How many DevOps do you need?

DevOps for Testers

(or DevTestOps for everyone)

Overview

- Collaboration (session 1)
  - Breaking down the structure we've created for ourselves
- Automation and processing change (session 2)
  - With an example

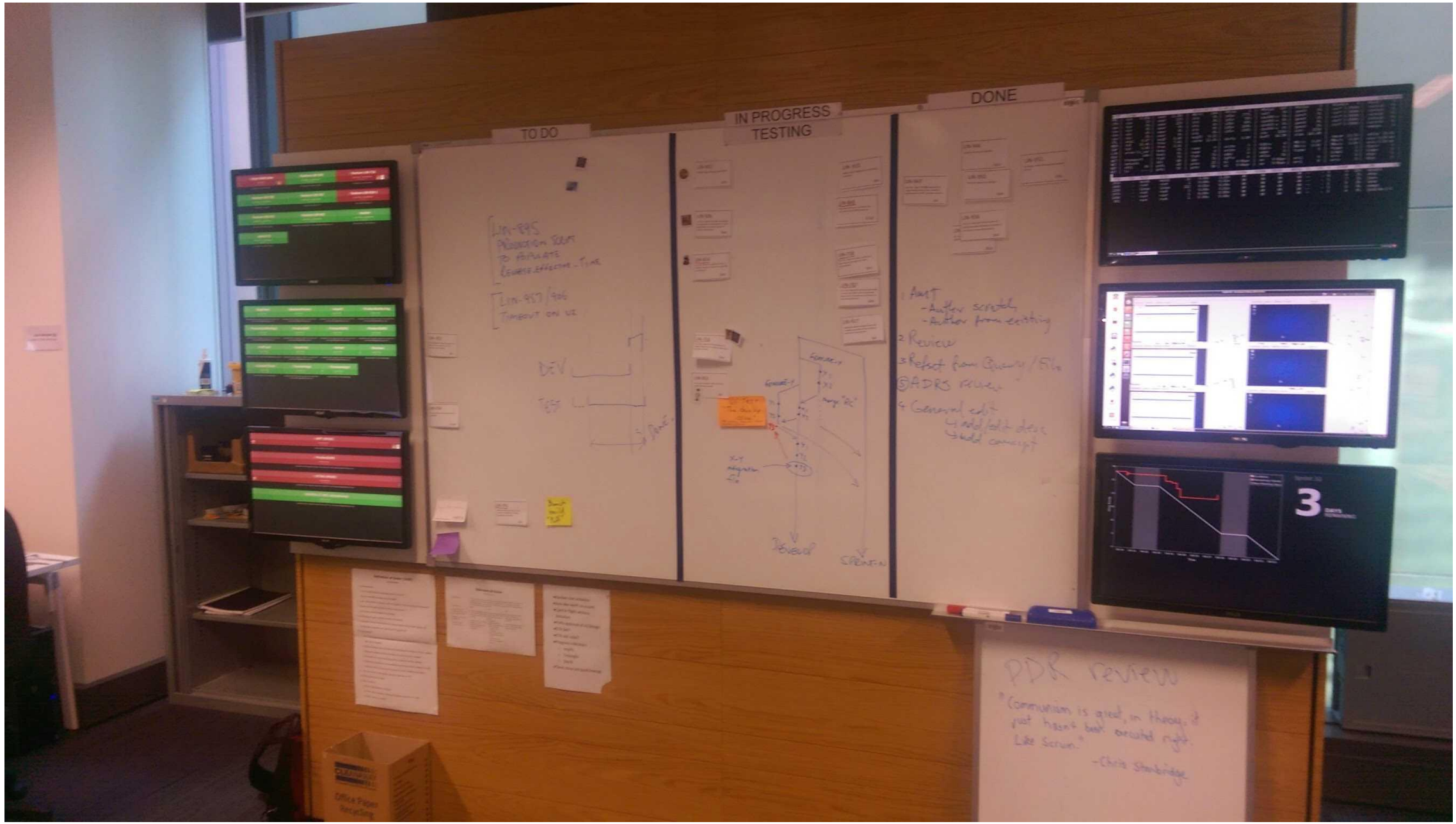
- DevOps for testers: The ways of working needed by the testing fraternity to address the changes in our industry brought through by the DevOps movement.

- This talk will put forward a case that testers can benefit greatly from understanding the process of releasing reliable software through build, test and deployment automation.

More than this… I actually think DevOps works best with Test

Trust me, I’m a test guy

- Google search : “DevOps for Testers”
- I have a personal perspective
- A broad & high level talk for this format
- Please hit up my colleagues for detail ☺


TO DO
IN PROGRESS
TESTING
DONE
DEV
TEST
Done
Review
Refact from Query / Fix
ADRS review
General edit
add/edit desc
add concept
PDR review
"Commission is great, in theory. It just hasn't been executed right. Like Scrum."
-Chris Shenbridge
3

What is it, again?

"a cross-disciplinary **community** of practice dedicated to the **study** of building, evolving and operating rapidly-changing resilient systems at scale."

"It's a **philosophy** of the close interaction between ops and developers"

DevOps is the practice of operations and development engineers **participating together** in the entire service lifecycle, from design through the development process to production support.

"The **union** of people, processes and technology to deliver continuous value to customers"

DevOps is the combination of **cultural philosophies, practices, and tools** that increases an organization's ability to deliver applications and services at high velocity.

