Analysis

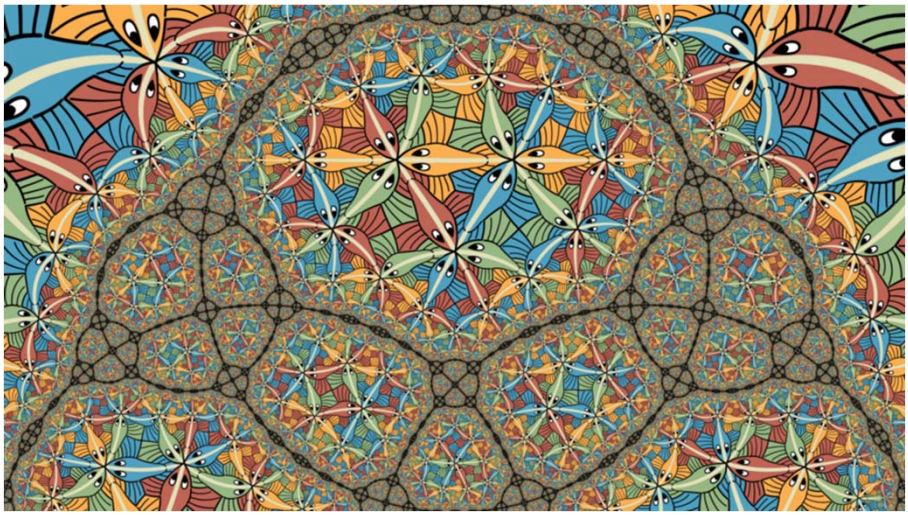
Michael Bolton
http://www.developsense.com
michael@developsense.com
Twitter: @michaelbolton

James Bach
http://www.satisfice.com
james@satisfice.com
Twitter: @jamesmarcusbach

Disclaimer

- This class will be largely experiential. We will do exercises and debriefs. (I will try—unsuccessfully—to avoid lecture). You'll do a lot of learning on your own, and with your group.
- I have some ideas, conclusions, and advice, but I'll have a lot more after the class, because YOU will generate them.
- I can help you to connect ideas to life in the working world, and I'll try to do so if you can't see how and if you ask for help. The colleagues around you can help too!
- If you don't like these ideas, it's okay with me if you go to someone else's class. Best to get it over with now.

Analysis is a Fractal Process!



Analysis - 10

Why Analysis for Testers?

- Testing, critical thinking, and analysis are all *intertwingled*.
- Each requires aspects of the other.
- Each helps to reinforce the others.
- They exhibit many similar patterns.
- They require many similar skills.
- They are not linear; they loop.

Analysis - 11

Analyzing Analysis

"decompositional"	*"regressive"*	*"transformative; interpretive"*
"up-loosening" (from the literal Greek meaning)	"working back to first principles"	"translation to correct logical form"
breaking down	working backwards to what is sought	
unpacking		
de-lumping		
drilling down		

***Stanford Encyclopedia of Philosophy*, "Analysis"**
https://plato.stanford.edu/entries/analysis

Analysis - 15

What Is an Analyst?

- **Analyst:** Anyone for whom analysis is an essential part of their work. (includes programmers, technical managers, system administrators, testers, tech. support people, scientists, mechanics, etc.)

What Does an Analyst Do?

- *Self-directed* exploration of systems
  - whether they are physical or abstract; natural or artificial; human or technological…
- Designing and using models of them
  - whether tacit or explicit, formal or informal, exact or approximate
- Designing and applying heuristics for that purpose
  - e.g. simple rules and shortcuts, comprehensive procedures, software tools…
- Developing an understanding of those systems

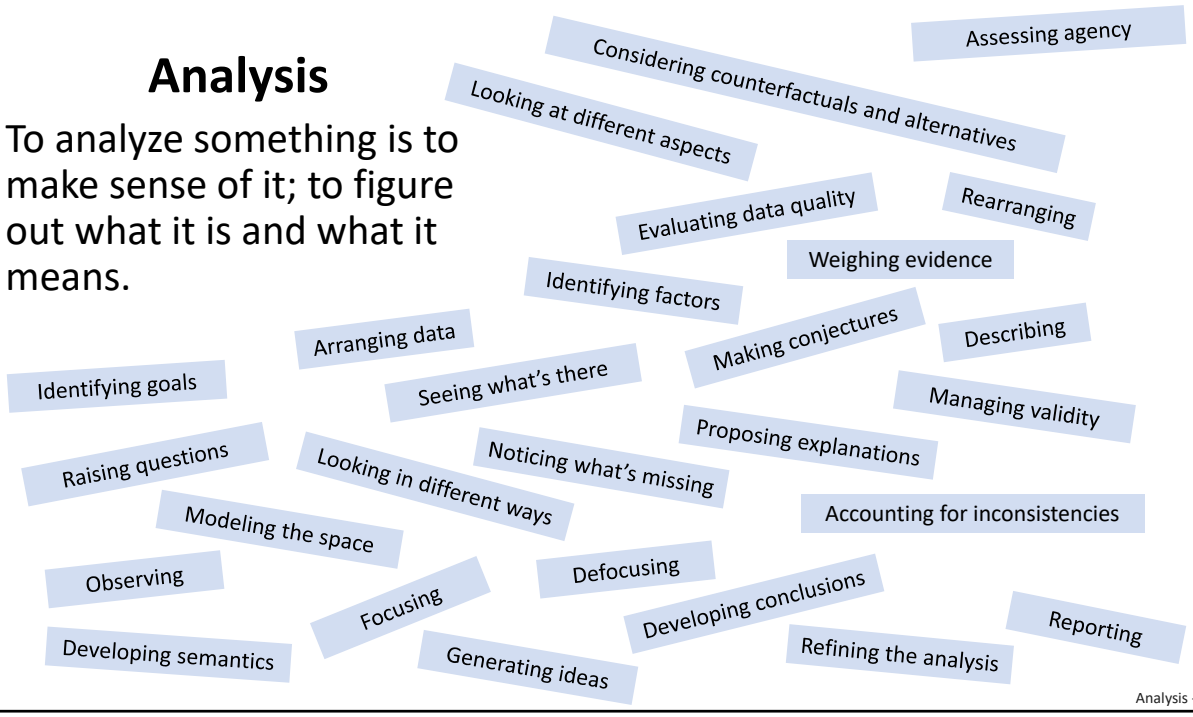
Plus

- Credible and relevant reporting
- Ongoing self-re-education
- Applying critical thinking

Analysis - 16

Analysis

To analyze something is to make sense of it; to figure out what it is and what it means.



Analysis - 18

Analysis is a *heuristic* process

Analysis - 19

Heuristics bring useful structure to problem-solving skill.

- **adjective:**

  "serving to discover."

- **noun:**

  "a fallible means of solving a problem."

A heuristic is not the same as a *rule*.

A heuristic ***can*** work but ***might*** fail.

"The engineering method is the use of heuristics to cause the best change in a poorly understood situation within the available resources."

"The engineer doesn't know where he is, where he is going, how he will get there, or if anyone will care when he does…The final state always has a reality that the engineer at the initial state cannot anticipate."

— Billy Vaughan Koen, Discussion of The Method

See "Heuristics for Understanding Heuristics"
http://www.developsense.com/blog/2012/04/heuristics-for-understanding-heuristics/

Analysis - 20

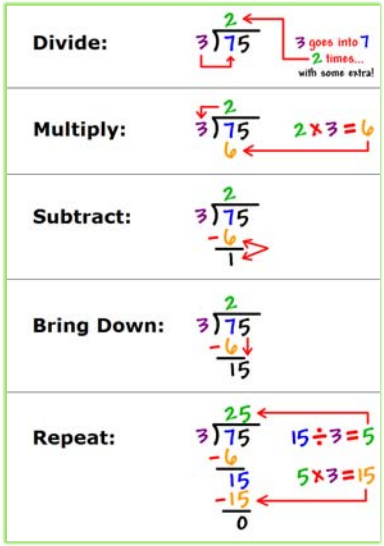
What is not a heuristic?

A deterministic process that solves a specific problem entirely within a formal system is an algorithm, **not** a heuristic.

Algorithms give guarantees only within formal systems. Therefore ask

- what is this method used for?
- does the formal system match real life?

