

Dealing with Defects: An Agile Approach



TISQA 2009

Lisa Crispin

With Material from Janet Gregory

Introduction

My defect tracking background:

- As programmer – no DTS
- Tech support – DTS developed in-house
- Traditional QA – vendor DTS, customized
- Agile teams – variety of approaches



Your Defect Tracking?

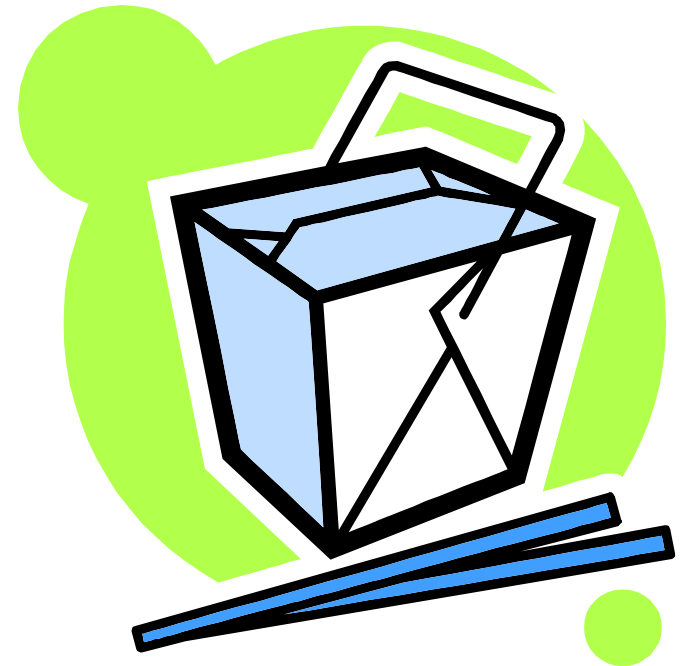
How about you?

- How many use a DTS?
- How many don't use one?



Takeaways

- Defects and technical debt
- When to use a Defect Tracking System
- Alternatives
- Defect metrics
- Focus on bug prevention



Traditional View

- Bugs must be documented
 - Especially those found in production
- Provide information for root cause analysis
- Knowledge base – did similar bug occur earlier?
- DTS provides useful metrics



Lean Development / Agile View

- Defect queues are queues of rework – waste (Poppendieck)
- Find, fix test-first, and forget
- Better yet, prevent
 - ❑ zero bug tolerance



Choices

An agile approach:

- Understand the problem
- Do what works for your team
- Focus on goals: bug prevention
- Start simple, add as needed
- Explore alternatives



Why Log Bugs

- Knowledge base
 - Place to keep supplemental docs
- Prioritizing
- Traceability
- Distributed/large teams
- Customer ease, visibility, convenience
- Metrics
 - Redundant with test-first, frequent releases



Why Not Log Bugs

- Overhead
- Duplicates test
- Impedes communication
- Waste
 - ❑ Inaccurate or inadequate reports
 - ❑ Caught during development
 - ❑ Won't get fixed
- Doesn't work towards goal
 - ❑ zero defects, bug prevention



Bugs - “Hidden Backlog”

- Acceptance tests = desired behavior
- Defect reports = misbehavior (Antony Marcano)
- Ignored defects = missed features
 - **Misbehavior not addressed**
- Defects prioritized over new features
 - **Misbehavior over desired features**
- Information can get lost in DTS

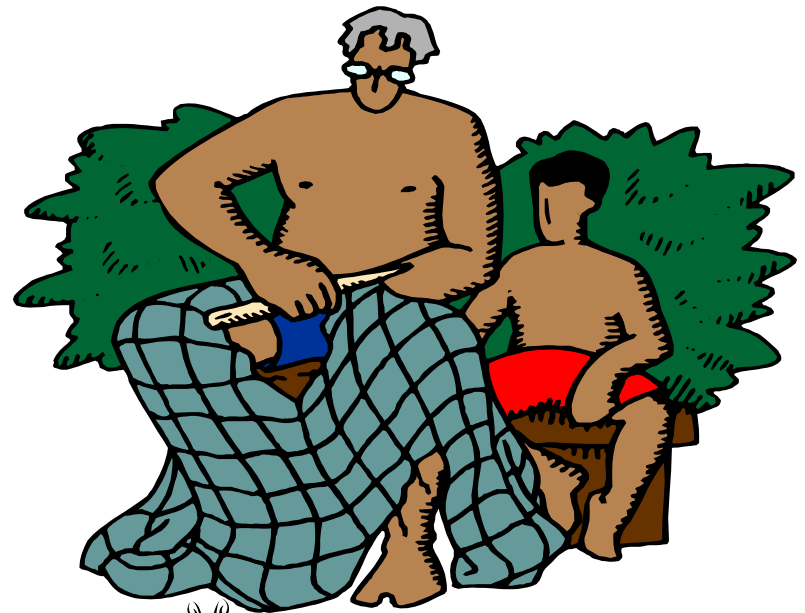


Copyright 2009: Lisa Crispin



Choosing to Not Log a Bug

- Unit-level bugs
 - ❑ Write a test, not a bug
- Higher-level regression failures
 - ❑ Test usually enough
- Bugs found during development
 - ❑ Fix immediately
 - ❑ Log on task card



