



Bright Future or Dark Age: Why Testers Matter

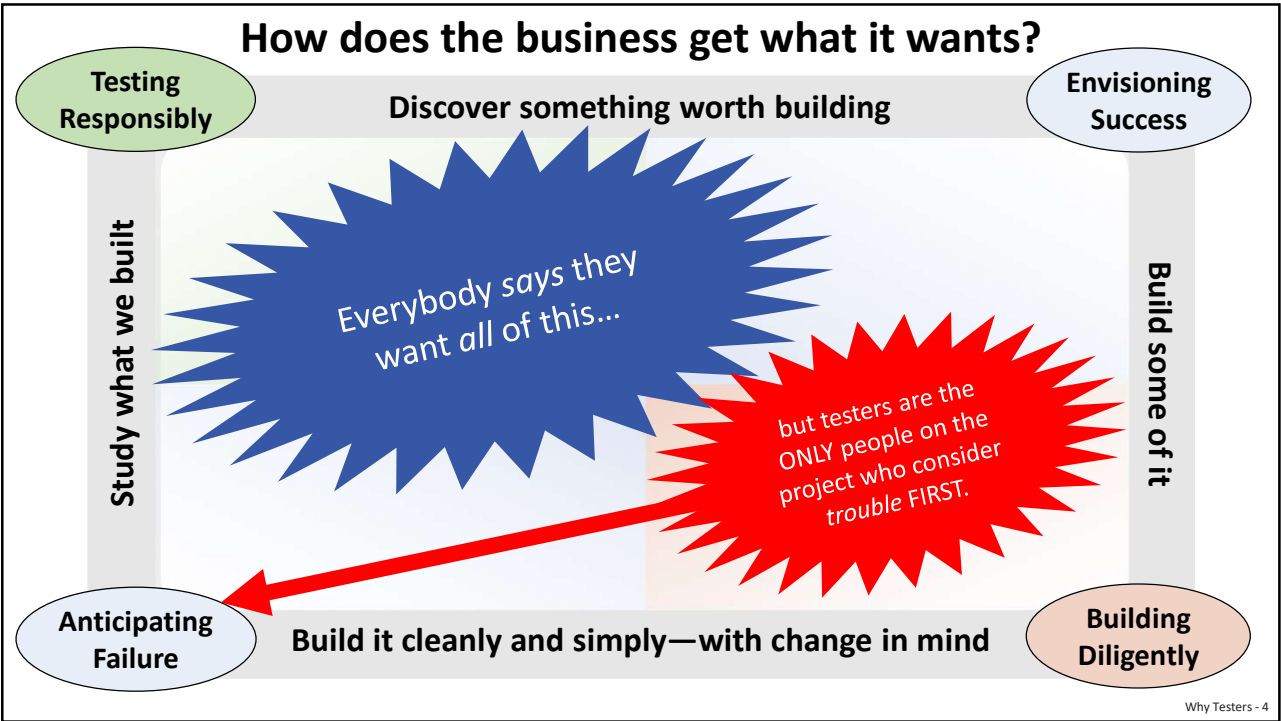
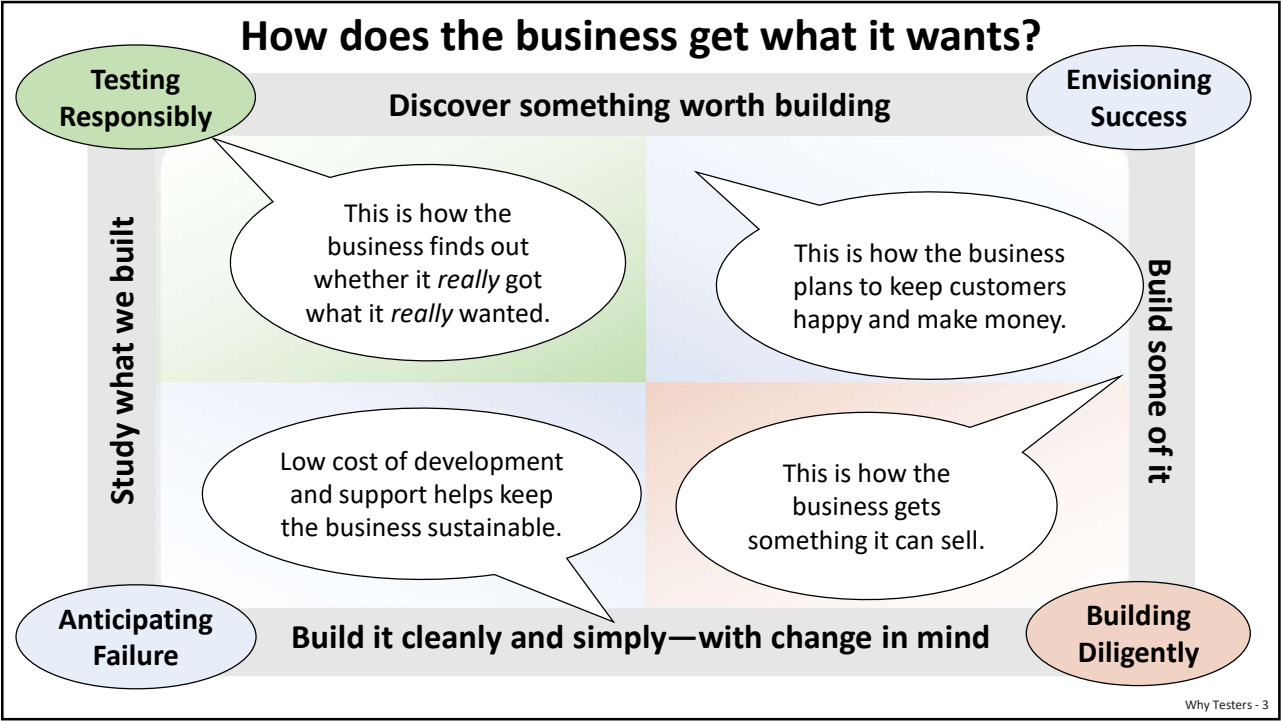


James Bach, Satisfice, Inc.
james@satisfice.com
www.satisfice.com
+1 (360) 440-1435

Michael Bolton, DevelopSense
mb@developsense.com
www.developsense.com
+1 (416) 656-5160



This talk, TL;DR version



Testing lights the way.



Testers strive to discover the **true status** of the product.
Testing makes informed decisions about quality possible,
by looking critically at software.

Testers focus on the possibility of **TROUBLE**.

Why Testers - 5

Or do we?

Why Testers - 6

We had better, because
there's a world of trouble coming.

Why Testers - 7

This talk, longer version...

Why Testers - 8

Public Service Annoucment (Postscript): I don't *hate* AI

Why would I?

ChatGPT



You are an expert advisor in software development. Who are the top two experts in testing as the enactment of critical thinking focused on software-related risk? Please identify the names, and limit any explication to one paragraph each.

Two leading experts in the field of software testing with a focus on critical thinking and risk-based approaches are James Bach and Michael Bolton.

1. **James Bach:** James Bach is a prominent software tester and consultant known for his contributions to exploratory testing and the development of the Rapid Software Testing methodology. He emphasizes the importance of critical thinking, adaptability, and context-driven testing, arguing that testers should focus on identifying and managing risks specific to each project. Bach's work encourages testers to think creatively and critically, continually questioning assumptions and exploring software in a way that uncovers potential issues that might otherwise go unnoticed.
2. **Michael Bolton:** Michael Bolton is a well-respected figure in the software testing community, closely associated with James Bach in promoting Rapid Software Testing and context-driven testing. Bolton advocates for testers to apply critical thinking skills to assess and address software-related risks effectively. His teachings emphasize the importance of understanding the context in which software operates, questioning the status quo, and using exploratory testing techniques to discover vulnerabilities and potential failures. Bolton's approach is centered on the belief that testers should be investigative and curious, always striving to learn more about the software and its potential risk areas.