"DevOps is a **set of practices** that automates the processes between software development and IT teams, in order that they can build, test, and release software faster and more reliably."

Value Propositions

- “Increase in flow of Value and Quality”
- “Better Value Sooner Safer Happier”
- “Productivity and Quality improvements”
- “Develop software with maximum efficiency”
- “Less complexity to manage”
- ….

Why might you want it…

- Testing fewer things per release is awesome
- Fewer combinations, more certainty
- Testing different things all the time
- Much less regression testing
- More time to get into a wider variety of things to test:
  - UX
  - Infrastructure
  - Accessibility
  - Performance
  - Security
  - …

Get towards Continuous Deployment

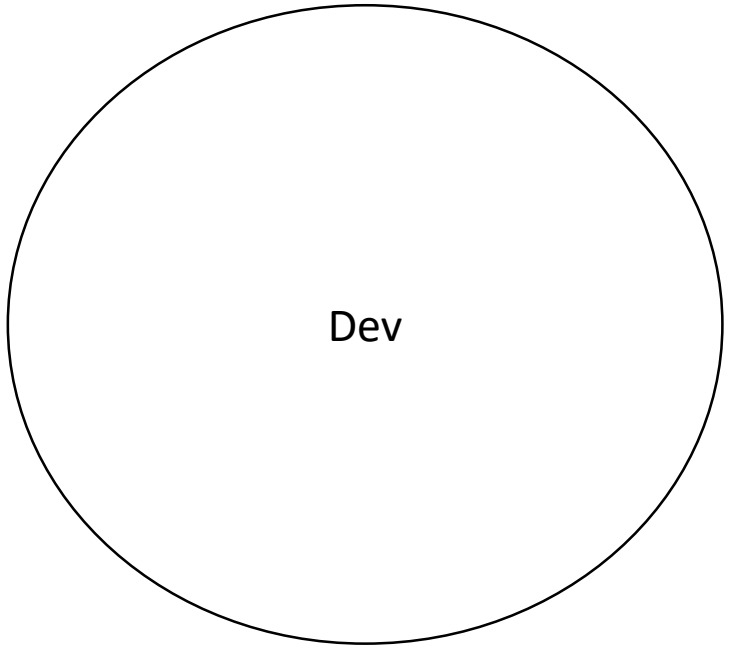
- High-performing IT organizations deploy **30 times** more frequently with **200 times** shorter lead times
- Etsy deploys to its production servers **50 times** a day
- Amazon engineers deploy code **every 11.7 seconds** on average
- Netflix engineers deploy code **thousands of times** per day
- …

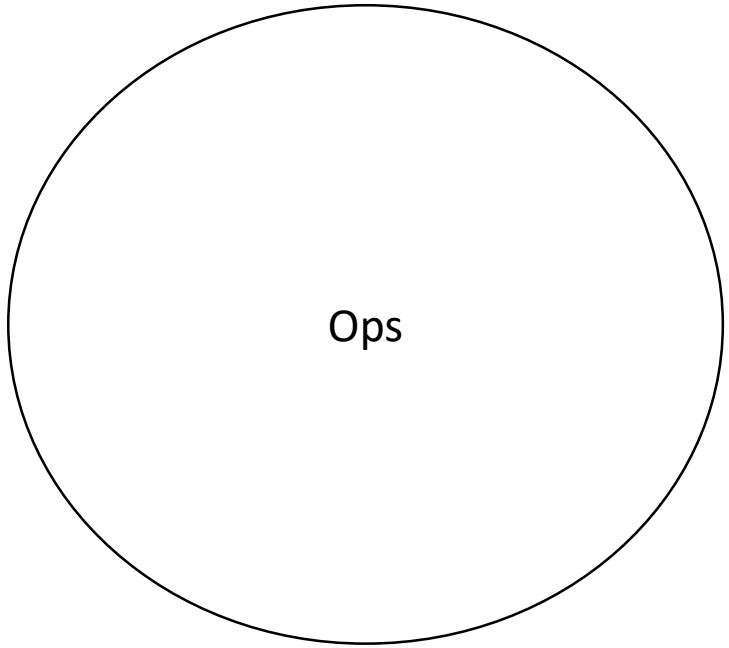
https://blog.newrelic.com/technology/data-culture-survey-results-faster-deployment/

More Releases… So what?

1. Reduces risk
2. Real Progress: The notion of "done“
3. Learning loop from user feedback
    1. Canary Testing
    2. A/B
    3. Blue/Green

Imagine doing this in waterfall


Dev

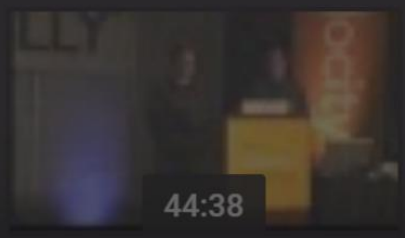

Ops

2009: John Allspaw and Paul Hammond

"10+ Deploys Per Day: Dev and Ops Cooperation at Flickr"

Little bit weird
Sits closer to the boss
Thinks too hard

Pulls levers & turns knobs
Easily excited
Yells a lot in emergencies



WHERE IS TEST?

…a sense of irascibility with real passion for life and doing the right thing.

- Wikipedia 2019

So what does **StarFleet** say about testers?

