



TESTABILITY

FOR *the* WIN

RAPID SOFTWARE TESTING AS A HABIT

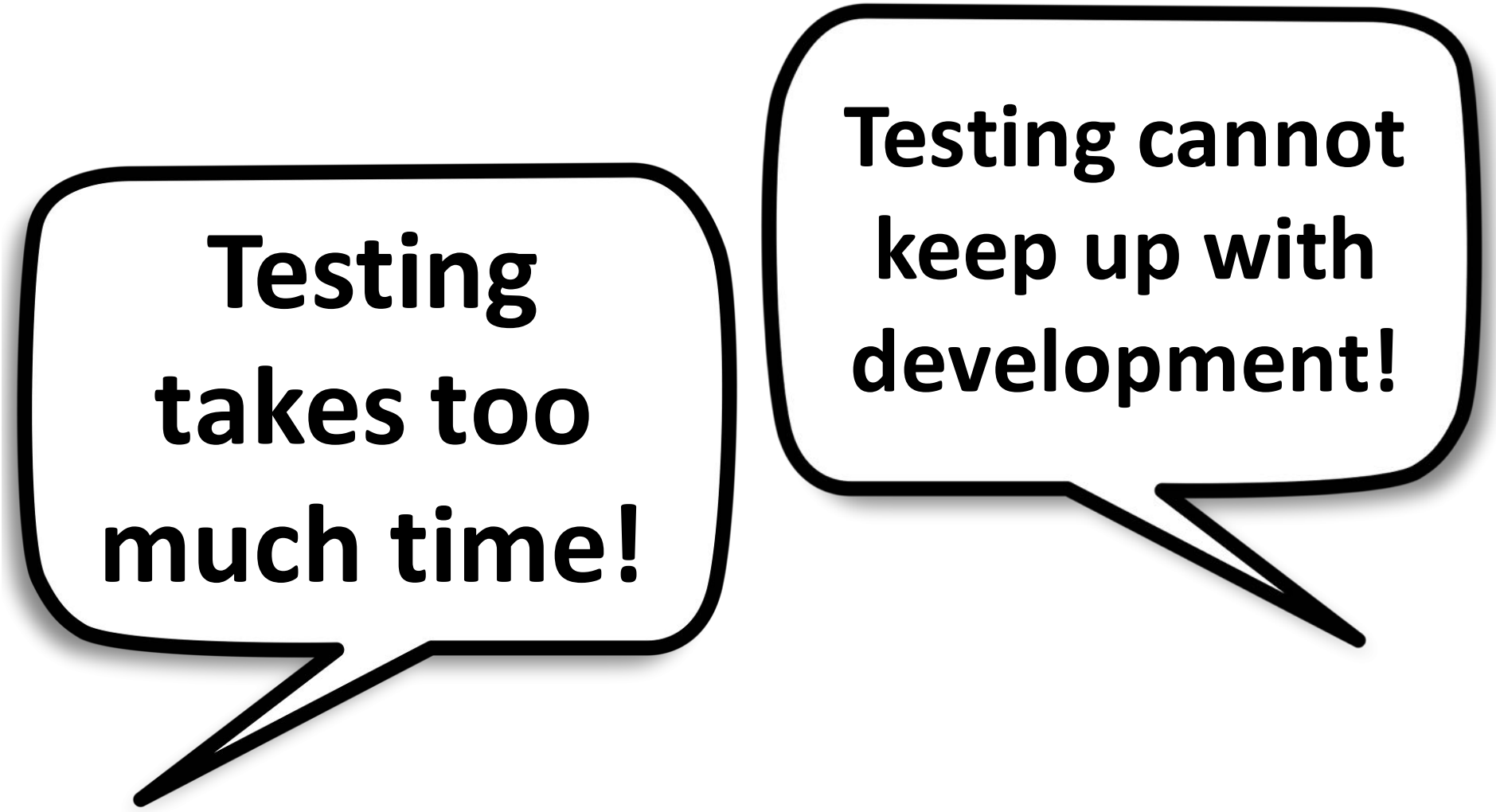
MAKING BETTER SOFTWARE FASTER

Who am I?

