

Simplicity: The Path to Achieving Agile Testing Efficiency

CHICAGO

Renaissance Hotel
1 West Wacker Drive
Chicago IL 60601

April 18th — April 22th



Speaker(s): Paul Herzog
Company: SPR Consulting

When: April 20, 2016
Time: 10:00 am - 11:00 am

SIMPLICITY: THE PATH TO ACHIEVING AGILE TESTING EFFICIENCY QUEST 2016



SPR CONSULTING
POWERING BUSINESS WITH TECHNOLOGY

MY INSPIRATION FOR THIS TOPIC



WHY BE AGILE?

Why do businesses adopt Agile Methodologies?





They want to develop software products faster!

**With rare exception, EVERY business decision is about going faster!
They just call it “efficient” instead...”faster” implies “sloppy”.**



AGILE – A LEVEL SET

- Project management methods for software
- From Lean manufacturing
- Small, self-sufficient teams
- A manifesto was developed on four values:
 - Individuals & Interactions over Processes & Tools
 - Working software over Comprehensive documentation
 - Customer collaboration over Contract negotiation
 - Responding to change over Following a plan
- Twelve Principles are behind the Values



MY FAVORITE AGILE PRINCIPLE

Principle #10

“Simplicity – the art of maximizing work not done –
is **essential**”

As testers, without a defined role on an Agile Team, we must look for Principle #10 in every aspect of our job



LIVING PRINCIPLE #10

- What you do as a tester should be analyzed:
 - Change it?
 - Do less of it?
 - Remove it entirely?
- Focus on what's left at each Agile milestone:
 - Inception (“Sprint 0”)
 - Sprint 1 – Test Planning
 - Sprint <n> - Test Execution & Defects
 - Sprint Retrospectives – Process Improvements

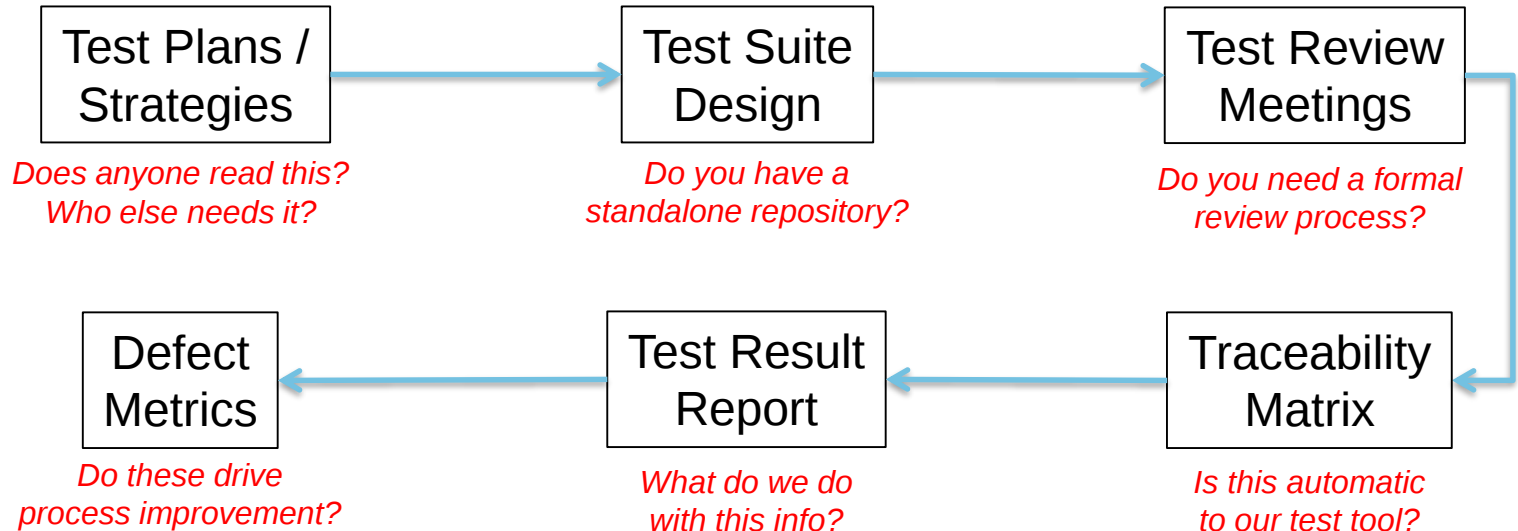


PROJECT INCEPTION: SPRINT 0

- Stakeholders create and estimate initial user story backlog
- What's Important:
 - Ask questions of stories and acceptance criteria
 - Plan test repository
 - Propose improvements to testing processes, milestones and deliverables
 - Identify project metrics – use the GQM model



SPRINT 0: ANALYZING DELIVERABLES



***If your team won't use it during the project...
and won't use it to improve later...
DON'T DO IT!***



BY THE WAY, DELIVERABLES ARE OK

- Agile DOES NOT MEAN “you don’t need or have documentation”
- Some projects greatly benefit from traditional documents
- Some industries require a high volume of documentation of every step of the process
- ...but documents are project overhead!