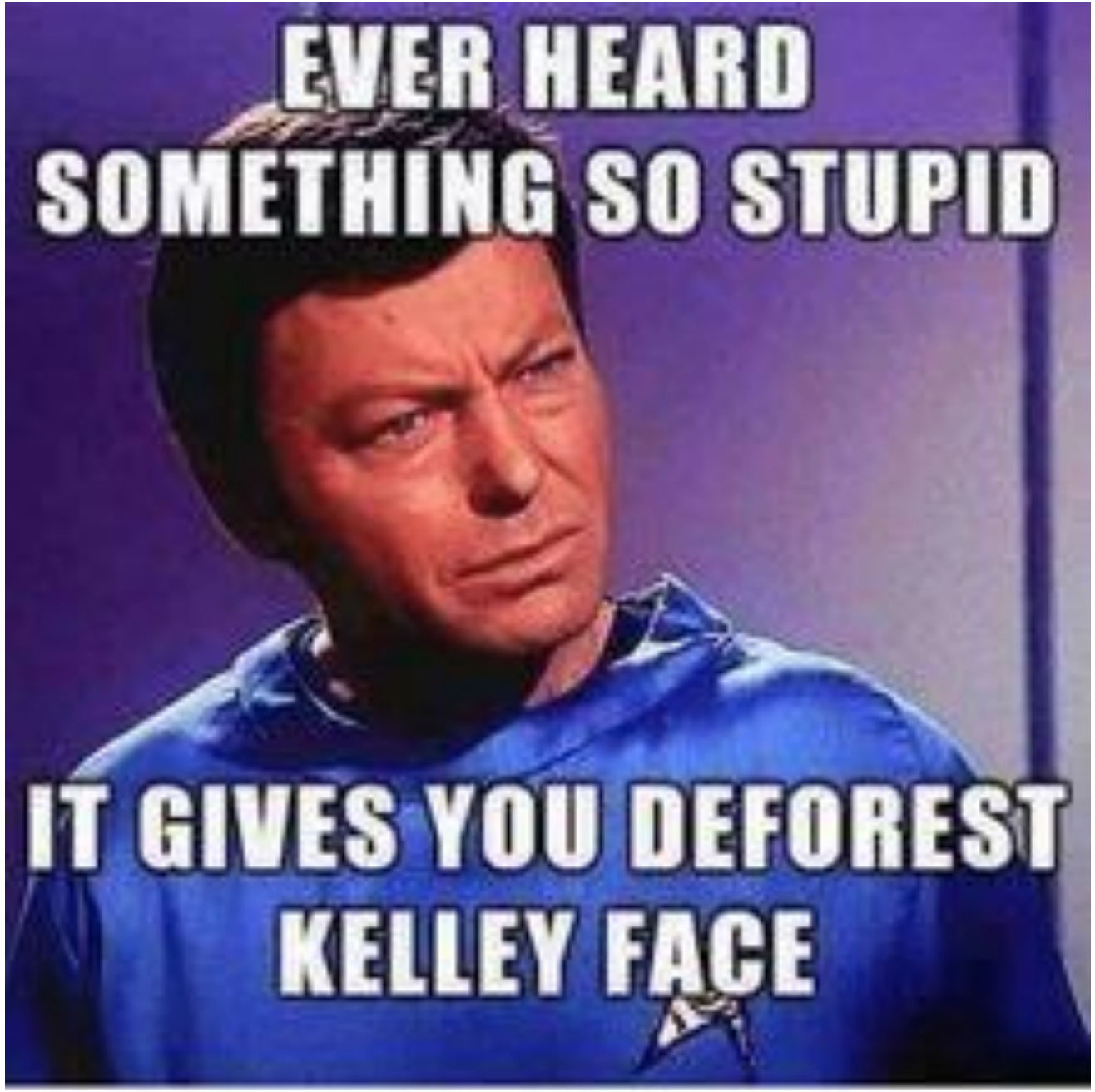
The 2018 “State of DevOps” report:

- “…testing and reuse of testing patterns can be one of the **harder challenges** to solve depending on your organization’s structure and complexity.
- …though we do see this practice adopted by highly evolved organizations in the early stages of a DevOps evolution, **it may not be the first thing you tackle**"

Then..

- “..If you have to prioritize, **we recommend waiting to tackle this one and focusing on the other practices first**."