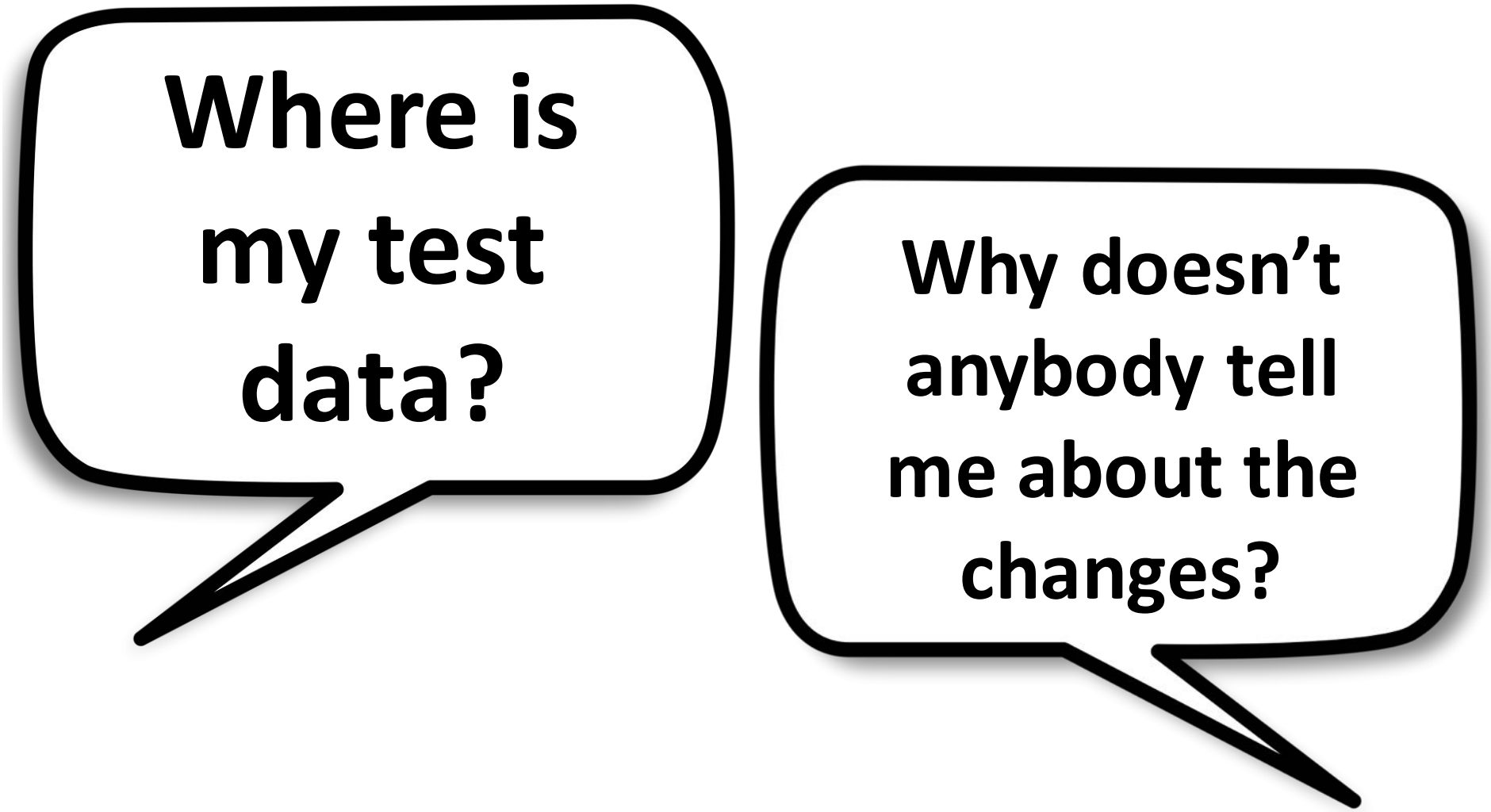
- Context-driven software tester
- Rapid Software Testing teacher
- Curious & lifelong learner
- Passionate & energetic Humanist
- Trainer, coach, writer, speaker & leader
- Books & Apple gadget collector
- Trombone player
- Gamer
- STAR WARS & LEGO freak
- Beer brewer

