



Develop Software with Confidence

Agitar @ Java Forum Stuttgart
Artur Hildebrandt, Solutions Consultant, July 6th 2006

Our Mission



Profoundly improve the **quality** of software
and the **economics** , **transparency** , and **agility**
of software development

**By making developer testing
effective for the enterprise**

Software Development Is Still Immature . . .

\$ 100+ Billion

Wasted annually on software bugs ¹

Only 29%

Of IT projects succeed ²

50%

Of all software projects
are total failures ³

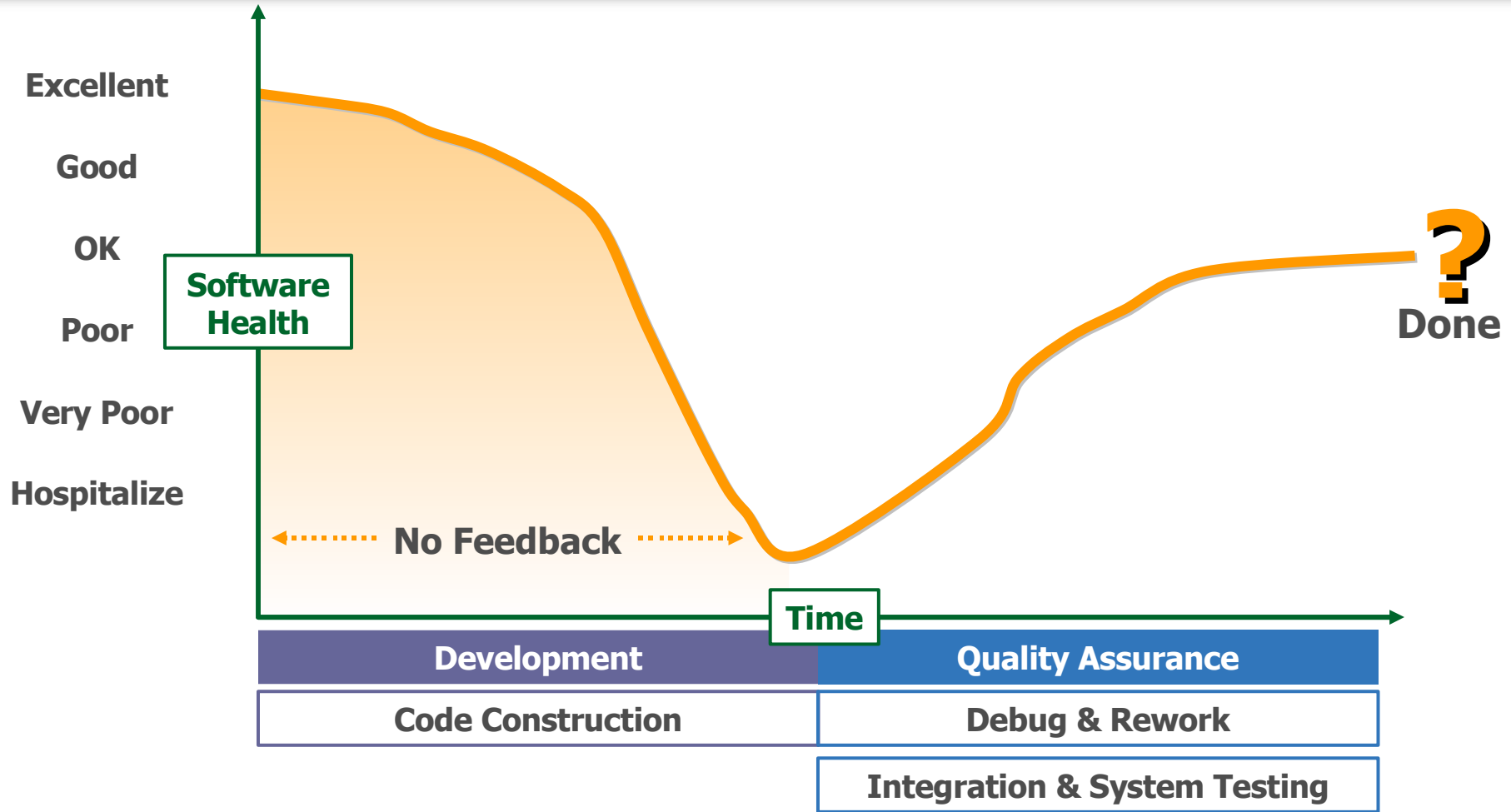
Projects take