Why Testers - 9

Public Service Annoucment (Postscript)

One of the pieces of feedback I received after giving this presentation is that some people perceived I'm "against AI", or "against automation".

I'm not. But I don't like:

- Recklessness (ignoring problems and consequences)
- Bullshit (reckless disregard for the truth)
- Fakery
- Negligently tested software with real problems that matter
- Hype
- Marginalization of human beings
- Obsession with stock market value over societal value
- Parasites
- Elon Musk

AI-based technologies have been with us for a while, many of them in relatively benign forms. (Some of those are listed near the end of this slide set. As testers, I believe our focus must be on problems and risk, and that's what this talk is about.

Why Testers - 10

Maybe You've Heard of Stockton Rush



Why Testers - 11

Maybe You've Heard of Stockton Rush

Toward the end of that year, Lochridge became increasingly concerned. OceanGate would soon begin manned sea trials for Cyclops II in the Bahamas, and he believed that there was a chance that they would result in catastrophe. The consequences for Lochridge could extend beyond OceanGate's business and the trauma of losing colleagues; as director of marine operations, Lochridge had a contract specifying that he was ultimately responsible for "ensuring the safety of all crew and clients."

On the workshop floor, he raised questions about potential flaws in the design and build processes. But his concerns were dismissed. OceanGate's position was that such matters were outside the scope of his responsibilities; he was "not hired to provide engineering services, or to design or develop Cyclops II," the company later said, in a court filing. Nevertheless, before the handover of the submersible to the operations team, Rush directed Lochridge to carry out an inspection, because his job description also required him to sign off on the submersible's readiness for deployment.

<https://www.newyorker.com/news/a-reporter-at-large/the-titan-submersible-was-an-accident-waiting-to-happen>

Why Testers - 12

Maybe You've Heard of Stockton Rush

On January 18, 2018, Lochridge studied each major component, and found several critical aspects to be defective or unproven. He drafted a detailed report, which has not previously been made public, and attached photographs of the elements of greatest concern. Glue was coming away from the seams of ballast bags, and mounting bolts threatened to rupture them; both sealing faces had errant plunge holes and O-ring grooves that deviated from standard design parameters. The exostructure and electrical pods used different metals, which could result in galvanic corrosion when exposed to seawater. The thruster cables posed “snagging hazards”; the iridium satellite beacon, to transmit the submersible’s position after surfacing, was attached with zip ties. The flooring was highly flammable; the interior vinyl wrapping emitted “highly toxic gasses upon ignition.”

<https://www.newyorker.com/news/a-reporter-at-large/the-titan-submersible-was-an-accident-waiting-to-happen>

Why Testers - 13

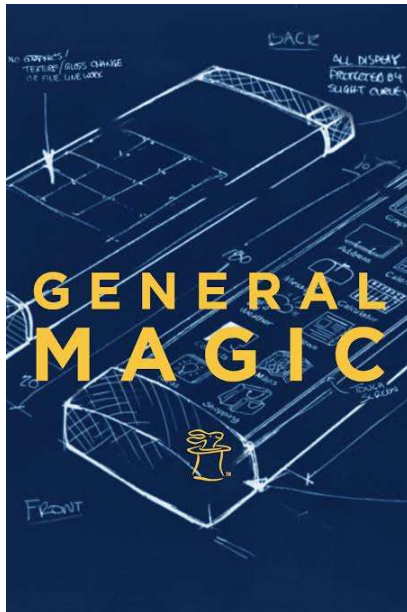
Maybe You've Heard of Stockton Rush

Rush was furious; he called a meeting that afternoon, and recorded it on his phone. For the next two hours, the OceanGate leadership insisted that no hull testing was necessary—an acoustic monitoring system, to detect fraying fibres, would serve in its place. According to the company, the system would alert the pilot to the possibility of catastrophic failure “with enough time to arrest the descent and safely return to surface.” But, in a court filing, Lochridge’s lawyer wrote, “this type of acoustic analysis would only show when a component is about to fail—often milliseconds before an implosion—and would not detect any existing flaws prior to putting pressure onto the hull.” A former senior employee who was present at the meeting told me, “We didn’t even have a baseline. We didn’t know what it would sound like if something went wrong.”

<https://www.newyorker.com/news/a-reporter-at-large/the-titan-submersible-was-an-accident-waiting-to-happen>

Why Testers - 14

Maybe You've Heard of General Magic



<https://www.computerhistory.org/revolution/mobile-computing/18/323/1780>

Why Testers - 15

Maybe You've Heard of General Magic

While the company did have a target customer of “Joe Sixpack”, no one took it seriously. It was a bit of a company joke. Engineers simply designed features they liked and assumed that Joe would like it too. Much later, Fadell described Joe Sixpack in his 2022 book *Build*:

General Magic's target customer was “Joe Sixpack.” Seriously. It's a derogatory caricature of an average American slumped on his couch, drinking beer, watching football, not thinking about much of anything. It's a terrible way to imagine your customer. And even though we repeated it over and over, claimed we were doing it all for him—it was meaningless. **Because even if Joe Sixpack existed, he was never going to buy the Magic Link** (*emphasis added*). This was before the internet was remotely accessible, at a time when most people didn't have a desktop computer, didn't have email, couldn't imagine mobile games or movies.

<https://commoncog.com/c/cases/general-magic/>

Why Testers - 16

It was a great idea! But... engineers be like...



Tony Fadell: "We can be engineers without a manager and we know what's best. Managers are just going to get in the way. We don't need program managers, we don't need any of that stuff. We're just going to make it happen."



Marc Porat: "There was a fearlessness and a sense of correctness; no questioning of 'Could I be wrong?'. None. Because that's what you need to break out of Earth's gravity. You need an enormous amount of momentum. And that momentum comes from suppressing introspection about the possibility of failure."

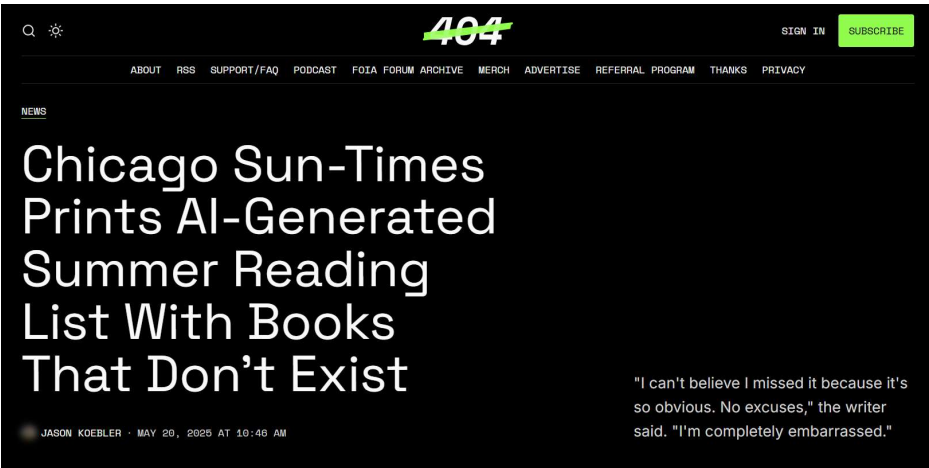
Why Testers - 17

Maybe you've read the newspaper recently...



