Testing Notes 2011 *(derived from james bach)*

Contents

Becoming a Software Testing Expert

Are you a testing expert

- Analyze these claims
  - You should write a test plan
  - It's important that testing be repeatable
  - Each test case should have an expected result
  - Test automation saves money and time
  - All testing is based on a model of what is being tested
  - Good enough quality is not good enough
  - An undocumented test cannot be improved
  - Exploratory tesing is a useful practice
  - It's better to use the term defect than bug
  - Ambiguity should be removed from requirements

Experts v.s Non-Experts

- For Any given claim or problem
  - Non-Experts are More Likely to Say
    - Yes, that's what the books say
    - This is right
    - This is wrong
    - I don't know (awkward silence)
  - Experts are More Likely to Say
    - Tell me more about the context
    - I can think of how that ight be gtrue and I can think of how it might be false. Let's think it through
    - Let me reframe that..
    - Here are some possible answers
    - Here's one way I've solved this
    - I don't know. Here's how I will find out..
- By "expert tester" I man any of the following
  - Someone who's very good at testing

    - Someone who's considered to be an expert
    - You may already be an expert testser – let's get better according to whatever standard is important to us.
- Perfect testing is
    - Testing infinite process of comparing the invisible to the ambiguous so as to avoid the unthinkable to the anonymous
    - …perfect testing is a challenge
- Testing is questioning a product in order to evaluate it
    - Through the cognition of the tester, good testing emerges from the infinite space of perfect testing.
- What is special About Testing
    - There are few people around to teach you how to test
    - Most of what is taugh as "Testing" is unreliable or misleading folklore
    - Testing is a complex problem-solving activity
    - Learning testing on your own doesn't cost you much, you don't need anyone's permission, and it generally poses no threat to life or property
    - However
        - It's hard to know if you are doing it well
        - Good testing varies quite a lot with the context
- Testing is not part of Computer Science

Context-Driven Testing Community:

- The Value of any practice depends on its context
- There are good practices in context, but there are no best practices
- People, working together, are the most important part of any project's context
- Projects unfold over time in ways that are often not predictable
- The product is a solution. If the problem isn't solved, the product doesn't work
- Good software testing is a challengeing intellectual process
- Only through judgment and skill, exercised cooperatively throughout the entire project, are we able to do the right things at the right times to effectively test our products

Some Cool Things That Experts Have and Do

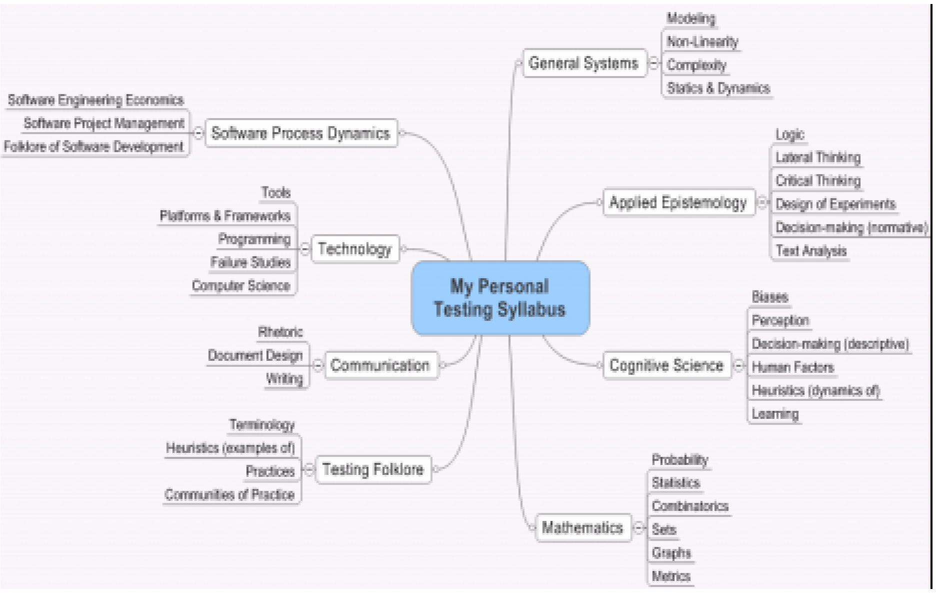
- Experts have
    - Situational awareness
    - Confidence in confusion
    - Collegue network
    - Trained reflexes
    - Awareness of limitations
    - Diverse experiences
    - Relevant Knowledge
    - Mental models for problem-solving
    - Reputation
- Experts do
    - Avoid traps and dead ends
    - Systematic inquiry
    - Confront authority and convention
    - Self-training and retraining
    - Self-criticism
    - Pattern matching on experience

    - Coherent explanations
    - Justify methodology
    - Write, speak, teach

Expert Performance is Situational

- Expertise is situated
    - Socially: if your clients and peers don't like your work, you won't be able to function as an expert in their eyes
    - Psychologically: If you're tired, angry, or indifferent, you won't perform well
    - Technically: If you don't know anything about databases, for instance, you will be limited in how well you test them.
- A Personal Vision of Testing Expertise
    - I can'
        - Test anything
        - Under any conditions
        - In any time frame
    - Ralative to
        - My standing in the local community
        - How hard I try
        - My technical insight
    - I can
        - Test anything
        - Under any condition
        - In any time frame
    - Such that
        - I deliver useful results in a usable form
        - I perform at least as well as any other expert would
        - I choose methods that fit the situation
        - I can explain and defend my work on demand
        - I collaborate effectively with the project team
        - I make appropriate use of the available tools resources
        - I advise client about the risks and limitation of my work
        - I advise clients about how they can help me do better work
        - I faithfully and ethically serve my clients