84% Longer

Than originally scheduled ⁴



The Traditional Approach Is Flawed



Source: "The Software Development Paradox", Alberto Savoia, 2004

There Is a Better Approach

Test Bugs Out



VS.

Build Quality In



Typical software development process

- Give QA the main responsibility for quality
- Test late
- Use unverified components
- Deliver, and fix defects later

Mature world-class manufacturing process

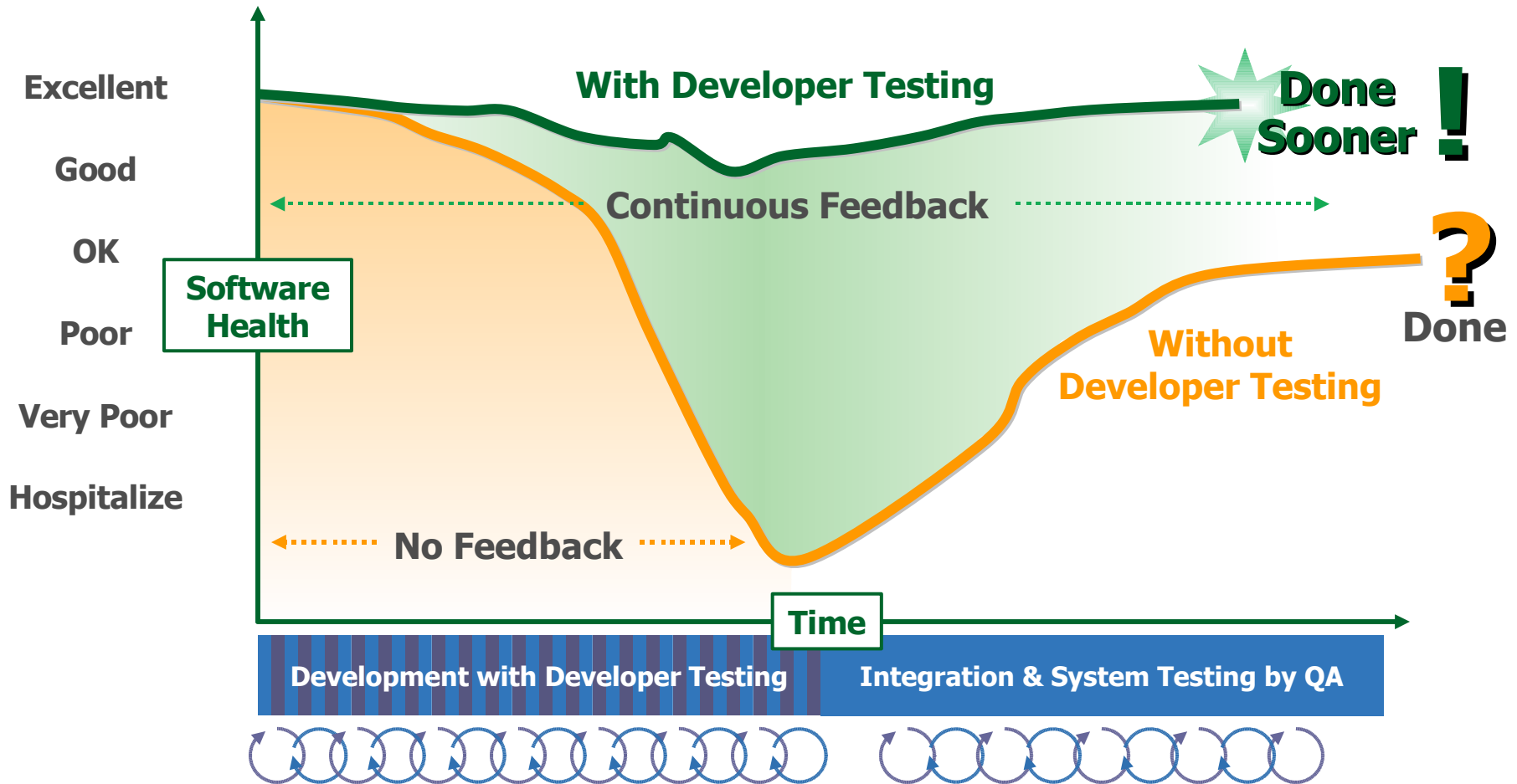
- Hold everyone responsible for quality
- Test early and often
- Use verified components
- Stop the assembly line