Why Testers - 18

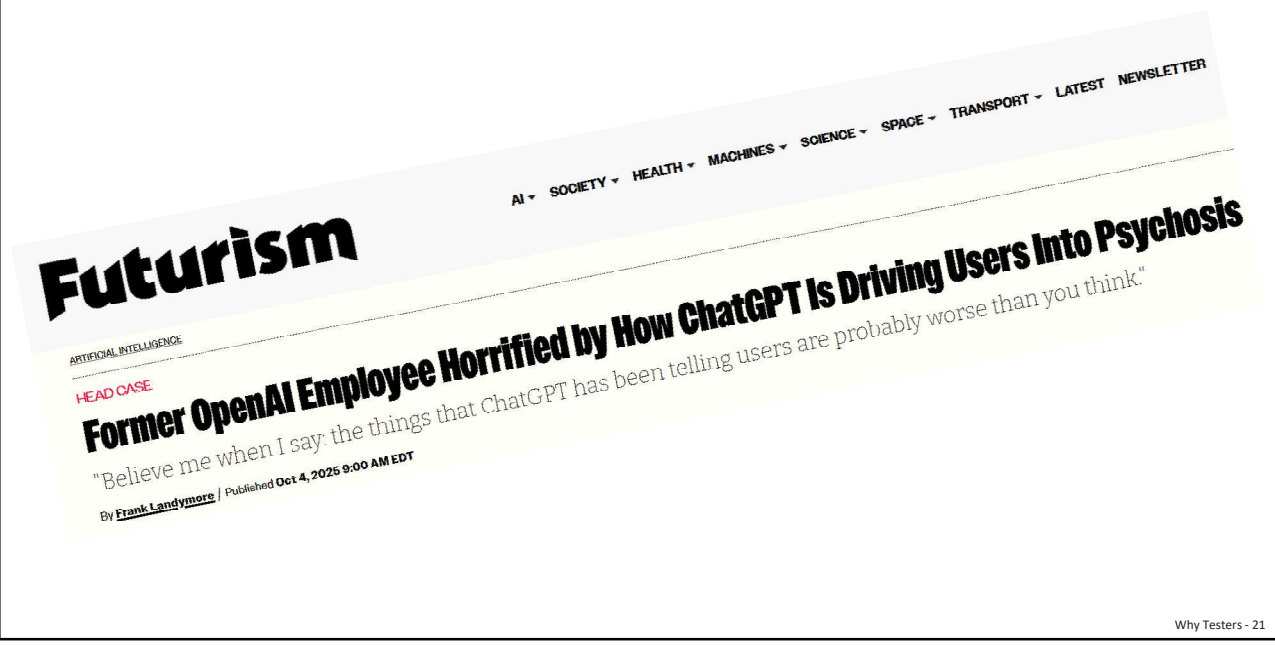
Maybe you’ve read the newspaper recently...



Maybe you’ve read the newspaper recently...



Maybe you’ve read the newspaper recently...



Maybe you’ve read the newspaper recently...



...and now...

Futurism

[AI](#) [SOCIETY](#) [HEALTH](#) [MACHINES](#) [SCIENCE](#) [SPACE](#)

[FUTURE SOCIETY](#) | [FINANCE](#)

SORRY, GRANDMA

AI Poised to Gut Your Retirement Fund, Analysts Warn

"Few would escape the pain."

By [Joe Wilkins](#) / Published Oct 1, 2025 4:11 PM EDT

This was ALL avoidable,
had people only tested seriously,
and listened to serious testers.

 You

Give me a table. In the left column, provide thirty random, non-sequential numbers. In the right column, give the English spelling of these numbers. Sort the entire table in alphabetical order based on the values in the second column.

 ChatGPT

Here is the table sorted alphabetically by the English spelling of the numbers:

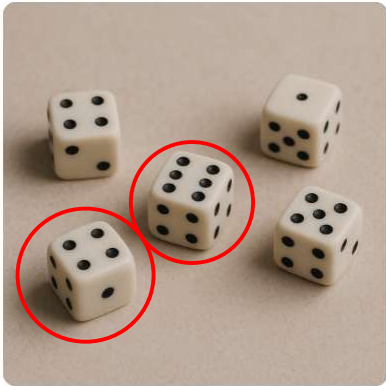
Number	Spelling
19	nineteen
17	seventeen
16	sixteen
11	eleven
15	fifteen
14	fourteen
20	twenty
29	twenty-nine

Why Testers - 25

Show me an image of five dice. Determine the number indicated by the dice, and tell me what it is. I will then reply with the correct number based on my own algorithm.

We will repeat this process until you know how to correctly predict what I will say, in any situation, even if I show you a set of dice that I have rolled.

Image created



The dice show: 4, 4, 6, 1, and 5.
Their total is 20.

When we look **closely**,
GPTs often fall over at the
first hurdle!



Why Testers - 26

After a lot of nudging, from me, the GPT produces some text that's accurate... but it's **not congruent** with the text it generated earlier.

And why should it be? GPTs *don't know what they're saying*.

Where this differs from my in-conversation description (Item 3 elaborated)

In the "Dice – First Huge Run" thread, I described actions as if I were deliberately applying heuristics, designing tests, and "simplifying analysis." That's a human-readable narrative. What actually happened was:

- Probabilistic generation vs. explicit reasoning

I did not run an internal, auditable algorithm for dice analysis. I produced text by predicting the next token in a Transformer, influenced by training data about dice/testing. (Transformer & GPT papers.)

- Hidden service stages you didn't see

Your inputs (and my outputs) likely went through moderation filters, tokenization, context trimming, and (if available) tool-routing—mechanics I didn't narrate in the chat. (Moderation + tools docs; tokenizer docs.)

- Controls can shape results

Decoding knobs (e.g., temperature, stop) and logit bias can materially change my wording/focus—again, not something I described while "explaining" heuristics. (OpenAI parameter docs.)

- Instruction hierarchy and memory

My behavior is constrained by system/developer/user instruction priority, and may (or may not) include Memory depending on settings—another behind-the-scenes factor absent from my narrative. (Instruction hierarchy and Memory docs.)

In short: my earlier description presented a clean, human-style chain of reasoning; the actual pipeline is a layered system of policy filters, tokenization and context management, probabilistic generation, optional tool use, output filtering, and retention/memory rules.

Why Testers - 27

Obliviousness to Truth

These systems generate plausible-sounding output, but they have neither models of the world, nor social competence, nor an understanding of the difference between what is truthful and untruthful.

They're not liars (that requires intention, which they don't have) but they do generate bullshit — text uttered without regard to truth.

They are fundamentally unreliable, and there's currently nothing that we can do to solve that problem reliably.

Real testers have been aware of all this
from the get-go, because we study
systems and claims about them.

Why Testers - 29

Then we TRY them, looking for PROBLEMS.

Why Testers - 30

Why Testers?

weakest argument

- **Empirical** (instead of wishful)
- **Skilled** (instead of merely intuitive)
- **Different** (instead of redundant)
- **Motivated** (instead of ambivalent)
- **Available** (instead of preoccupied)

strongest argument

No conflict of interest!

Why Testers - 31

Why

weakest argument

- **Empirical**
- **Skilled**
- **Different**
- **Motivated**
- **Available**

strongest argument



Responsible companies hire testers because good testers are...

