

Eliminate the QA Role Completely (!) ~~Fire QA~~ to Lower Risk and Go Faster in Agile

Paul Carvalho

Gatineau-Ottawa Agile Tour
November 2019

Tweet your insights!: @can_agile #GOAT19



Background: About Me

- 30+ years in IT; many industries: Public Sector, Finance, Healthcare, Agriculture, Engineering, Logistics, Telecommunications, and more..
- 5+ years as a Programmer; Tech Support; Project Manager
- 20+ years in QA/Test-related roles, including:
 - Director of QA, Auditor (ISO, FDA), Tester, Test Designer, Performance specialist, Security certification, Internationalisation specialist, Exploratory Tester, Automation (web, API, mainframe)
- 10+ years as Agile Coach, trainer & consultant
 - Enterprise Coach; Build Digital Start-ups for Fortune 100 companies; DevOps Coach
- International speaker for 20 years on topics related to Quality, Agile, Development
- Give back to community: published articles, coach camps, mentor, conferences
- Science & Teaching (Education) degrees (former high school Science teacher)

© Paul Carvalho 2019

Thesis:

X: Eliminate your traditional testers and QA completely when moving to Agile Development

Y: So that you can (i) raise quality, (ii) lower product and system risk, and (iii) increase delivery speed

Z: Otherwise you will be stuck in waterfall QA practices, controls and blame culture that will **never allow you to** realize the **benefits of Agile**.

@can_agile #GOAT19

© Paul Carvalho 2019

Fail Recipe: Bad Agile Checklist:

We will:

- ☒ Change how the teams are arranged (co-located, ...)
- ☒ Hire and train Scrum Masters & Product Owners
- ☒ Ask programmers to use new tools, pair, work with Business daily, deliver more frequently
- ☒ Change how the teams manage work, communicate
- ☐ Let the Testers keep using their same old Waterfall, independent testing tools and practices



@can_agile #GOAT19

© Paul Carvalho 2019

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Most of your agile team (Green box with green checkmarks over the left items)

Your Testers - WHY?!? (Red box with red X marks over the right items)

@can_agile #GOAT19 © Paul Carvalho 2019

History: Why Do We Have QA?

- Traditional [independent] Testing and QA is a form of Quality **Control**
- It is a *reactive* activity that only identifies mistakes *after* you make them
 - In *Lean Manufacturing*, this is “waste”
 - **Question: what is the cost here?**
- QA and Testing are the primary control mechanisms (the eyes and hands) for Governance, Audit & Regulatory teams
 - **Question: Why do we have Governance, Audit & Regulatory?**

