Schools of Software Testing

Bret Pettichord

bret@pettichord.com
www.pettichord.com

March 2007

What is a School?

Defined by

- Intellectual affinity
- Social interaction
- Common goals

Made up of

- Hierarchies of values
- Exemplar techniques
- Standards of criticism
- Organizing institutions
- Common vocabulary

Schools are *not* defined by

- Common doctrine
- Specific techniques

5 Views of Testing

Analytic School
sees testing as rigorous and technical with many proponents in academia

Standard School
sees testing as a way to measure progress with emphasis on cost and repeatable standards

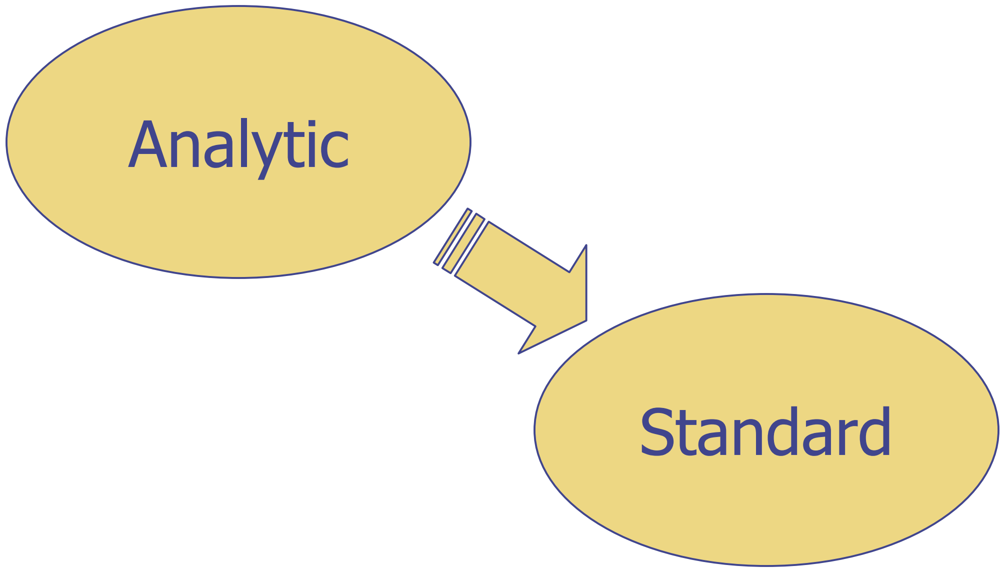
Quality School
emphasizes process, policing developers and acting as the gatekeeper

Context-Driven School
emphasizes people, seeking bugs that stakeholders care about