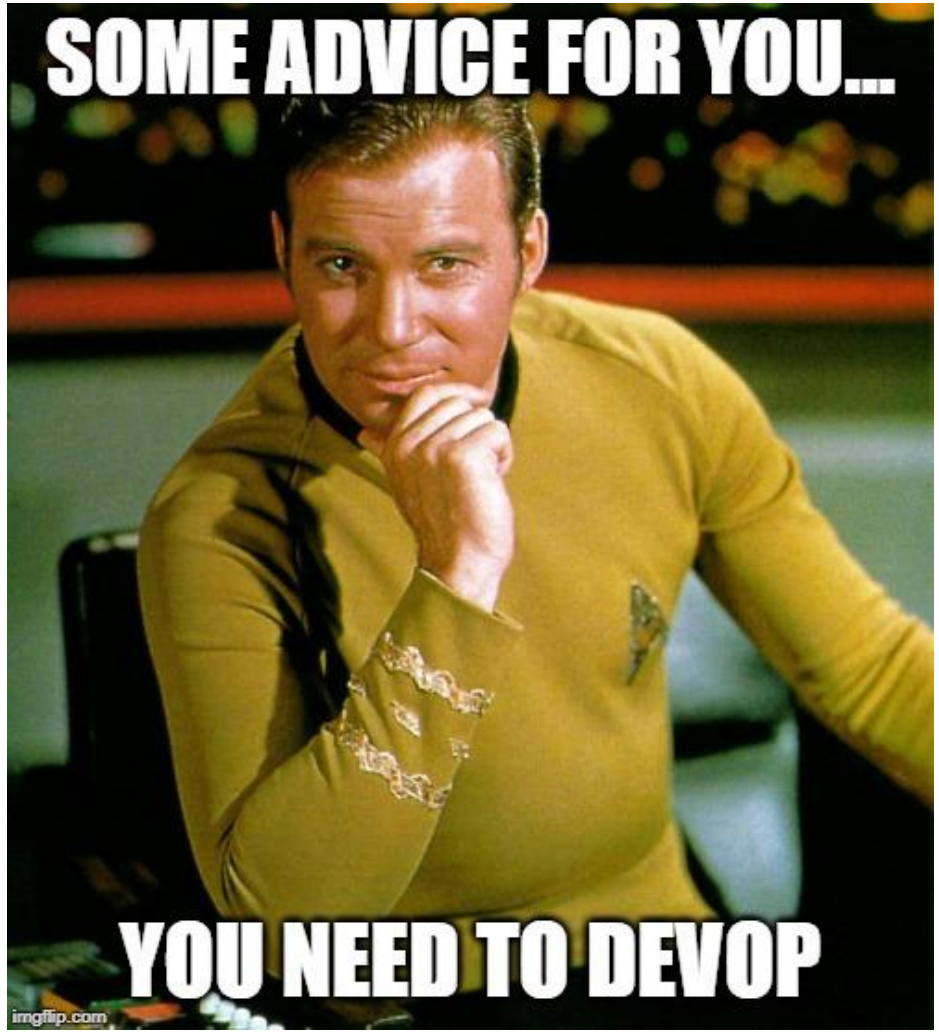
https://www.thinkahead.com/wp-content/uploads/2018/10/State-of-DevOps-Report.pdf


EVER HEARD
SOMETHING SO STUPID
IT GIVES YOU DEFOREST
KELLEY FACE

How about Leadership?

- If you're not yet supported by high level leadership, DevOps is just not going to work
- We're asking people to change the way they work, who they work with and who they work for

This is a **big** deal.



Bug Fix Bingo

Rules of the Game

1. Bingo squares are marked off when a developer makes the matching statement during bux fix sessions.
2. Testers must call "Bingo" immediately upon completing a line of 5 squares either horizontally, vertically or diagonally.
3. Statements that arise as a result of a bug that later becomes "deferred", "as designed", or "not to be fixed" should be reclassified as not marked.
4. Bugs that are not reported in an incident report can not be used.
5. Statements should also be recorded against the bug in the defect tracking system for later confirmation.
6. Any tester that marks off all 25 statements should be awarded 2 weeks stress leave immediately.
7. Any developer found using all 25 statements should be seconded into the test group for a period of no less than 6 months for re-education.

1. "It works on my machine."	2. "Where were you when the program blew up?"	3. "Why do you want to do it that way?"	4. "You can't use that version on your system."	5. "Even though it doesn't work, how does it feel?"
6. "Did you check for a virus on your system?"	7. "Somebody must have changed my code."	8. "It works, but it hasn't been tested."	9. "THIS can't be the source of THAT."	10. "I can't test everything!"
11. "It's just some unlucky coincidence."	12. "You must have the wrong version."	13. "I haven't touched that module in weeks!"	14. "There is something funky in your data."	15. "What did you type in wrong to get it to crash?"
16. "It must be a hardware problem."	17. "How is that possible?"	18. "It worked yesterday."	19. "It's never done that before."	20. "That's weird..."
21. "That's scheduled to be fixed in the next release"	22. "Yes, we knew that would happen"	23. "Maybe we just don't support that platform"	24. "Its a feature, We just haven't updated the specs"	25. "Surely nobody is going to use the program like that"

Read what people are saying about Bug Fix Bingo !!!

"Thanks Bug Fix Bingo, You have made defect review meetings fun again !!!"

"I used to fall alseep when meeting with developers, now with BFB, I anticipate every word"

http://www.kjross.com.au/bingo

Defects and Testers

- Testers have a complex relationship with Defects
  - We HATE them (see bug fix bingo)
  - We LOVE them (it is our KPI)
- We discourage change
- But the business needs change
- But change is the root cause of defects
- ARGH! Cognitive Dissonance ensues

Defect Classification

ANSI/IEEE Std 1044-1993 Standard Classification for Software Anomalies:

“Impact” can be classified by the following fields:

- Severity (5 levels)
- Priority (5 levels)
- Customer Value (6 levels)
- Mission safety (5 levels)
- Project Schedule (4 levels)
- Project Cost (4 levels)
- Project Risk (4 levels)
- Project quality/reliability (4 levels)
- Societal (4 levels)

Which means if you use all these fields to classify a defect, it will finally fall into 1 of **768,000** unique 'impacts'

How many times have you debated the “Priority" vs "Severity” vs “Impact”?