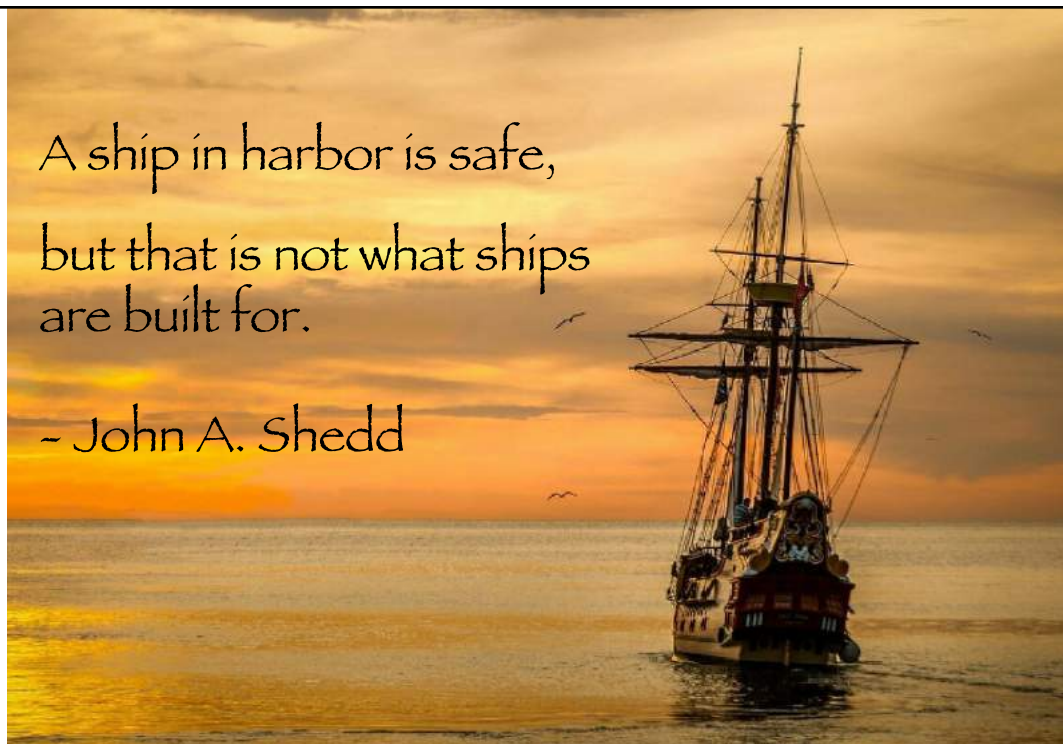


@can_agile #GOAT19

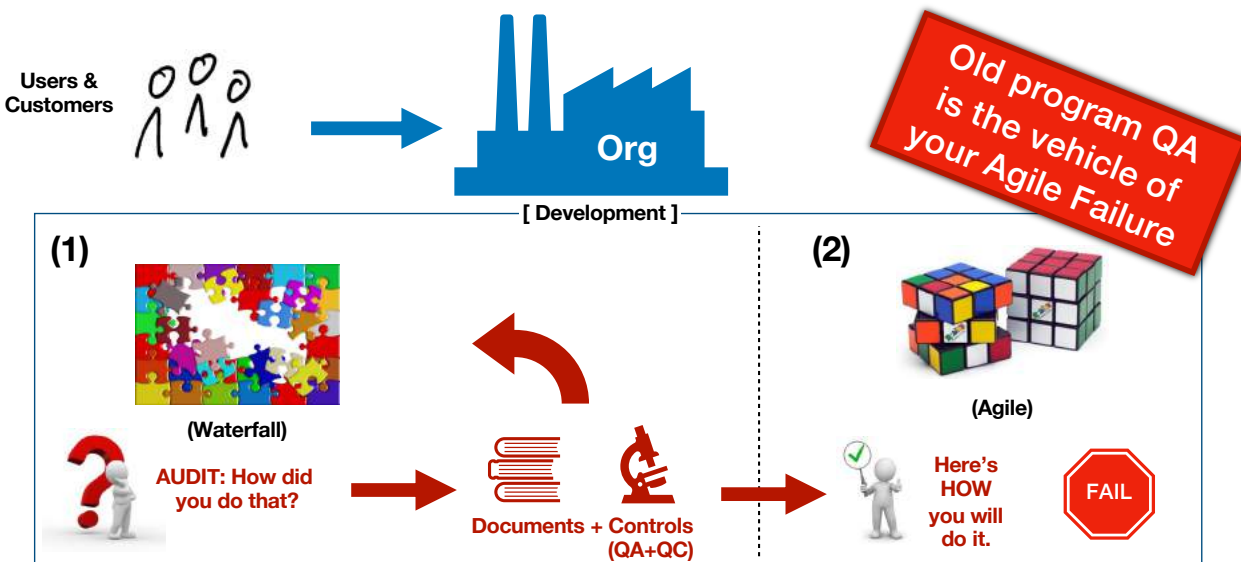
© Paul Carvalho 2019

A ship in harbor is safe,
but that is not what ships
are built for.

~ John A. Shedd



How Does Audit (& QA) Work?