My Personal Testing Syllabus



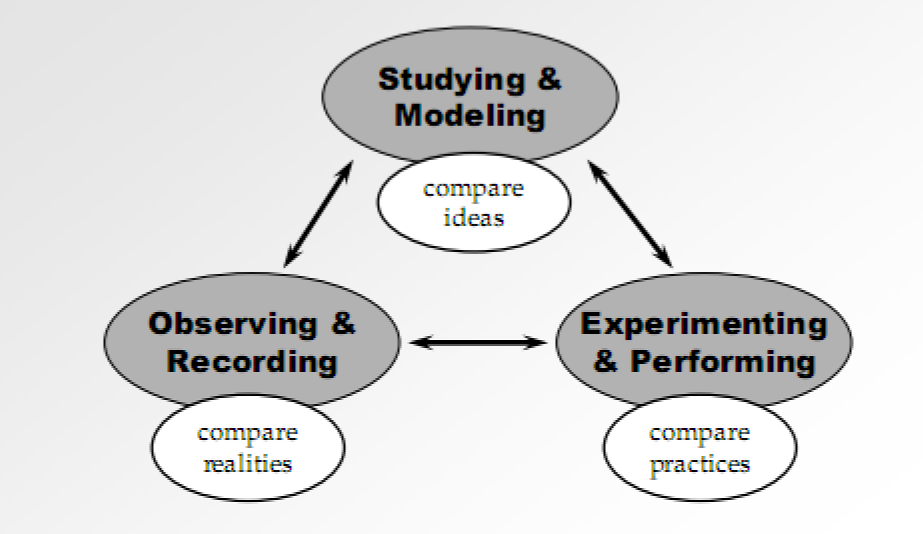
Notice No Testing Books

__ Gödel, Escher, Bach: An Eternal Golden Braid , Douglas Hofstadter.
__ The Sciences of the Artificial, 3rd Ed., 1996, Herbert A. Simon
__ Introduction to General Systems Thinking, 1975, Gerald M. Weinberg.
__ Secrets of Consulting, 1986, Gerald M. Weinberg
__ More Secrets of Consulting, 2002, Gerald M. Weinberg
__ Quality Software Management, Vol. 1: Systems Thinking, 1991, Gerald M. Weinberg
__ General Principles of Systems Design, 1988, Gerald M. Weinberg, Daniela Weinberg
__ Tools of Critical Thinking, 1997, David A. Levy
__ Exploring Requirements: Quality Before Design, 1989, Don Gause, Gerald M. Weinberg
__ The Social Life of Information, 2000, John Seely Brown, Paul Duguid
__ How to Solve It, 1945, George Polya
__ How to Read and Do Proofs, 1990, Daniel Solow
__ Judgment and Decision Making, 2000, Terry Connolly, et al
__ Cognition in the Wild, 1996, Edwin Hutchins
__ Thinking and Deciding, 1994, Jonathan Baron
__ Lateral Thinking: Creativity Step by Step, 1990, Ed De Bono
__ Abductive Inference: Computation, Philosophy, Technology, 1996, John R. Josephson, Susan G. Josephson
__ Time Pressure and Stress in Human Judgment and Decision Making, 1993, Ola Svenson, A. John Maule
__ Conjectures and Refutations: The Growth of Scientific Knowledge, 1992, Karl Popper
__ Proofs and Refutations, 1976, Imre Lakatos
__ The Pleasure of Finding Things Out, 1999, Richard Feynman
__ Rethinking Systems Analysis and Design, 1988, Gerald M. Weinberg
__ Quality Software Management, Vol. 3: Congruent Action, 1994, Gerald M. Weinberg
__ Becoming a Technical Leader: An Organic Problem-Solving Approach, 1986, Gerald M.Weinberg

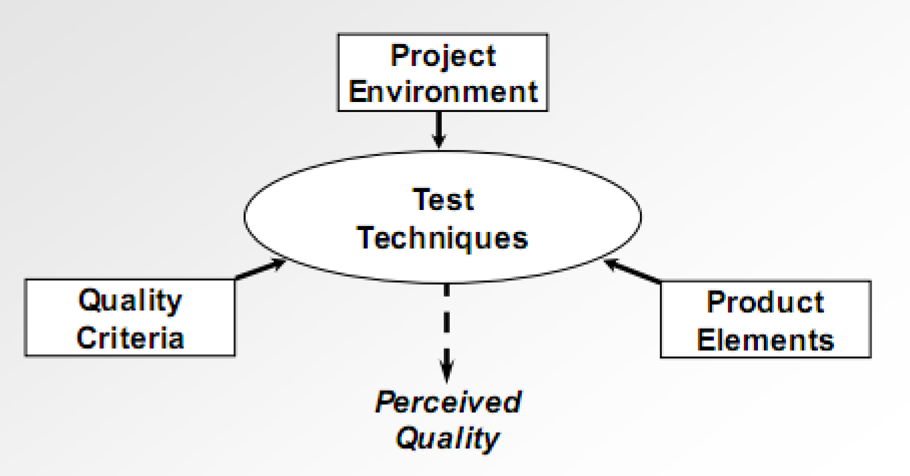
What Level of Learning

- Level 0: I overcame obliviousness
  - I now realize there is something here to learn
- Level 1: I overcame intimidation
  - I fell I can learn this subject or skill. I know enough about it so that I am not intimidated by people who know more than me.
- Level2: I overcame incoherence
  - I no longer feel that I'm pretending or hand-waving. I fell reasonably competent to discuss or practice. What I say sounds like what I think I know.
- Level3: I overcame competence
  - Now I feel productively self-critical, rather than complacently good enough. I want to take risks, invent, teach, and push myself. I want to be with other enthusiastic students.

Developing your expertise



All Product Testing is Something Like This



Thirty Six Heuristics

- Customers
- Information
- Developer relations

- Team
- Equipment and Tools
- Schedule
- Test Items
- Deliverables
-
- Structures
- Functions
- Data
- Platforms
- Operations
- Time
-
- Capability
- Reliability
- Usability
- Security
- Scalability
- Performance
- Installability
- Compatibiliity
- Supportability
- Testability
- Maintainability
- Portability
- Localizability
-
- Function testing
- Domain Testing
- Stress Testing
- Flow testing
- Scenario testing
- Claims testing
- User testing
- Risk testing
- Automatic testing

What about Certifiction

- A lot of people feel that college degrees comprise some kind of certification. I haven't felt a need for them.
- I have not seen a "Certified Tester" program that I respect. The one I've seen are an embarrassment to the craft and an insult to skilled testers everywhere.
- Certification by association has worked well for me.
- Certification by body of work also works well.
- When you make a name for yourself, your name trumps any "certification" you might also have.

My Advice to New Experts

- Practice, practice, practice!

- Don’t confuse experience with expertise
- Don’t trust folklore – but learn it anyway
- Take nothing on faith. Own your methodology
- Drive your own education – no one else will
- Reputation = Money – Buld and protect your reputation
- Relentlessly gather resources, materials, and tools
- Establish your standards and ethics
- Avoid certifications that trivialize the testing craft
- Associate with demanding colleagues
- Write, speak, and always tell the truth as you see it
-