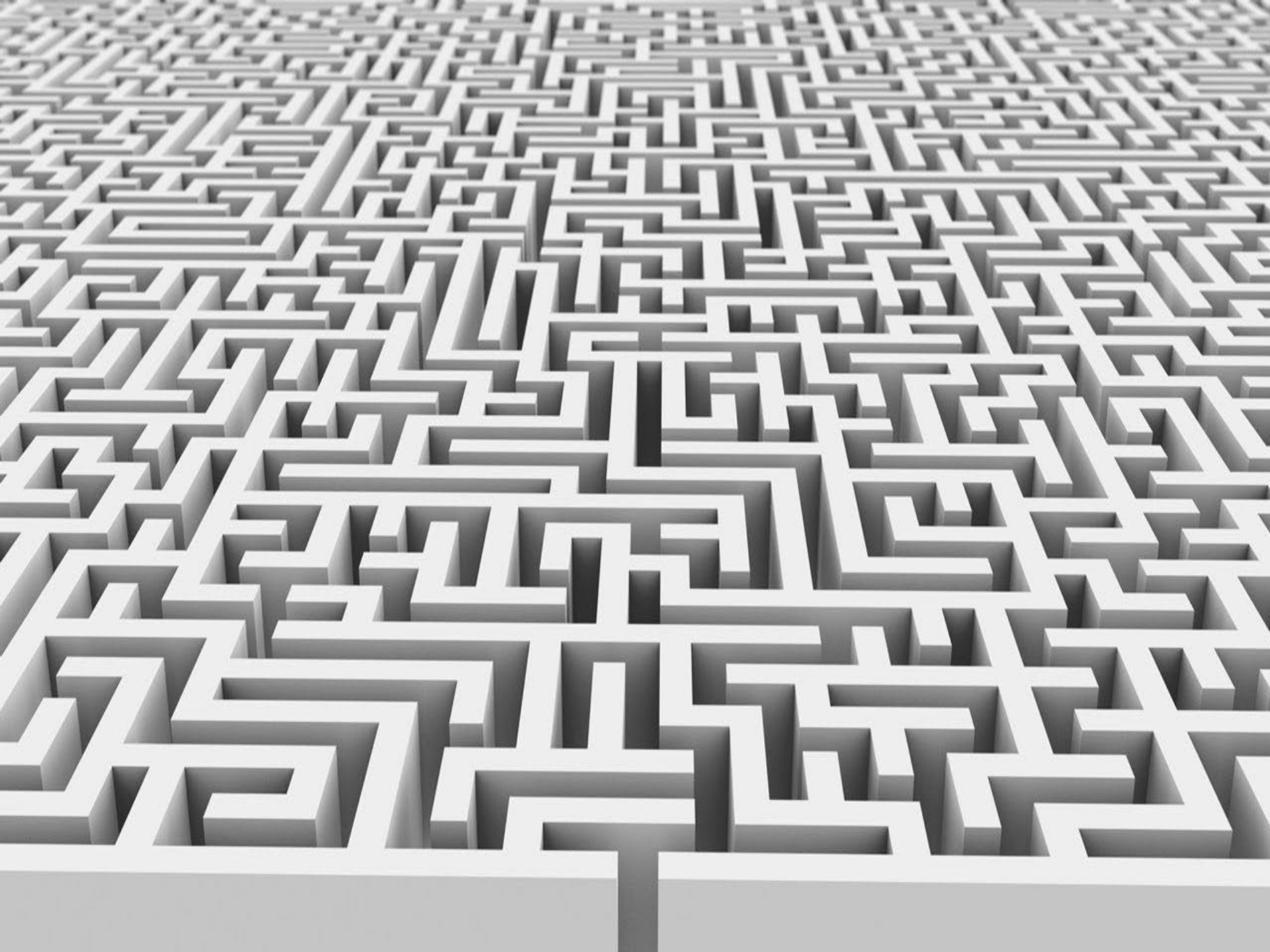
1. A Rapid Software Testing view on testing in an agile context.
2. What is testability and how can teams take maximum benefit from it?
3. What teams can do to get the most out of testing!

Testing


Testing takes too much time!
Testing cannot keep up with development!

In the meanwhile…





Testable?

Imagine a space shuttle under a glass shield. You are not allowed to come near it or poke or probe it…

Can you test this?

Agile: a manifesto …

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Source: http://agilemanifesto.org/

… and 12 principles

1. Customer satisfaction by early and continuous delivery of useful software
2. Welcome changing requirements, even in late development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity - the art of maximizing the amount of work not done - is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstance

Source: http://agilemanifesto.org/

But why?