@can_agile #GOAT19

© Paul Carvalho 2019

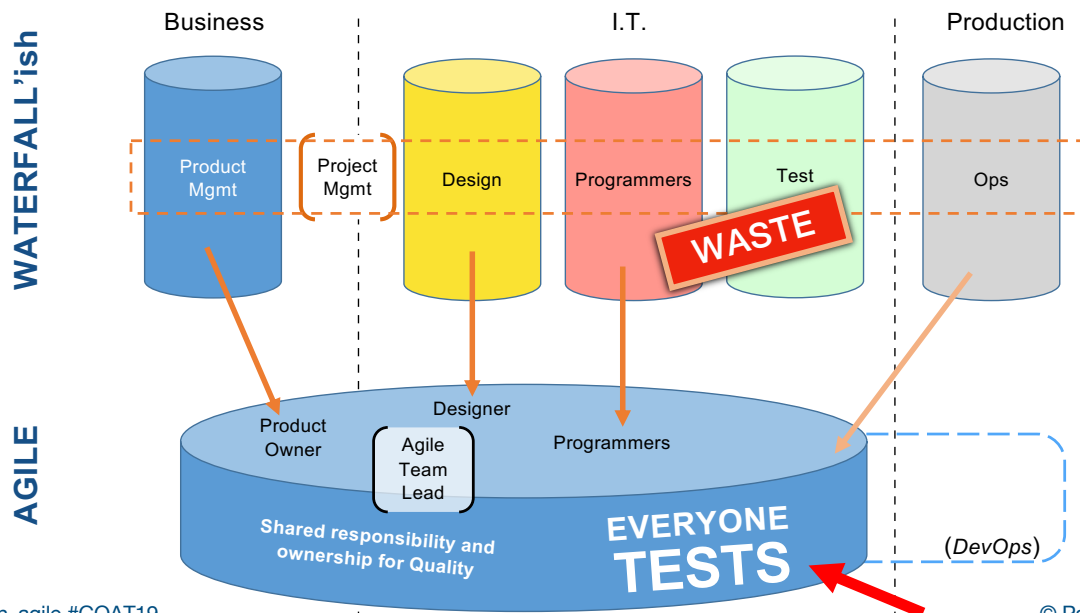
How Does Audit (& QA) Work?

Problem 1: Audit creates an organisational system structure that prevents agile teams from collaboration and addressing risk needs in new ways.
Traditional QA == Traditional Project Mgmt
=> SYSTEM BROKEN.

@can_agile #GOAT19

© Paul Carvalho 2019

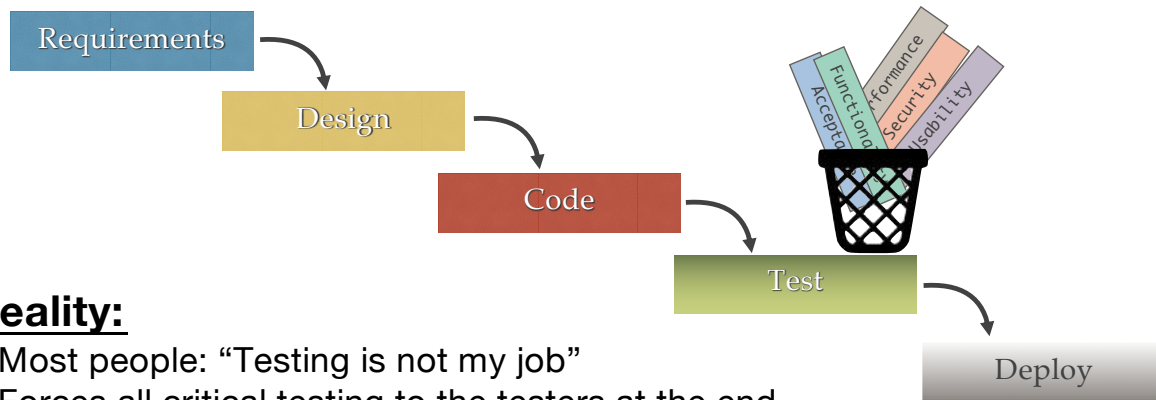
How Does IT Work?



@can_agile #GOAT19

© Paul Carvalho 2019

Responsibility for Quality in Waterfall



Reality:

- Most people: “Testing is not my job”
- Forces all critical testing to the testers at the end
 - Irresponsible; Scapegoat culture