…History isn’t kind…

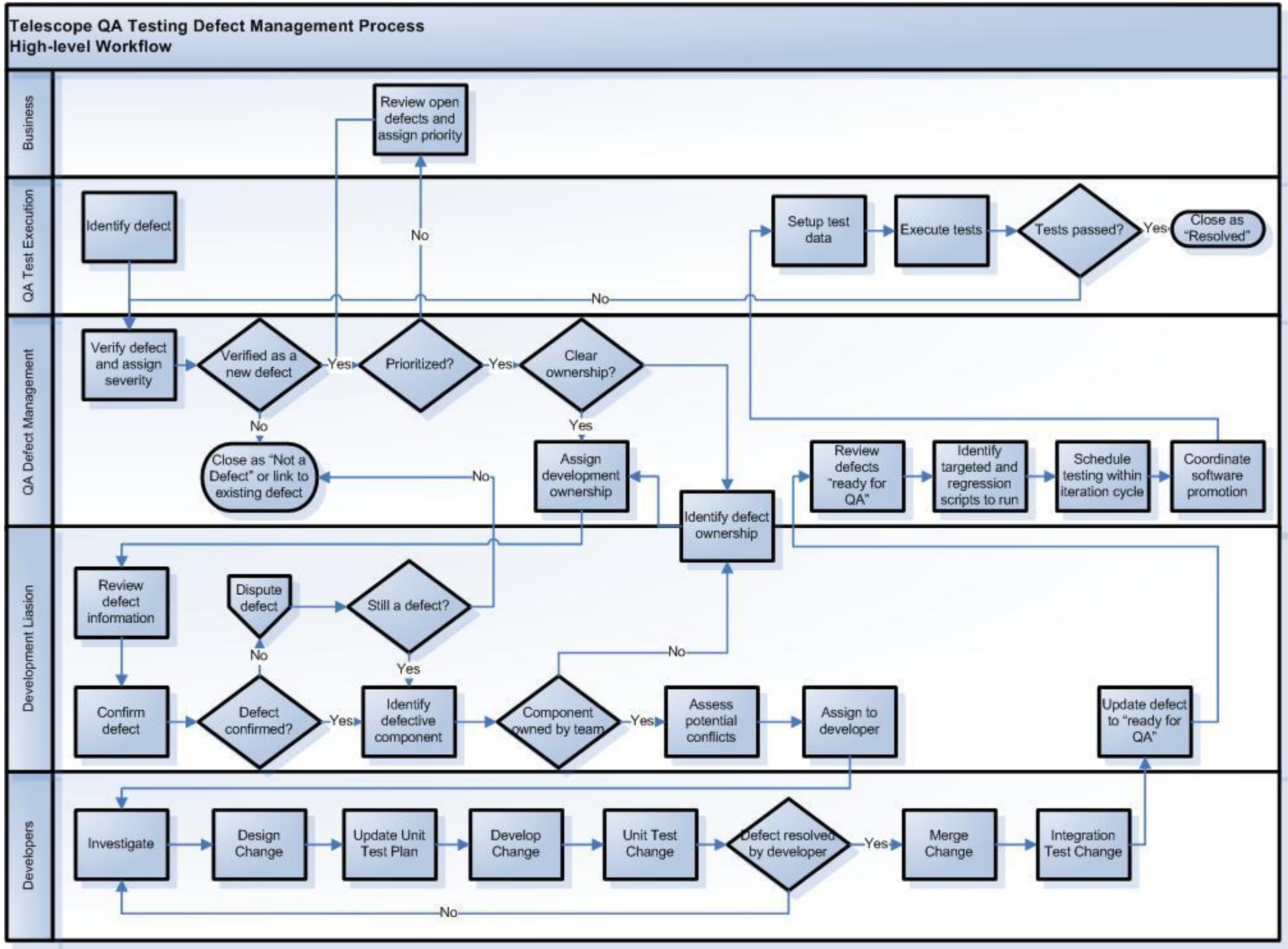
www.csiro.au

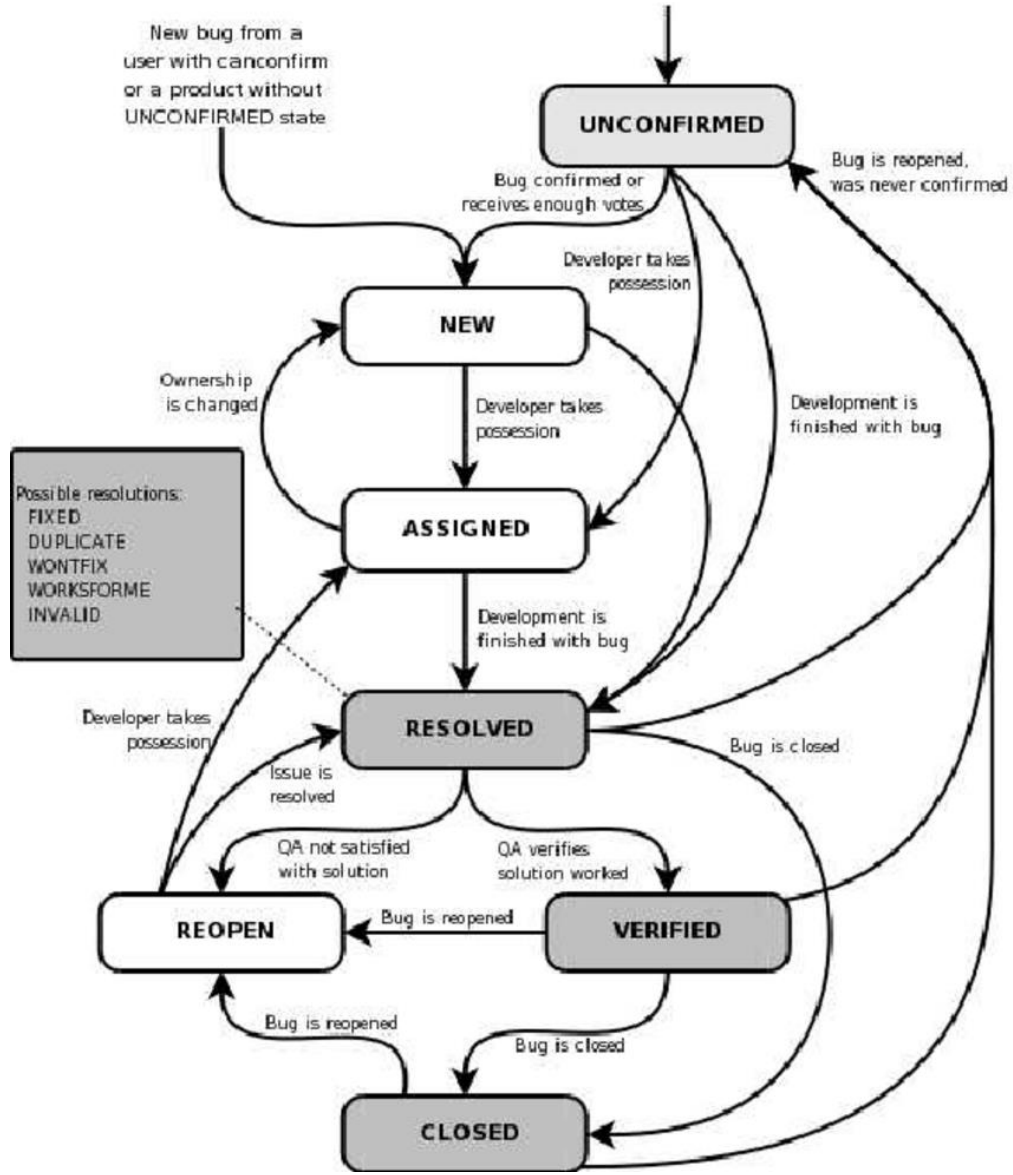
Triage Techniques for Defect Classification

Chris Stanbridge
Research Engineer (Software Quality) ,
The Australian e-Health Research Centre
ANZTB 2009

Donna Powell RN RM
Emergency Nurse
The Sunshine Coast Private Hospital
ANZTB 2009

CSIRO


Telescope QA Testing Defect Management Process
High-level Workflow
Business
QA Test Execution
QA Defect Management
Development Liasion
Developers
Review open defects and assign priority
Identify defect
No
Setup test data
Execute tests
Tests passed?
Yes
Close as "Resolved"
No
Verify defect and assign severity
Verified as a new defect
Yes
Prioritized?
Yes
Clear ownership?
No
Yes
Close as "Not a Defect" or link to existing defect
No
Assign development ownership
Identify defect ownership
Review defects "ready for QA"
Identify targeted and regression scripts to run
Schedule testing within iteration cycle
Coordinate software promotion
Review defect information
Dispute defect
Still a defect?
No
Yes
No
Confirm defect
Defect confirmed?
Yes
Identify defective component
Component owned by team
Yes
Assess potential conflicts
Assign to developer
Update defect to "ready for QA"
Investigate
Design Change
Update Unit Test Plan
Develop Change
Unit Test Change
Defect resolved by developer
Yes
Merge Change
Integration Test Change
No


New bug from a user with canconfirm or a product without UNCONFIRMED state
UNCONFIRMED
Bug is reopened, was never confirmed
Bug confirmed or receives enough votes
Developer takes possession
NEW
Ownership is changed
Developer takes possession
Development is finished with bug
