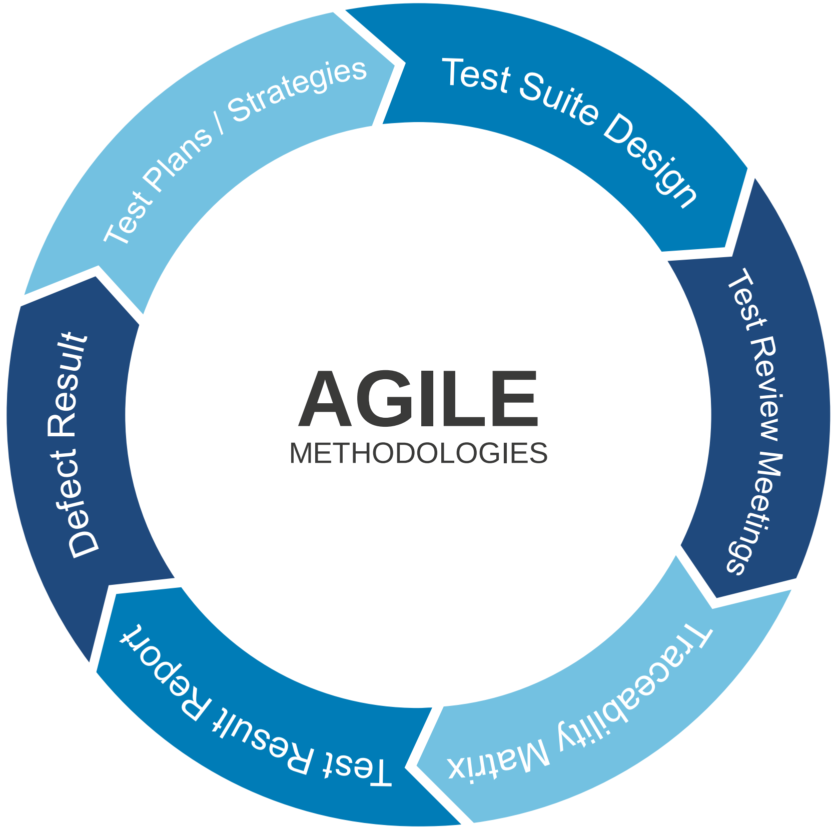
SIMPLICITY: THE PATH TO ACHIEVING AGILE TESTING EFFICIENCY

SPR CONSULTING
POWERING BUSINESS WITH TECHNOLOGY

Matt Paxton – My Inspiration

Why Be Agile?



Why do businesses adopt Agile Methodologies?

Why be Agile?

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:

1. Individuals & interactions over processes & Tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

Twelve Principles 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 this principle – Simplicity – 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

Test Plans / Strategies

Does anyone read this?
Who else needs it?

Test Suite Design

Do you have a
standalone repository?

Test Review Meetings

Do you need a formal
review process?

Traceability Matrix

Is this automatic
to our test tool?

Test Result Report

What do we do
with this info?

Defect Metrics

Do these drive
process improvement?

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: 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