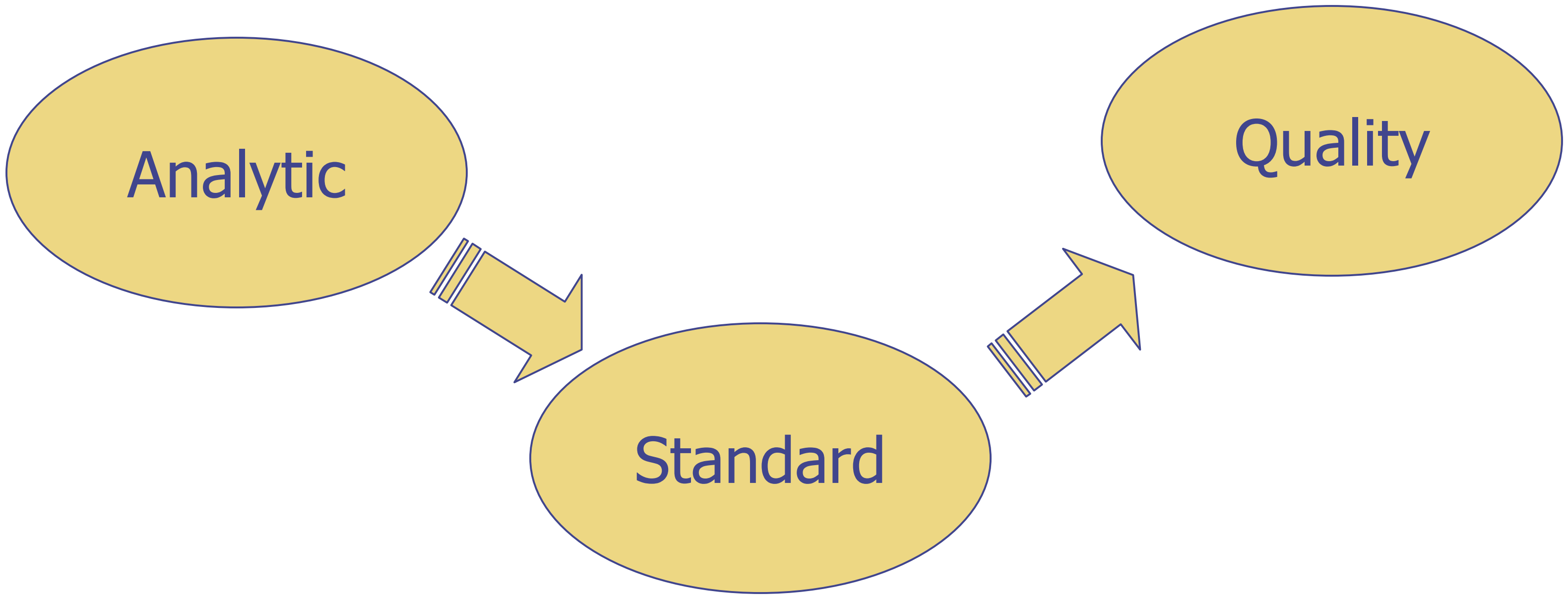
Agile School
uses testing to prove that development is complete; emphasizes automated testing