Possible resolutions:
FIXED
DUPLICATE
WONTFIX
WORKSFORME
INVALID
ASSIGNED
Development is finished with bug
Developer takes possession
RESOLVED
Bug is closed
Issue is resolved
QA not satisfied with solution
QA verifies solution worked
REOPEN
Bug is reopened
VERIFIED
Bug is reopened
Bug is closed
CLOSED

With Test Case count as a KPI, we've been incentivised to type

Test Case Template

A test case can have the following elements. Note, however, that a test manage
normally used by companies and the format is determined by the tool used.

Test Suite ID	The ID of the test suite to which this test case belongs.
Test Case ID	The ID of the test case.
Test Case Summary	The summary / objective of the test case.
Related Requirement	The ID of the requirement this test case relates/traces to.
Prerequisites	Any prerequisites or preconditions that must be fulfilled prior executing the test.
Test Procedure	Step-by-step procedure to execute the test.
Test Data	The test data, or links to the test data, that are to be used wh conducting the test.
Expected Result	The expected result of the test.
Actual Result	The actual result of the test; to be filled after executing the te
Status	Pass or Fail. Other statuses can be 'Not Executed' if testing is performed and 'Blocked' if testing is blocked.
Remarks	Any comments on the test case or test execution.
Created By	The name of the author of the test case.
Date of Creation	The date of creation of the test case.
Executed By	The name of the person who executed the test.
Date of Execution	The date of execution of the test.
Test Environment	The environment (Hardware/Software/Network) in which the executed.