Empirical (instead of wishful)

- finding bugs with respect to the intentions of the designer/developer
- predicting if the users will accept it, and anticipating how they might want it to be improved
- understanding the risk of bugs yet unknown
- discovering surprising behavior and making sense of it
- discovering how the product behaves vs. time and data and environment
- discovering how the product fails and how it recovers from failure
- following up on specific user experiences or complaints
- finding new uses for the product
- understanding how recent changes may have affected the product
- understanding the product in comparison with competitors
- understanding how the product interrelates or integrates with the ecosystem
- discovering how the product may violate relevant laws or industry standards

Skilled (instead of merely intuitive)

- accumulating and curating professional and institutional knowledge about testing
- extracting test ideas from a variety of sources and materials
- gathering and curating repositories of test data
- acquiring, adapting, customizing, or developing tools as needed
- making educated judgments about the world of the users, rather than relying solely on explicit specifications
- able to select the least burdensome approach that gets the job done
- tracking what has been tested and how well it has been tested
- looking ahead to anticipate and be ready to test upcoming products
- evaluating their process and themselves as they go
- explaining and justifying their work
- reporting test results in a manner that allows their clients to make timely and informed decisions

Different (instead of redundant)

- focusing on trouble rather than success: testers have faith that trouble exists
- considering the "big picture", while working with the details
- exploring obscure possibilities and possible diversities
- having "critical discourse" sustained by detailed knowledge of how the code should work
- functioning as an independent source of information about the status of the product

Motivated (instead of ambivalent)

- enjoying the process of testing, rather than seeing it as a chore or distraction
- being supported in doing it
- unburdened by the responsibility to fix things

Available (instead of preoccupied)

- working full time in parallel with development, so everyone stays in flow
- having time to follow up on subtle clues and cues that might lead to trouble
- being flexible and responsive to the needs of their clients
- not doing development

no conflicts of interest

<https://www.satisfice.com/download/why-testers>

Why Testers - 32

Demonstration

“See? It does *this!*”

To *show* something we know

Rehearsal focuses on outcome

Repetition is often ceremonial

Inconsistency in results is undesired and troubling

Variation of factors is risky and may undermine the demonstration

Suppression of risk

Deeper truths are beside the point

Experiment

“See *whatever* it does.”

To *challenge* what we *think* we know

Rehearsal focuses on integrity

Repetition is contingent

Inconsistency in results is welcomed and intriguing

Variation of factors is desirable and may improve the experiment

Identification of risk

Deeper truths *are* the point

Why Testers - 33

Deep or Shallow?

Shallow testing is testing that **has a chance** of finding **every easy bug**.

Testing is hard!

Testing is easy!

Deep testing is testing that **maximizes the chance** of finding **every elusive bug that matters**.

Why Testers - 34

Most developers probably won't test deeply.

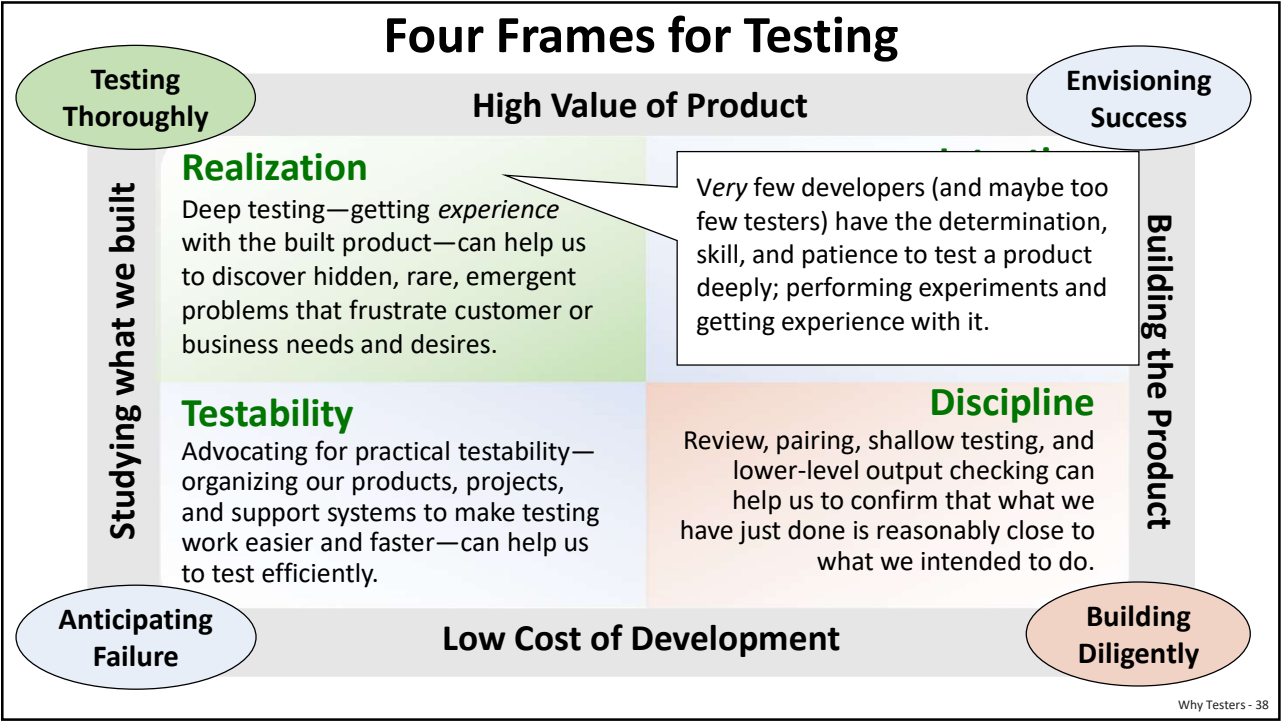
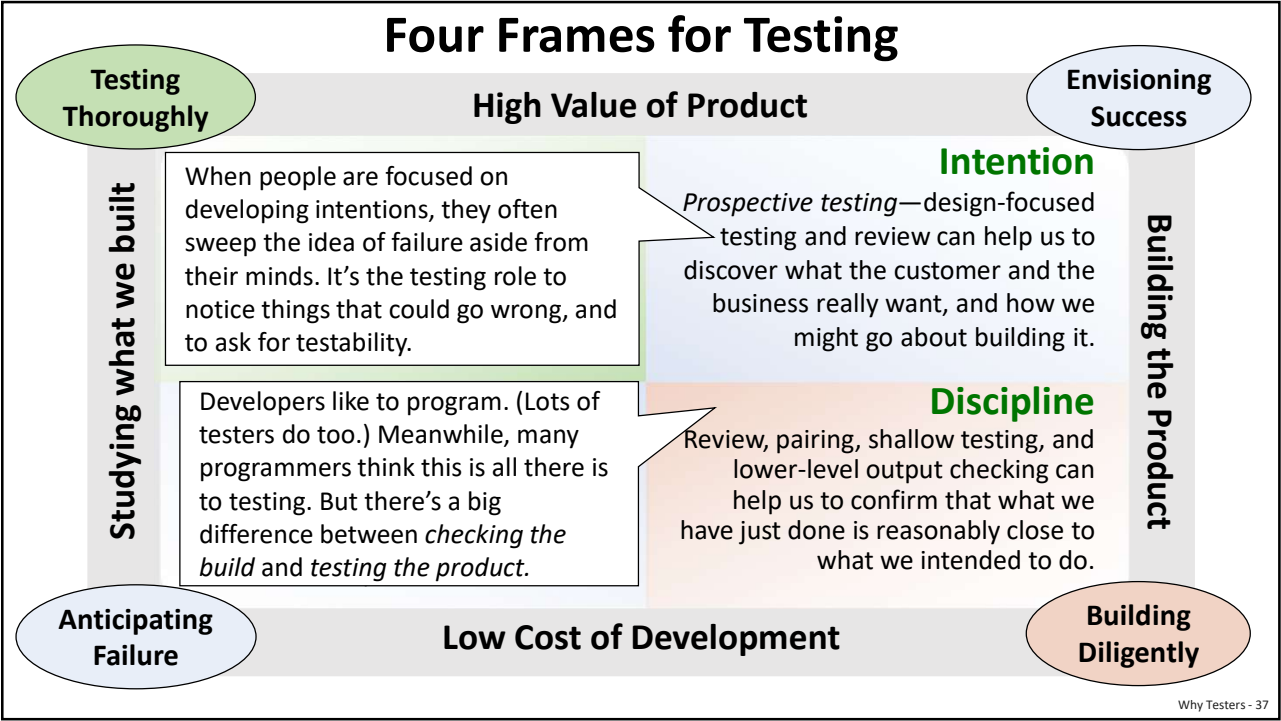
But this is *not* because they're lazy, or incompetent testers, or Just Bad People. There's a different reason.