Development of the Schools

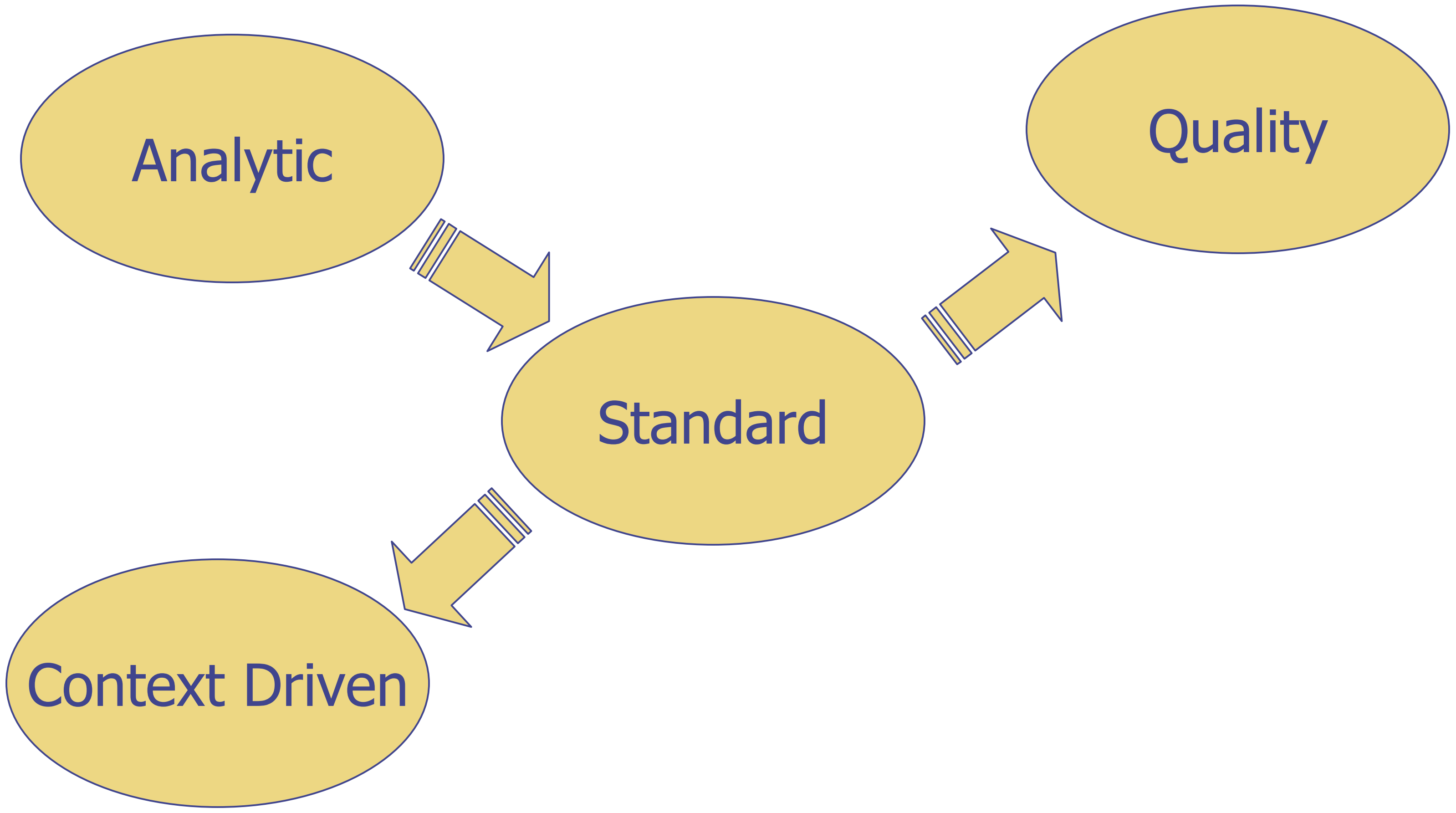
Development of the Schools



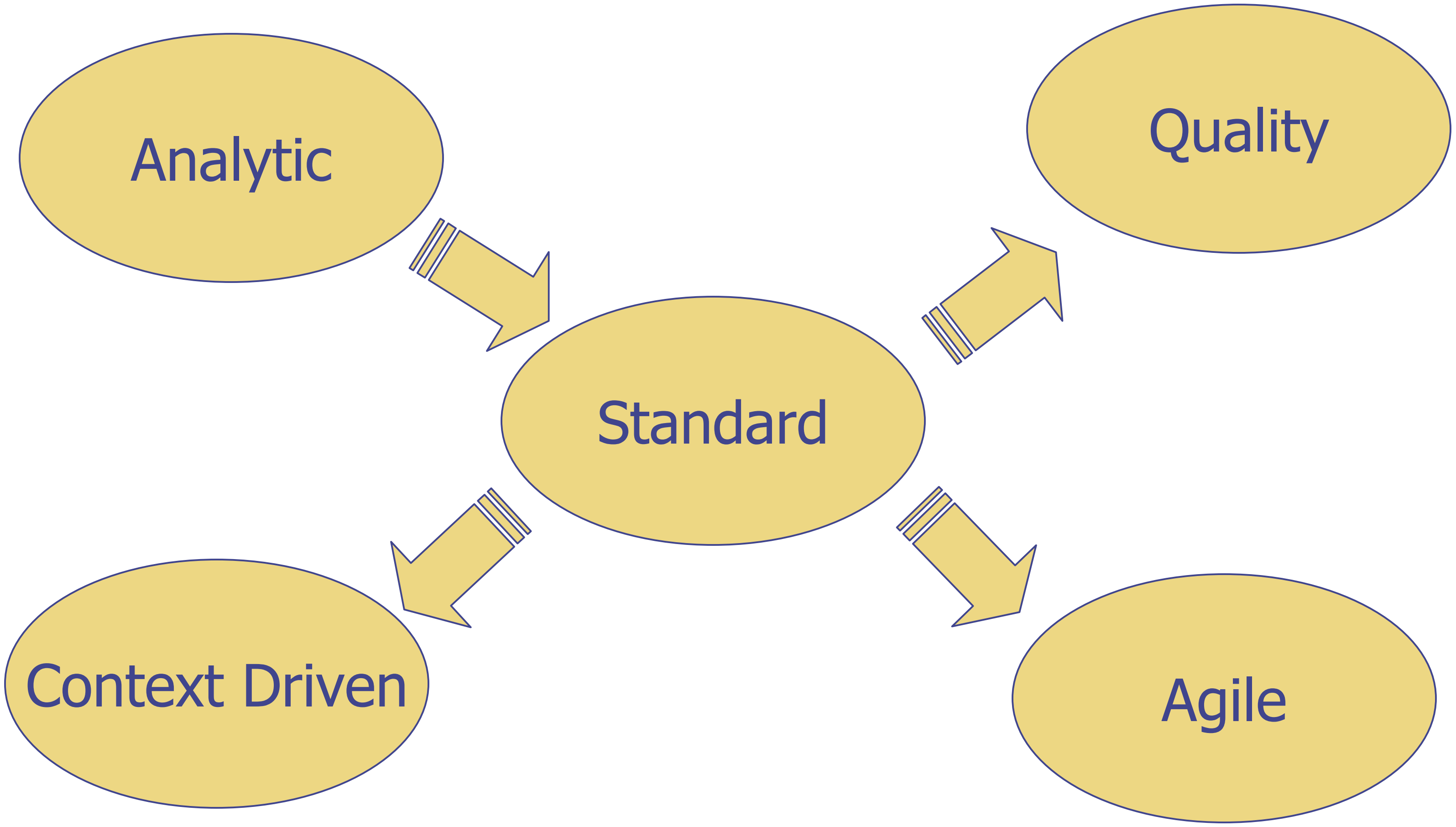
Development of the Schools



Development of the Schools



Development of the Schools



Why Classify Testing Ideas into Schools?

- Understand why testing experts disagree
  - Not simply a matter of personality or experience
  - There are often underlying reasons for disagreement
- Improve the basis for debate
  - Differences in values may explain why we favor different policies

But it can also be used to dismiss ideas you don't agree with.

Schools of Psychology

- Structural
  - James
- Behavioral
  - Watson, Skinner, Pavlov
- Gestalt
  - Wertheimer
- Psychoanalytic
  - Freud, Jung
- Cognitive
  - Piaget
- Humanistic
  - Rogers, Maslow

Analytic School Core Beliefs

- Software is a logical artifact
- Testing is a branch of CS/Mathematics
  - Objective, rigorous, comprehensive
- Testing techniques must have a logico-mathematical form
  - "one right answer"
- Testing is technical
- Key Question:
  Which techniques should we use?

Analytic School Exemplar

Code Coverage

- aka “Structural” testing
- Dozens of code-coverage metrics have been designed and compared
- Provides an objective measure of testing

Analytic School

- Implications
  - Require precise and detailed specifications
  - Testers verify whether the software conforms to its specification
  - Anything else isn't testing
- Most prevalent
  - Telecom
  - Safety-Critical
- Institutions
  - Academia

Standard School Core Beliefs

- Testing must be managed
  - Predictable, repeatable, planned
- Testing must be cost-effective
  - Low-skilled workers require direction
- Testing validates the product