Software development

Factory

or

Research & development

Unknown unknows

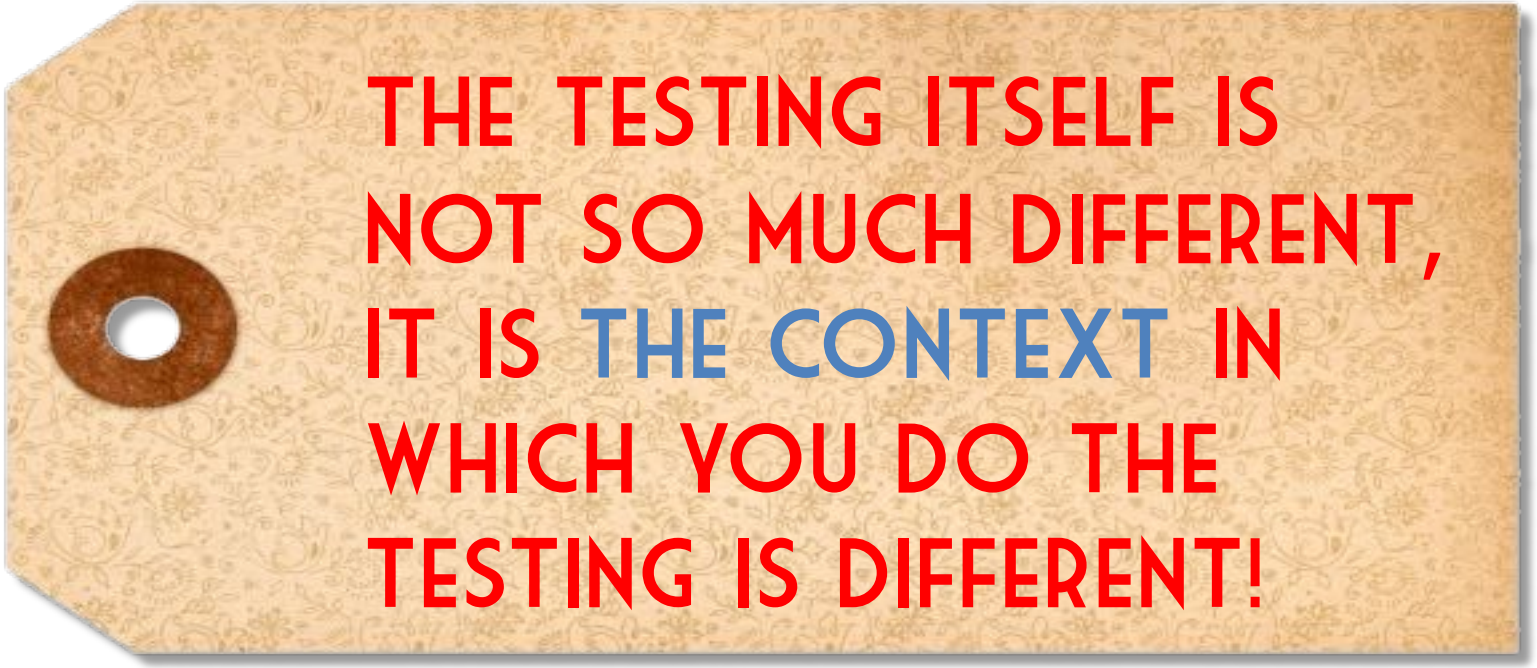
- Capture everything upfront vs. building new insights
- Customers don’t know what they want
- Cope with complexity, confusion, change, new insights and half answers
- Classic waterfall and "command and control" in the way of insight and serendipity

Agile testing to some…

- TDD
- ATDD, BDD, Specification by example
- 100% test automation
- No more testers
- Built-in quality
- Faster, shorter, better!

Agile testing?

- What makes testing agile?



Source: http://www.huibschoots.nl/wordpress/?p=1072

Some differences…

- Less time to prepare, execute and report
- Iterative and incremental approach
- Focus on test automation
- Role change: less testing, more coaching
- Less certainty: change is common
- Team work
- Continuous critical thinking

Source: http://www.huibschoots.nl/wordpress/?p=1072

Rapid Testing

Rapid testing is a mind-set
and a skill-set of testing
focused on how to do testing
more quickly,
less expensively,
with excellent results.

This is a general testing methodology. It adapts to any kind of project or product.

Source: Rapid Software Testing – James Bach & Michael Bolton

The Premises of Rapid Testing

1. Software projects and products are relationships between people, who are creatures both of emotion and rational thought.
2. Each project occurs under conditions of uncertainty and time pressure.
3. Despite our best hopes and intentions, some degree of inexperience, carelessness, and incompetence is normal.
4. A test is an activity; it is performance, not artifacts.
5. Testing's purpose is to discover the status of the product and any threats to its value, so that our clients can make informed decisions about it.
6. We commit to performing credible, cost-effective testing, and we will inform our clients of anything that threatens that commitment.
7. We will not knowingly or negligently mislead our clients and colleagues.
8. Testers accept responsibility for the quality of their work, although they cannot control the quality of the product.