Exploratory Testing

Contrasting Approaches

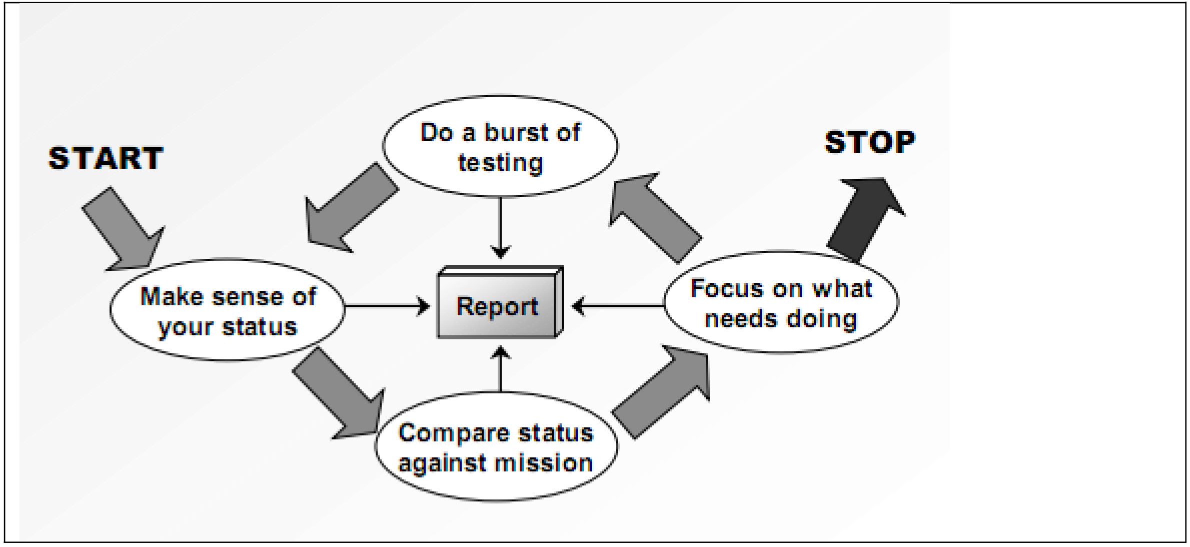
- Scripted Testing
  - Tests first designed and recorded, then executed later
  - Re-usable
- Exploratory Testing
  - Tests designed and executed at same time
  - Often not recorded

ET Is Good For

- it is not obvious what the next test should be.
- want to go beyond the obvious tests

What is ET?

- Cycle of self management


START
Do a burst of testing
STOP
Make sense of your status
Report
Focus on what needs doing
Compare status against mission

- Concurrent, interacting test tasks
  - Burst Testing:

  - Study the product
  - Model the test space
  - Select what to cover
  - Determine test oracles
  - Configure the test system
  - Operate the test system
  - Observe the test system
  - Evaluate the test results
  - Notice issues
  - Organize notes
- The core practice of a skilled tester
  - Challenge constraints and negotiate their mission
  - Alert their clients to project issues that prevent good testing
  - Spontaneously coordinate and collaborate, as needed
  - Train their minds to be cautious, curious, and critical
  - Know how to design questions and experiments
  - Know the difference between observation and inference
  - Have developed resources and tools to enhance performance
  - Adapt their test stratey to fit the situation
  - Tolerate substantial ambiguity, indecision, and time pressure
  - Take notes and report results in a useful and compelling way
  - Have earned the trust placed in them

Getting the Most Out of ET

- Augment ET with scripted tests and automation
  - Usually it's best to use a diverisified strategy
- Exploit inconsistency
  - Let yourself be distracted by anomalies and new ideas
  - Avoid repeating the exact same test twice
- Exploit the human factor
  - Encourage variablilty among testers
  - Exploit subject matter expertise
  - Use your confusion as a resource
  - Work in pairs or groups, whenever possible
  - Get to know your developers
  - Test in short bursts
- Learn the logic of testing
  - Conjecture and refutation
  - Abductive inference
  - Correlation and causality
  - Design of experiments
  - Forward, backward, and lateral thinking
  - Biases, heuristics, and human error
  - Risk, benefit , and the meaning of "good enough"
- Practice critical reading and interviewing
  - Analyzing natural language specifications
  - Analyzing and cross-examining a developer's explanation

- Exploratory questions
    - How well does your car work?
    - How do you know how well your car works?
    - What evidence do you have about how your car works?
    - Is that evidence reliable and up to date?
    - What does it mean for your car to work?
    - What facts would cause you to believe that your car doesn't work?
    - In what ways could your car not work, yet seem to you that it does?
    - In what ways could your car work, yet seem to you that it doesn't?
    - What might cause your car to not work well (or at all)?
    - What would cause you to suspect that your car will soon stop working?
    - Do other drivers operate your card? How does it work for them?
    - How important is it for your car to work?
    - Are you qualified to answer these questions? Is anyone else qualified?
- Adpot a clear and consistent testing vocabulary
    - Bug, specification, risk, test strategy, test case, test plan
- Learn to model a product rapidly
    - Flowcharting; data flows; state model
    - Matrices and outlines
    - Function/data square
    - Study the technology
- Use a "grid search" strategy to control converage
    - Model the product in some way, then specify broad test areas in terms of that model (not specific test cases). Keep track of what areas you have and have not tested in.
- Learn to take reviewable notes
    - Take consice notes so that they don't interrupt your work
    - Record at least what your strategy was, what you tested, what problems you found, and what issues you have
- Practise responding to scrutiny
    - Why did you test that?
    - What was your strategy?
    - How do you know your strategy was worthwhile?
- Learn to spot obstacles to good testing
    - Not enough information, yet
    - The product isn't testable enough
    - Don't have enough of the right test data
- Develop and use testing heuristics
    - Try the heuristic test strategy model, on **my web site**
    - Practice critiquing test techniques and rules of thumb
    - Notice the test ideas you use that lead to interesting results
- If you're in a highly structured environment, consider using session-based test management to measure and control exploratory testing.
    - Package ET into chumks that can be tracked and measured
    - Protects the intuitive process; gives bosses what they want
    - See "how to measure Ad Hoc Testing" on Friday
    - See "Session-based Test Management" article in STQE

- Books to Help You Be an ET Savant
  - Tools for Critical Thinking,DavidLevy
  - Rethinking Systems Analysis and Design,GeraldM.Weinberg
  - Exploring Requirements, Don Gause & Gerald M. Weinberg
  - Lateral Thinking, Edward De Bono
  - Abductive Inference, John R. Josesphson
  - Proofs and Refutations, Imre Lakatos
  - Conjectures and Refutations, Karl Popper
  - Explanation Patterns,RogerS.Schank
  - The Sciences of the Artificial, Herbert Simon
  - Sensemaking in Organizations, Karl E. Weick
  - The Social Life of Information, John Seely Brown & Paul Duguid

Software Engineering Coming of Age? Not Yet.

Coming of Age What does that mean?

- We'ew finally finguring it out
- We know what we're doing
- The major debates are over
- Experts agree
- Academia and industry are in sync
- The good guys won

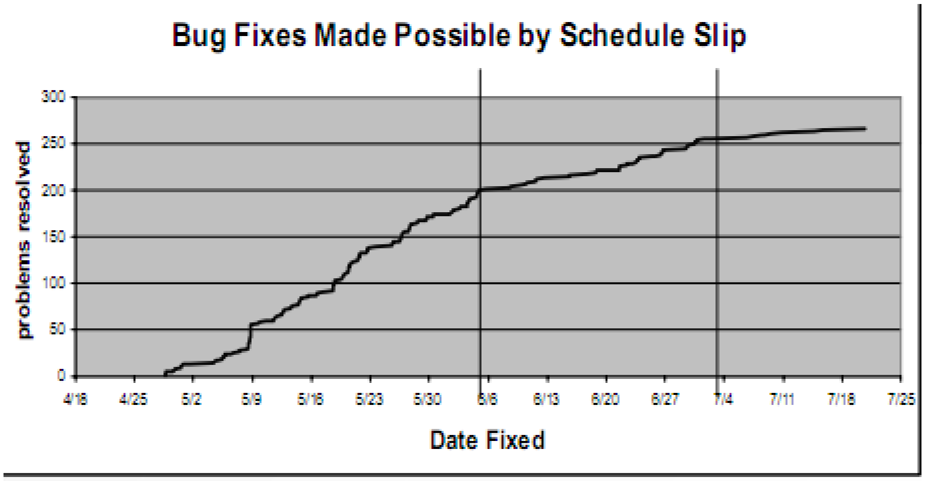
What's immature about SE?