Long division is a heuristic when used for equitably dividing a restaurant check.



Analysis - 21

Fallibility implies that wisdom and responsibility are critical.

Many heuristics are so effective we might be excused for treating them as infallible…

- "Don't hurt other people."
- "Breathe now."
- "Don't eat things that you find in the street."

But in engineering, when we treat heuristics as "best practices" we suffer. [note: that is a heuristic, too] For instance…

- "ALWAYS document your work with this template."
- "ALWAYS write a plan before starting development."
- "ALWAYS avoid planning before starting development."
- "ALWAYS use test driven design to write code."

Analysis - 22

Why is it dangerous to treat heuristics as "best practices?"

- "Best practices" are simply commandments.
- Commandments are mechanical, uncontrolled, and binary: "do or not do."
- Anything beyond "just doing it" is therefore illegible, invisible, and seemingly gratuitous.
- Thus, the analyst is incented to use any "commanded heuristic" in the **cheapest, laziest way**…
- … and to have **no concern** for outcome.
- This syndrome is very common. It is called:

Pathetic Compliance

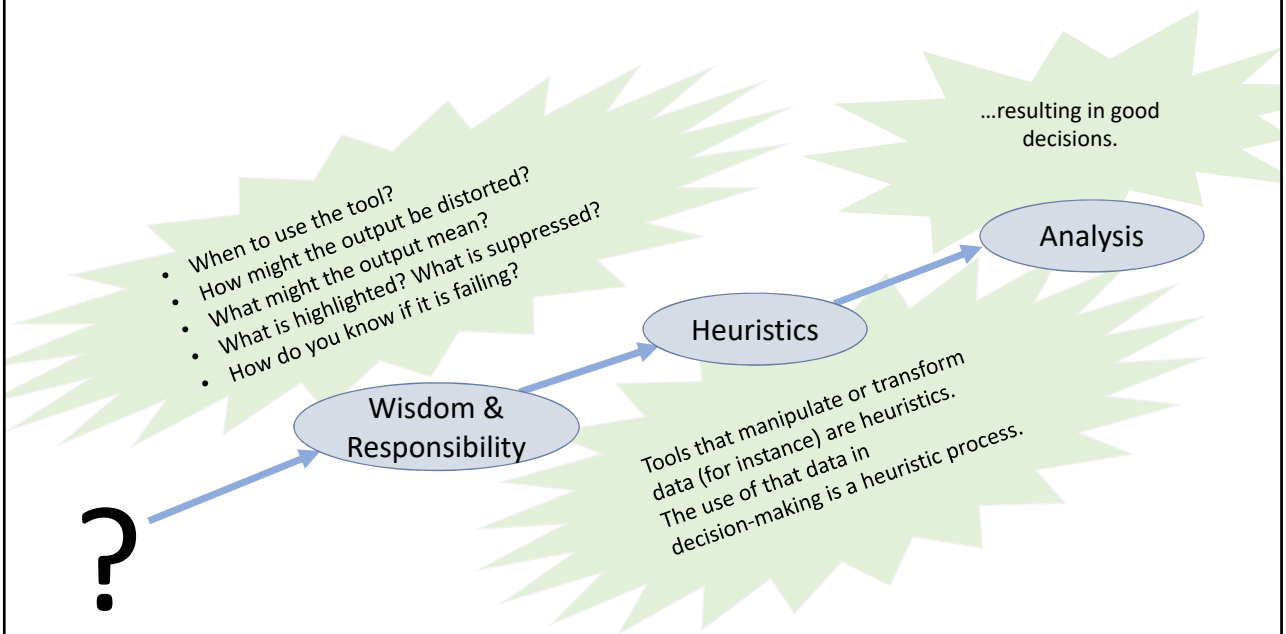
Analysis - 23

Two Ways to Achieve Safety

1. **Eliminate freedom**
   (Danger: Treats people as objects, requiring expensive vigilance, and inviting rebellion. It also means they have no power to do **good** things.)
2. **Instill responsibility & wisdom**
   (Danger: Lecture isn't enough to do that, not even this lecture. Those things require *development*.)

Analysis - 24

Development of wisdom and responsibility is critical. But how?



Analysis - 25

RESPONSIBILITY

"I CHOOSE to help (and not harm)."

WISDOM

"I KNOW how to help (and not harm) with my decisions."

FREEDOM

"I am ABLE to harm people." "...or help people."

Analysis - 26

Responsibility and wisdom require freedom to develop. Analysis without freedom is nothing but marketing.

FREEDOM

"I am ABLE to make decisions that may help or harm."

Analysis - 27

Managerial (and Parental) Compromises

Curtail freedom only as a **temporary contract** to prevent conflict, or in matters of imminent or extreme danger.

Proceed via **challenges to solve problems**, rather than assignments to perform tasks.

Create lower risk **"sandboxes"** in which high risk activities may be tried.

Cultivate responsibility through **unjustified** respect.

Celebrate the wisdom gained from **mistakes** responsible and free testers (and children) make.

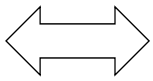
Analysis - 28

Analysis is a *synthesizing* process

Analysis - 29

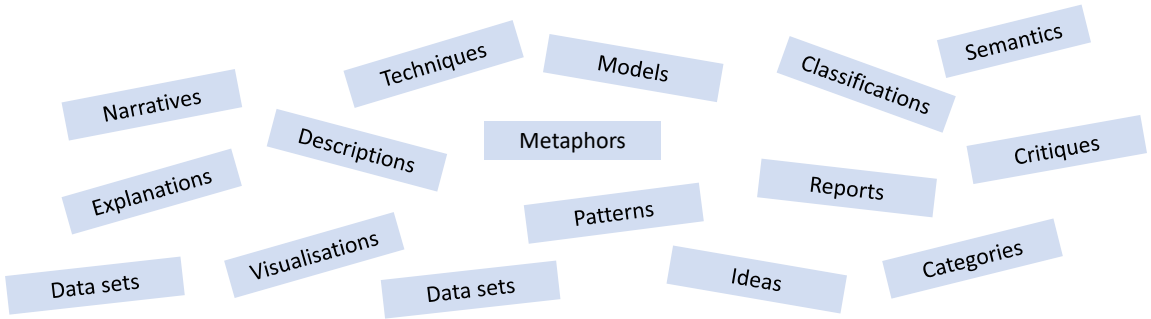
Analysis

To analyze something is to make sense of it; to figure out what it is and what it means.



Synthesis

Because we are trying to *make* sense of something, analysis also entails *synthesis*.



Analysis - 30

Sensemaking

Sensemaking in Organizations, by Karl Weick

1. Grounded in identity construction
2. Retrospective
3. Enactive of sensible environments
4. Social
5. Ongoing
6. Focused on and by extracted cues

7. Driven by plausibility rather than accuracy

Analysis - 31

Sensemaking for Testers

Sensemaking in Organizations, by Karl Weick

1. Applying our skillset and mindset as testers to make sense of…
2. What has happened…
3. As we create conditions, tools, and stories…
4. In collaboration with other people…
5. Continuously throughout the project…
6. To discover what we need to know about the product, and how we might learn about it…
7. Using effective, yet imperfect methods.

