Zero Bugs: State of the Practice

Stephen Vance

Agile Alliance Technical Conference

April 21, 2017

Zero Bugs?

Zero “Bugs”!

- Why? vs. What?
  - Weinberg: “Faults” and “Failures”
  - There is rarely only one “why” even at the root
  - “Why” drives improvement
- When?
  - Does it meet expectations, intentions, and needs?
  - Prior to release, it’s rework
  - After, it’s unrealized value

Frame of Reference

- Lean
  - Focus on value
  - Seek perfection
  - Build feedback loops

https://www.lean.org/WhatsLean/Principles.cfm

- Key waste types for software quality
  - Inventory
  - Waiting
  - Over-processing
  - Over-production
  - Defects

Build Quality In

Any approach that catches issues after they occur has a lighter weight alternative that prevents issues from happening

Culture

Blameless

Everyone accountable

Collaborative

Learning

Intentional

Characteristics

Automated		Manual
Repeatable		Ad Hoc
Predictable	**OVER**	Variable
Fast		Slow
Reliable		Error-prone
Emergent		Planned

What's Typical?

(Highly Subjective and Anecdotal)

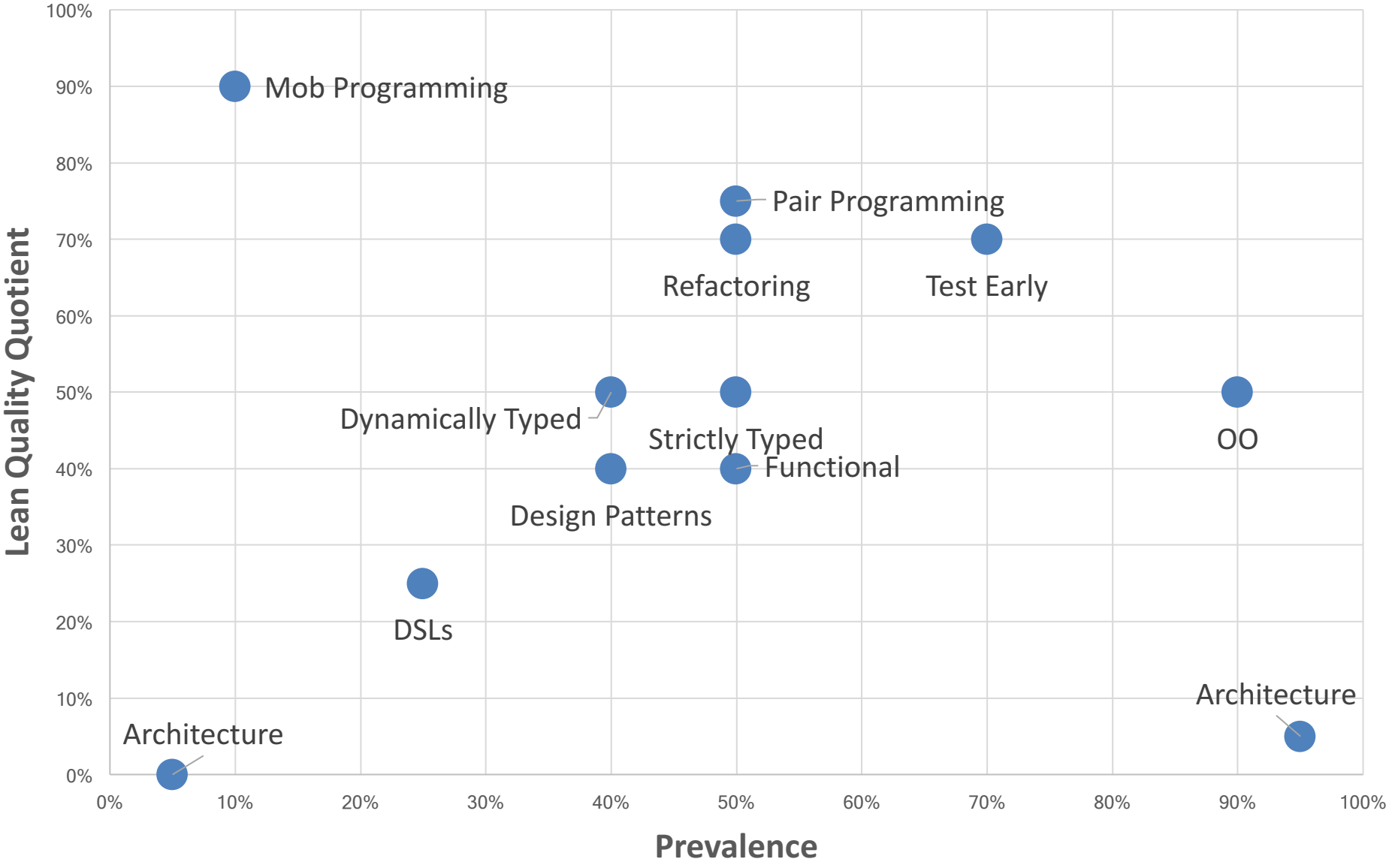
- Coding Practices
  - Limited to moderate use of pair programming with little mob programming
  - DSLs if they are supplied
  - Limited understanding of design patterns
  - Limited understanding of refactoring
  - OO as default with some functional constructs
  - Dynamic or strictly typed languages
  - Mostly test early with some TDD or BDD
- Architecture
  - Little to no architecture or big up-front architecture
- Tools
  - Low to moderate code coverage
  - Moderate use of static analysis
  - Code generation if supplied
- Testing
  - Moderate to high manual testing
  - Increasing dev-test collaboration but still limited
  - Light to moderate use of exploratory testing
  - No to little use of advanced testing techniques like QuickCheck, fuzz, mutation
- Build and Deploy
  - CI with some CD
  - Light to moderate application monitoring
  - Light to moderate DevOps and infrastructure as code
  - Moderate use of automated rolling deployment, heavily tied to use of cloud at scale