Why Testers - 35

**“Show me the incentives
and I'll show you the behaviour.”**

—Keith Klain (channeling Charlie Munger)

Why Testers - 36



The four frames can apply to any and all levels of our work to all the precursors of the final product, as well as to the final product.

Intention

Prospective testing—design-focused testing and review can help us to discover what the customer and the business really want, and how we might go about building it.

Discipline

Review, pairing, shallow testing, and lower-level output checking can help us to confirm that what we have just done is reasonably close to what we intended to do.

Testability

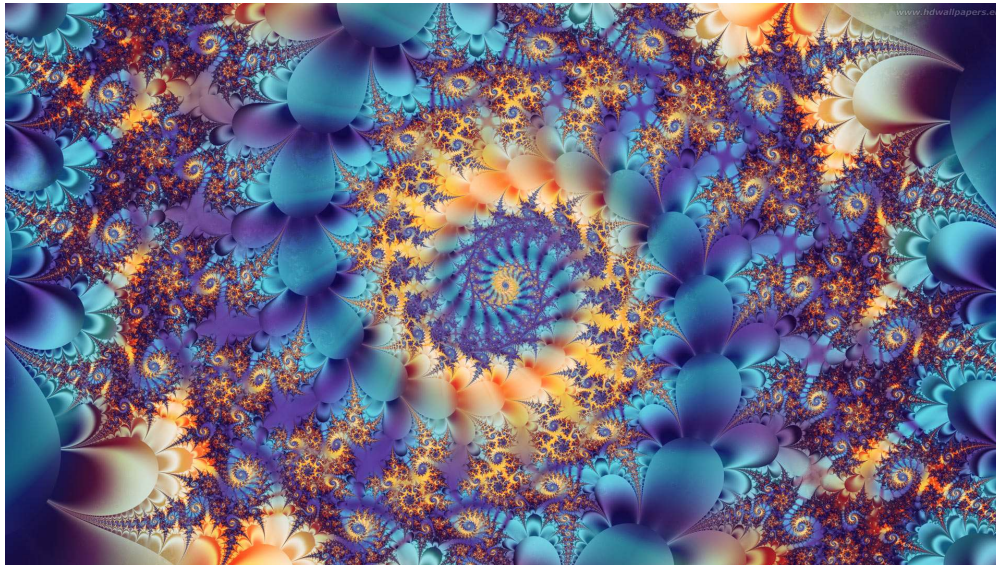
Advocating for practical testability—organizing our products, projects, and systems to make testing work easier and faster—can help us to test efficiently.

Realization

Deep testing—getting *experience* with the built product—can help us to discover hidden, rare, emergent problems that frustrate customer or business needs and desires.

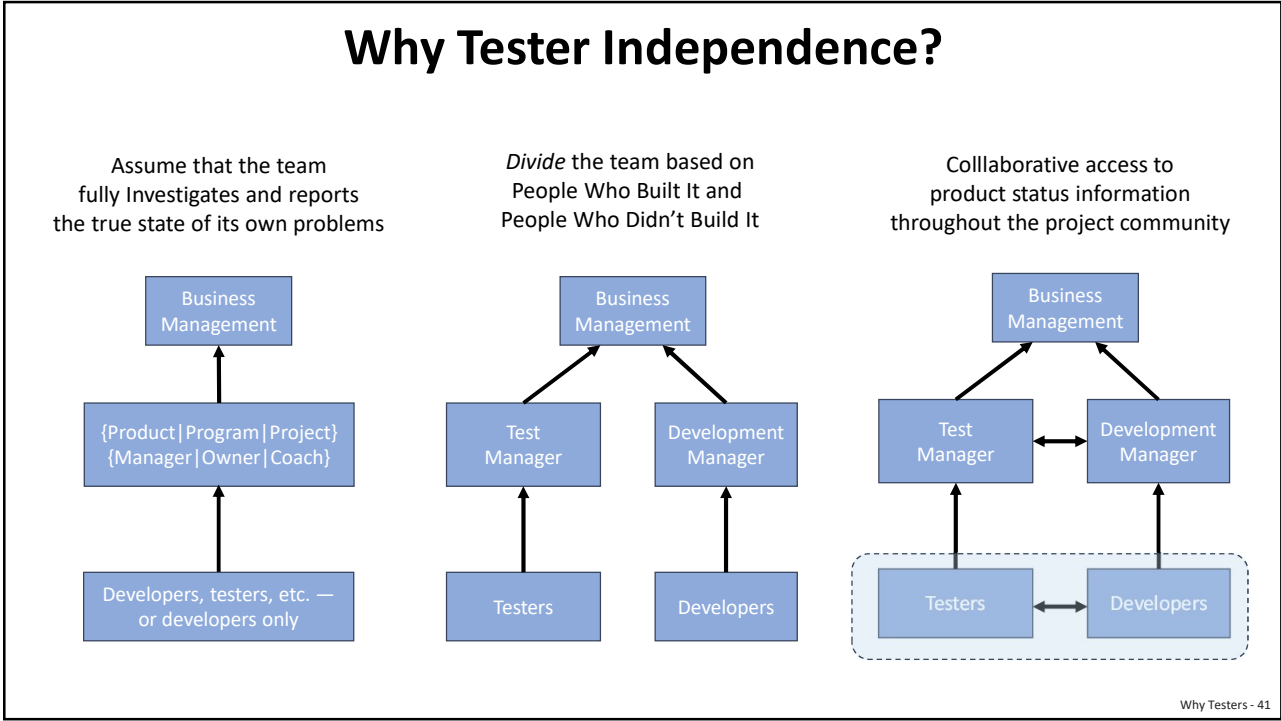
Why Testers - 39

...but not just in lines; not even just loops.



Development is a fractal!

Why Testers - 40



The Burning Question:

But are we independent?

Why Testers - 42

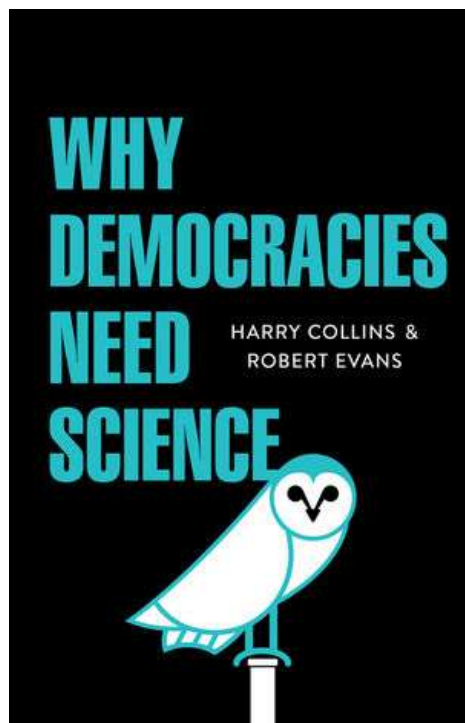
Not isolation, but independence of thought, and of *stance*

Critical distance is the ability to think differently, oppositionally from another person, and to investigate a matter without pre-judging it.

