing, but manual testers (and customers) still find issues. Are we hearing a false story here?

Q: Why are defects missed by QA .. yet our customers are reporting them in PROD ?

QA:

We hear about companies that do continuous testing, continuous integration, automated testing, but manual testers (and customers) still find issues. Are we hearing a false story here?

CEOs are finding the defects - they find unexpected behaviors and question their existence.

Q: Why are defects missed by QA .. that our customers are finding defects in PROD ?

What can you say?

Q: Why are defects missed by QA .. that our customers are finding defects in PROD ?

- Resource constraints
 - QA team is too small
 - Need more time to test (execute Test Plan)
 - Testing is rushed
- Poor requirements/user stories
- Test plan was flawed
 - Reliability Testing was lacking
 - Focused on wrong test areas
- QA Environment / Build Issues

*Q: Why are defects missed by QA .. that our customers are finding defects in PROD ?

Leverage UAT - share the responsibility .. the product belongs to everyone.

- Involve Product Managers/Owners
- Involve DEV
- Key (Willing) Customers

*Q: Why are defects missed by QA .. that our customers are finding defects in PROD ?

As a QA leader, where, the last thing we should be is **defensive** if there's a bug report. Each one is an **opportunity** to diagnose the system, and each time we conduct root cause analysis answer **3 questions**:

1. Reach understanding as to “how was the bug introduced?”

*Q: Why are defects missed by QA .. that our customers are finding defects in PROD ?

As a QA leader, where, the last thing we should be is **defensive** if there's a bug report. Each one is an **opportunity** to diagnose the system, and each time we conduct root cause analysis answer **3 questions**:

1. Reach understanding as to “how was the bug introduced?”
2. How did the bug escape the checks and quality/**policy** gates?
 - code reviews
 - system reviews
 - UAT
 - etc

*Q: Why are defects missed by QA .. that our customers are finding defects in PROD ?

As a QA leader, where, the last thing we should be is **defensive** if there's a bug report. Each one is an **opportunity** to diagnose the system, and each time we conduct root cause analysis answer **3 questions**:

1. Reach understanding as to “how was the bug introduced?”
2. How did the bug escape the checks and quality/policy gates?
 - code reviews
 - system reviews
 - UAT
 - etc
3. What do we do differently?

*Q: Why are defects missed by QA .. that our customers are finding defects in PROD ?

What more can QA do?

GAP Analysis

- Where are our current test GAPS?
 - What tests are actually within scope?
 - What tests are out of scope?
 - What does the data/defects tell us?
 - Do we have a process for defect management?
 - Do we have a process to manage our GAPS?
- What's our quality gates actually covering?
- Are we doing enough testing for the other test types?
 - Restart
 - Configuration

*Q: Why are defects missed by QA .. that our customers are finding defects in PROD ?

What more can QA do?

GAP Analysis

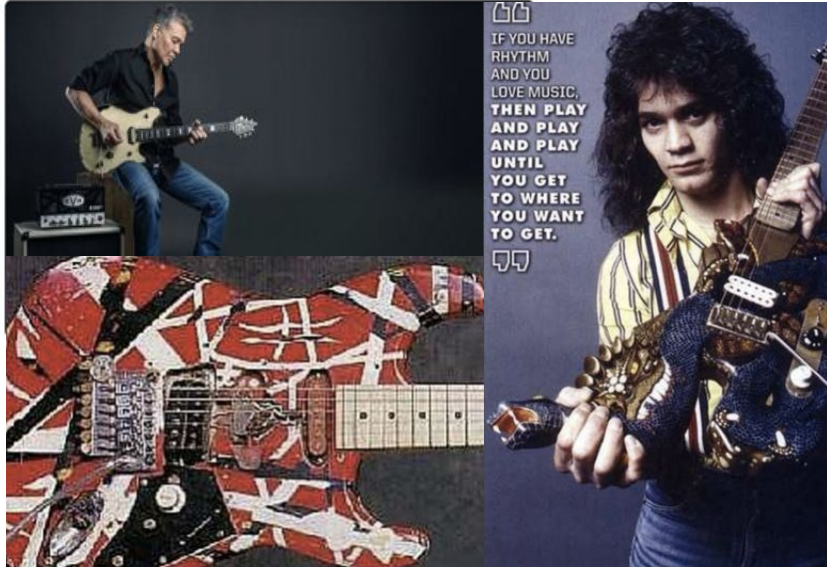
- Where are our current test GAPS?
 - What tests are actually within scope?
 - What tests are out of scope?
 - What does the data/defects tell us?
 - Do we have a process for defect management?
 - Do we have a process to manage our GAPS?
- What's our quality gates actually covering?
- Are we doing enough testing for the other test types?
 - Restart
 - Configuration
 - **Reliability/Rebustness**
 - **Load/Stress**