Developer Testing Is the Solution

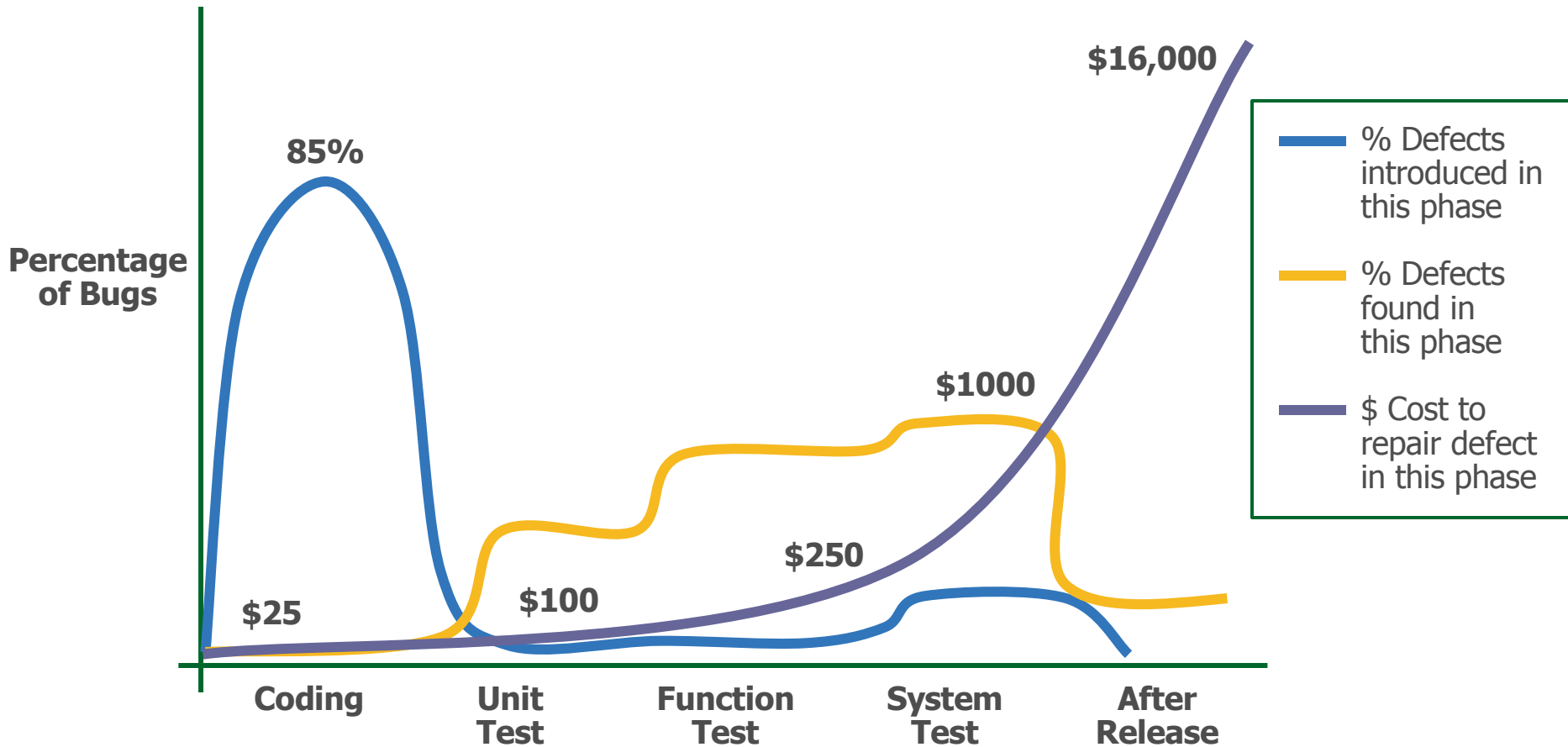
Also known as “unit testing” or “programmer testing”