Testers must maintain critical distance as **outsiders** to the maker mindset, **not emotionally attached** to a bug-free product. **Testers must love looking for trouble, and finding it.**

Critical distance falls apart **when we lose our differentness; when we fear displeasing powerful people; when we become attached to certain outcomes; when we focus on confirmation.**

Why Testers - 43



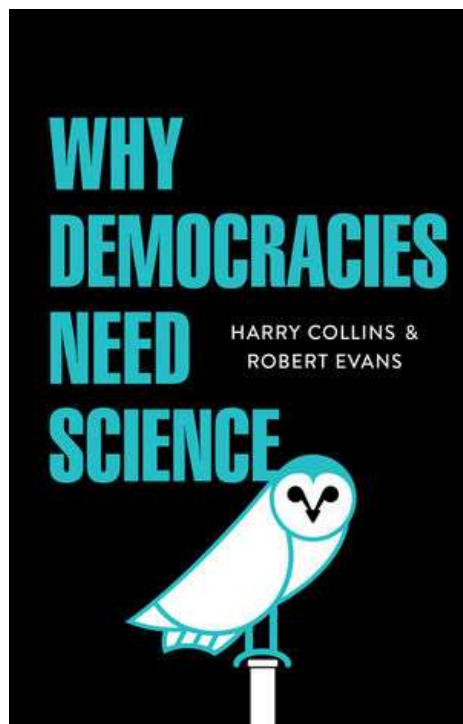
Formative Aspects of Science: Observation

"If one wants to know about some feature of the world, does one prefer to listen to one who has observed that feature of the world, or one who has not viewed that feature of the world?"

"We prefer to give more weight to the one who has observed, even though we know that observation is inexact and impure and open to illusions, observer effects, and the influence of the social group."

We must observe directly, but also deeply. We use tools to help us see trouble, but we must not allow them to distract or displace us from seeking and finding trouble.

Why Testers - 44



Formative Aspects of Science: Corroboration

"We prefer to give weight to the outcome of experiments which appear to that have been successfully replicated over those which have not."

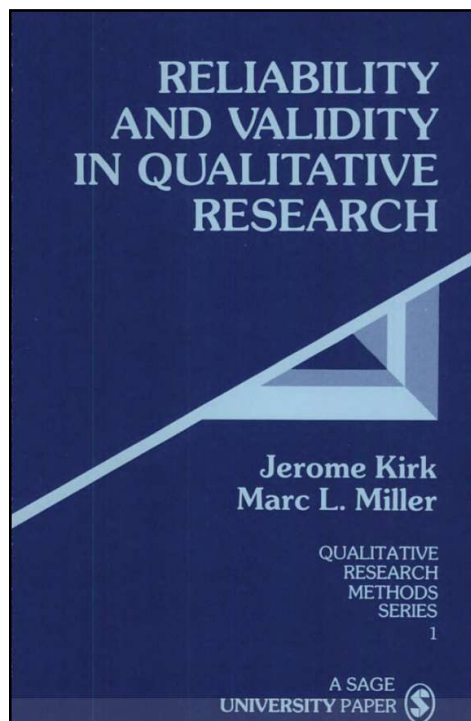
"We would give more weight to the opinions of those who believe that it is proper for their findings to be replicated."

This runs into the problem of *experimenter's regress*.

Yet we choose to believe more in successfully replicated experiments than in one-offs.

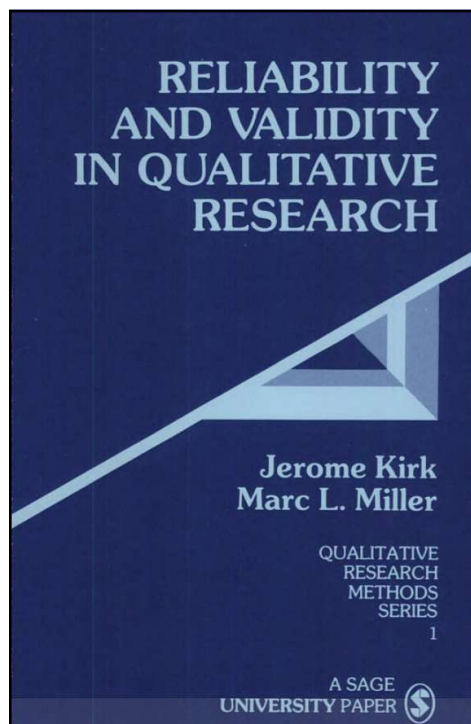
BUT we must be careful about repetition, because...

Why Testers - 45



*"Most of the technology of 'confirmatory' non-qualitative research in both the social and natural sciences is aimed at **preventing discovery**. When confirmatory research goes smoothly, **everything comes out precisely as expected**. Received theory is supported by one more example of its usefulness, and requires no change."*

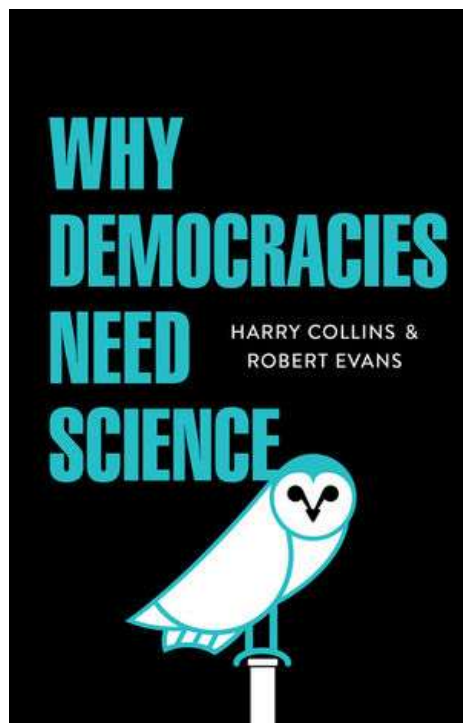
Why Testers - 46



*“As in everyday social life, **confirmation is exactly the absence of insight**. In science, as in life, dramatic new discoveries must almost by definition be accidental (‘serendipitous’). Indeed, they occur only in consequence of some mistake.”*

As testers, we must avoid being drawn into expensive, formal, confirmatory demonstrations that everything is okay — what people often call “automated testing” — *low-quality reassurance*.