The Lean Quality Quotient™

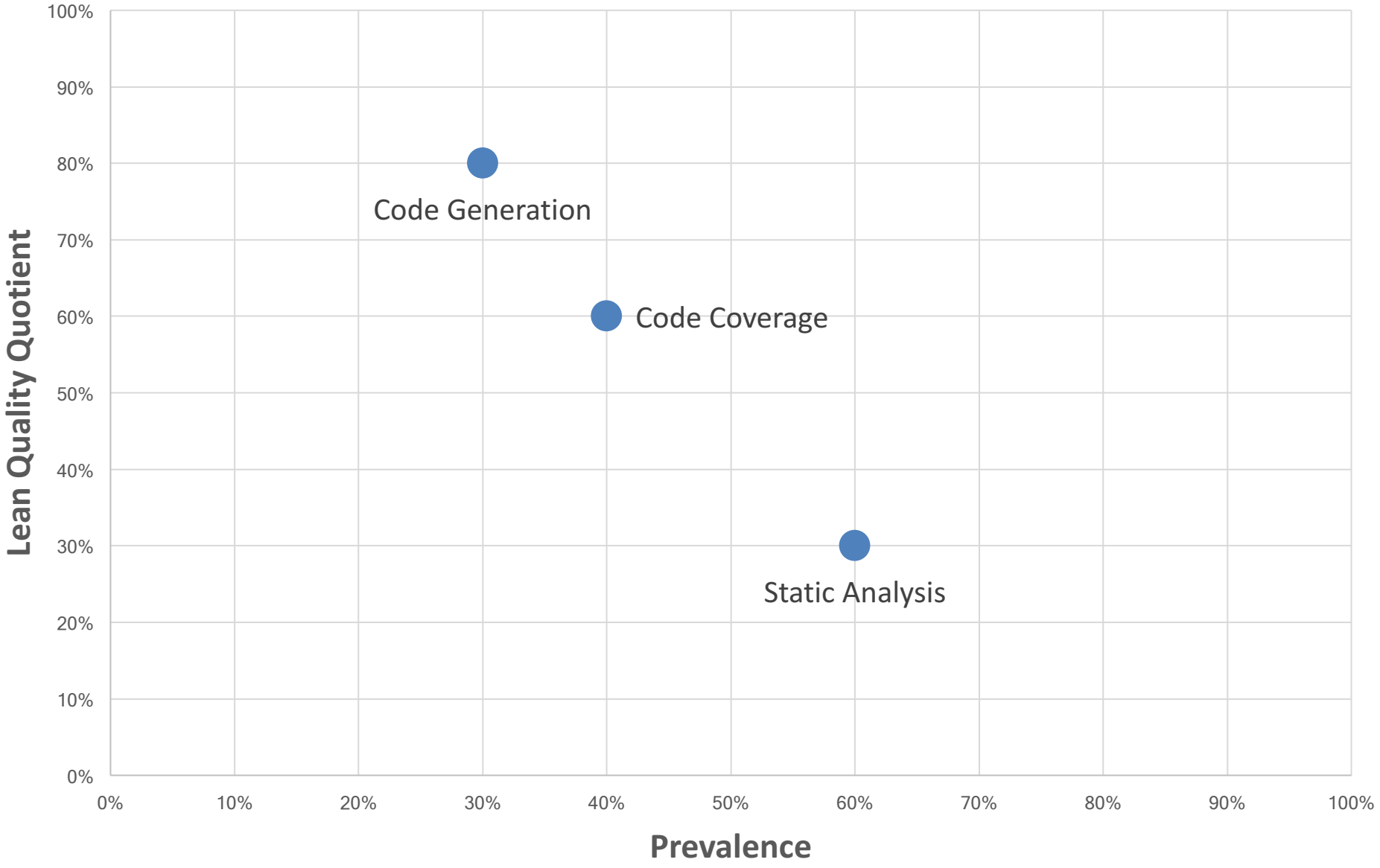
(Not actually trademarked)