Passion for quality ... guitar legend EVH does QA

MusicRadar

Eddie Van Halen talks building the Frankenstein, honing the 5150 and EVH Gear

Guitar legend goes in-depth on the guitar gear behind his career (53 kB) ▾



We heard an interesting story about you bench testing the 5150 III with feedback...

"Oh yeah! I left it feeding back for a month! And then I put a bass through it and left it for another month, because I wanted a really low frequency to see how the amp and the cabinet would hold up. [I'd try] different feedback frequencies, really high, then I'd muffle the other strings once I got the note that I wanted it to feedback at, and I'd just leave it.

"I'll never forget, we had a Fender meeting with the powers that be to talk about something, and we're walking up the hill to the studio here and they hear this, 'Ooooh'. Then we open the door and it's 'OOOOOH' then we open another door and it's just fucking screaming!

"And they all went 'What the fuck are you doing?' and I'm like 'I'm crash-testing the amp!' I don't like stuff to blow up. So then I take it out on tour for a whole cycle before it was released to the public. Nothing goes out until I've totally fucking crash-tested it."

So did it break down?

"No, it held up!"

QA is a place where it's easy to lose ground and hard to regain it. When things are going down hill, how do you regain that ground?

You have to get good data because your decisions are data driven. You're going to have to make some choices, you have to take the number of bugs and break it up into a priority list that is **relevant to the business impact**.

Take that data and translate it into real dollars **lost**, or shifted costs elsewhere (support/dev \$ increases). Consider scoping those bugs against the most important customers:

If customer X sees this, would we keep risk our contract?

QA is a place where it's easy to lose ground and hard to regain it. When things are going down hill, how do you regain that ground?

You need to make sure your priorities reflect the best possible outcomes for the company.

How does QA gain the respect from DEV?

You shouldn't let this distract you. QA doesn't need to play up to DEV, to get their respect.

How does QA gain the respect from DEV?

Partner with product and be an expert in the business.

How does QA gain the respect from DEV?

You shouldn't let this distract you. QA doesn't need to play up to DEV, to get their respect.

Anytime you take part in a design discussion or requirements analysis, you can, with your expertise, eliminate problems before code is written and you **will be respected.**

How much effort into test plans?

How much effort into test plans?

Should they be really flushed out, or super basic?

How much effort into test plans?

Should they be really flushed out, or super basic?

- Writing test cases should consume < 20%
- How detailed? ... it depends

How much effort into test plans?

Should they be really flushed out, or super basic?

- Writing test cases should consume < 20%
- How detailed? ... it depends
 - If other people are using the assets you create you have to up your game on the detail

How much effort into test plans?

Should they be really flushed out, or super basic?

- Writing test cases should consume < 20%
- How detailed? ... it depends
 - If other people are using the assets you create you have to up your game on the detail
 - **The test cases belong to the stakeholders**

How much effort into test plans?

Should they be really flushed out, or super basic?

- Writing test cases should consume < 20%
- How detailed? ... it depends
 - If other people are using the assets you create you have to up your game on the detail
 - The test cases belong to the stakeholders
 - They need to be detailed enough so your team understands them but not so detailed that they break when things change

How much effort into test plans?

Should they be really flushed out, or super basic?

- Writing test cases should consume < 20%
- How detailed? ... it depends
 - If other people are using the assets you create you have to up your game on the detail
 - The test cases belong to the stakeholders
 - They need to be detailed enough so your team understands them but not so detailed that they break when things change
 - They cement the team into the plan, vs. being more agile and making it flexible so you can change it as you go ... adhere to the high level goals.

How much effort into test plans?

Should they be really flushed out, or super basic?