Choosing to Log a Bug

- Found in later iteration
 - Log if not fixed immediately
- Legacy bugs
 - Write new story/feature
 - Don't log if it won't be fixed
- Found in production
 - Fix immediately or
 - Prioritize, estimate, plan
- Non-bug production issues
 - “production support requests”



When to Fix Bugs

- **Fix now**
 - way cheaper
 - prevents technical debt
- **Estimate, prioritize and fix later**
 - complex, high effort
 - legacy code, may destabilize
 - customer decision
- **Never fix**
 - about to be rewritten
 - low risk
 - don't pretend you'll fix if you won't!



Choosing Media for Logging Bugs

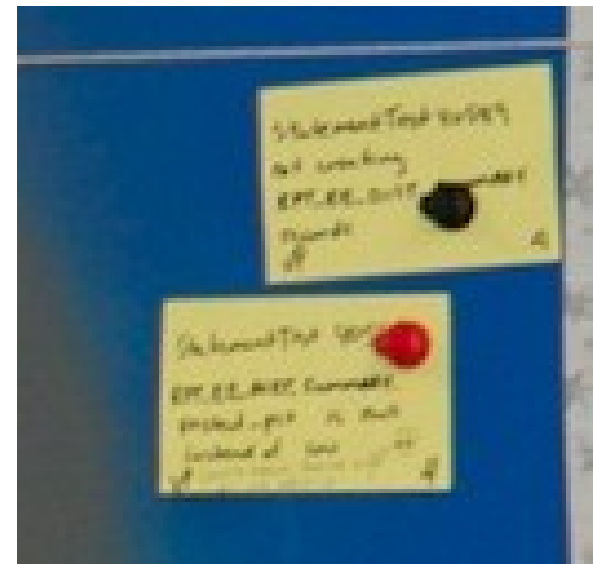
- Cards
- DTS
- Tests



Logging Bugs on Cards

Story, task or bug cards – physical or online

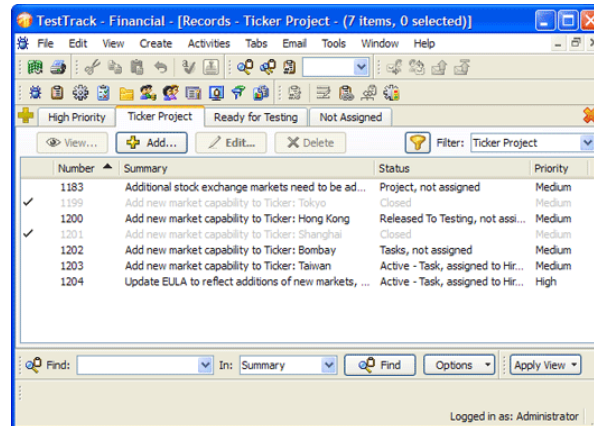
- Tangible
- Visible – color coding
- Attach screen prints
- Use when:
 - Disciplined team
 - Bugs need high visibility



Logging Bugs in DTS

Use a DTS when

- Team is distributed
- Audit requirements
- Release notes
- Not all bugs will be fixed immediately
- Legacy system with lots of defects



The screenshot shows the TestTrack Financial interface with a table of bug records. The table has columns for Number, Summary, Status, and Priority. The records are as follows:

Number	Summary	Status	Priority
1183	Additional stock exchange markets need to be ad...	Project, not assigned	Medium
1199	Add new market capability to Ticker: Tokyo	Closed	Medium
1200	Add new market capability to Ticker: Hong Kong	Released To Testing, not assi...	Medium
1201	Add new market capability to Ticker: Shanghai	Closed	Medium
1202	Add new market capability to Ticker: Bombay	Tasks, not assigned	Medium
1203	Add new market capability to Ticker: Taiwan	Active - Task, assigned to Hir...	Medium
1204	Update EULA to reflect additions of new markets, ...	Active - Task, assigned to Hir...	High



Logging Bugs with Tests

Use the test written to reproduce the bug and verify the fix when:

- Everything learned from fixing bug is captured in test
- Test can be automated
- Team is disciplined and writes tests for every bug found

Check Employee Fixture				
userId	isHce?	isEligible?	adr?	acr?
1001	true	true	12.682927	15.61
1002	true	true	12.682927	12.68
1003	true	true	25.00	25
1004	true	true	10.00	10
1005	true	true	20.00	20
1006	true	true	8.666667	8.67
1007	true	true	0	0
1008	false	true	6	6



Explore Alternatives

Start new project with no DTS, and see what you need.