SPRINT 0: GQM

- Goals have:
 - Purpose
 - Issue
 - Object
 - Viewpoint
- Questions are about characterizing the objects
- Metrics are numbers to answer the questions



WHAT'S A VALUABLE METRIC?

Metrics must either:

Assist in day-to-day project management

Drive process improvements

Fulfill industry requirements



SPRINT 1: TEST PLANNING

- Get involved
 - DO NOT wait until Sprint Planning to see User Stories
 - Read...review acceptance criteria...estimate...digest...start thinking about how to test...ASAP
- Ask the right questions
 - Where are my data sources?
 - What APIs or other interfaces are implemented?
 - Schedule of deployments to a test environment?



SPRINT 1: DON'T SWEAT THE DETAIL

- Business-oriented testing, not functional
 - Don't wait for comprehensive requirements
- Why have detailed requirements?
 - Adds overall value, or just easier for you?
- Make less work for yourself...and others!



SPRINT <N>: TEST CASE CREATION

- Necessary component of the process
- Greatest time waster (if you do it wrong!)
- What's Important:
 - Clear detail for future automation
 - Coverage to support each user story or requirement
 - Don't write tests for code that won't change again
 - Purposeful negative conditions



THE NEGATIVE CONDITION “PARTY”

It can be really fun to try to break the software for the sake of breaking it...
“FUN” does not equal “PRODUCTIVE”

Test for ROBUSTNESS!



SPRINT <N>: NEW WORK OR BUGS?

- Know ahead of time how to prioritize development
 - New features?
 - Bug fixes?
- If you wait until there's a backlog, you're too late
- “Zero backlog” approach

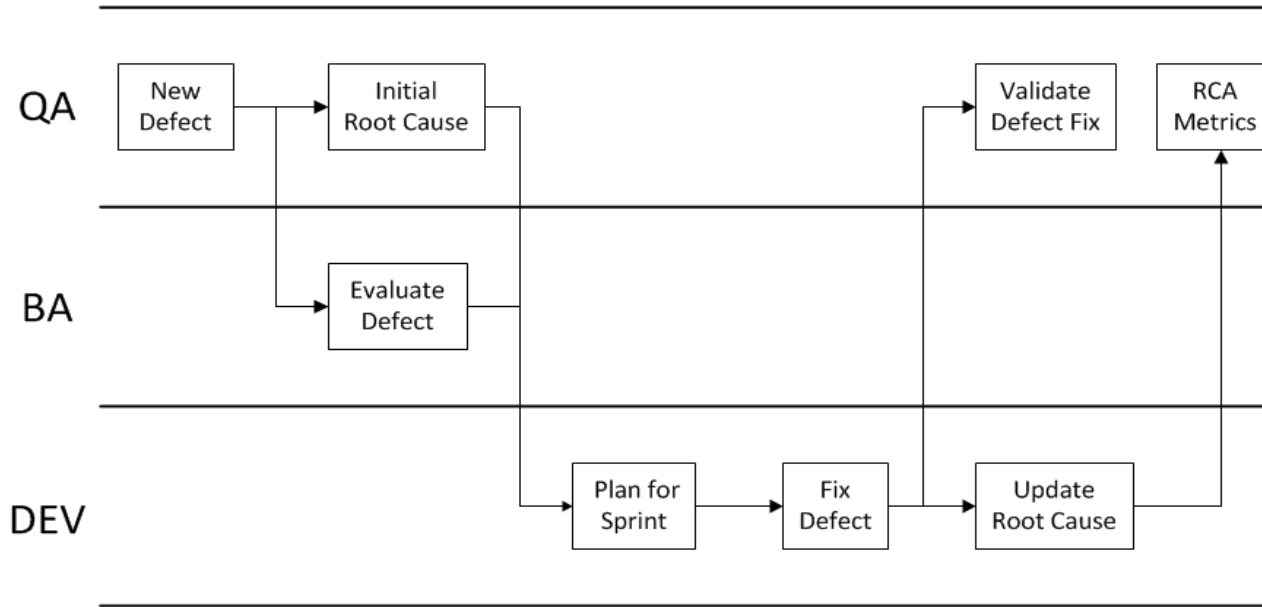


SPRINT <N>: TEST EXECUTION

- Basic stats - Pass/Fail counts, remaining tests
 - *Does your team need any more than that?*
- Root Cause Analysis metadata for failed tests
- Connect RCA to your GQM model
- RCA can drive great process improvements
...BUT YOU HAVE TO USE IT!!



ROOT CAUSE SWIM LANES



***Multiple people, multiple conversations for each failure!
How are you using this data to improve your process?***



RETROSPECTIVE: RCA IMPROVEMENT

Failure Types – why did this test fail?

Resolution – what changed to fix the problem?

How do we prevent either from happening again?