Sprint <n>: Test Case Creation

It can be really fun to try to break the software for the sake of breaking it…

FUN $\neq$ 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

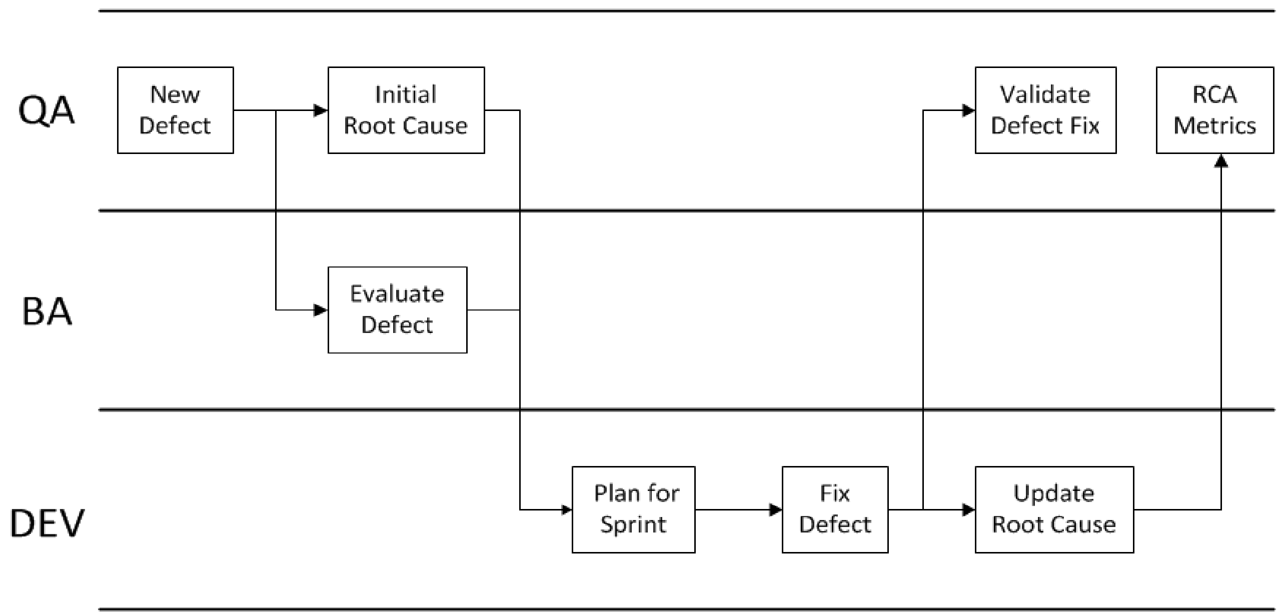
“Zero backlog” approach

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

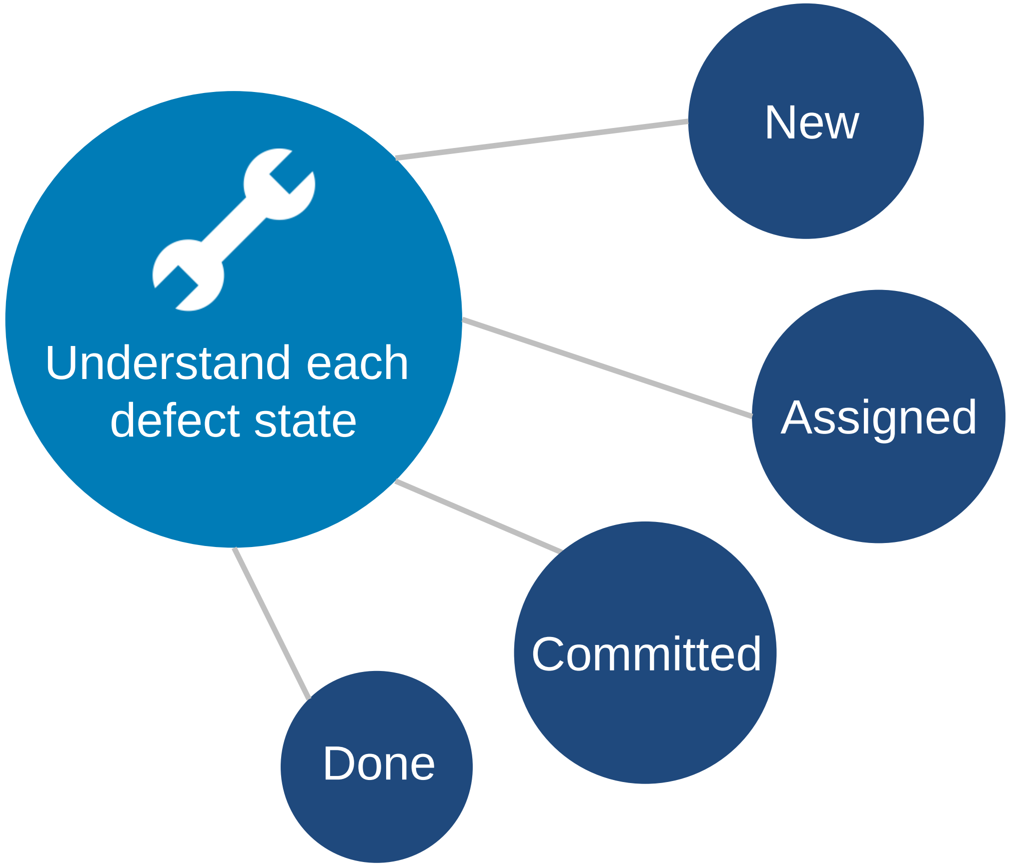
Why did this test fail?