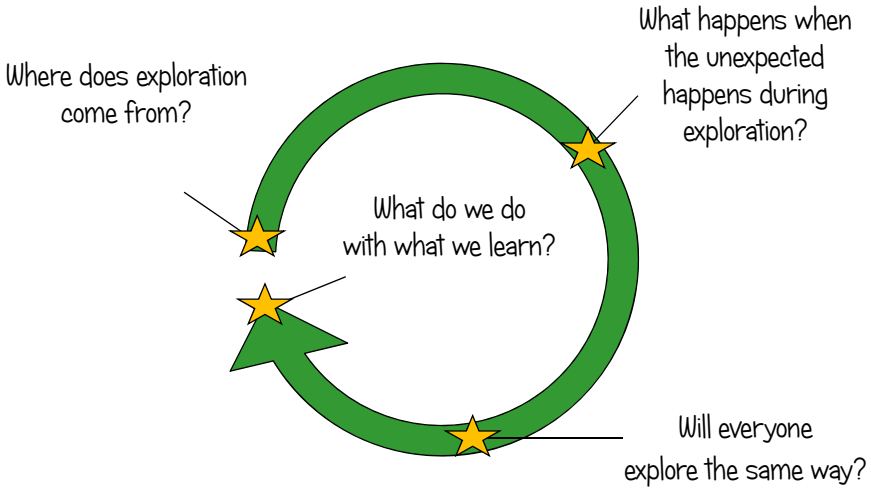
Analysis - 33

Analysis is an *exploratory* process

Analysis - 34

Questions About Exploration…

arrows and cycles



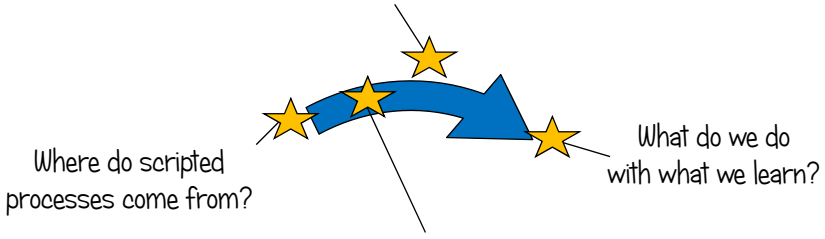
(value seeking)

Analysis - 35

Questions About Scripted Processes…

arrows and cycles

What happens when the unexpected happens during a scripted process?

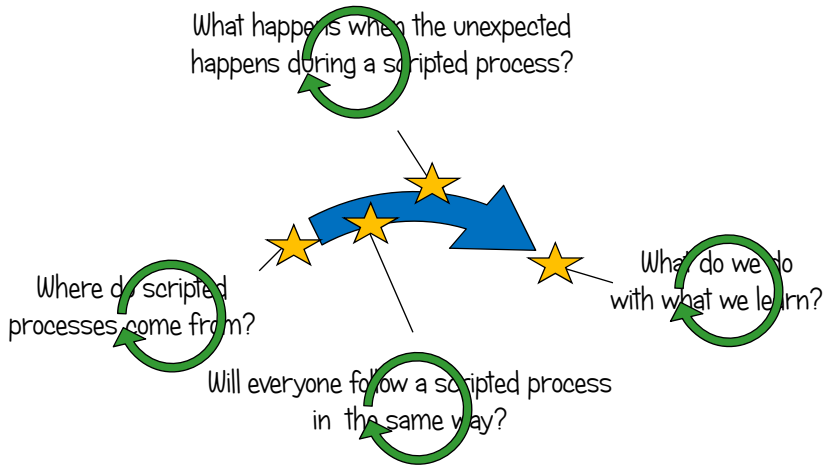


(task performing)

Analysis - 36

Answers About Scripts…

arrows and cycles

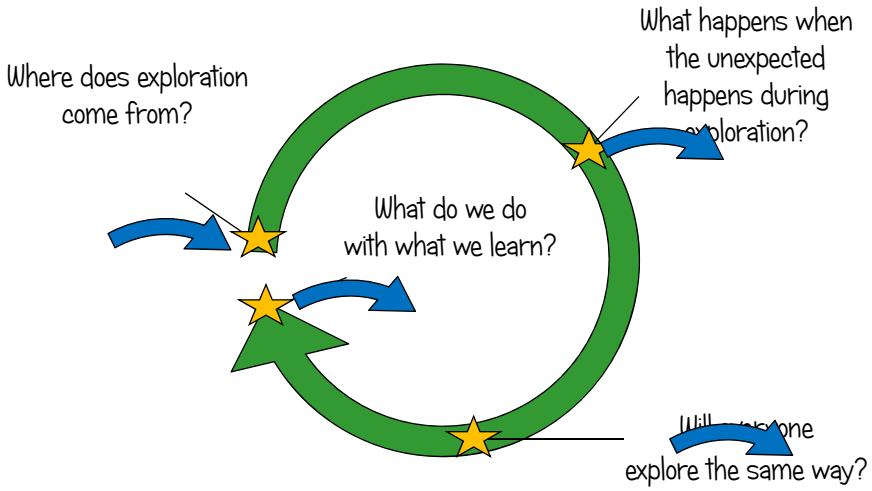


(task performing)

Analysis - 37

Questions About Exploration…

arrows and cycles

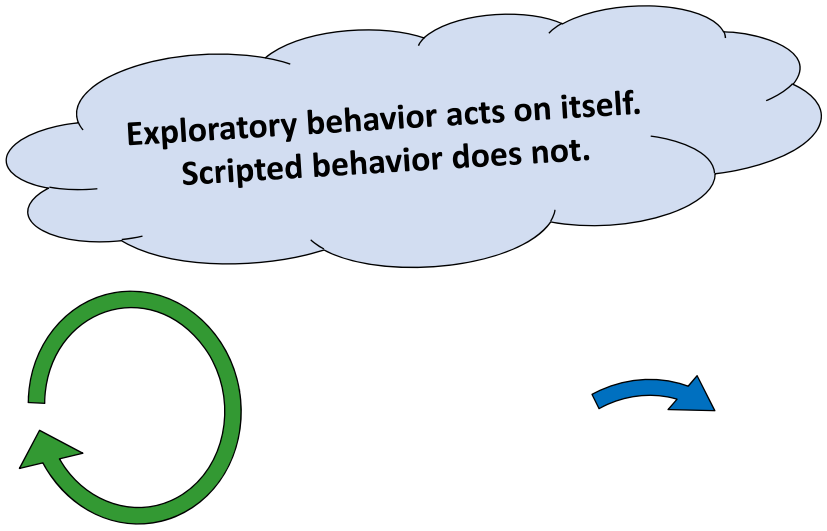


(value seeking)

Analysis - 38

Exploration is Not Just Action

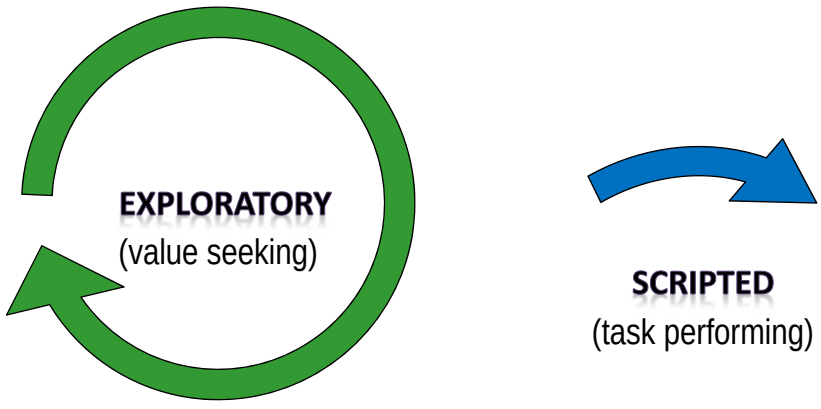
arrows and cycles



Analysis - 39

You can put them together!

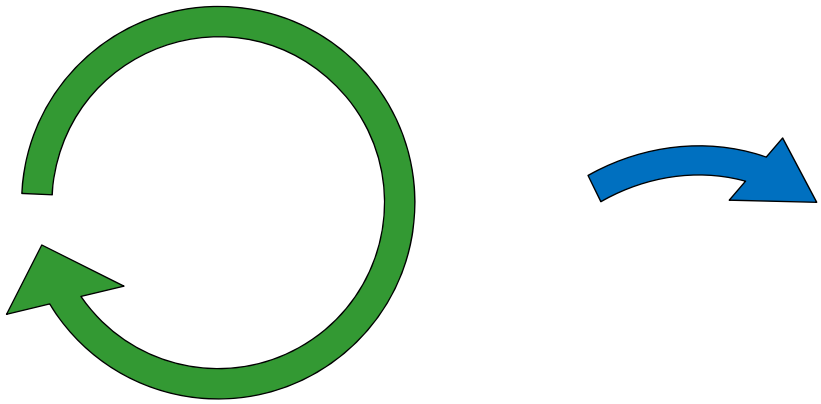
arrows and cycles



Analysis - 40

You can put them together!