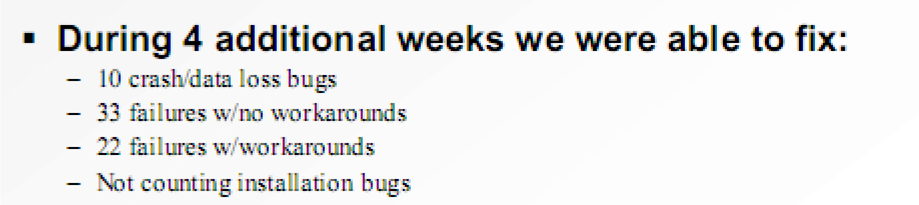
- Isolated enclaves
- Indoctrination without experimentation
- Experimentation without experimentation
- Experiminetation without critism
- Self-training or no training
- Dominated by short-term economics
- Perfunctory discourse
- Disinterest in related fields
- Unintelligible data
- Paradigm blindness

Do Our Students Study These Fields?

- Cognitive Psychology (thinking and perception)
- Family Psychology (small group behavior)
- Economics (allocating scarce resources)
- Graphic Design (effective documentation)
- Decision Theory (rationality despite uncertainty)

- Genera Systems Theory (patterns in systems)
- Qualitative Research (observing behavior)
- Measurement Theory (limits of quantification)
- Who has the time?



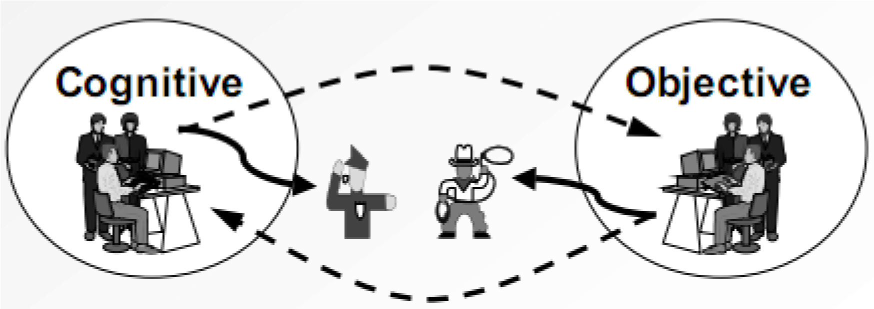


- **During 4 additional weeks we were able to fix:**
  - 10 crash/data loss bugs
  - 33 failures w/no workarounds
  - 22 failures w/workarounds
  - Not counting installation bugs

Contrasting Paradigms

- Objective
  - Defined process
  - Measurable progress
  - Traceability to specs
  - Documented work
  - Repeatable work
  - Organization focused
  - Compliance praised
  - Popular in Contract and Regulated communities
- Cognitive
  - Problem-solving
  - Meaningful results
  - Relevance to mission
  - Peer communication
  - Effective work
  - People focused
  - Technical insight praised

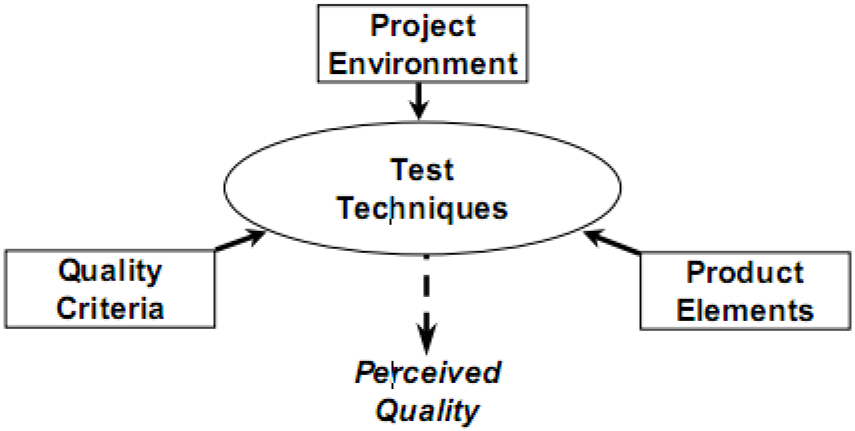
    - Popular in Market community
- Paradigm Blindness
    - Objective: process accountability matters
    - Cognitive: solution accountability matters

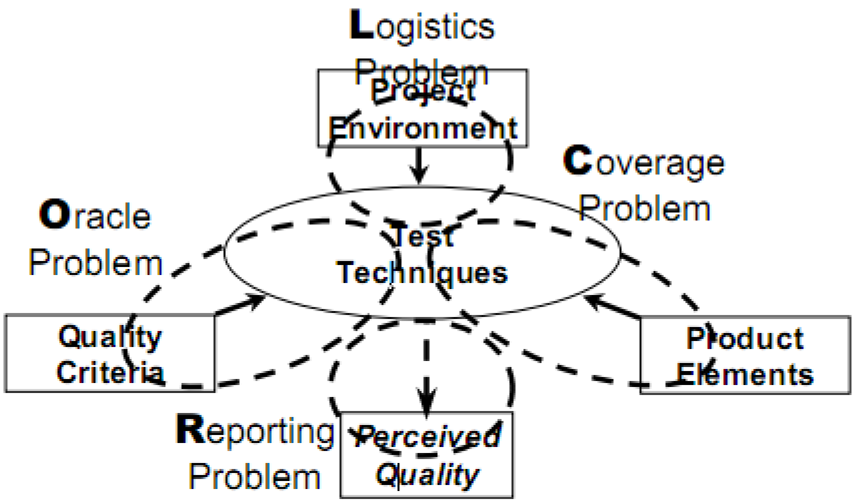


Explaining Testing to Anybody

- Basics of Explaining
    - Be quick, then stop and check in
    - Be humble – Exceptions and alternatives abound
    - You can be wrong, as long as you're thoughtful
    - Be Real. Own the explanation
    - Be respectful. Honor objections and questions
    - Be patient. Often, experience must come first
    - Be relevant. How does it relate to them? What's the bottom-line? What do you want them to do?
- Be Prepared
    - Quick point: a few sentences at most
    - Similie & Metaphor: relate to something familiar
    - Vocabulary check: are you using words the same way?
    - 5 minute whiteboard talk: diagrams are powerful
    - Concrete examples or demonstration: in-house data
    - Anticipate common questions and objections
    - References & supporting material: have it handy
- Explanations
    - Confusions
        - What is testing?
        - What's so hard about testing?
        - How do you know when you're done?