- Each code unit is accompanied by unit tests that validate and document its correct behavior
- The developer creates these unit tests while coding
- Each unit test is self-sufficient and self-evaluating so it can be run and checked automatically
- The unit tests are run very frequently
- Developer testing may also include functional (scenario) testing done by developers, often using JUnit

Find and Fix Defects Sooner



The Economics Are Compelling



Source: Applied Software Measurement, Capers Jones, 1996

Why Is Developer Testing Not Yet Pervasive?

Poor Division of Labor

- Developers deliver code, not knowing if it works
- QA verifies, integrates, and does system tests, but cannot fix problems
- Detection and repair is deferred until long after the bug is created

Too Much Manual Work

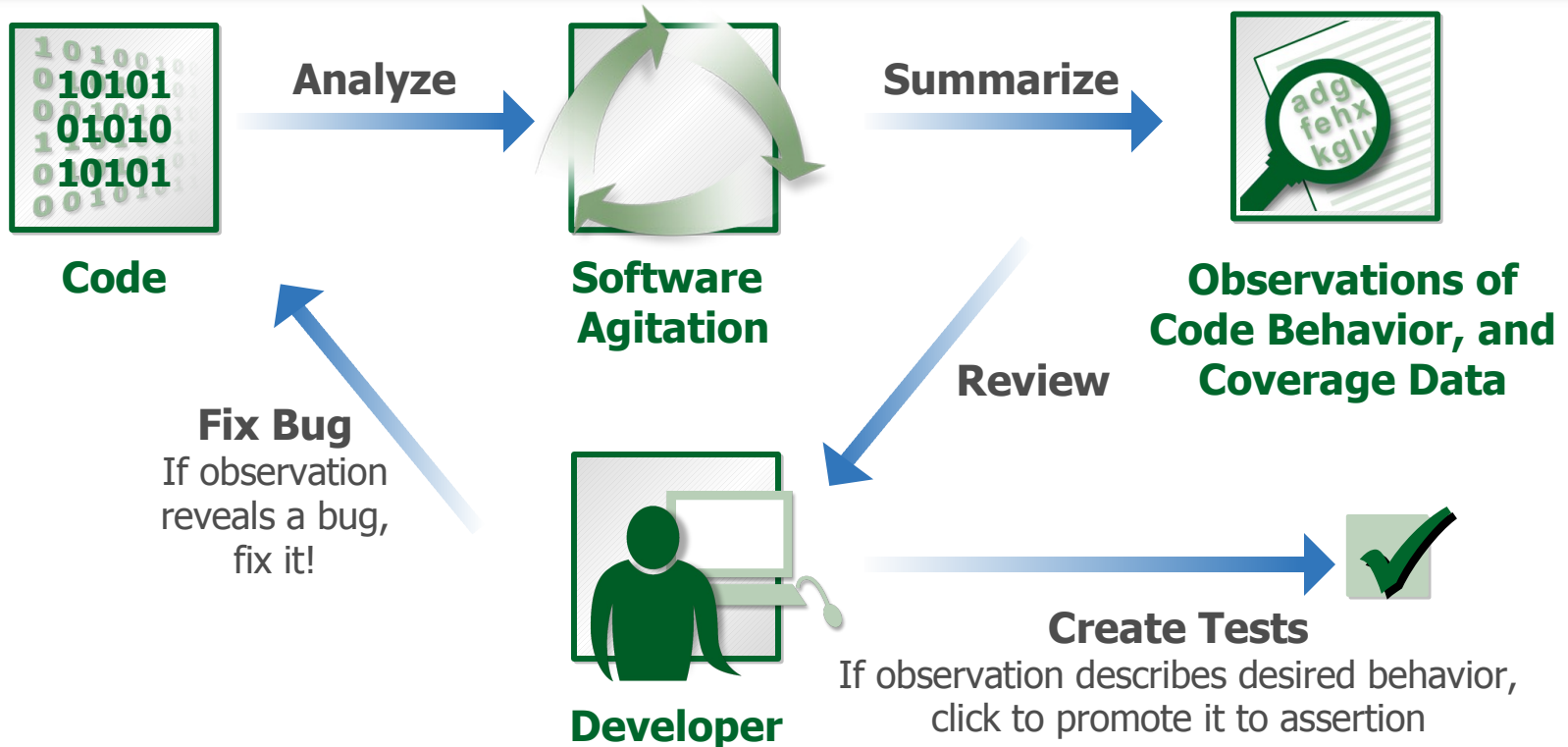
- A combinatorial problem, so full coverage is hard for unit tests or debugger
- Often 2x – 4x more test code than code to test
- Most time spent in tedious setup and data creation
- Tests get stale
- Tough to be thorough and meet the schedule