Resolution

What changed to fix the problem?

How do we prevent either from happening again?

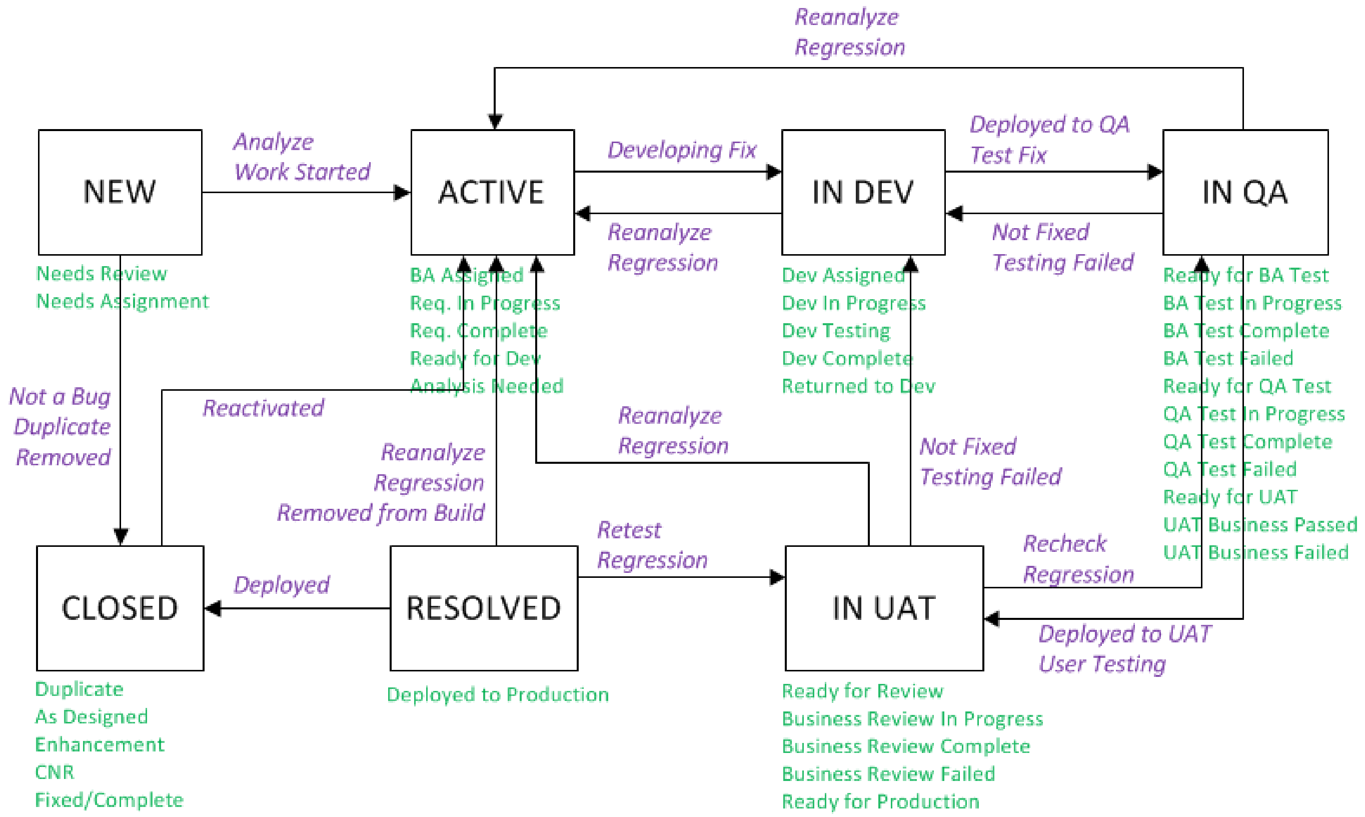
Sprint <N>: DEFECT STATE MANAGEMENT



Your team can have more… but why?

Tie it back to your GQM model

Have you ever seen this?



DEFECT INFORMATION: NO HOARDING!

Each data point leads to decision & dissent.