The STLC:

STLC Phases and Activities

Phase	Activity	Deliverables	Necessity
Requirements/Design Review	You review the software requirements/design (Well, if they exist.)	• 'Review Defect' Reports	Curiosity
Test Planning	Once you have gathered a general idea of what needs to be tested, you 'plan' for the tests.	• Test Plan • Test Estimation • Test Schedule	Farsightedness
Test Designing	You design/ detail your tests on the basis of detailed requirements/design of the software (sometimes, on the basis of your imagination).	• Test Cases/Test Scripts/Test Data • Requirements Traceability Matrix	Creativity
Test Environment Setup	You setup the test environment (server/ client/ network, etc) with the goal of replicating the end-users' environment.	• Test Environment	Rich company
Test Execution	You execute your Test Cases/ Scripts in the Test Environment to see whether they pass.	• Test Results (Incremental) • Defect Reports	Patience
Test Reporting	You prepare various reports for various stakeholders.	• Test Results (Final) • Test Metrics • Test Closure Report • Who Worked Late & on Weekends (WWLW) Report [Depending on how fussy your Management is]	Diplomacy

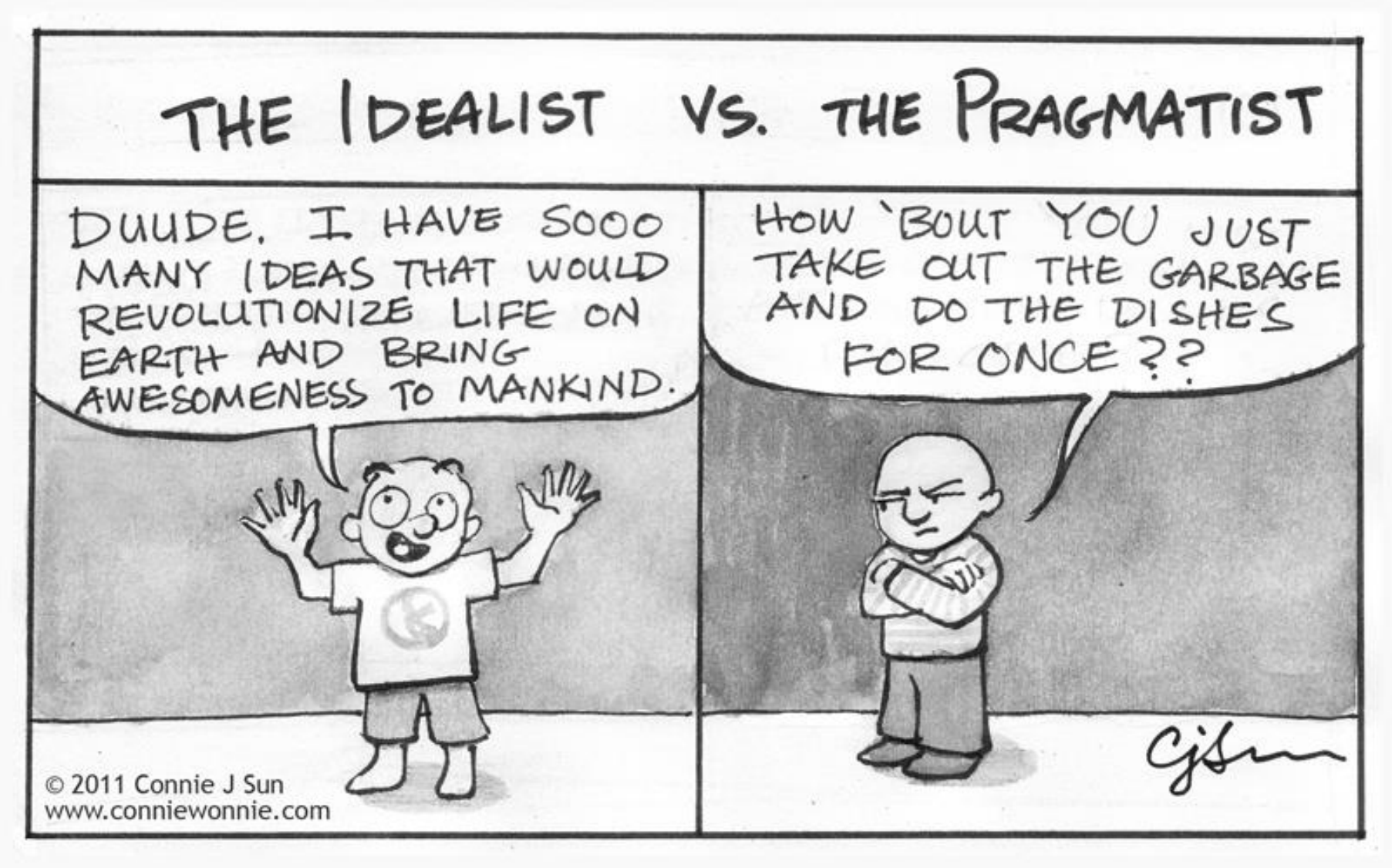
Test Case Management

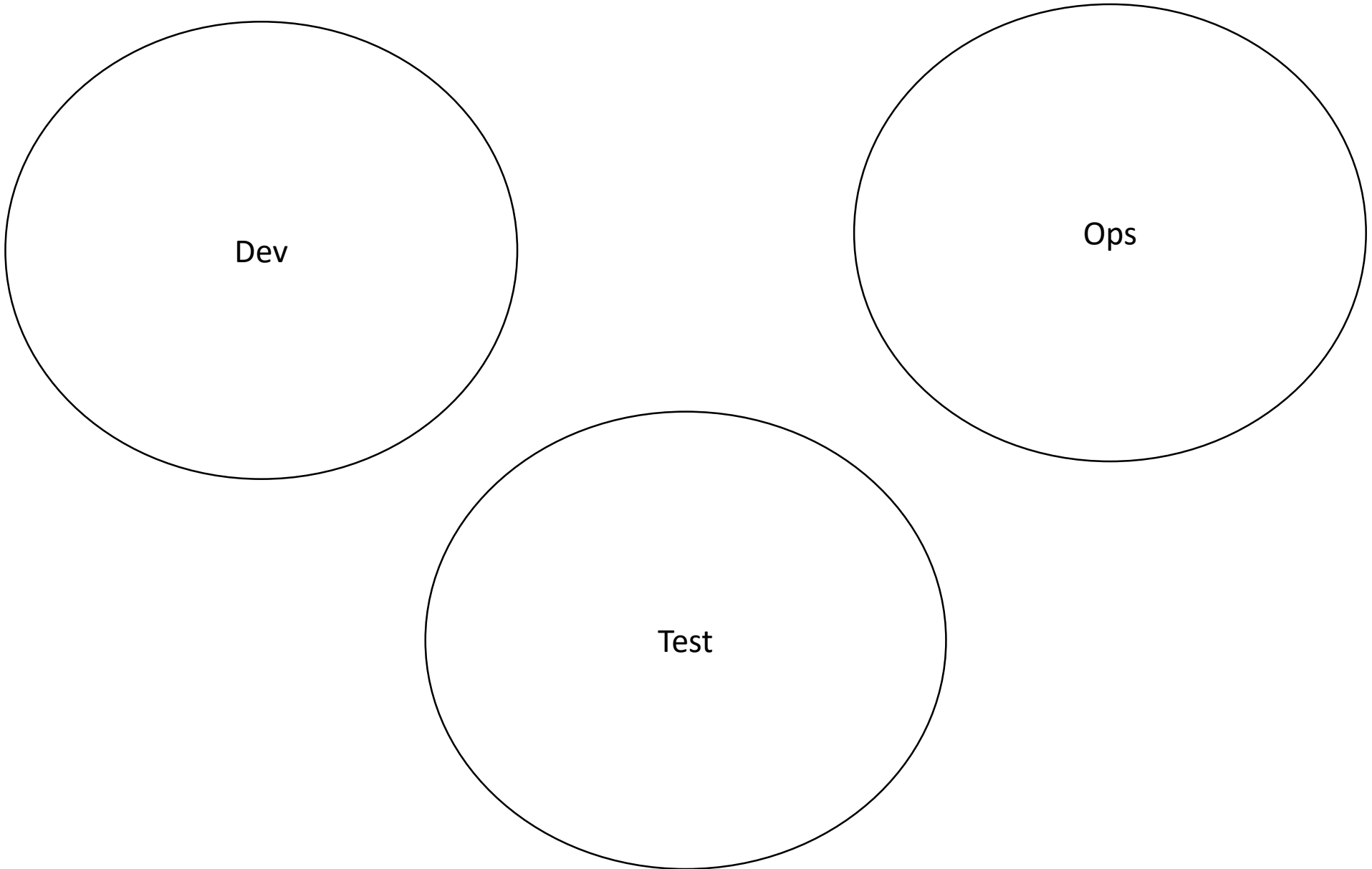
- By design, we have enforced a culture of independence in testing (the STLC)
- We have Centre's Of Excellence, bodies of authority (TMMI)
- Governance separate to the SDLC
- Independence is deemed as valuable because it allows untainted scrutiny of our (untrusted) peer's work