Uncertain Status and Goals

- Are you done, and have you done enough?
- What has been tested and what has not?
- Have you focused enough on the riskiest or most complex code?

Manual approaches (usually JUnit-based) tend to stall.

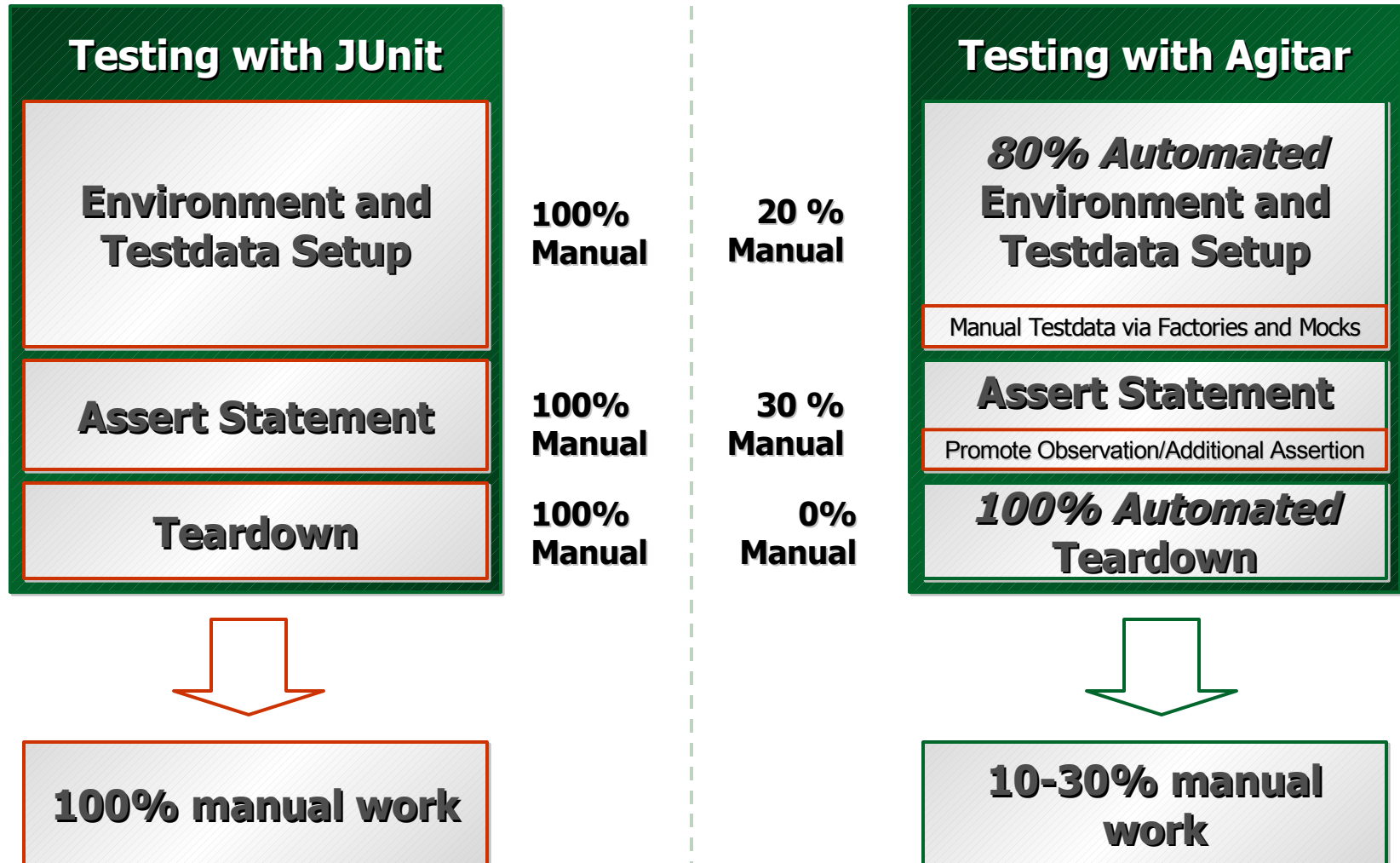
Agitator®: Unprecedented Automation



Validate intended code behavior

- Discover invariants (or specify and validate them)
- Achieve high data and state coverage without manual setup
- Test without needing application server, database, etc.