- Writing test cases should consume < 20%
- How detailed? ... it depends
 - If other people are using the assets you create you have to up your game on the detail
 - The test cases belong to the stakeholders
 - They need to be detailed enough so your team understands them but not so detailed that they break when things change
 - They cement the team into the plan, vs. being more agile and making it flexible so you can change it as you go ... adhere to the high level goals.
 - **The things that are overlooked are the things the team doesn't experience - performance, robustness, etc.**

SPECIAL THANKS

DCAST - DC Agile Software Testing Meetup



Jennifer Galvin

@jenn_galvin



Peter Kim

@peter_kim777



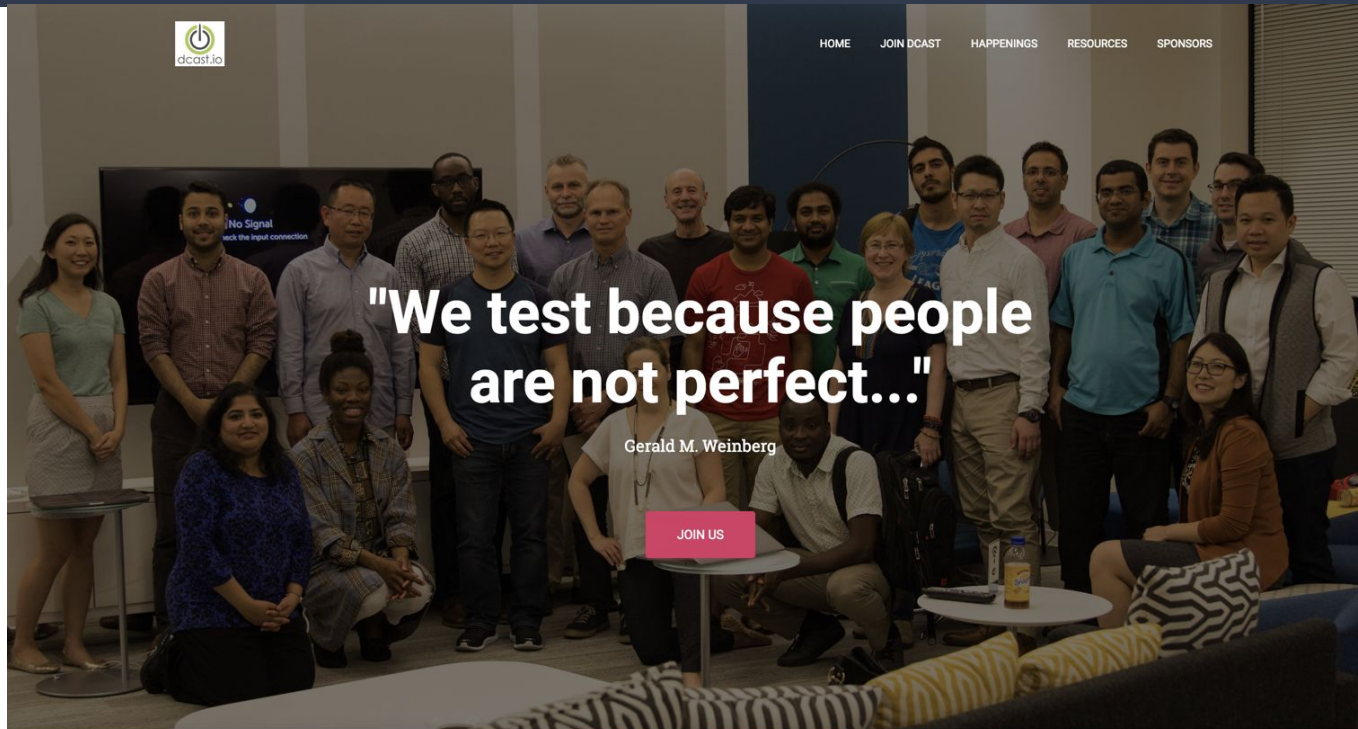
Gene Gotimer

@CoverosGene



Cynthia Wu

@eat_test_love



DC Agile Software Testing Group (DCAST)

SPECIAL THANKS



Andrew Csontos



John Ruberto



Edward Peterson

Peter Kim

LinkedIn: peterkim777

Twitter: peter_kim777

Email: h20dragon@outlook.com

dcast.io