- Testing measures development progress
- Key Questions:
  How can we measure whether we're making progress? When will we be done?

Standard School Exemplar

Traceability Matrix

- Make sure that every requirement has been tested

Standard School

- Implications
  - Require clear boundaries between testing and other activities (start/stop criteria)
  - Resist changing plans (complicates progress tracking)
  - Software testing assembly line (V-model)
  - Accept management assumptions about testing
  - Encourage standards, “best practices,” and certification
- Most Prevalent
  - Enterprise IT
  - Government
- Institutions
  - IEEE Standards Boards
  - Tester Certifications

Quality School Core Beliefs

- Software quality requires discipline
- Testing determines whether development processes are being followed.
- Testers may need to police developers to follow the rules.
- Testers have to protect users from bad software.
- Key Question:
  Are we following a good process?

Quality School Exemplar

The Gatekeeper

- The software isn’t ready until QA says it’s ready

Quality School

- Implications
  - Prefer "Quality Assurance" over "Testing"
  - Testing is a stepping stone to "process improvement"
  - May alienate developers
- Most Prevalent
  - Large bureaucracies
  - Organizations under stress
- Institutions
  - American Society for Quality
  - Software Engineering Institute (CMM)
  - ISO

Context-Driven School Core Beliefs

- Software is created by people. People set the context.
- Testing finds bugs. A bug is anything that could bug a stakeholder.
- Testing provides information to the project
- Testing is a skilled, mental activity
- Testing is multidisciplinary
- Key Question:
  What testing would be most valuable right now?