Agitator – Automate what makes Sense



Improve the Developer's Daily Work

CREATE

High-quality
new code

FIX

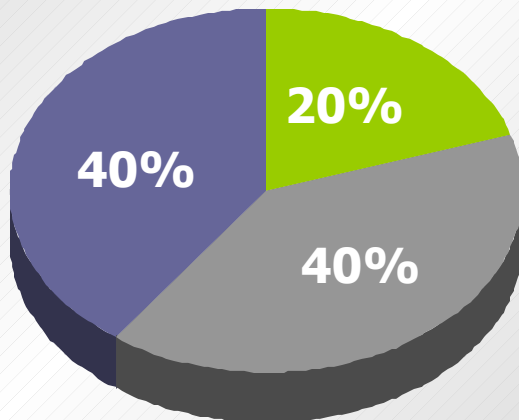
Defective code
quickly

CHANGE

Existing code
without fear

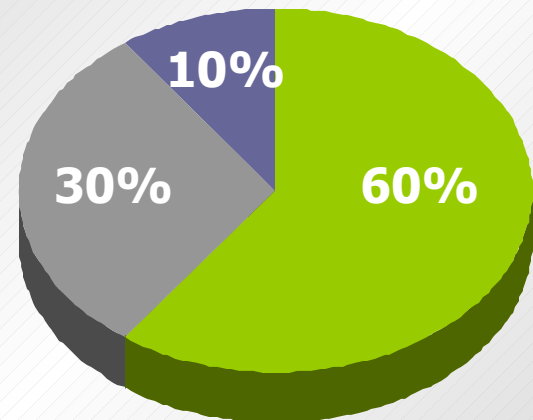
Goal: code that works

Typical Development



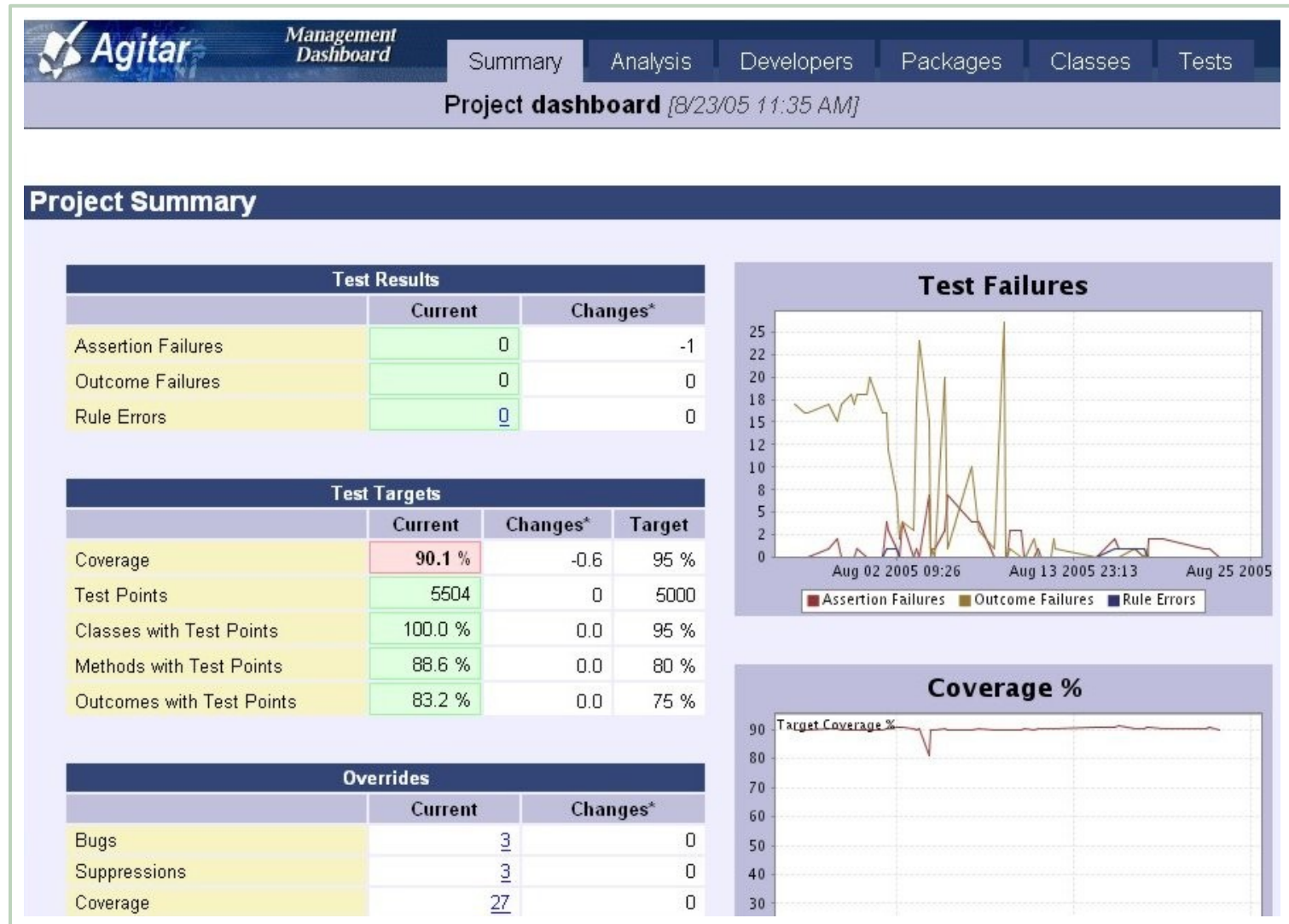
■ Coding
■ Testing
■ Rework

Developed with Agitator



■ Coding
■ Testing
■ Rework

Real Visibility into Unit-Level Quality



Metrics for High-Performance Teams

Project dashboard

Test Targets

	Current	Changes*	Target
Coverage	89.8 %	0.0	95 %
Test Points	5273	5	5000
Classes with Test Points	100.0 %	0.0	95 %
Methods with Test Points	75.5 %	-0.1	80 %
Outcomes with Test Points	76.2 %	-0.0	75 %