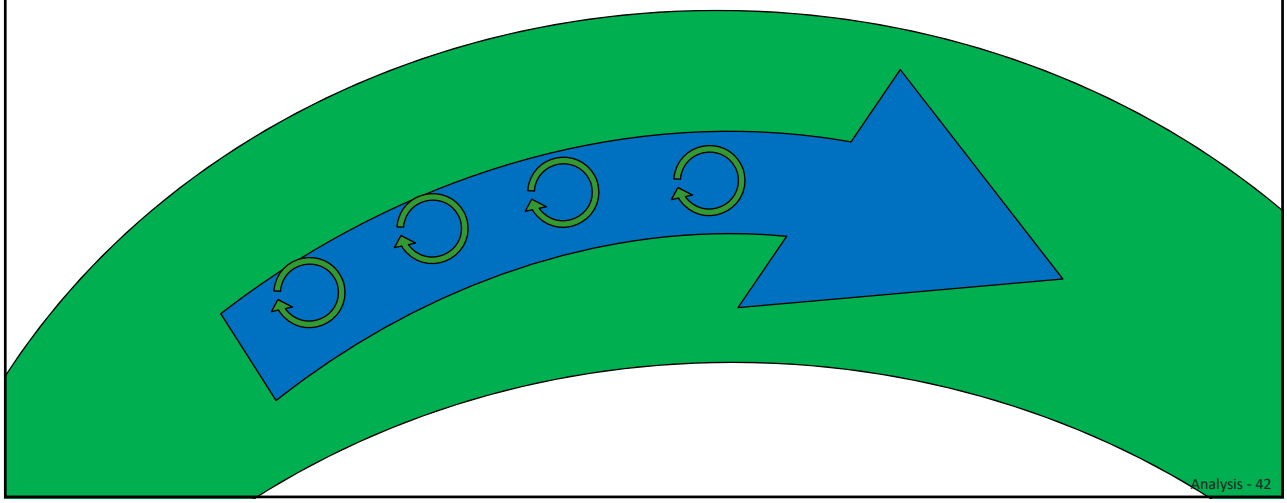
arrows and cycles



Analysis - 41

You can put them together!

arrows and cycles



Analysis - 42

Looking at something in different ways requires managing *distance*.

Analysis - 46

Distance Means Several Things

the first four definitions from O.E.D.

1. The condition of being at variance; discord, disagreement, dissension; dispute, debate. (original definition in English)
2. Difference, diversity
3. The fact or condition of being apart or far off in space; remoteness
4. The extent of space lying between any two objects; the space to be passed over before reaching a[illegible]bject.

Each of these first three definitions is part of what I am talking about today.

Analysis - 47

Critical Distance and Social Distance

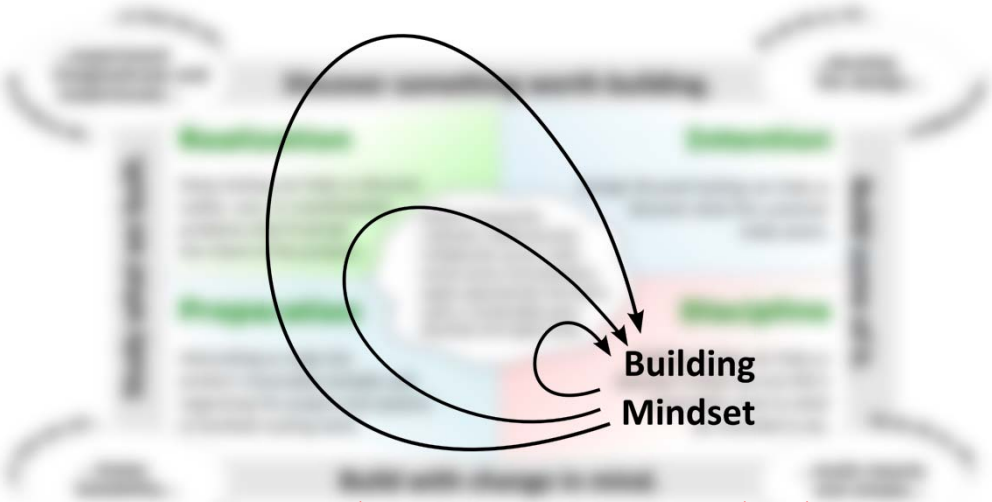
By **critical distance** I mean
A difference between two ways of thinking about some thing,
or *an absence of knowledge about some thing in favor of other things.*

By **social distance** I mean
any barrier to or absence of harmony and cooperation among people.

Cultivate critical distance.
Eliminate social distance.

Analysis - 48

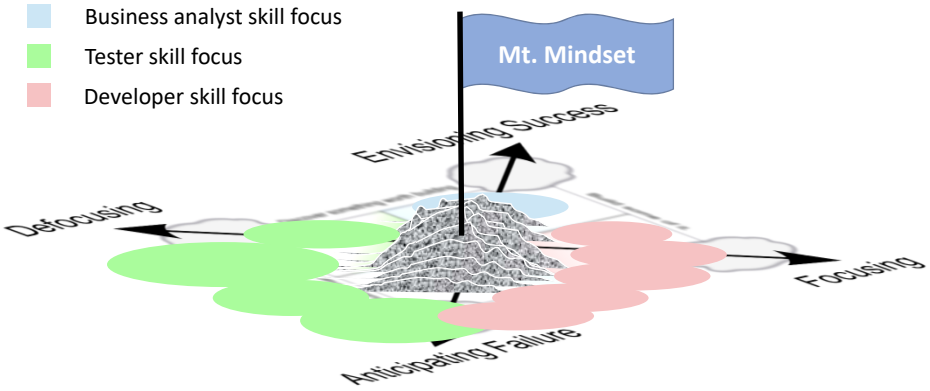
Deep testing requires critical distance.



Shallow testing doesn't need critical distance, but deeper or naturalistic long-form testing tends to **require** or **create** more distance from the builder's mindset.

Analysis - 49

Why roles? Because changing mindsets is HARD.

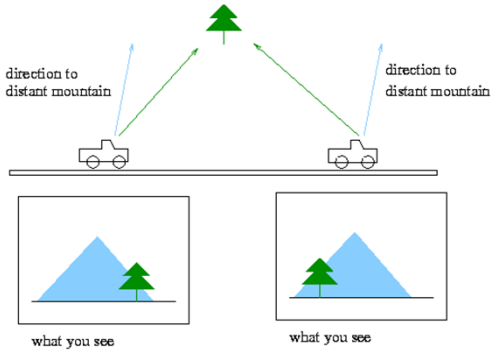


NOTE: We do NOT claim that different kinds of work *must* be done by different people, or that the people *must* have different titles.

We DO claim that having skilled people *focused* on testing is a powerful heuristic for addressing the mindset switching problem.

Analysis - 50

Critical Distance can be Helpful



- Critical distance can vastly **improve** the ability of one process to test, check, or analyze another process.
- It helps **prevent shared errors** by reducing shared sources of error.
- It helps **detect errors that do occur** by reducing shared blindness to errors.
- It helps **increase innovation**, by increasing the variety of our ways of working and increasing the probability of happy accidents.

Analysis - 51

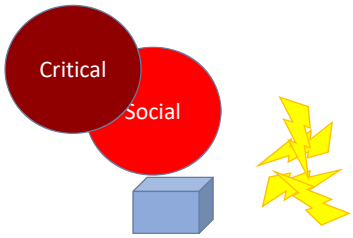
Social Distance can be Harmful

Social distance can **degrade** the ability of one process to test or check another process.

- It can **cause errors** and **hide errors** by reducing collaboration and sharing of critical information.
- It can **reduce motivation for error prevention or detection** by reducing empathy and sense of responsibility to outsiders, while increasing defensiveness.
- It can **reduce innovation** by because of less spontaneous exchange of contrasting ideas.
- It can **reduce organizational flexibility and resilience** by discouraging different parties from helping each other.

Analysis - 52

But… Critical Distance and Social Distance are Chained Together



Eliminating social distance also tends to get everyone thinking the same way.

Creating critical distance also tends to reduce the points of connection between people.

Separating them requires ongoing work.

Analysis - 54

In SOME ways, reducing critical distance can make testing better!

- A highly technical tester who understands the underlying code is automatically going to think **MORE** like the developer who wrote it.
- This reduces critical distance, which **HARMS** testing.
- But it also greatly increases his understanding of the product, which **HELPS** testing.
- My advice: **Use open box AND closed box testing!**

Although ignorance can be a good short-term tactic, it is a terrible longer-term strategy.