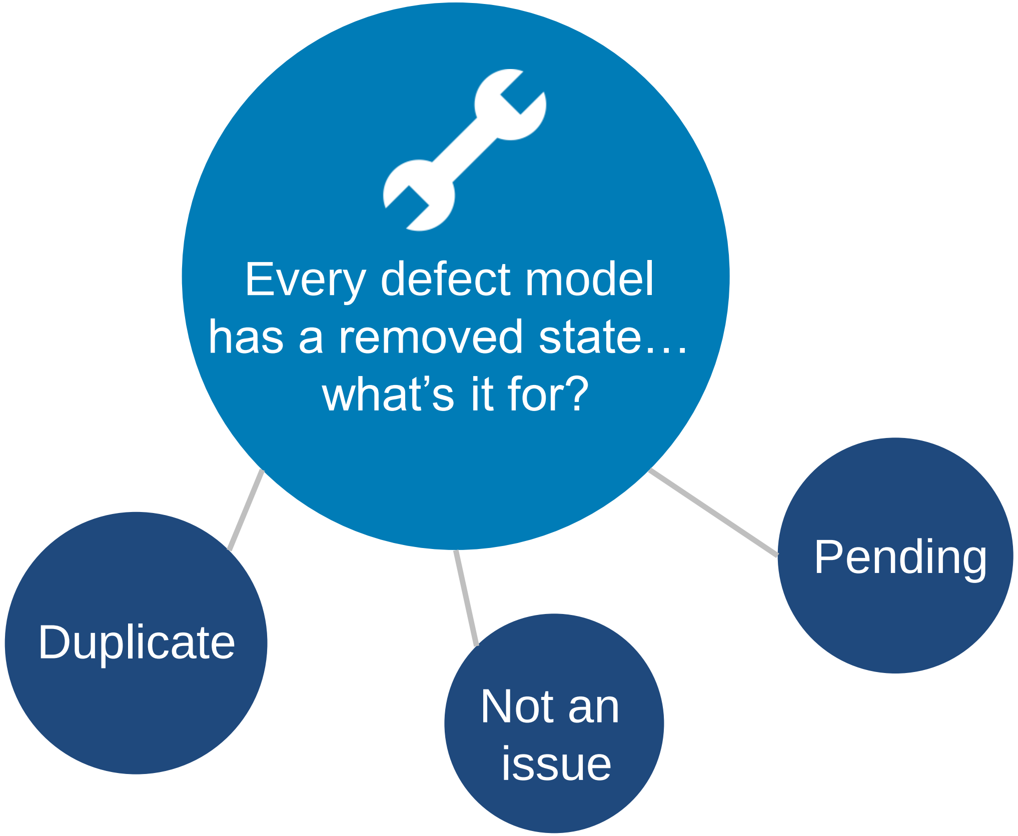
The more you have, the more you add.

10 metadata changes per bug

X 1 minute each X 500 bugs

= OVER 80 MAN-HOURS

DEFECTS: Remove “REMOVED”?



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!

Closing

“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.herzog@spr.com
312.756.1760 x410

SPR CONSULTING
POWERING BUSINESS WITH TECHNOLOGY