Project Code Metrics

	Current	Changes*
Project Classes	131	0
Test and Harness Classes	189	0
Project Methods	1355	1
Test Methods	416	0
Project Executable Lines	5279	1
Test Executable Lines	7070	0

High code coverage

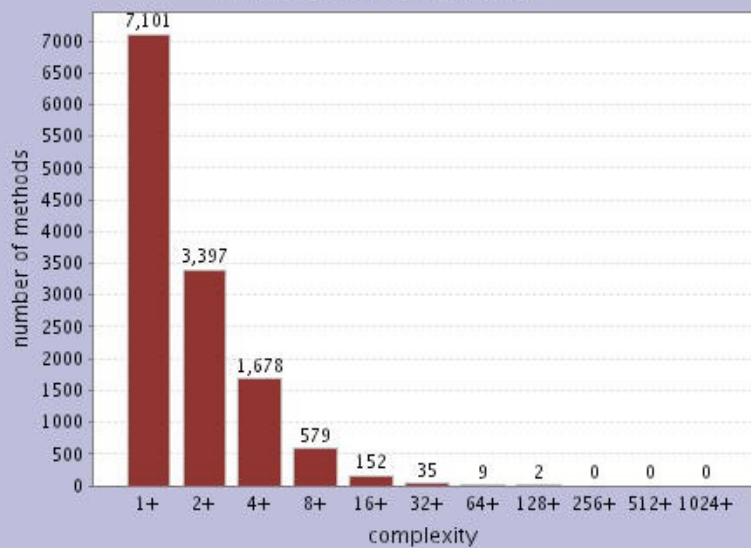
Every class has assertions

High % of methods and outcomes have assertions

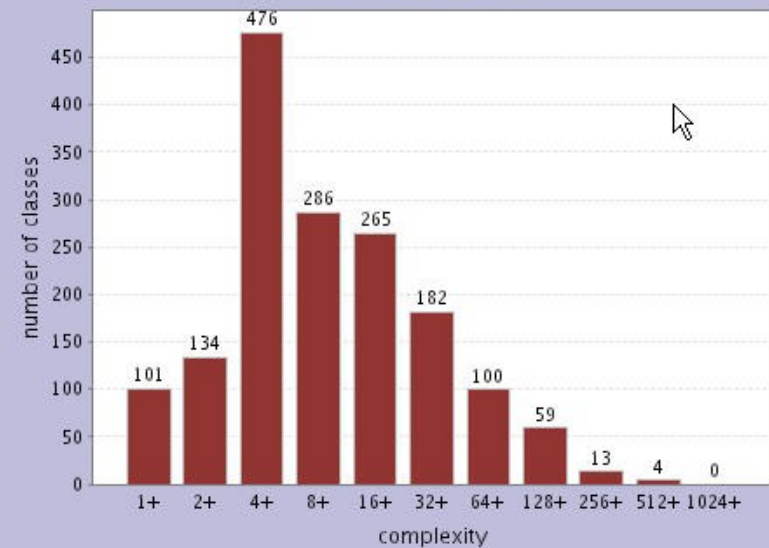
***1:1 ratio of assertions to lines of code

Manage Complexity

Method Complexity



Class Complexity



Demo Placeholder

Add Agitator to Your JUnit Testing

Kent Beck



Agitar Fellow

JUnit



Agitator

- 95% of our customers start with JUnit
- JUnit is not required to use Agitator
- Agitator and Dashboard understand and leverage existing JUnit

The Leader in Developer Testing

Top-Tier Investors



SEQUOIA CAPITAL®

NEA® New Enterprise Associates®

GLOBESPAN
CAPITAL PARTNERS

Unprecedented Recognition



Gartner

IDC
Analyze the Future