Analysis - 59

In SOME ways, increasing social distance can make testing better!

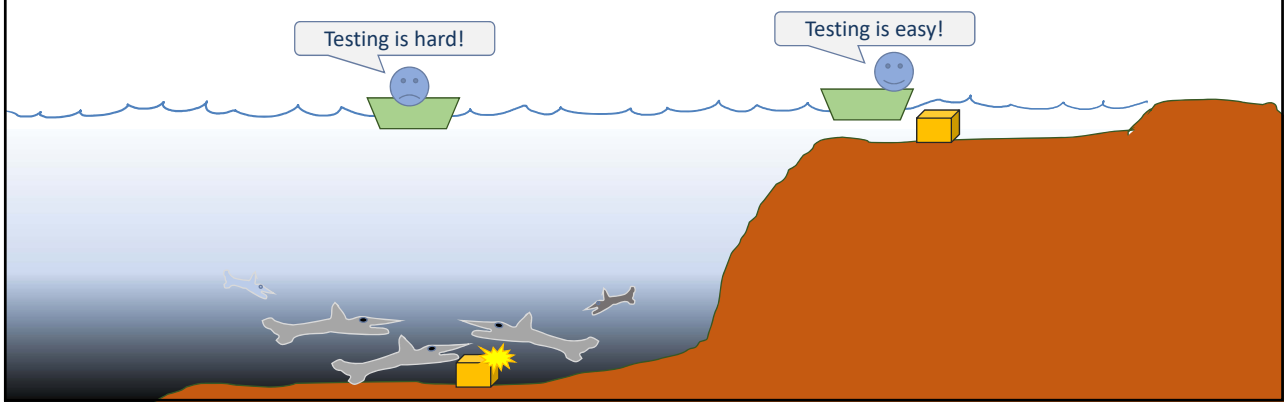
- A tester who is remote from and protected from development people does not experience social pressure from them.
- This increases social distance, which **HARMS** testing.
- But it also greatly reduces coercion and shared motivation to accept a low standard of quality, which **HELPS** testing.
- My advice: **Manage coercion without allowing social disconnection.**

This is why managers matter!

Analysis - 60

Not Everything Needs Deep Testing

- Simple things inside simple systems
- Things where problems don't matter
- Things that are easy to test

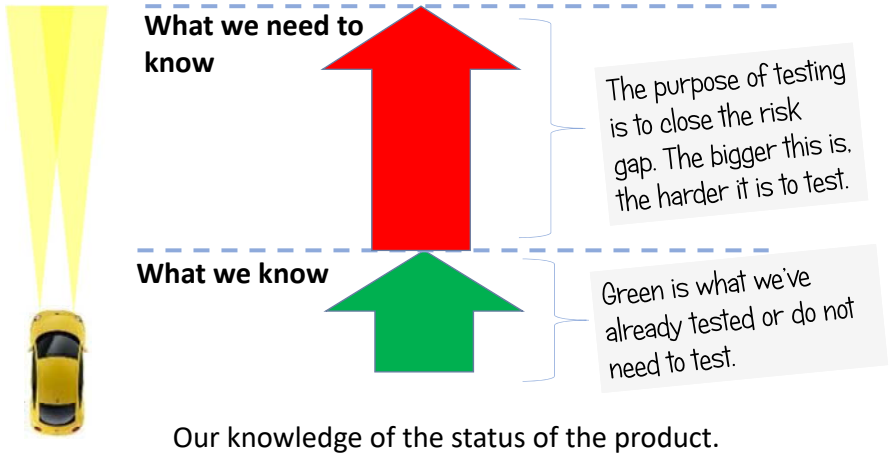


Key Elements for Analyzing Claims

- Look for the deep structure of the claim
- Bring supressed premises and buried assumptions to light
- Negotiate semantics and terms of communication in the trading zone (that is, where people gather to solve problems, but may not share a language)
  - This takes practice! The more you practice the more quickly and easily it happens.
  - Collective tacit knowledge helps! The more you practice as a group, the more quickly and easily it happens.
  - This takes some degree of managing
    - your heuristics and biases
    - your feelings about the process
    - your feelings about controversy

Analysis - 69

The Product Risk Knowledge Gap:



Our knowledge of the status of the product.

Analysis - 74

You can test one thing in one way…

Example: "Let's look for performance problems across the site."

Testing is the headlights of the project...

but where should you point them?

Analysis - 75

Or a different thing in a different way…

Example: "Let's look for incorrect shipping calculations during checkout."

Testing is the headlights of the project...

but where should you point them?

Analysis - 76

You can test shallowly…

Example: "I will see if I can log in."

Testing is the headlights of the project...

but where should you point them?

Analysis - 77

Or you can test more deeply…

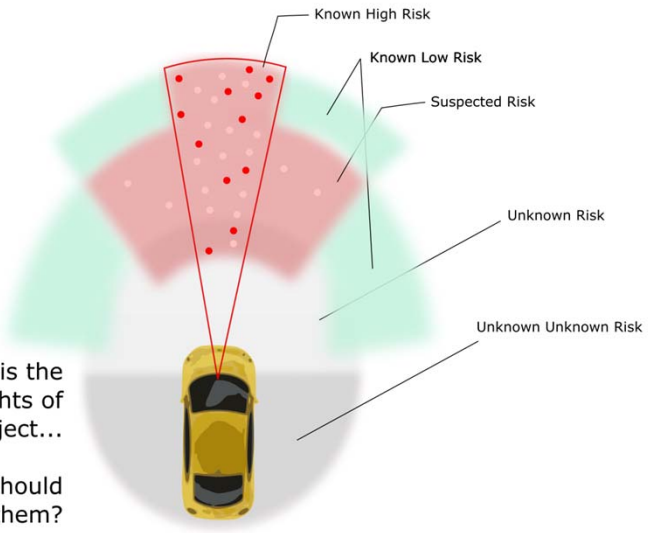
Example: "I logged in and tested every accessible screen for each of 16 user permission configurations and user account states."

Testing is the headlights of the project...

but where should you point them?

Analysis - 78

"Risk-Based Testing" means organizing testing around suspected risk.



Testing is the headlights of the project...

but where should you point them?

Analysis - 79

How do you know you are in the "green zone?"

You don't know.

There's no way to be certain.

We live with that uncertainty.

Analysis - 80

How do you ~~know~~ decide you are in the "green zone?"

- If you **know enough** about the product to judge it…
- …and you feel that its **known problems aren't bad**…
- …and you believe there are (probably) **no hidden problems** left (that matter)…
- …and **you can explain why** you know enough…
- …and **your story is compelling**.

Analysis - 81

Important Problems?
PROBABILITY

Definitely affects everyone continuously

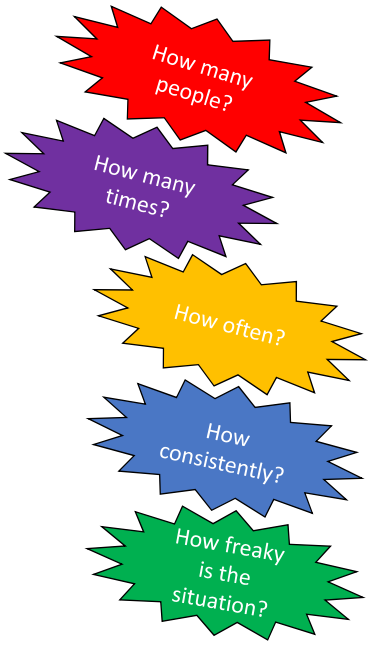
Affects most people occasionally

Affects some people occasionally

Affects few people occasionally

Affects few people rarely

Happens only under bizarre conditions



Analysis - 82

Important Problems?
SEVERITY

Corporate collapse event

Irreparable damage

Expensive repairable damage

Inexpensive repairable damage

Functionally limiting

Functionally limiting with workaround

Merely irritating

Liveable

Analysis - 83

Important Problems?
ETHICS and REPUTATION

Shocks public conscience

Violates law

Violates standard

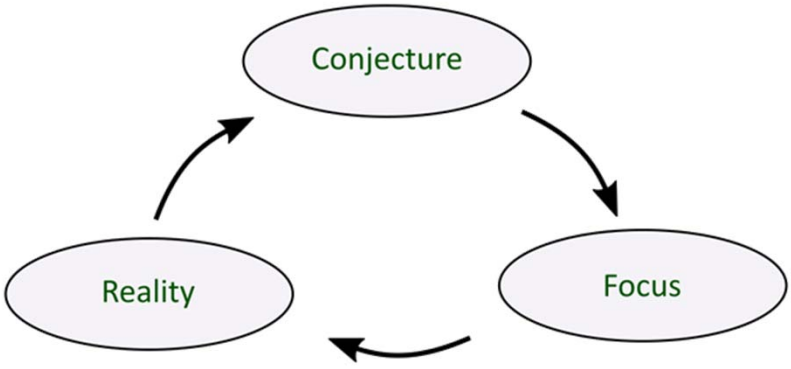
Diminishes or destroys trust

Makes people wonder “Why didn’t you find that?!”