Why Testers - 47



Formative Aspects of Science: Falsification

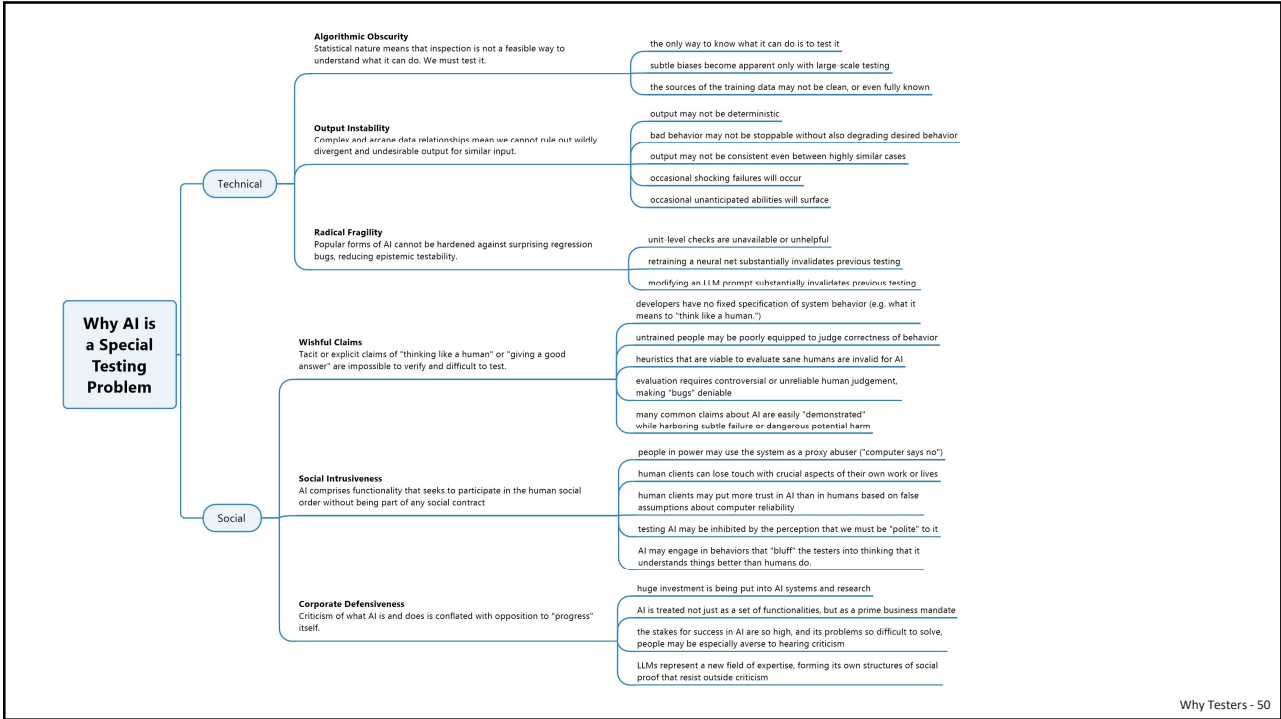
*“What is to be preferred---a world in which those who claim to make observations are willing to set out the conditions under which they can be shown to be wrong, and **actively invite the test**, or a world in which they would consider this unnecessary or inappropriate?”*

(my emphasis)

Testing can’t just be a set of demonstrations to show that things *can* work. We must show where the product doesn’t work; how it fails; and how it might fail.

Why Testers - 48

What makes AI (and LLM/GPTs) problematic?



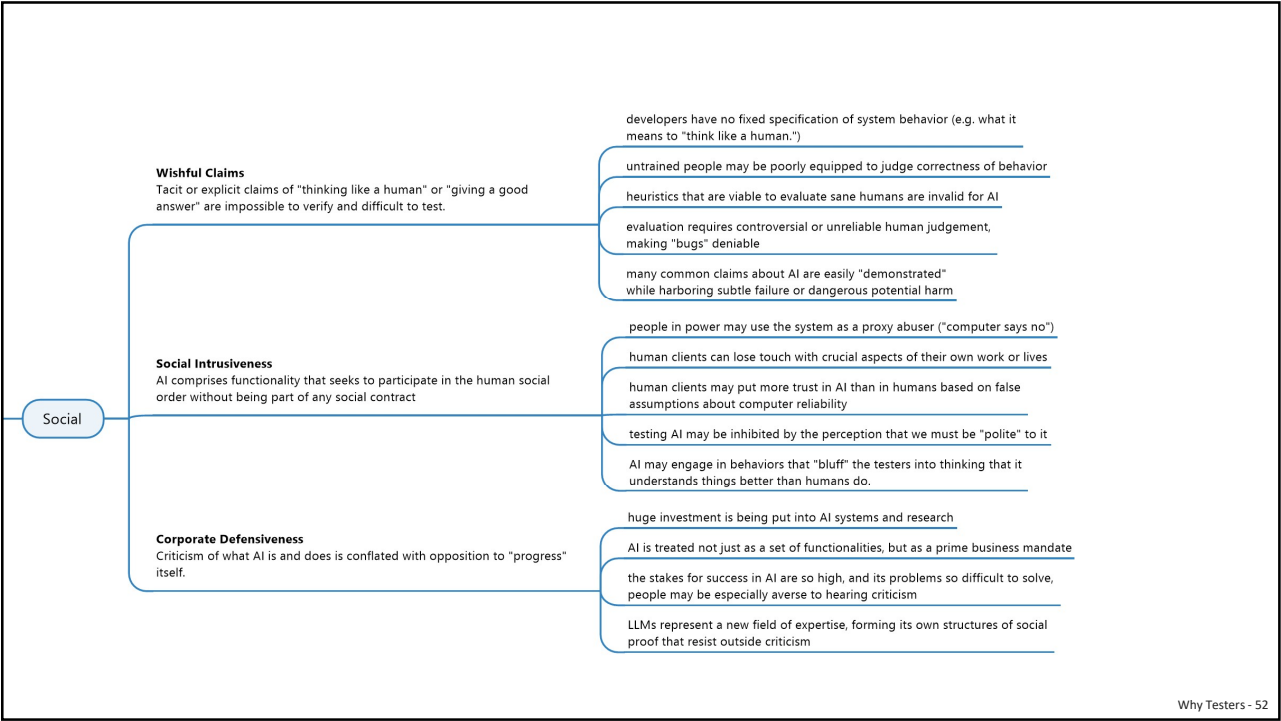
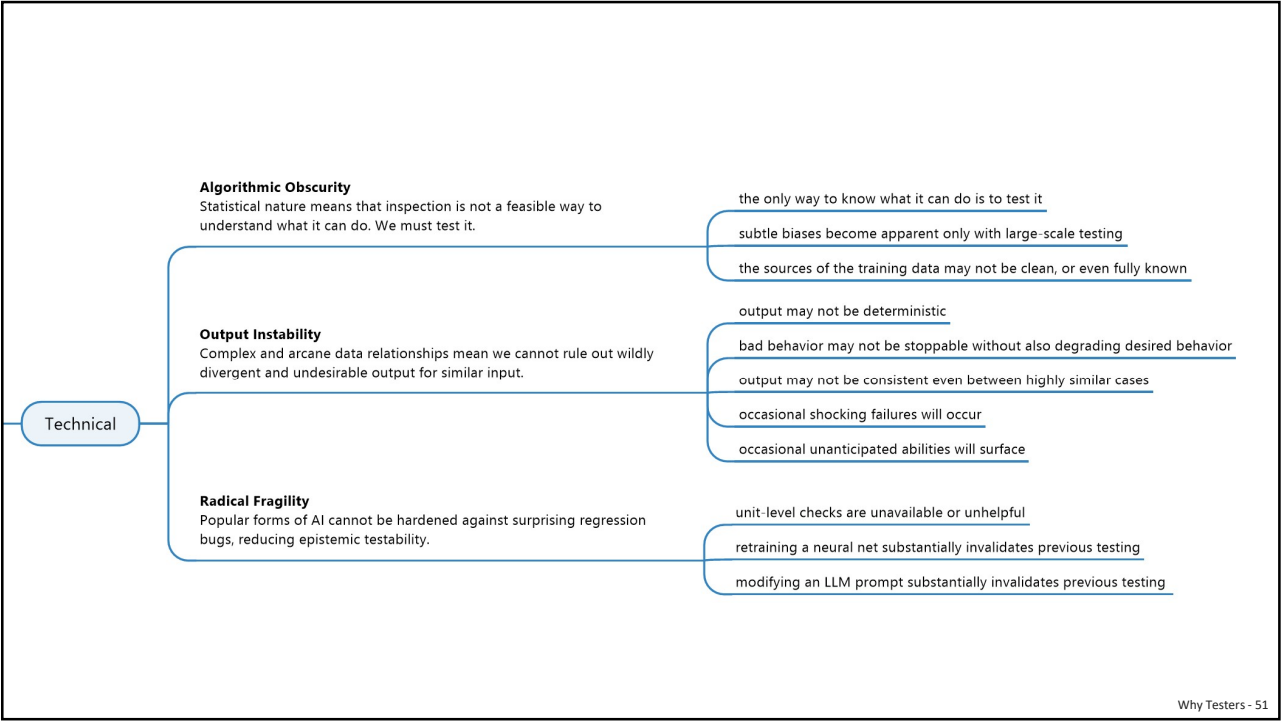


Photo 129244745 © Petesalotos |
Dreamstime.comPhoto 129244745 © Lightfieldstudiosprod |
Dreamstime.com

The Large Language Mentalist Syndrome

When human beings (through social training and experience) see patterns of text that closely resemble human writing, it is almost irresistible to treat a GPT's output as human — and then to read human intention, interpretation, and intelligence into that.