        - Why didn’t you find that bug?
    - Lets {do my crazy plan}, okay?
        - …change the product at the last minute…
        - …specify those tests in advance {and automate them}
- What is Testing?
    - Quick points (choose one):
        - Testing is organized skepticism. It’s the belief that things may not be as they seem; that things could be different.
        - Testing is comparing the ambiguous to the invisible, so as
        - to avoid the unthinkable happening to the unknown.
        - – Testing brings vital information to light, so we can see
        - where we are and make better decisions.
        - – Testing is a support function that helps developers look
        - good by finding their mistakes before anyone else does.
    - Similes and Metaphors
        - Police patrol (scanning for trouble in a complex world)
        - Secret Service (making risky activities safer)
        - Radar (threat detection while avoiding false alarms)
        - Copy editing (saving authors from embarrassment)
        - Soccer/hockey defense (we defend the goal)
        - Science (conjecture and refutation)

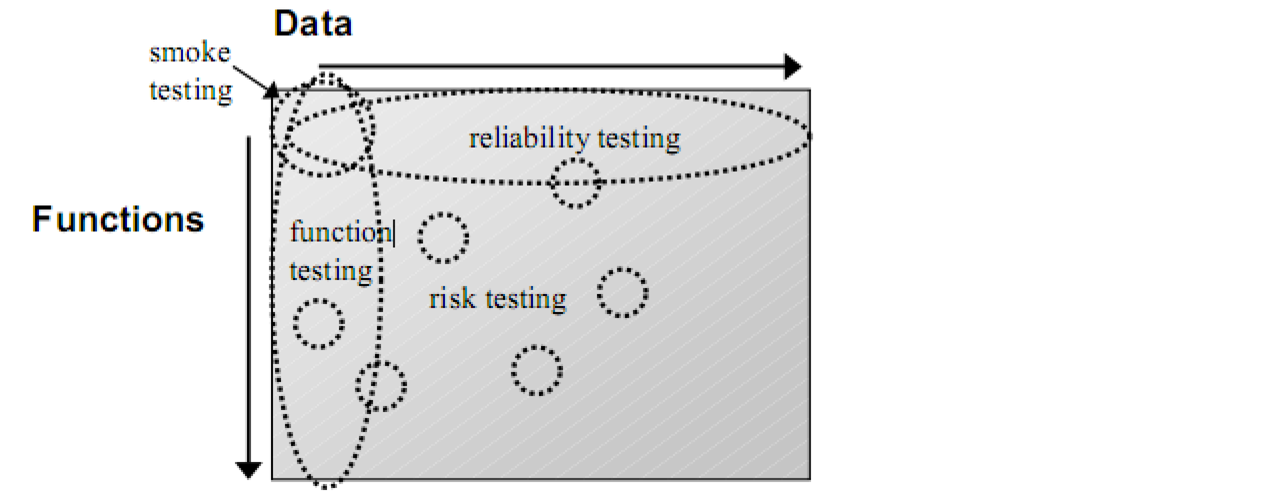




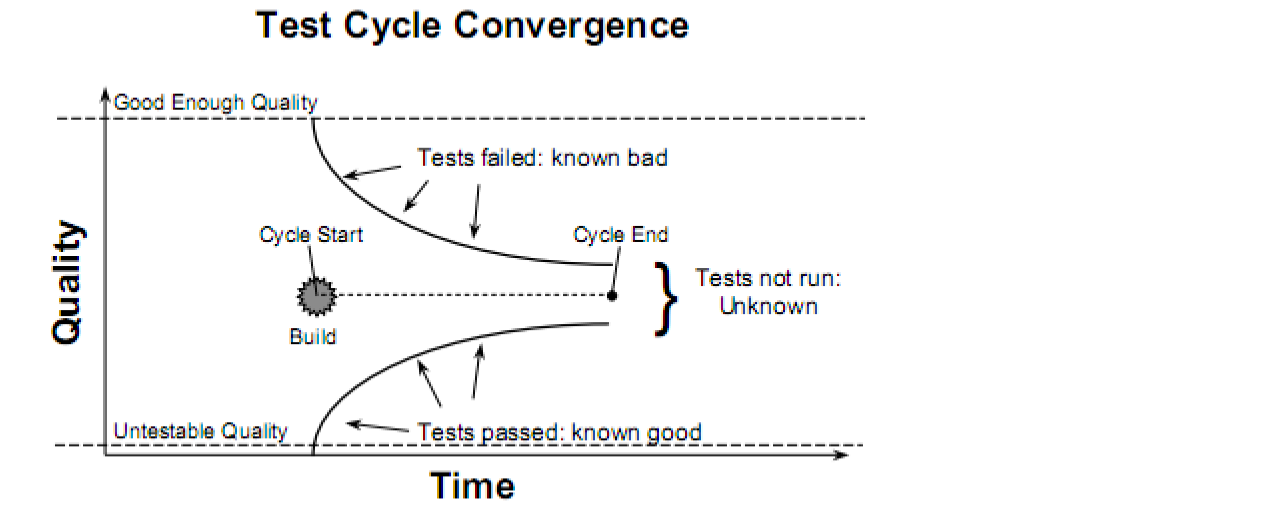
Supporting Material
– http://www.kaner.com/imposs.htm
Explains the impossibility of complete testing.
– http://www.satisfice.com/articles/test_automation_snake_oil.pdf
– http://www.kaner.com/lawst1.htm
Explain the challenges of regression test automation.
– http://www.satisfice.com/articles/good_enough_testing.pdf
Explains testing in the context of how much is enough.
– http://www.rstcorp.com/marick/classic/mistakes.html
A gold mine of marvelous little explanations