Just unreasonably bad

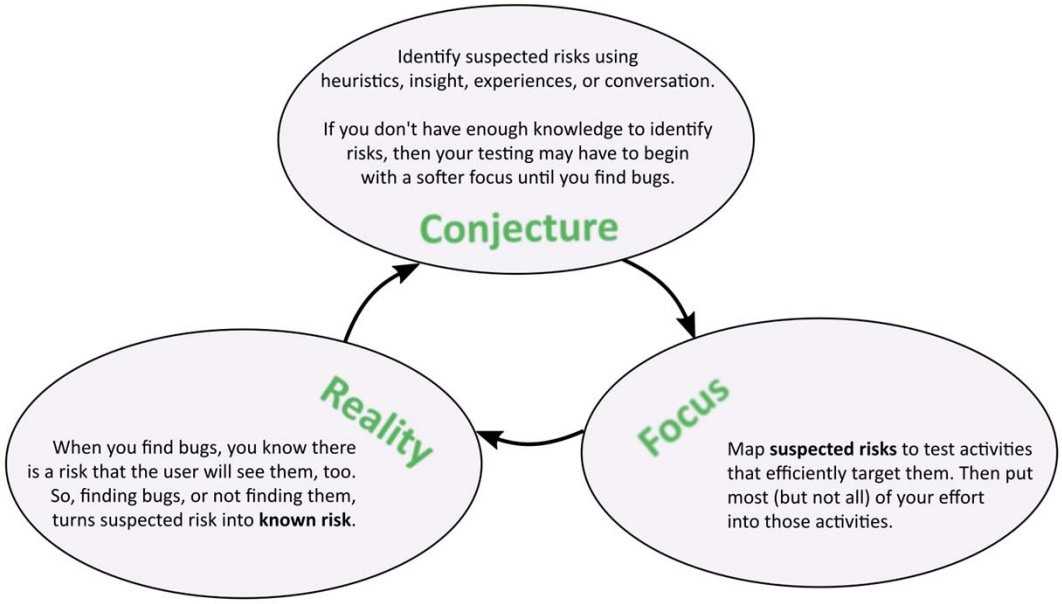
Analysis - 84

“Risk-Based Testing” means organizing testing around suspected risk.



Analysis - 85

"Risk-Based Testing" means organizing testing around suspected risk.



Analysis - 86

Unknown Product Risk

"Known unknown risk" is when you are **conscious of not knowing** about the risk; you don't know what to say about the risk other than "this thing might not work."

Example: "I don't know if performance is an issue with the online ticket-selling system. I guess I will just have to look at it."

"Unknown unknown risk" is when **you think you know** where all the high risk is, **but are mistaken**. This leads to overconfidence.

Example: "We run our system on big servers in the cloud. No need for performance testing!"

Analysis - 87

Suspected Product Risk

- A risk that you are worried **might** exist regarding the product.
- A risk that you know exists, but **that you don't yet know enough about.**
- A feeling of concern about some aspect of the product based on generalities or indicators but **without specific, compelling evidence that there is any actual bug in that area.**

Example: "I'm worried about poor performance when people use our ticket-buying system for very popular rock concerts. Peak loads in such cases are astronomical."

Analysis - 88

Known Product Risk

A product risk that you can **describe and explain** with **sufficient evidence and specificity** that any reasonable person would **accept that it exists**.

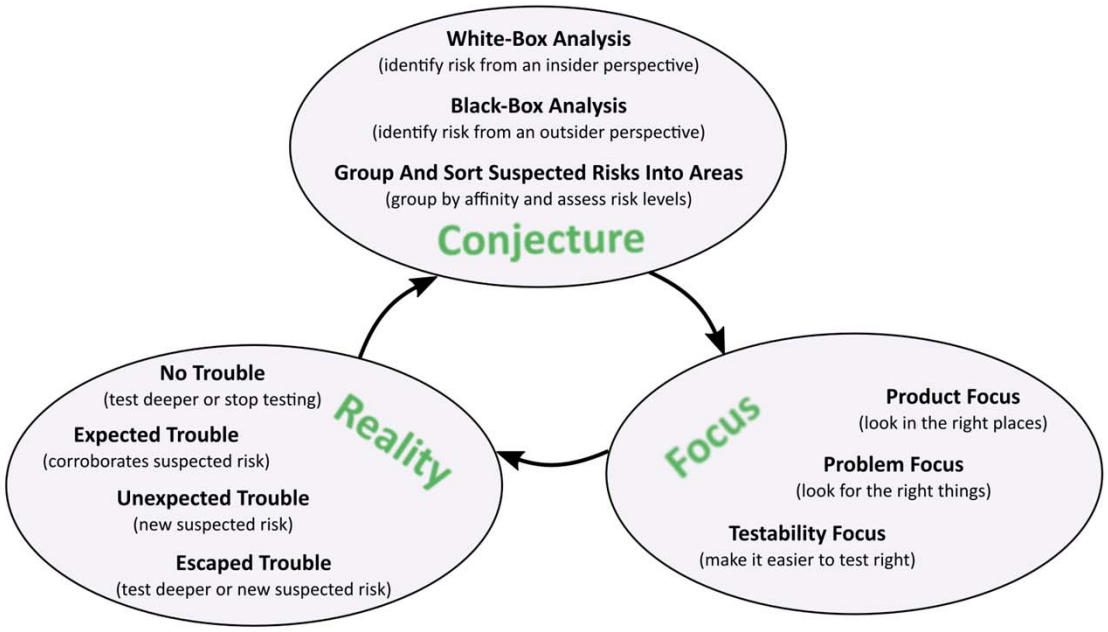
Usually, known high risk is based on known open bugs; known low risk is based on good testing that didn't find bugs.

Most testing is motivated by the intention to go from speculation about suspected risk to evidence of actual risk.

Example: "We tested performance under realistic conditions and we now know that it cannot reliably handle more than 150 users per minute. Our servers will crash when Taylor Swift comes to town."

Analysis - 89

"Risk-Based Testing" means organizing testing around suspected risk.



Analysis - 90

Actual Product Risk

Almost all testing results in the assessment of actual risk.

- A product risk that you can **describe and explain** with **sufficient evidence and specificity** that any reasonable person would **accept that it exists**.
- The presence of any actual bug implies the presence of actual risk.
- We almost never quantify risk in terms of hard probabilities or cold cash because there are usually **too many uncertainties** to model risk in that way.
- In that case, we can use our mental models and our feelings as a heuristic (fast and frugal; letting **plausibility** stand in for accuracy)
- Meanwhile, treat any mathematical or seemingly "complete" and "rational" risk analyses with **great skepticism.**

Analysis - 91

Typical Trouble with Risk Analysis

- Testers afraid of anything that has the word "analysis" in it, and looking for easy answers.
- Having no systematic method of risk analysis.
- Conceiving of big huge categories, or tiny specific bugs, but nothing in between.
- Instead of product risks, focusing instead on *project* risks.

See "Risk Analysis Heuristics (for Digital Products)", in the class materials.