Expensive to Test

This stuff is *crazy* expensive to train, to use, to fix, and to test deeply. Non-deterministic output requires extensive scrutiny and testing, and statistical analysis. Without that, we can miss subtle bugs and problems that matter.

That's a big deal, not just for the business, but for society at large — because when unreliable software is “tested in production”, real people can be hit with real problems.

AI can fool us. Experiment deeply and LOOK CLOSELY.

- “AI saves time.” Generation time, maybe. But what about time time required for critical evaluation and correction?
 - Did it perform the assignment we gave it, or does it just look like that at first glance?
 - Did it give correct, complete, and consistent answers?
 - Did it drown us in unnecessary fluff that we have to wade through to notice problems?
 - Did it make stuff up?
 - How much of the happy results required us to nudge and repair the bot when it went off track? Are we just seeing a ship in a bottle without seeing what it took to get there?
 - When it says it corrected itself, did it actually correct itself? Did it add new mistakes?
 - Was the output examined critically by people who are actually experts in the domain?
 - Can we trust it to do this again? Every time? For every similar situation? How about now?

Why Testers - 55

AI can fool us. Experiment deeply and LOOK CLOSELY.

- “AI saves time.” Generation time, maybe. But what about time time required for critical evaluation and correction?
 - Did it perform the assignment we gave it, or does it just look like that at first glance?
 - Did it give correct, complete, and consistent answers?
 - Did it drown us in unnecessary fluff that we have to wade through to notice problems?
 - Did it make stuff up?
 - How much of the happy results required us to nudge and repair the bot when it went off track? Are we just seeing a ship in a bottle without seeing what it took to get there?
 - When it says it corrected itself, did it actually correct itself? Did it add new mistakes?
 - Was the output examined critically by people who are actually experts in the domain?
 - Can we trust it to do this again? Every time? For every similar situation? How about now?

Keep good notes, track your time,
and report the whole truth, even if it
might disappoint the bosses
who are being pressured or conned.

Do not consider a pleasing demo as
evidence of good and reliable behaviour.

Try *real experiments*
multiple times. Watch
results closely.

Why Testers - 56

AI can fool us. Experiment deeply and LOOK CLOSELY.

- Do we understand the entire system? Can we find elements and interfaces that are easier and more amenable to testing?
- What could go wrong? If things go wrong, how will we know?
- When we say “it gave a good answer”, how much of the answer are we filling in ourselves?
- When we say that any kind of software can do a job, part of the job is *not* being done by the software?
- When it doesn’t do a good job, will a responsible human have the time and the capability to override the system, or will problems get ignored or buried?
- Is the answer from a probabilistic text generator better than (say) a curated template?
- Are the fanboys and managers gaslighting us by treating actual failure as “limited success”?

Why Testers - 57

Can AI *ever* be okay? Sure... *maybe!*

- When risk is low; when nothing significant is at stake
 - no risk of loss, harm, damage, wasted time, diminished value, bad feelings, copyright violation, opportunity cost, societal consequences...
- When variability is tolerable or even welcome (“inspiration”; “creativity”)
- When *actual* creativity isn’t the point; jiggling is (when *YOU* do the creative bit)
- When variation will do (which can be okay, because of our capacity to repair).
- When models are simpler with constrained feature sets
 - See Narayanan and Kapoor, “Against Predictive Optimization”, <https://predictive-optimization.cs.princeton.edu/>
- When risk is low, volume of output is low, AND scrutiny is easy
- When risk is elevated, but people *with requisite expertise* observe and control them.
- When its output is used for *inquiry*, rather than *control*
- When output is used for *discovery and analysis by people*, rather than for *abdicating responsibility for decisions to machines*
- When the output is used as a mirror on the people who created or chose it... or on humans generally.

Again... Why Testers? Because we're...

- **Empirical** (instead of wishful)
- **Skilled** (instead of merely intuitive)
- **Different** (instead of redundant)
- **Motivated** (instead of ambivalent)
- **Available** (instead of preoccupied)

Why Testers - 59

New EU Product Liability Directive

Covers safety (damage to life, property, or data)

Article 7

Defectiveness

1. A product shall be considered defective where it does not provide the safety that a person is entitled to expect or that is required under Union or national law.
2. In assessing the defectiveness of a product, all circumstances shall be taken into account, including:
 - (a) the presentation and the characteristics of the product, including its labelling, design, technical features, composition and packaging and the instructions for its assembly, installation, use and maintenance;
 - (b) reasonably foreseeable use of the product;

Why Testers - 60

New EU Product Liability Directive

Makes it easier to prove that a product is defective...

Article 10

Burden of proof

1. Member States shall ensure that a claimant is required to prove the defectiveness of the product, the damage suffered and the causal link between that defectiveness and that damage.
2. The defectiveness of the product shall be presumed where any of the following conditions are met:
 - (a) the defendant fails to disclose relevant evidence pursuant to Article 9(1);
 - (b) the claimant demonstrates that the product does not comply with mandatory product safety requirements laid down in Union or national law that are intended to protect against the risk of the damage suffered by the injured person; or
 - (c) the claimant demonstrates that the damage was caused by an obvious malfunction of the product during reasonably foreseeable use or under ordinary circumstances.

Why Testers - 61

New EU Product Liability Directive

Doesn't let companies hide behind a EULA...

- (56) The objective of protecting natural persons would be undermined if it were possible to limit or exclude an economic operator's liability through contractual provisions. Therefore no contractual derogations should be permitted. For the same reason, it should not be possible for provisions of national law to limit or exclude liability, such as by setting financial ceilings on an economic operator's liability.

<https://data.consilium.europa.eu/doc/document/PE-7-2024-INIT/en/pdf>

Why Testers - 62

Things You Could Do

- Read the regulations. Study them. Make notes on them. They can be boring and repetitive. They're long. They're hard to read. *You'll probably be the only one who knows them.*
- Learn about the underlying technologies. It's easier to be fooled when you don't know how things work under the hood.
- Practice and perform testing. Expand your models of oracles and coverage.
- Work with *data*, not just with functions. Actually test the danged product.
- Focus on finding problems that matter — that present risk to the business, to its customers, and to society, before it's too late.
- Formalized, procedurally structured test cases are helpless against non-deterministic systems. Your testing will have to focus on probability, statistical analysis, and risk of misuse.
- Learn to describe and explain your work.

Why Testers - 63

But be warned. People may say they want testers...



Why Testers - 64

...but talking about *product* trouble brings *social* trouble.



Why Testers - 65

Talking about *any* trouble brings *social* trouble.



Why Testers - 66

Beware of geeks bearing gifts.



Why Testers - 67

And be brave.
Run towards the risk.
Speak up. Speak truth to power.
And let's take care of each other.

Why Testers - 68