@can_agile #GOAT19

© Paul Carvalho 2019

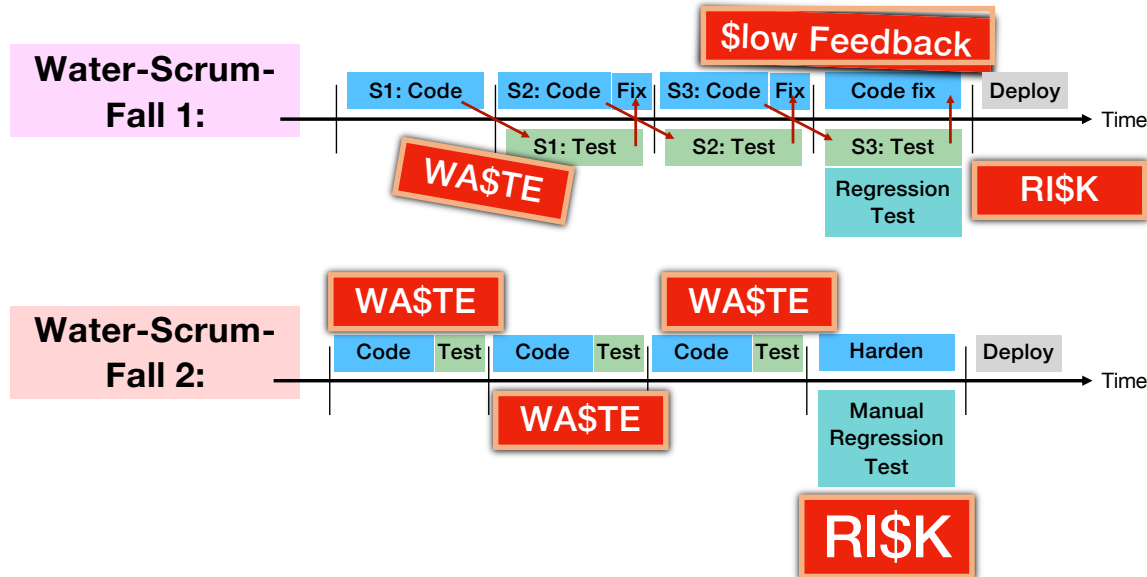
Responsibility for Quality in Waterfall

Problem 2: Blame and Scapegoat culture.
 In the original Waterfall model, Quality and Testing are owned at every phase. In practice, no one does it that way.
 => YOU AREN'T EVEN WATERFALL!

@can_agile #GOAT19

© Paul Carvalho 2019

What Does Popular Bad Agile Look Like?



@can_agile #GOAT19

© Paul Carvalho 2019

What Does Popular Bad Agile Look Like?

Problem 3: Stop allowing programmers to put bugs *into* the code.

Manual testing afterwards == slow feedback and deployments, expensive time & effort

Manual Regression test at end is always too short (time-boxed) => Many tests skipped

=> INTRODUCES RISK

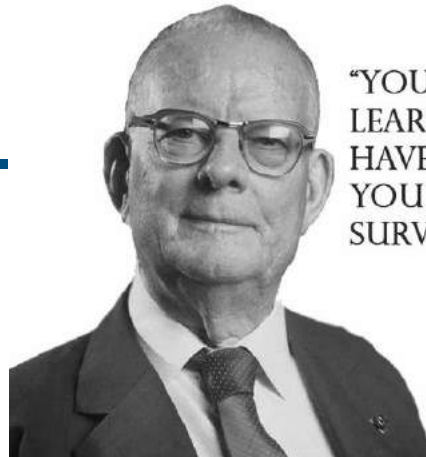
=> GUARANTEES POOR QUALITY

@can_agile #GOAT19

© Paul Carvalho 2019

Q: What is the Alternative to Traditional Testing and QA Practices?

A: Build Quality In.



"YOU DON'T HAVE TO LEARN NOR DO YOU HAVE TO CHANGE. YOU DON'T NEED TO SURVIVE EITHER!"