Context-Driven School Exemplar

Exploratory Testing

- Concurrent test design and test execution
- Rapid learning

Context-Driven School

- Implications
  - Expect changes. Adapt testing plans based on test results.
  - Effectiveness of test strategies can only be determined with field research
  - Testing research requires empirical and psychological study
  - Focus on skill over practice
- Most Prominent
  - Commercial, Market-driven Software
- Institutions
  - LAWST Workshops & Spin-offs

Agile School Core Beliefs

- Software is an ongoing conversation
- Testing tells us that a development story is complete
- Tests must be automated
- Key Question: *Is the story done?*

Agile School Exemplar

Unit Tests

- Used for test-driven development

Agile School

- Implications
  - Developers must provide automation frameworks
  - Slow to appreciate value of exploratory testing
- Most Prevalent
  - IT Consulting
  - ASP Development
- Institutions
  - Pattern Workshops

What is Testing?

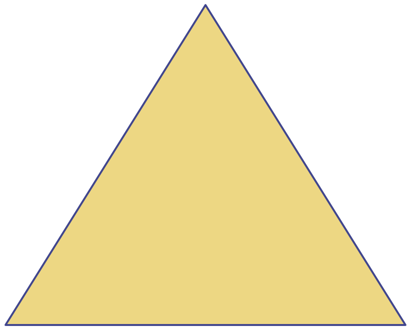
- Analytic School says
  - A branch of computer science and mathematics
- Standard School says
  - A managed process
- Quality School says
  - A branch of software quality assurance
- Context-Driven School says
  - A branch of development
- Agile Schools says
  - Part of the customer role

Testing the Triangle Program

The Triangle Program

- Takes three inputs:
  - the sides of a triangle
- Determines the type of triangle:
  - isosceles,
  - scalene, or
  - equilateral

How many tests should you run?



Author	School	Tests
Hetzel	Standard	28
Jorgenson	Analytic	35
Binder	Analytic	65
Beck	Agile	6

Four Views of Risk-Based Testing

- Analytic
  - Use operational profiles
  - Calculate reliability
- Standard
  - Key risk: failure to meet schedules (project risk)
  - Top down assessment of feature risks
- Quality
  - Uncover *project* risks
  - Prove that project is out of control
- Context-Driven
  - Testing develops team understanding of risks
  - Develop testers' ability to design tests for identified risks

Controversy: Testing Without Specs

FOR

- Context-Driven School
  - Do what you can to be useful
  - Ask questions if necessary
  - Dig up “hidden” specs
- Agile
  - Conversation is more important than documentation

AGAINST

- Analytical School
  - Impossible
- Standard School
  - Some kind of spec is necessary
- Quality School
  - Force developers to follow the process

Controversy: Tester Certification

FOR

- Standard School
  - Make testers easier to hire, train and manage
- Quality School
  - Increase status

AGAINST

- Context-Driven School
  - Existing certifications are based on doctrine, not skill
- Analytic School
  - Prefer university degrees over certification

Open Questions

- What happens when people of different schools work together?
- Can we cross-fertilize between schools?
- Do I have to pick a school?

Origin of this Analysis

- This is not my idea
- I learned it from Kaner, Bach and Marick
- It was a working concept behind “Lessons Learned in Software Testing”
- But each of us prefers different labels and characterizations for the different schools.

Thanks for reviews and discussions: James Bach, Rex Black, Michael Bolton, Ross Collard, Kathy Iberle, Cem Kaner, Brian Marick, Wayne Middleton, Tim Van Tongeren, and the participants of the 2003 Workshop on Teaching Software Testing at Florida Tech, the 2003 Pacific Northwest Software Quality Conference, the 2003 Software Testing Australia New Zealand conferences, and the Austin SPIN.

Learn more about Context-Driven Testing

- *Lessons Learned in Software Testing: A Context-Driven Approach*
  - Cem Kaner, James Bach & Bret Pettichord
- Mailing List
  - http://groups.yahoo.com/group/software-testing/