How to Analyze Product Risk

- When you don't know much
  - You will just be able to get a vague sense of risk.
  - Move toward the product… learn everything you can about the product and its context.
  - Talk to people who know about it.
  - Does it have a history? Learn about it.
  - Perform *survey testing* (testing with an emphasis on learning)
  - Use *circumstantial* and *general systems heuristics*

Some General Systems Analysis Heuristics

- **Cardinality:** Can there be 0, 1, or more than one object?
- **Boundaries:** Is there a limit? More than one? Are different limits consistent?
- **Extrapolation:** If we can go THIS far, can we go FARTHER?
- **Interpolation:** If two things exist in different places, does something exist between them?
- **Intersections:** Do components collide? Can one contaminate another?
- **Surface Integrity:** Does behavior change correctly as input changes in any given dimension?
- **Symmetry/Asymmetry:** If a behavior exists for A, does a corresponding behavior exist for B?
- **Pattern Completion:** Is a pattern apparent that has not yet been completed, or is obscured (all customary parts of a shape)?
- **Negation:** Whatever is there might disappear or reverse.

Analysis - 70

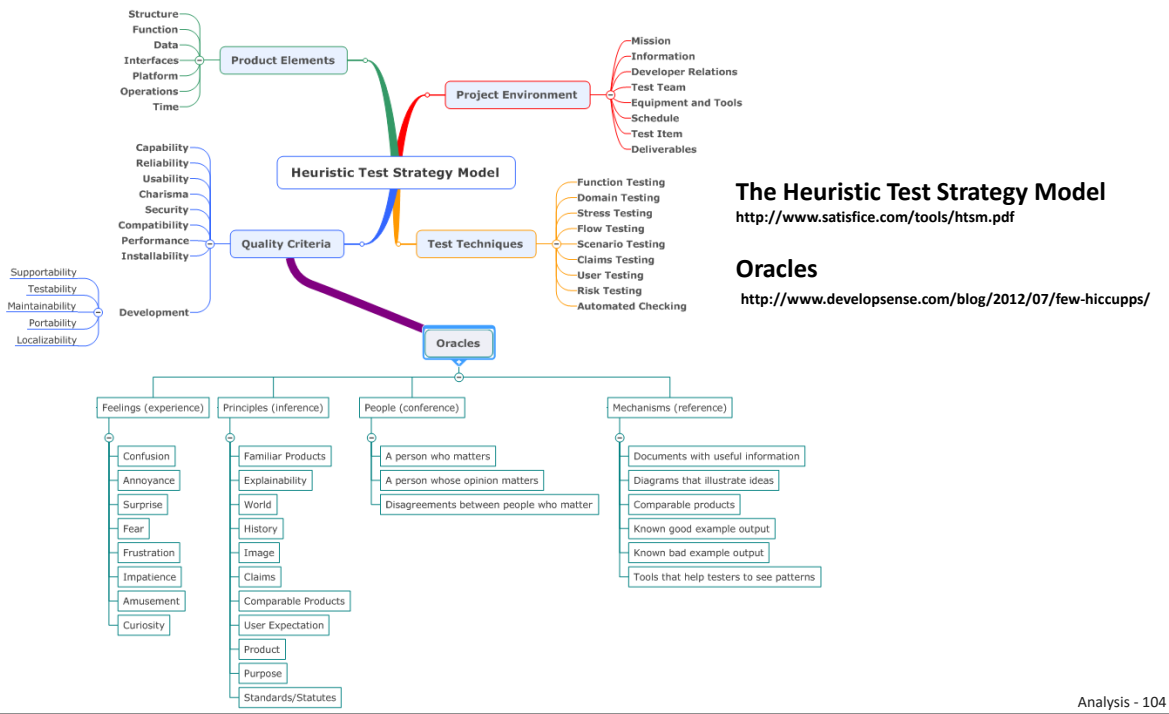
Some General Systems Analysis Heuristics

- A condition in the world that interacts with your product may happen **rarely** or **frequently**.
- A problem may be **detectable** or **undetectable** at the time it first occurs or begins to occur (relative to our means of detection).
- A problem may occur in a **result**, **process**, **component**, or **environment**.
- In the realm of technology and society, problems fall into diverse categories: **usability, performance, concurrency, legality, correctness, compatibility, etc.**
- A problem may impact a **user**, a **bystander**, or the **business** that produced the product.
- A problem may cause **inconvenience**, **annoyance**, or **harm**.
- …

Analysis - 71

How to Analyze Product Risk

- When you know **a lot**
  - Use the *Quality Criteria Categories* list from the HTSM to systematically consider different kinds of bugs.
  - Brainstorm risks with the team.
  - List the specific ways the product may fail based on knowledge of the code, knowledge of functionality, knowledge of the contexts in which people will use the product.
  - List the kinds of problems that users care most about.
  - Consider the problems that have happened before.
  - Use the *four-part risk* story to fill out your analysis.



Analyzing Risk

- Consider creating and maintaining a *product* risk list and a *project* risk list, especially if no one else on the team is doing it.
- Brainstorm a list of risks, and rank them in approximate order of significance.
- Then compare this list to coverage and quality criteria areas in the Heuristic Test Strategy Model, or your own taxonomies.
- Identify tasks associated with investigating and managing risks.
- **Make risks and tasks public**, and advertise when and where you need help.
- Do some testing *not* focused on specific risks, in order to discover unrecognized risks.

See Rapid Software Testing Appendices for several examples, including "Install Risk Catalog", "Risk-Based Test Plan (OWL)", "Risk-Based Test Plan #2".

Analysis - 106

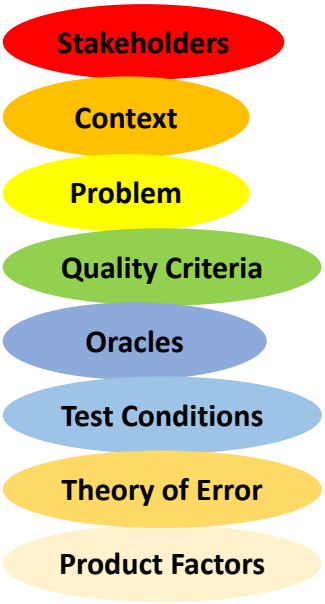
The Product Risk Story

"Some person(s) will experience a problem with respect to something desirable that can be detected in some set of conditions because of a vulnerability in the system."

Analysis - 107

Risk Story Elements

- Some PERSON(S)
  - user, customer, developer, tester, businessperson, bystander…
  - (a group, a business, a community, society at large…)
- will EXPERIENCE
  - be affected, in the context of an event or situation, at least once by …
- a PROBLEM
  - that leads to bad feelings (annoyance, frustration, confusion), loss, harm, or diminished value…
- with respect to SOMETHING DESIRABLE
  - like capability, reliability, performance…
- that CAN BE DETECTED
  - by a feeling, principle, tool, comparison to a document or picture…
- in SOME SET OF CONDITIONS
  - perhaps always, perhaps only sometimes,…
- because of a VULNERABILITY
  - a bug, a missing feature, an inconsistency…
- in the SYSTEM
  - some result, process, component, feature, environment…