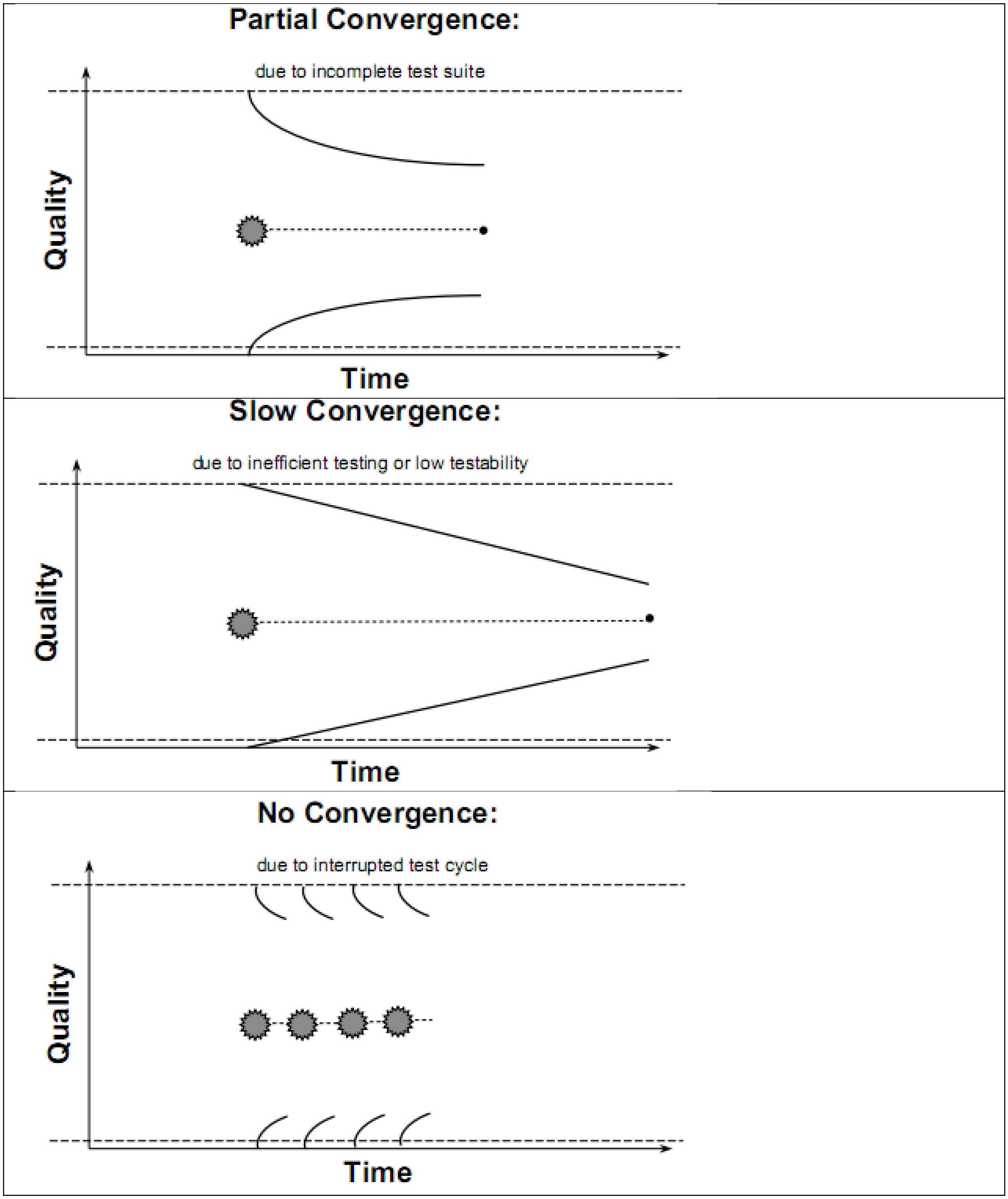
- Explanation Hints
  - Try highlighting one key dynamic, instead of explaining the whole issue.
  - Try beginning by explaining things in terms of the mission, then get more concrete from there.
  - Use a diagram that helps you examine different scenarios and issues as part of the dialog.
  - Be ready to explain things from more than one angle, in case they aren't receptive to your first try.
  - Use analogies that come from everyday experience, when you can. Then you can appeal to common sense.
  - It helps to validate the crazy plan even while you're opposing it. For one thing, no plan is totally crazy. For another, it's how they know you understand it.
  - The success of your explanations ultimately rests on your passion, your curiosity, and your humanity, not on your logic. (still… logic helps)

- What's so hard about testing?



- How do you know when you're done?
  - 1. When management has enough information to make an informed decision about the benefits and risks associated with the product.
  - 2. Management decides to release or cancel the project.
  - Both must be true or testing is not done.





Partial Convergence:

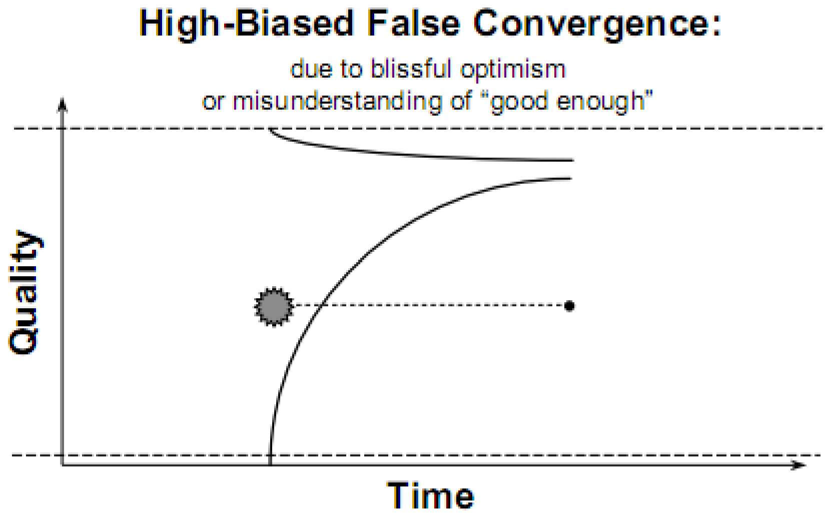
due to incomplete test suite
Quality
Time

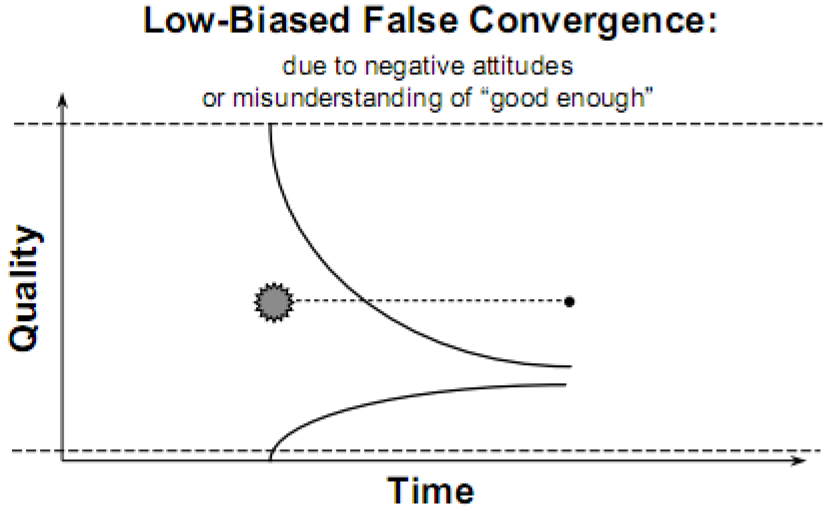
Slow Convergence:

due to inefficient testing or low testability
Quality
Time

No Convergence:

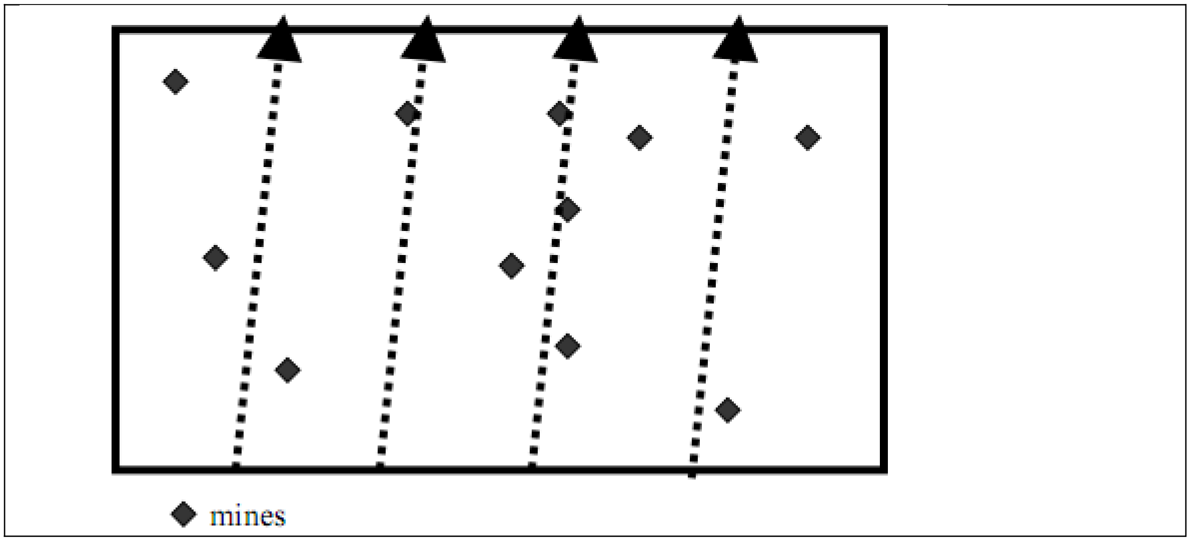
due to interrupted test cycle
Quality
Time

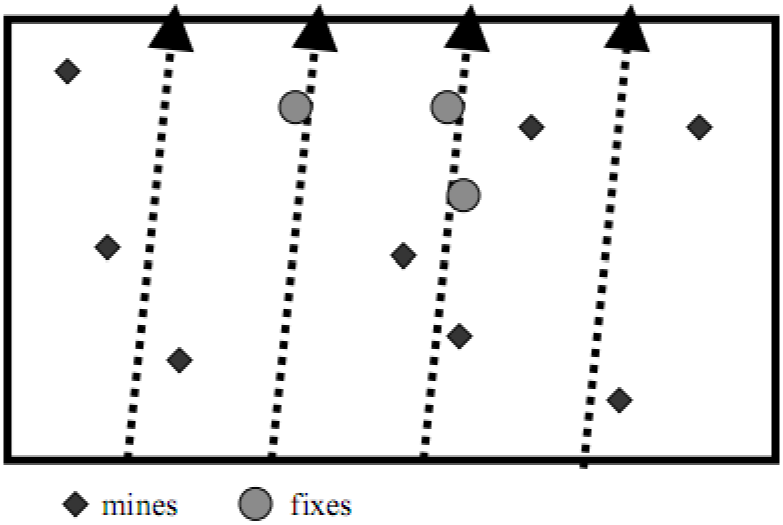

High-Biased False Convergence:
due to blissful optimism
or misunderstanding of "good enough"
Quality
Time

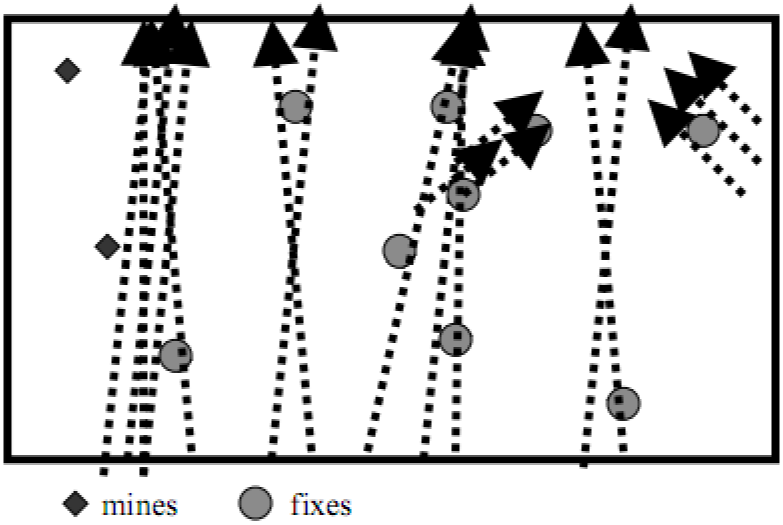

Low-Biased False Convergence:
due to negative attitudes
or misunderstanding of "good enough"
Quality
Time

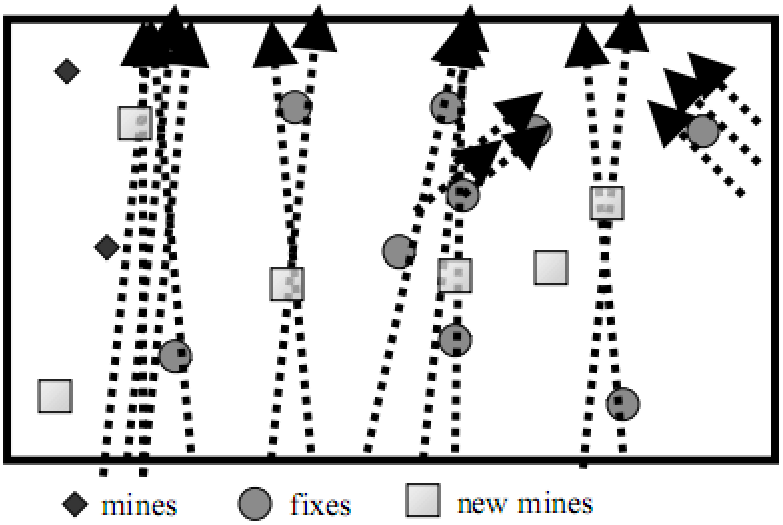
- Why didn't you find that bug?
  - Highway Patrol analogy: Do you realize how hard it would be to patrol the highways so that all speeders are always caught?
  - Risk-based improvement argument: Our goal is to find important problems fast. Was that bug important? Could we economically have anticipated it? If so, we'll modify our approach.
  - Testability reframe: We didn't find it because the developers didn't make the bug obvious enough for us to notice. The developers should put better bugs into the code. Let's make a more testable product.
- Responding to "Crazy Plans"
  - Discover the need that motivates their plan.
  - Honor something good in it. It's good practice to find points in favor even of those things you don't believe.
  - Say something like…