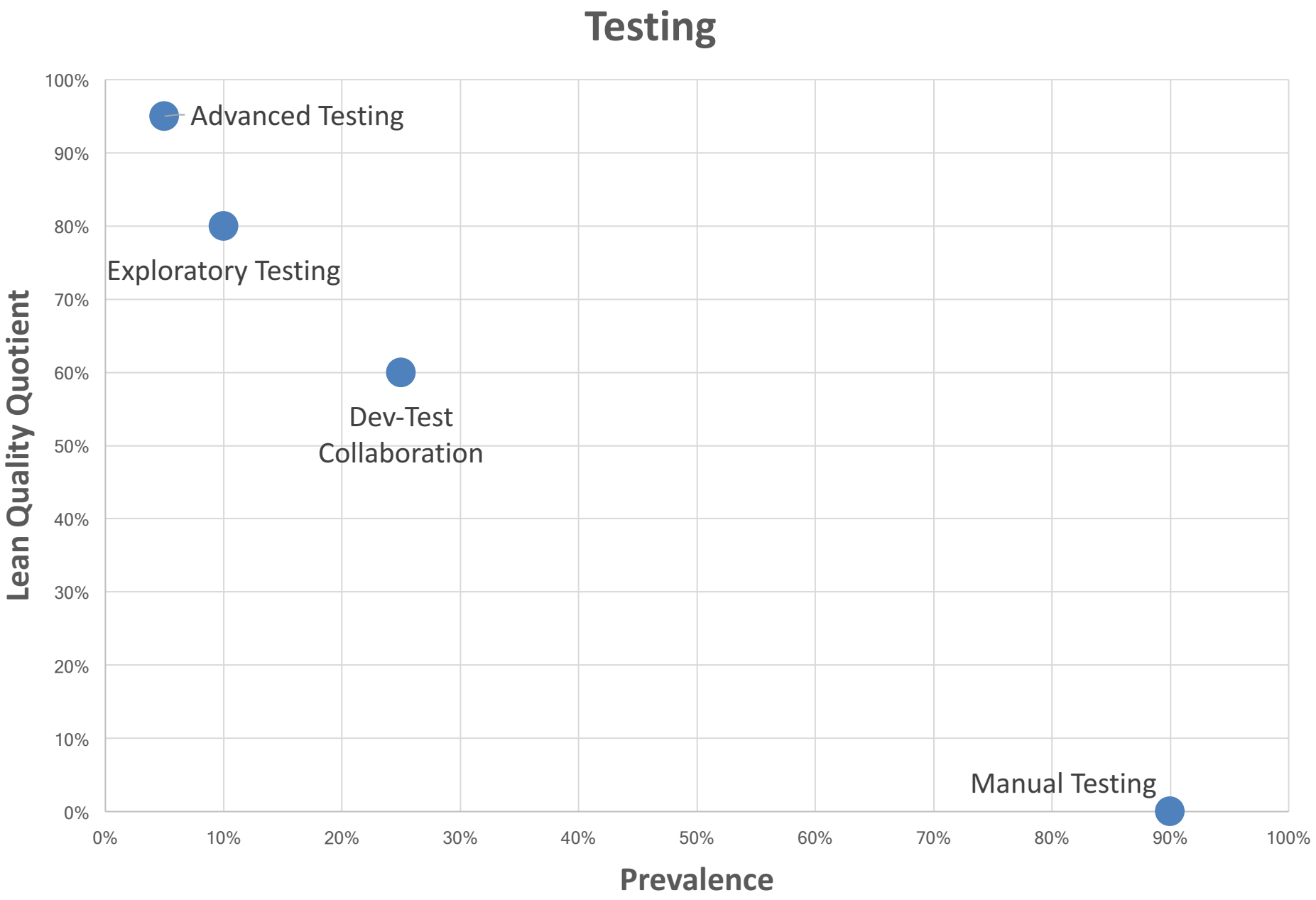
- The degree to which a practice is used in a manner that is
  - Emergent
  - Iterative
  - Adaptive
  - Informed
  - Pre-emptive