Source: Rapid Software Testing – James Bach & Michael Bolton

Testers light the way

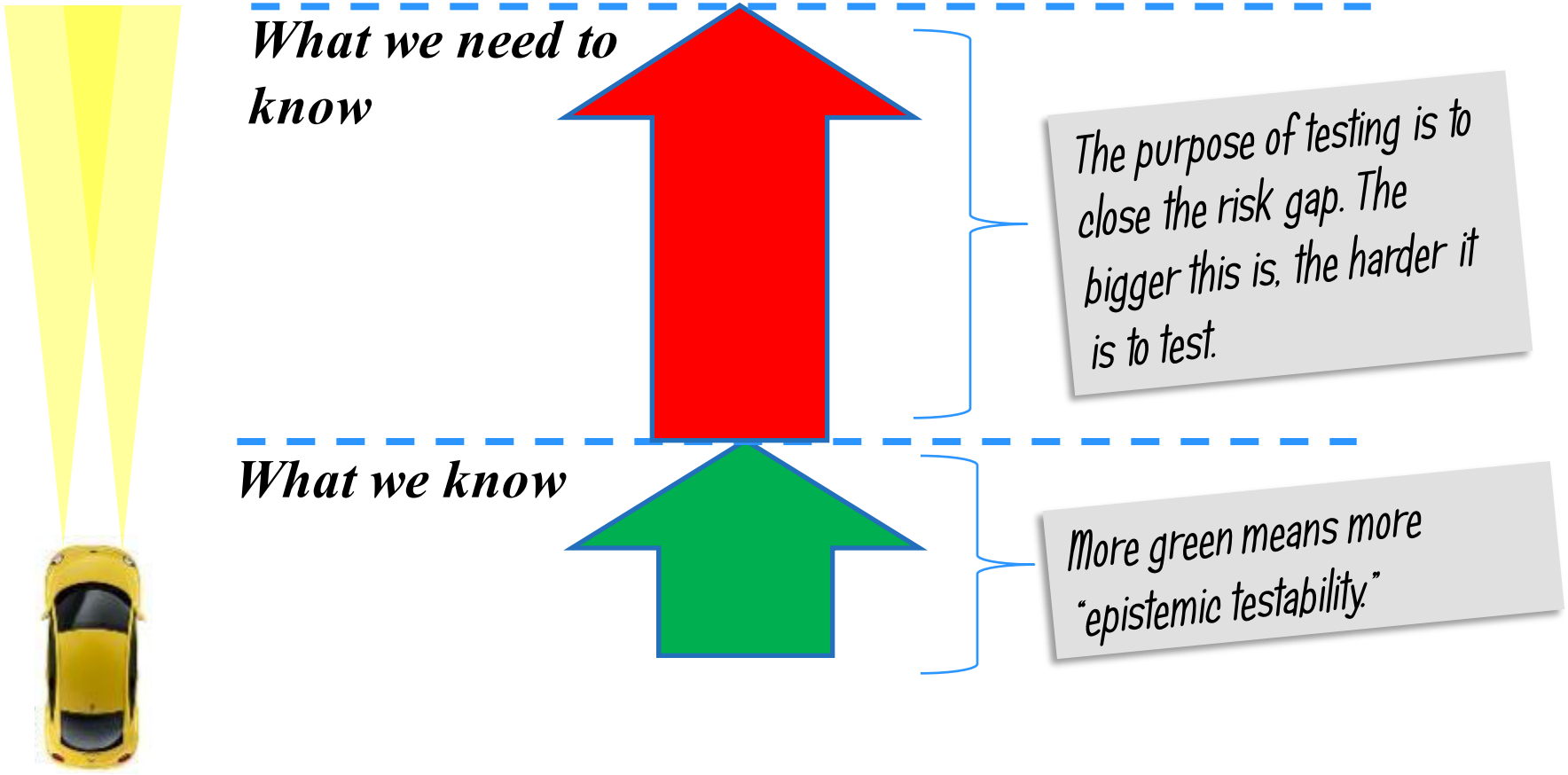
This is our role.

We see things for what they are.
We make informed decisions about quality possible,
because we think critically about software.

Source: Rapid Software Testing – James Bach & Michael Bolton

The Product Risk Knowledge Gap

How much do we already know?



Our knowledge of the status of the product.

Source: Rapid Software Testing – James Bach & Michael Bolton

Call this “Checking” not Testing

operating a product to check specific output…

means

Interact with the product in specific ways to collect specific observations.

Apply algorithmic decision rules to those observations.

Report any failed checks.

Source: Rapid Software Testing – James Bach & Michael Bolton

A Check Has Three Elements

1. An *observation* linked to…
2. A *decision rule* such that…
3. both observation and decision rule can be applied algorithmically.