    - "I'm confused about your idea. Let me explain my confusion and then you can help me see how your plan would work."
    - "I don't know how to do what you propose without also creating the following problems…"
    - "I'm concerned about your proposal, because I haven't yet seen it work. What I have seen work is this…"
    - Then, explain the issue as you see it.
- Change the product at the last minute
    - Driving analogy
        - When on the freeway, go 65mph.
        - When on a suburban artery, go 40mph.
        - When in a neighborhood, go 25mph.
        - When in a parking lot, go 5mph.
        - Park very slowly.
    - Change Control Funnel
        - The closer you are to shipping, the more reluctantly you should change code.
    - To minimize retesting, carefully review proposed changes.
    - "You can change what you like, but change may invalidate everything we thought we knew about quality."
- Specify those tests in advance [and automate them]









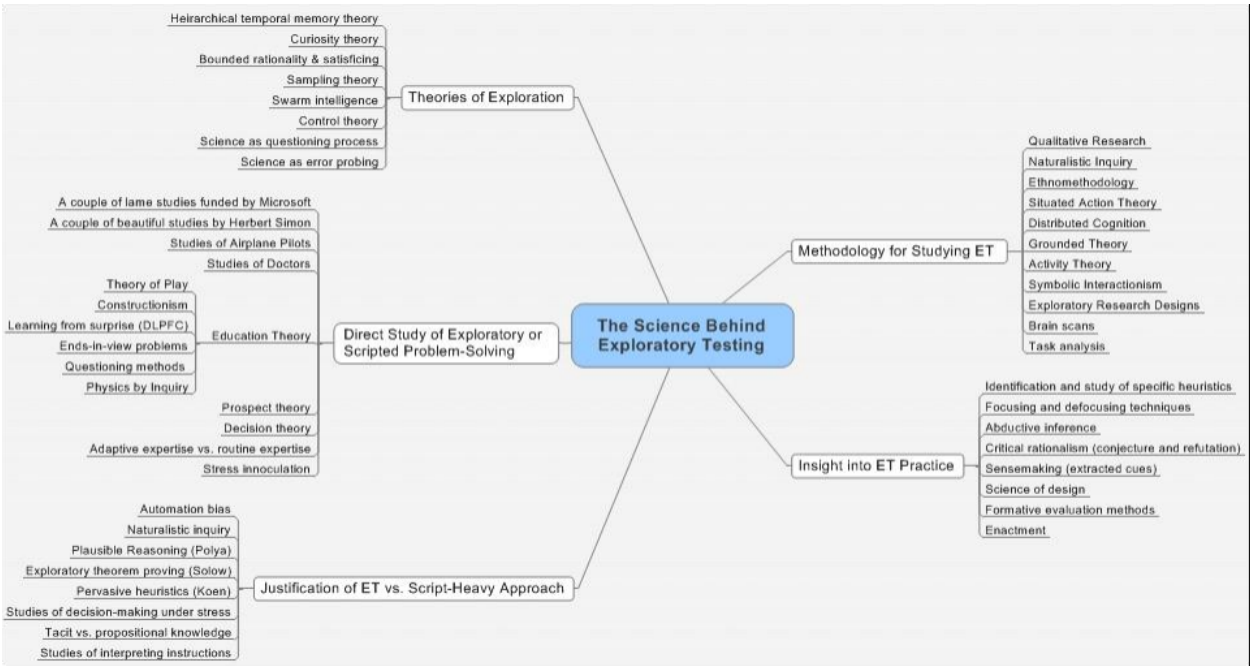
- Mine clearing analogy
  - To avoid mines, repeat steps exactly.
  - A repeatable test strategy can minimize the chance that you'll discover new

problems.
    - A variable test strategy will at least not avoid finding bugs.
- Principle of diverse half-measures
    - It’s usually better to do many different test techniques than it is to do one technique really well.
    - Maximize diversity in all dimensions your testing, if you want to find more problems and uncover more risks.
- Kaner’s Tour Bus: A test script is just a tour bus. Don’t fall asleep on the bus. Get off and look around.

The 5% Future of Testing

- Not much has changed in 10 years
- Self-Development/Earning Reputation
    - Project experiences
    - Writing
    - Teaching
    - Speaking
    - Innovation
    - Vision
    - Study
- Buccaneer-Scholar
    - Anyone whose love of learngin is not muzzled, yoked or shackled by any institution or authority; whose mind is driven to wander and find its own voice and place in the world.
- Three kinds of credentials
    - Reputation (public,local, or instant)
    - Portfolio
    - Performance
- Lots of new Ideas
    - Session-based test management
    - Thread-based test management
    - Exploratory/scripted continuum
    - Automatic test matrices
    - Video bug reports
    - Paired testing
    - Guideword heuristics
    - Gocusing/Defocusing heuristics
    - Testing playbooks
    - Sapient testing vocabulary

  - Testing games
- New skills
  - Test framing
  - Test factoring
  - Test chartering
  - Test narration
  - Test notetaking
  - Prospective testing
  - Test heuristic design
  - Bug advocacy
- What is the 5% Future?
  - Outrun and out-think the certification and standardization zombies. Make them irrelevant.
  - Sapient tester blogging and tweeting communities.
  - Practical self and peer-education (e.g. Weekend Testers, Peer Conferences, “Rat Pack”, IM coaching, online BBST course).
  - A strong ethical focus.
  - Testing organizations embracing pluralism, such as Association of Software Testing and the Context Driven testing community.
  - ..is a much smaller testing field. Where people who don’t care to educate themselves get outsourced away and then the outsource contracts canceled.
  - A much smaller community of highly skilled and ethical testers can work with domain experts and programmers to serve all the testing needed in large companies that want to do a great job.



The Science Behind Exploratory Testing
Theories of Exploration
Heirarchical temporal memory theory
Curiosity theory
Bounded rationality & satisficing
Sampling theory
Swarm intelligence
Control theory
Science as questioning process
Science as error probing
Methodology for Studying ET
Qualitative Research
Naturalistic Inquiry
Ethnomethodology
Situated Action Theory
Distributed Cognition
Grounded Theory
Activity Theory
Symbolic Interactionism
Exploratory Research Designs
Brain scans
Task analysis
Direct Study of Exploratory or Scripted Problem-Solving
A couple of lame studies funded by Microsoft
A couple of beautiful studies by Herbert Simon
Studies of Airplane Pilots
Studies of Doctors
Education Theory
Theory of Play
Constructionism
Learning from surprise (DLPFC)
Ends-in-view problems
Questioning methods
Physics by Inquiry
Prospect theory
Decision theory
Adaptive expertise vs. routine expertise
Stress innoculation
Insight into ET Practice
Identification and study of specific heuristics
Focusing and defocusing techniques
Abductive inference
Critical rationalism (conjecture and refutation)
Sensemaking (extracted cues)
Science of design
Formative evaluation methods
Enactment
Justification of ET vs. Script-Heavy Approach
Automation bias
Naturalistic inquiry
Plausible Reasoning (Polya)
Exploratory theorem proving (Solow)
Pervasive heuristics (Koen)
Studies of decision-making under stress
Tacit vs. propositional knowledge
Studies of interpreting instructions