Coding Practices

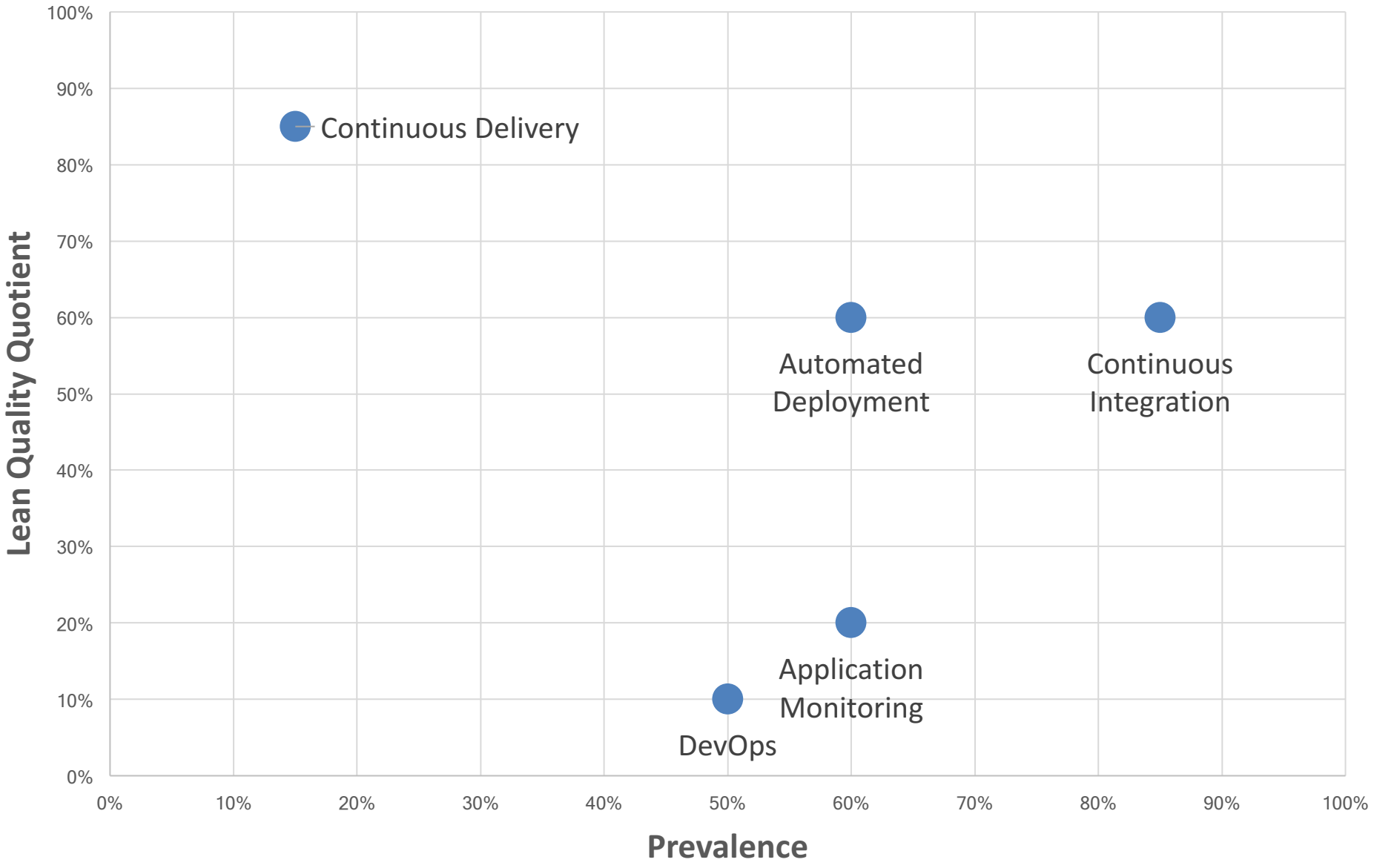


Tools

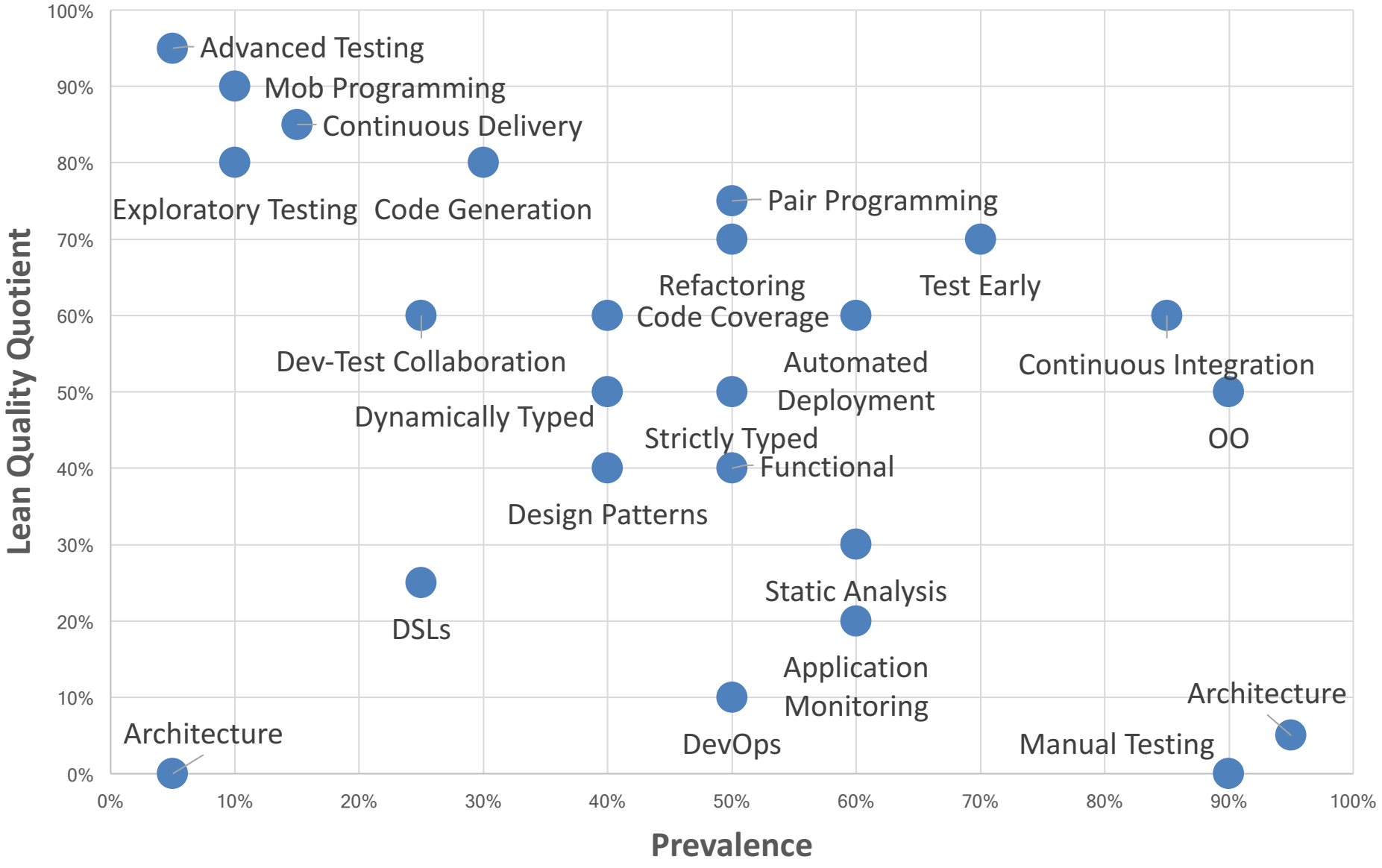



Testing
Advanced Testing
Exploratory Testing
Dev-Test Collaboration
Manual Testing
Lean Quality Quotient
Prevalence
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%