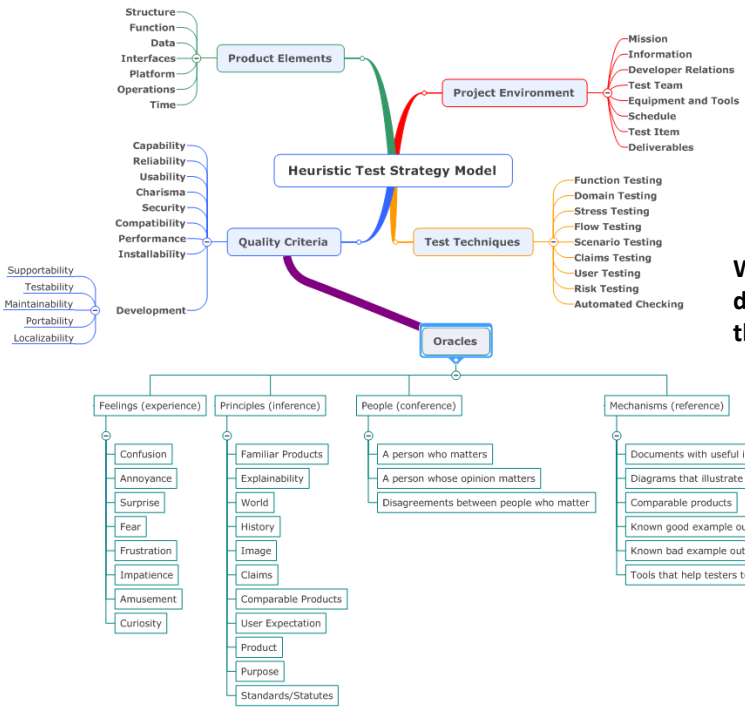


Analysis - 108

An Example Table of Risk Elements

Where is the problem?	Can we see it?	When would we see it?	How often does it happen?	How regularly?	What is affected?	Quality criterion?	Who is affected?	How do they feel?
Structure	Obvious	Immediately	Frequently	Consistent	Structure	Capability	Society	Impatient
Function	Obscure	Later	Rarely	Intermittent	Function	Reliability	Public	Confused
Data	Invisible	Never			Data	Usability	Consumer	Annoyed
Interfaces	Can others?				Interfaces	Charisma	Network Admin	Surprised
Platform					Upstream Platform	Security	External Developer	Disappointed
Operations					Downstream Platform	Scalability	Internal Developer	Angry
Time					Operations	Coexistence	Tester	Afraid
					Time	Inter-operability	Tech writer	Suspicious
						Performance	Malicious user	Uninformed
						Installability	Incompetent	Paralyzed
						Configurability	Bystander	
						Supportability		
						Testability		
						Maintainability		
						Portability		
						Localizability		

10-Risk - 7



When you can internalize and describe all this like a professional, the amateurs don't hassle you.

Analysis - 117

Activities in Analysis

- Identifying goals
- Modeling the analysis space
- Collecting observations and data
- Evaluating the quality of the data
- Identify factors that influence evaluations
- Identify factors that influence each other
- Classifying and coding (in the qualitative research sense) data
- Comparing and contrasting
- Seeking patterns and connections
- Identifying consistencies and inconsistencies
- Accounting for inconsistencies
- Developing explanations
- Managing assumptions and validity

Analysis - 120

Managing Your Data

- Gathering it
- Arranging it
- Refining it
- Representing it
- Organizing it
- Storing it
- Retrieving it
- Processing it
- Presenting it

Analysis - 121

Analysis Meta-Methods

- Developing descriptions
  - narrating, writing
  - seeing what's being said and noticing what's being left out
- Visualizing the data and relationships
  - drawing, sketching
  - arranging maps, tables, charts, schematics…
  - focusing; defocusing; refocusing
  - seeing what's there and noticing what's missing
- Managing validity
- Reflecting
- Revising

Analysis - 122

Analysis Methods: Examples of Alternation

- Forward Analysis ⇔ Backward Analysis
  - Open Coding vs. Closed Coding
  - Forward Chaining vs. Backward Chaining Reasoning
- Incubating ⇔ Progressing
- Focusing ⇔ Defocusing
- Touring ⇔ Sensemaking
- Lone ⇔ Social
- Talking ⇔ Doing
- Planning ⇔ Doing
- Suppression ⇔ Exaggeration
- Naturalistic ⇔ Artificial ⇔ Pathological
- Easy ⇔ Hard

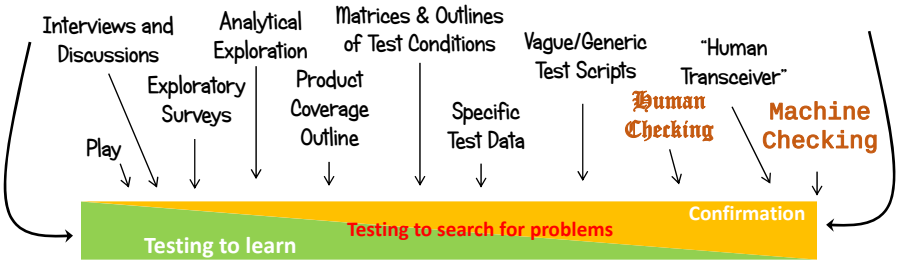
Analysis - 123

Analysis Method: Scaling and Scoping

- Examine an object, phenomenon, or idea at different scales.
- Look at the whole.
- "Reduce it" to its elements.
- Reduce its elements to *their* elements.
- Examine the relationship of the object to things around it.
  - What are those things?
  - What are the relationships?
- What is missing or invisible?
- What changes when you change the scale at which you're looking?
- Observe how things change over time.

Analysis - 124

The Testing Formality Continuum



Loops of testing start with informal, exploratory work. If you want to do excellent formal testing (like automated checking), it must begin with excellent informal work. Formality tends to intensify over time.

Analysis - 125

Problems in Analysis

Any analysis is vulnerable to problems by attending to some factors and ignoring or ruling out others in…

- Conceptual frameworks
- Research questions
- Sampling
- Examination of instances
- Instrumentation
- The nature of your data
- Data quality

Analysis - 126

Readings

- *Stanford Encyclopedia of Philosophy*, "Analysis"
  - https://plato.stanford.edu/entries/analysis/
- Austin, Robert. *Measuring and Managing Performance in Organizations*. Dorset House, 1996.
- Collins, Harry. *Tacit and Explicit Knowledge*. University Of Chicago Press, 2010.
- Collins, Harry M., and Martin Kusch. *The Shape of Actions: What Humans and Machines Can Do.* The MIT Press, 1999.
- Collins, H. M. Rethinking Expertise. Chicago: University of Chicago Press, 2007.
- Dorner, Dietrich. *The Logic Of Failure: Recognizing And Avoiding Error In Complex Situations.* 1st ed. Basic Books, 1997.
- Elkins, James. How to Use Your Eyes.
- Federman, Mark, and Derrick deKerchkhove. *McLuhan for Managers: New Tools for New Thinking*. Viking Canada, n.d.
- Gigerenzer, Gerd. *Calculated Risks: How to Know When Numbers Deceive You.* 1st ed. Simon & Schuster, 2003.

Readings

- Hofstader, Douglas R. *Surfaces and Essences: Analogy as the Fuel and Fire of Thinking.* New York: Basic Books, 2013.
- Kahneman, Daniel. *Thinking, Fast and Slow*. Penguin, 2011.
- Kirk, Jerome, and Marc L Miller. *Reliability and Validity in Qualitative Research*. Beverly Hills: Sage Publications, 1986.
- Koen, Billy Vaughn. *Discussion of the Method: Conducting the Engineer's Approach to Problem Solving*. Oxford University Press, USA, 2003.
- Maxwell, Joseph Alex. *Qualitative Research Design: An Interactive Approach*. Thousand Oaks, Calif.: Sage Publications, 2005.
- Meadows, Donella H. *Thinking in Systems: A Primer*. Chelsea Green Publishing, 2008.
- Petroski, Henry. *To Engineer Is Human: The Role of Failure in Successful Design.* Vintage, 1992.
- Polya, George. *How To Solve It: A New Aspect of Mathematical Method.* Ishi Press, 2009.

Readings

- Polya, George. *How To Solve It: A New Aspect of Mathematical Method.* Ishi Press, 2009.
- Shadish, William R., Thomas D. Cook, and Donald T. Campbell. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference.* 2nd ed. Wadsworth Publishing, 2001.
- Taleb, Nassim Nicholas. *The Black Swan: Second Edition: The Impact of the Highly Improbable: With a New Section: "On Robustness and Fragility."* 2nd ed. Random House Trade Paperbacks, 2010.
- Tufte, Edward R. *Envisioning Information.* Graphics Pr, 1990.
- Tufte, Edward R. *The Visual Display of Quantitative Information.* 2nd ed. Graphics Pr, 2001.
- Weick, Karl E. *Sensemaking in Organizations*. Sage Publications, Inc, 1995.
- Weinberg, Gerald M. *An Introduction to General Systems Thinking*. 25 Aniversary Edition, Dorset House, 2001.

A Word from Our Sponsor (Me)

- Rapid Software Testing is a course, a mind-set, and a skill set about how to do excellent software testing in a way that is very fast, inexpensive, credible, and accountable. I co-author RST with James Bach.
- I teach RST in classes for testers, developers, managers, business analysts, documenters, DevOps people, tech support…
- I also offer advice and consulting on testing and development to those people, and also to managers and executives.

http://www.developsense.com