A ***check*** can be performed

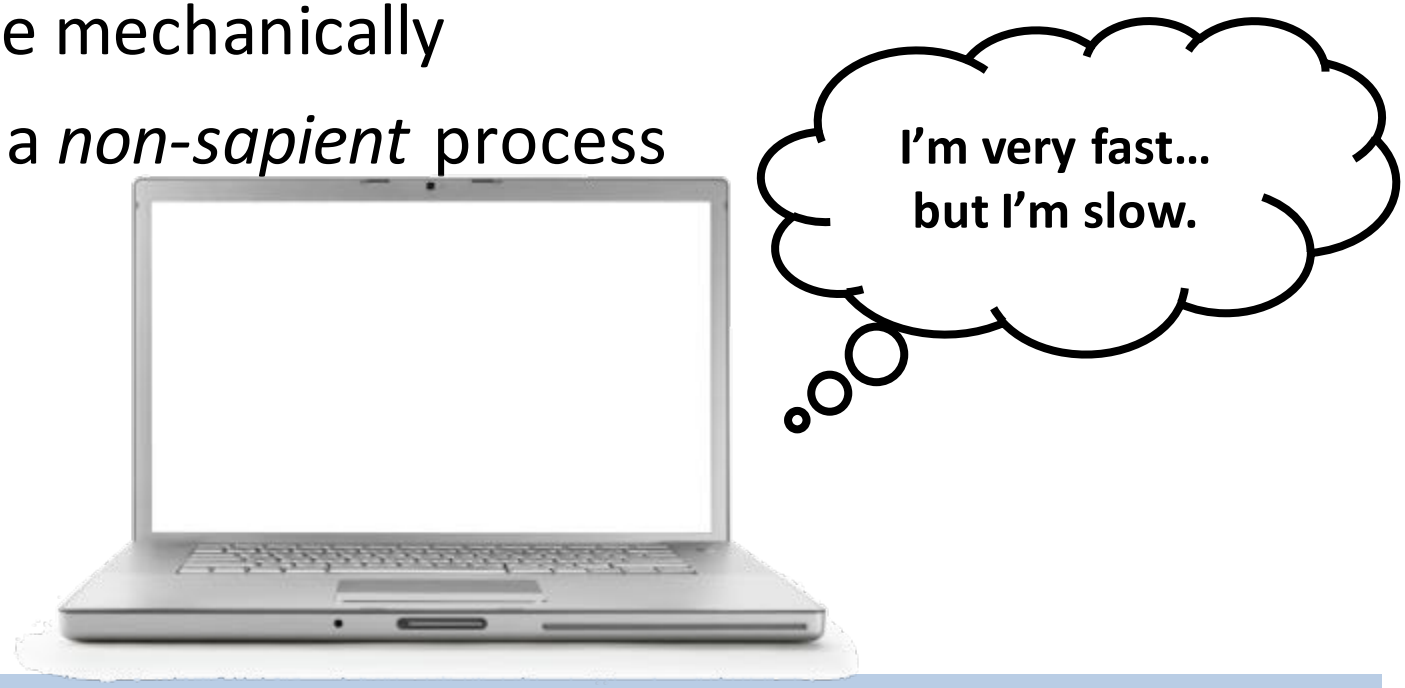
by a machine
that *can't* think
(but that is quick and precise)

by a human who has been
instructed *not* to think
(and who is slow and variable)

Source: Rapid Software Testing – James Bach & Michael Bolton

Testing Is More Than *Checking*

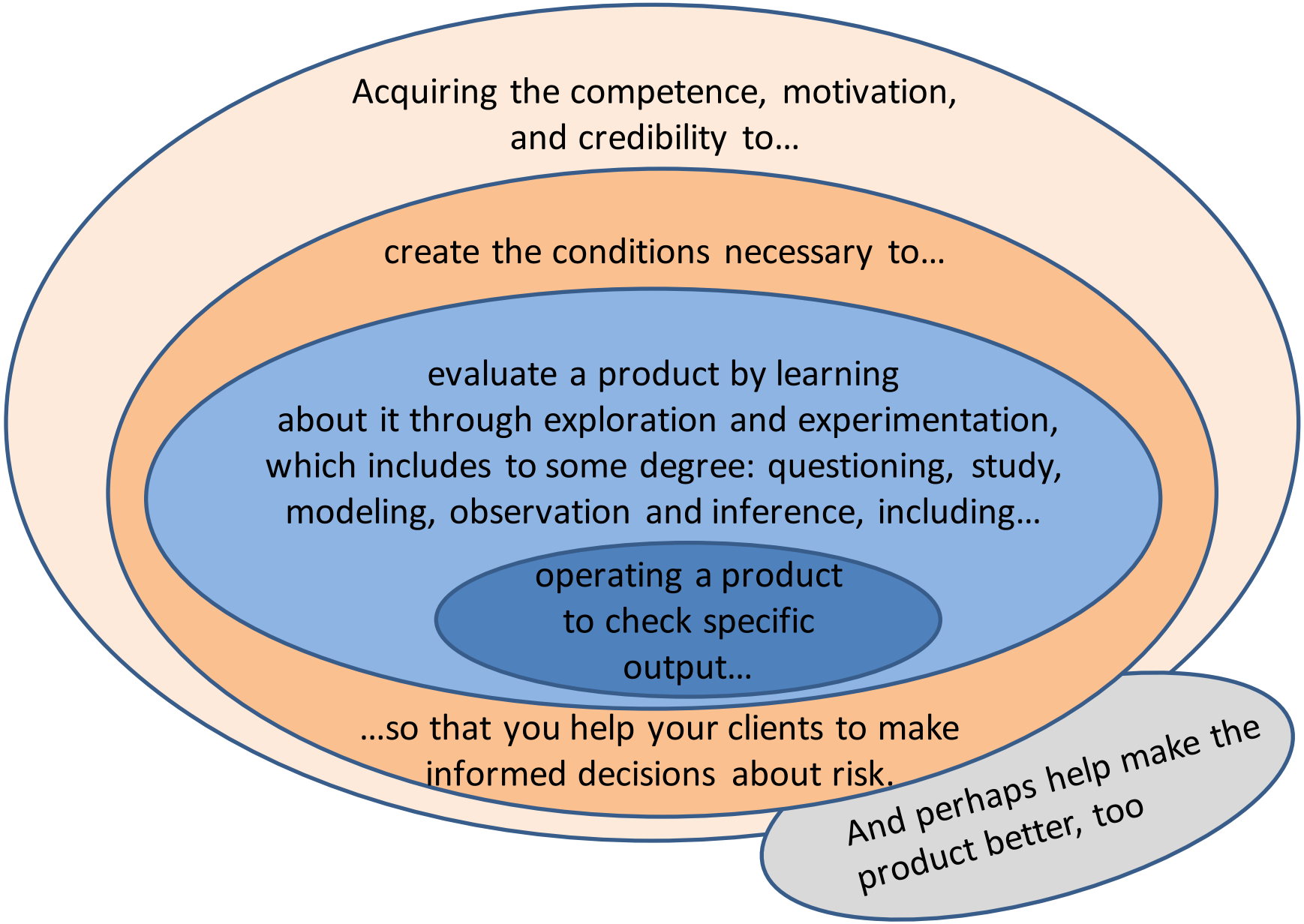
- *Checking* is a process of confirming and verifying existing beliefs
  - Checking can (and we argue, largely should) be done mechanically
  - It is a *non-sapient* process