Build and Deploy



Composite


Lean Quality Quotient
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Advanced Testing
Mob Programming
Continuous Delivery
Exploratory Testing
Code Generation
Pair Programming
Refactoring
Test Early
Code Coverage
Dev-Test Collaboration
Automated
Deployment
Continuous Integration
Dynamically Typed
Strictly Typed
OO
Functional
Design Patterns
Static Analysis
DSLs
Application
Monitoring
Architecture
DevOps
Manual Testing
Architecture
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Prevalence

So What Next?

- Do more testing
- Figure out how to kill testing
- Bring the skills to the front to build quality in
- Extend the practicum
  - Real-world design patterns and refactoring
  - Evolutionary architecture
- Enhance collaboration
- Improve the tools
- Use more tools

Should We …?

- Create more DSLs and teach how to create good DSLs?
- Move toward strong typing?
- Move toward functional programming?
- Do more mob programming?
- Expand or eliminate code generation?
- Teach the machines to do our jobs?

A Survey

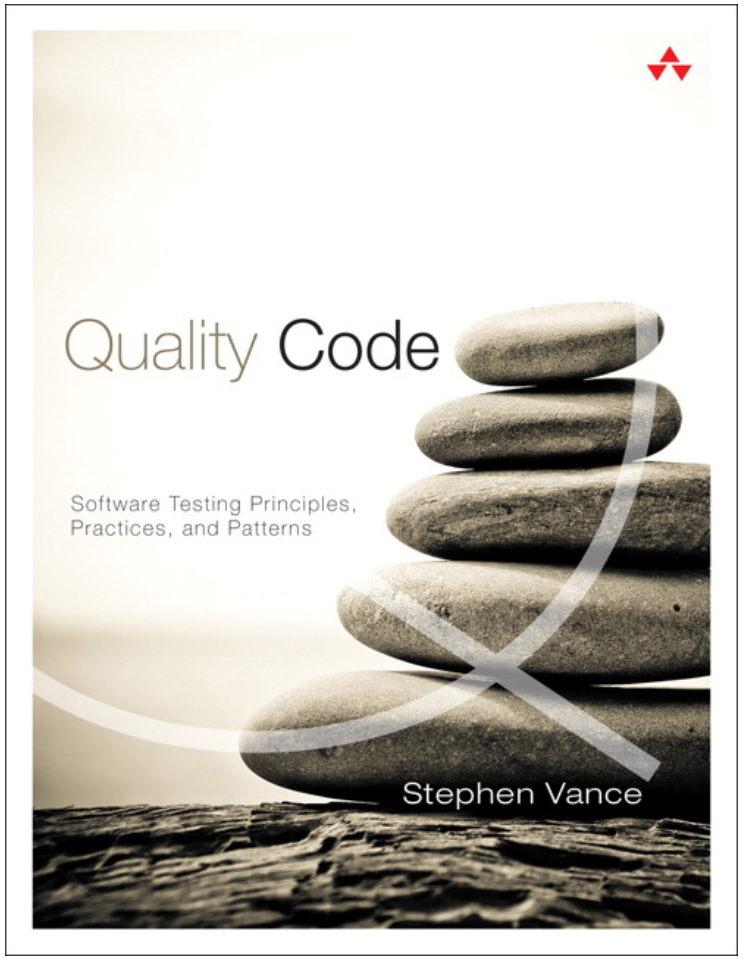
Want to inform the next generation of tools?

https://www.surveymonkey.com/r/longreen

References

- Crosby, Philip, “Quality is Free,” McGraw-Hill
- Tarlinder, Alexander, “Developer Testing: Building Quality into Software,” Addison-Wesley
- Thompson, Kate, “Zero Bugs and Program Faster,” Kate Thompson Publishing
- Weinberg, Gerald M., “Errors: Bugs, Boo-boos, Blunders,” Leanpub

Informit.com/aatc



Save 40% through April 30, 2017

- Use code AATC
- Good on print & eBook
- eBook files include PDF, EPUB, and MOBI

Offer only good at informit.com/aatc

Pearson

informIT®
the trusted technology learning source

Contact Me

Stephen Vance

https://www.vance.com

steve@vance.com

@StephenRVance

srvance on GitHub and LinkedIn