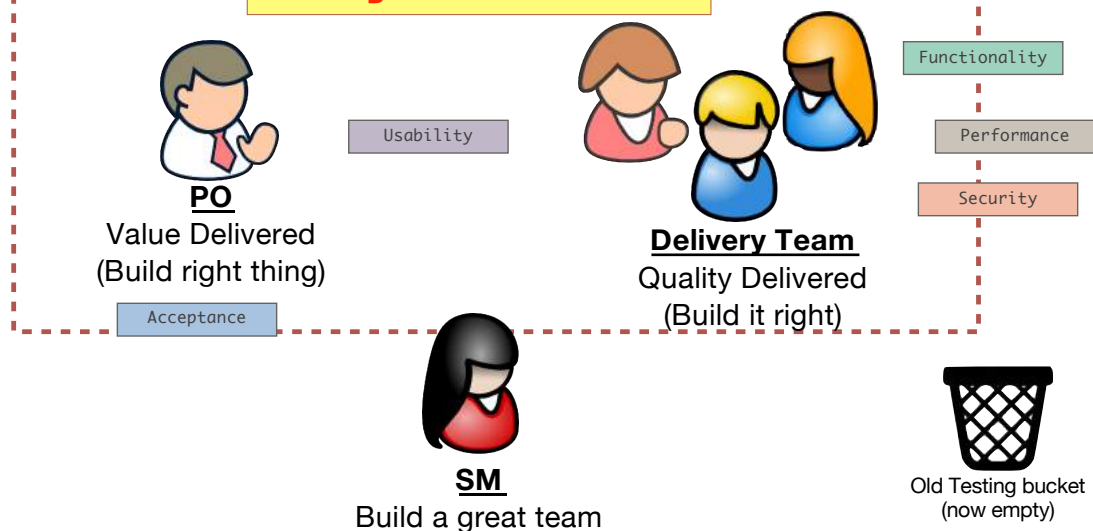
~W. EDWARDS DEMING



@can_agile #GOAT19

Agile Responsibilities

Everyone is a Tester!



@can_agile #GOAT19

© Paul Carvalho 2019

Agile Responsibilities

**Problem 4: Nothing left for traditional QA/Tester to do on a real agile team.
You aren't a programmer.
What value do you bring?
You will only slow everyone down.**

@can_agile #GOAT19

© Paul Carvalho 2019

**So, what would happen
if we get rid of QA?**

@can_agile #GOAT19

<https://spectrum.ieee.org/view-from-the-valley/computing/software/yahoos-engineers-move-to-coding-without-a-net>



@can_agile #GOAT19

Article highlights:

- Software engineers at Yahoo are no longer permitted to hand off their completed code to another team for cross checking.
- The code goes live as-is; if it has problems, it will fail and shut down systems, directly affecting Yahoo's customers.
- [This] forced engineers to develop tools to automate the kinds of checks previously handled by teams of humans.
- The shift wasn't easy.. It required some tough parenting, with no exceptions.
- "We said 'No more training wheels,' and it made a huge difference. We forced excellence into the process."
- "It turns out that when you have humans everywhere, checking this, checking that, they add so much human error into the chain that, when you take them out, even if you fail sometimes, overall you are doing better."

Recap of What We Know

- We know the Agile Engineering models to develop better software faster (e.g. Extreme Programming, Continuous Delivery, etc.) => Use them.
- Need new management expectations on Programmers/Developers
 - No more training wheels - take responsibility for the quality of your code
 - No exceptions
- Humans in the process introduce errors. Traditional QA practices forces bad Waterfall => You will never be agile with traditional testers and QA on your team.
- Leadership and Audit needs to build a new organisational system that aligns with Agile ways of working
 - Partner with your agile teams to understand the new ways of delivering high quality systems with confidence

@can_agile #GOAT19

© Paul Carvalho 2019

So what's the downside?

*Oh, right. Testers & QA people
lose their jobs.*

@can_agile #GOAT19

What Do You Think?

*Q: What should people in traditional
tester/QA roles do if they lose their
[waterfall, unnecessary] jobs in agile
teams?*



@can_agile #GOAT19

Further Reading

Articles and experiences that formed the basis for this presentation:

- Can Governance and Agile Get Along? (July 2017)
 - <https://www.projectmanagement.com/articles/391738/Can-Governance-and-Agile-Get-Along->
- Agility and Quality Start with No Manual Test Cases (April 2018)
 - <https://www.projectmanagement.com/articles/423702/Agility-and-Quality-Start-with-No-Manual-Test-Cases>
- Deploy Faster by Getting Rid of End-of-Development Testing (November 2016)
 - <https://www.projectmanagement.com/articles/352184/Deploy-Faster-by-Getting-Rid-of-End-of-Development-Testing>
- Four Elephants in the Agile Testing Room (March 2016)
 - <https://www.projectmanagement.com/articles/324099/Four-Elephants-in-the-Agile-Testing-Room>
- Agile Testing for Rapid Feedback (June 2015)
 - <https://www.projectmanagement.com/articles/296721/Agile-Testing-for-Rapid-Feedback>

© Paul Carvalho 2019

What Questions Do You Have For Me?



Contact Me:

Paul@Quality-Driven.com

www.Quality-Driven.com

Twitter: @can_agile