See http://www.developsense.com/2009/08/testing-vs-checking.html

Source: Rapid Software Testing – James Bach & Michael Bolton

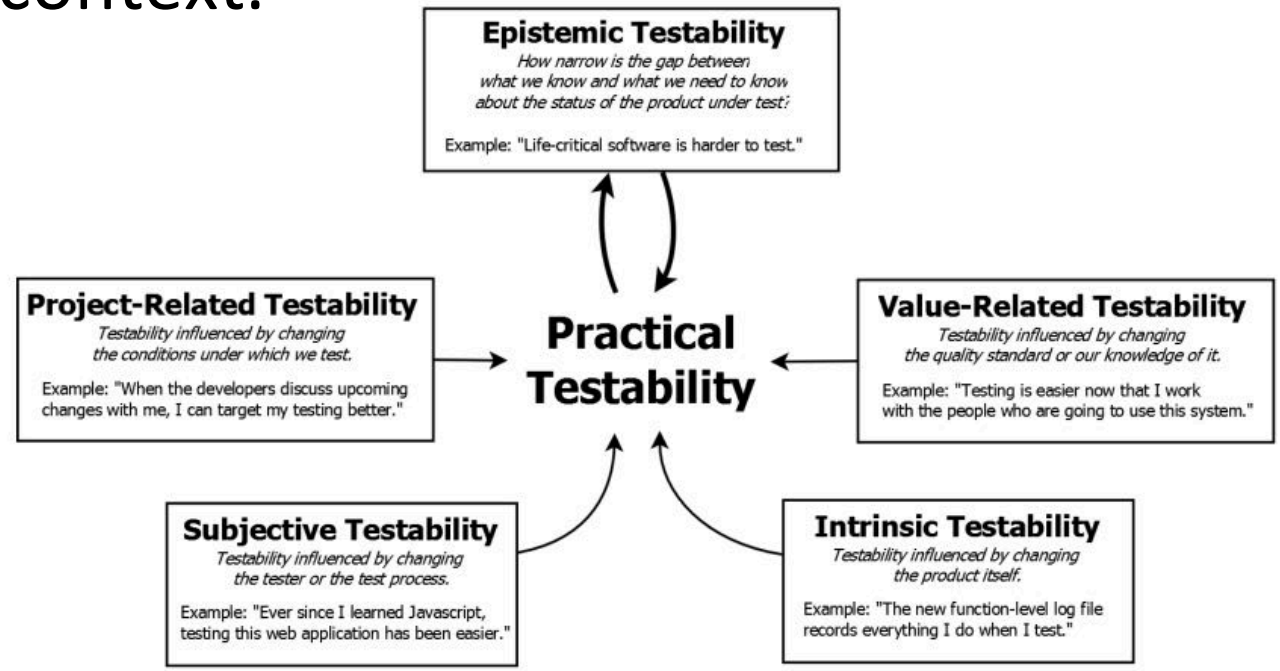
Testing is…



Source: Rapid Software Testing – James Bach & Michael Bolton

Testability

"The practical testability of product is how easy it is to test by a particular tester and test process, in a given context."



Source: http://www.satisfice.com/tools/testable.pdf

Testability? Why?