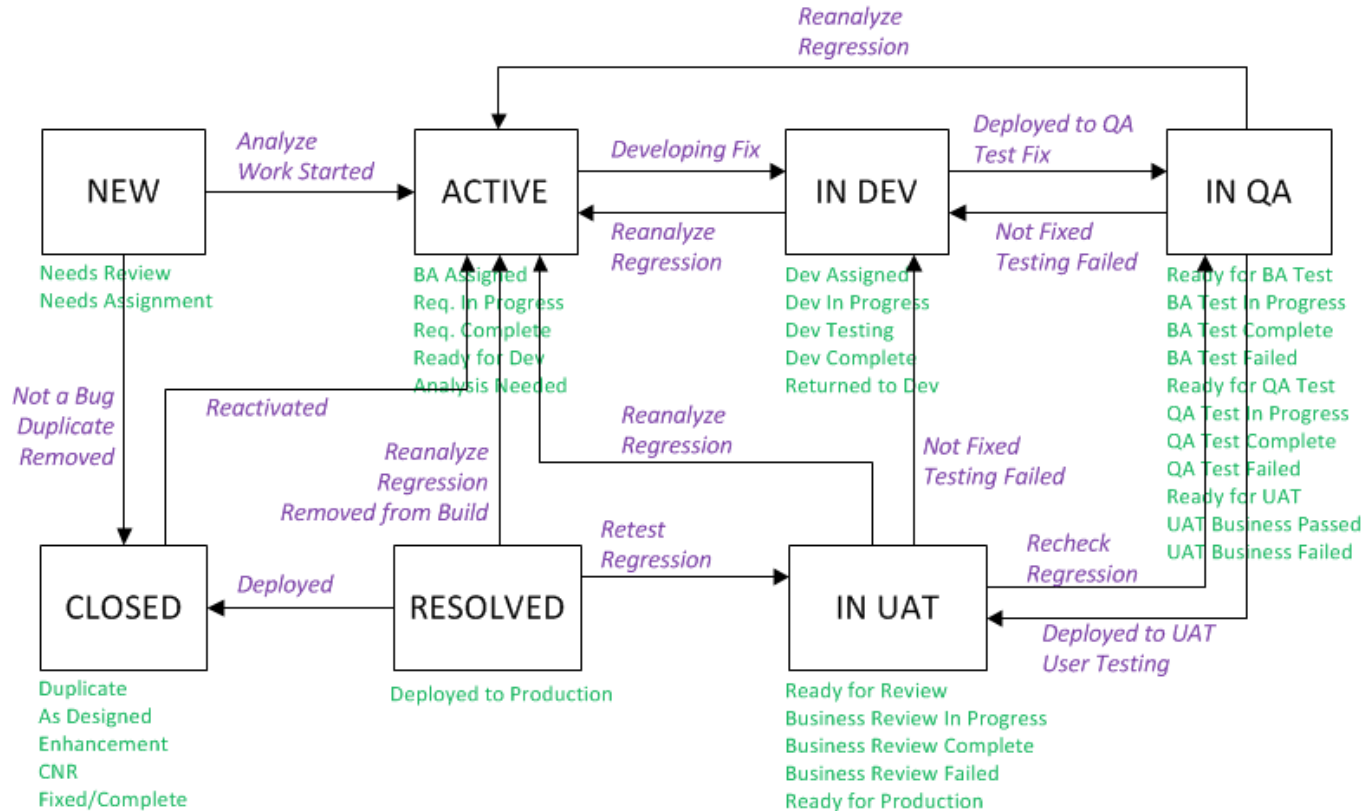


SPRINT <N>: DEFECT STATE MANAGEMENT

- Understand each defect state
 - New
 - Assigned
 - Committed
 - Done
- Your team can have more...but why?
- Tie it back to your GQM model



HAVE YOU EVER SEEN THIS?



RETROSPECTIVE: NO HOARDING!

Each data point leads to decision & dissent.
The more you have, the more you add.

*10 metadata changes per bug X
1 minute each X 500 bugs =
OVER 80 MAN-HOURS*



RETROSPECTIVE: REMOVE “REMOVED”?

- Every defect model has a Removed state...
what's it for?
 - “Duplicate”
 - “Pending”
 - “Not an Issue”
- Your team has all the data. Use it to improve!



YOUR “CHECKLIST”

- ☐ Involved from Inception
- ☐ Necessary Deliverables
- ☐ Evaluate Acceptance Criteria ASAP
- ☐ No Perfect Requirements
- ☐ Manual Tests for Automation
- ☐ No “Tests for Bugs”
- ☐ Purposeful Negative Conditions
- ☐ Bug Fix Priority
- ☐ Root Cause Improvements
- ☐ Streamlined Defect States
- ☐ Improve from Removed
- ☐ KEEP CLEANING!



IF YOU ONLY REMEMBER ONE THING...

If you can't tell me WHY you're doing something...

...and HOW it helps your Agile team...

STOP DOING IT!!



Testing is no longer the last step in the Software Development Life Cycle. It is our responsibility to make sure we add value to every aspect of quality software. Be efficient. Be undeniably important to your projects.

Paul Herzog
Delivery Manager, Testing Services Practice
paul.hertzog@spr.com
312.756.1760 x410



SPR CONSULTING
POWERING BUSINESS WITH TECHNOLOGY