Current State

- Our persona for testing is missing!
- The DevOps authority is dragging it’s feet on testing
- Do leaders even support a change
- The structure we’ve built isn’t helping:
  - Defect Classification and workflow
  - Test Case Management
  - Software Test Life Cycle & TMMI
  - Value in Independence

Idealism?

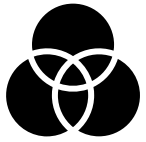

THE IDEALIST VS. THE PRAGMATIST
DUUDE. I HAVE SOOO MANY IDEAS THAT WOULD REVOLUTIONIZE LIFE ON EARTH AND BRING AWESOMENESS TO MANKIND.
HOW 'BOUT YOU JUST TAKE OUT THE GARBAGE AND DO THE DISHES FOR ONCE??
© 2011 Connie J Sun
www.conniewonnie.com


Dev
Ops
Test

IT'S SOFTWARE DEVELOPMENT JIM
BUT NOT AS WE KNOW IT
imgflip.com

What do we need to do?

- Break the Silos
- Use a common toolset
- Do automation (see later)
- Upskill
- Learn to love the change
- All those lovely Agile things (AKA Common Sense, Transparency, Collaboration…)
- And Venn diagrams ☺



Be Lean

- Test's job is to articulate confidence. Not to find bugs
- Let’s focus on quality software – find the cause of the bug and fix that
- We (the team) provide “informed consent” so that someone understands the risk

Take Control Of Our Stuff

- DevOps needs to have **control of code and tests and infrastructure**
- We are masters of our own domain

Let the Team Test!

- We all want change ‘done’
- Let the team work towards that goal – you’re the DevTestOps team!
- Inform, Check, Recommend, Supervise, Advise, Help – but also trust

Questions?