- Make testing easier
- Make testing faster
- Make testing cheaper
- Less non-reproducible bugs

Project-Related Testability

"Changes in conditions under which we test"

- Change control
- Information availability
- Tools availability
- Test item availability
- Environmental controllability
- Time

Source: http://www.satisfice.com/tools/testable.pdf

Subjective Testability

"Changes of the tester or test process"

- Product, technical and domain knowledge
- Testing skills
- Engagement & collaboration
- Test strategy

Source: http://www.satisfice.com/tools/testable.pdf

Intrinsic Testability

"Changes of the product itself"

- Observability
- Controllability
- (Algorithmic) Simplicity
- Unbugginess, stability & availability
- Smallness

Source: http://www.satisfice.com/tools/testable.pdf

Epistemic Testability

"The gap between what we know and what we need to know"

- Prior knowledge of the quality of the product
- Tolerance for failure (less quality required or risk can be taken)

Source: http://www.satisfice.com/tools/testable.pdf

Value-Related Testability

"Changes of the quality standard or your knowledge about it"

- Oracle availability, authority & reliability
- User availability
- User data availability
- User environment availability

Source: http://www.satisfice.com/tools/testable.pdf

Dynamics

Increasing one type of testability might decrease another…

When developers deliver faster, project-related testability will increase, but intrinsic testability might decrease because the product has more bugs…

Who is responsible?

- **The tester must *ask* for testability.**
  We cannot expect any non-tester to seriously consider testability. (James Bach)

- Really?

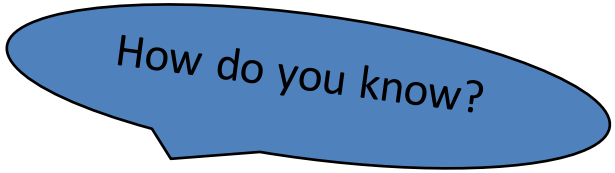
The Testing story: three stories

Level 1: A story about the status of the PRODUCT...

...about how it failed, and how it *might* fail...
...in ways that matter to your various clients.

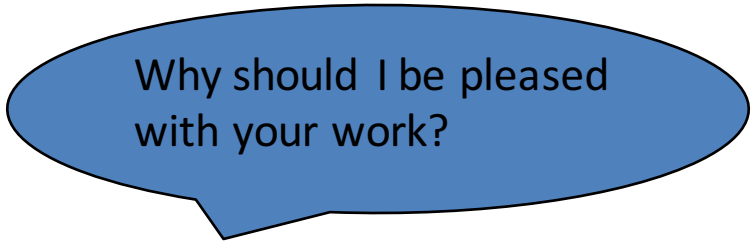
Level 2: A story about HOW YOU TESTED it...

...how you configured, operated and observed it...
...about what you haven't tested, yet...
...and won't test, at all...



Level 3: A story about the VALUE of the testing...

...what the risks and costs of testing are...
...how testable (or not) the product is...
...things that make testing harder or slower...
...what you need and what you recommend...

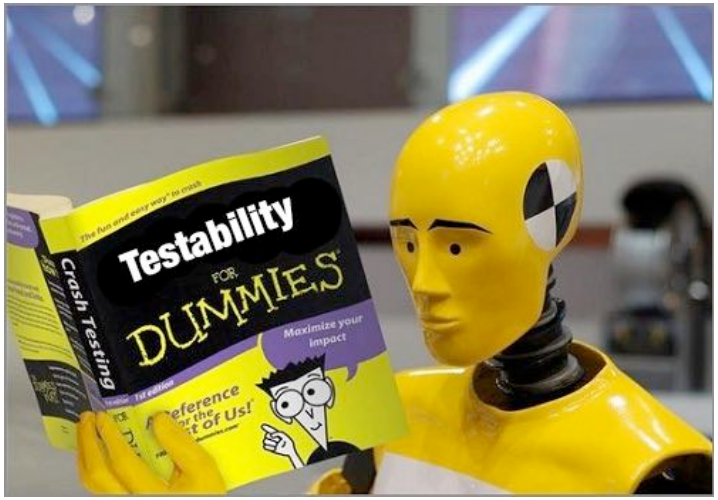


Source: Rapid Software Testing – James Bach & Michael Bolton

1. Testing is way faster, easier and cheaper with high testability! And more fun!
2. Everybody in the team benefits from high testability!
3. Testability is a team responsibility.
4. Make testability a subject for sprint planning & retrospective.

Read more

- Rapid Software Testing - http://www.satisfice.com/info_rst.shtml
- Testing & Checking - http://www.satisfice.com/blog/archives/856
  Test Cases Are Not Testing: Toward a Culture of Test Performance" by James Bach & Aaron Hodder - http://www.testingcircus.com/documents/TestingTrapeze-2014-February.pdf
- A test is a performance - http://www.satisfice.com/blog/archives/1346
- What makes agile testing different? - http://www.huibschoots.nl/wordpress/?p=1072
- Testability - http://www.satisfice.com/tools/testable.pdf
- Heuristics of Testability - http://oredev.org/2013/wed-fri-conference/heuristics-of-testability
- Testability awakens: moving testability into new dimensions - http://www.testingtrapezemagazine.com/wp-content/uploads/2015/12/TestingTrapeze-2015-December.pdf
- Visualizing Testability - http://youtu.be/_VR8naRfzK8
- More great resources: http://www.huibschoots.nl/links



huib.schoots@improveqs.nl
@huibschoots
www.huibschoots.nl/blog

IMPROVE
Quality Services