- Set rules - “no more than 10 bugs at once”
 - Analyze and address bad trends
- Fix all bugs
 - Low priority usually quick to fix
 - Unfixed bugs build technical debt quickly
- Treat bugs as stories
 - Estimate and prioritize
- Color-coded cards and stickers
 - Visible, promotes communication



Defect Metrics

- Traditionally, tell us state of quality, release readiness
- Using tests to drive incremental, iterative development tells us this
- Trends more important than numbers
- Who cares how many bugs found in development?
- May be useful for team goals
- Careful how you use metrics



Explore Alternatives

- Understand your needs
- Choose by team consensus
- Cards and stickers – physical or online
 - ❑ Story board
- Online DTS
- Combine DTS and cards
 - ❑ Visibility, planning
- Get recommendations
- Experiment



Focus On Your Goals

- Zero defects is achievable
- Focus on bug prevention
- Test-driven development
- Use retrospectives



Questions?

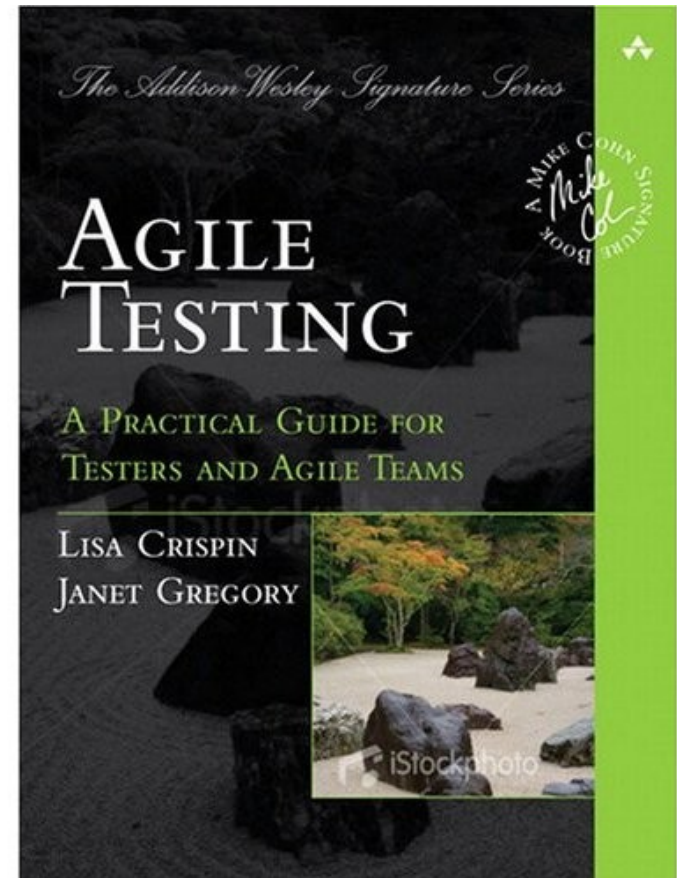


Now Available

Agile Testing: A Practical Guide for Testers and Agile Teams

By Lisa Crispin and Janet Gregory

www.agiletester.ca



Copyright © 2009: Lisa Crispin



Some Agile Testing Resources

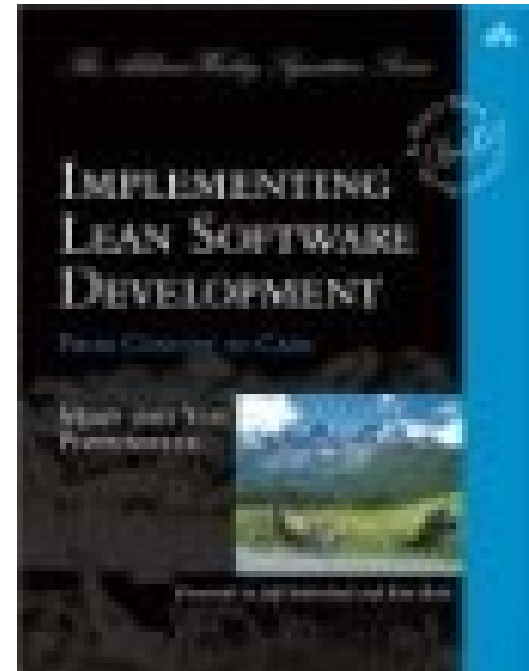
- lisacrispin.com
- janetgregory.ca
- agilealliance.org
- exampler.com
- agile-testing@yahooogroups.com
- testobsessed.com
- testingreflections.com



Lean Development

*Implementing Lean Development:
From Concept to Cash*

Mary and Tom Poppendieck



Copyright © 2009: Lisa Crispin

