TRICENTIS

Exploratory Testing

Rediscover the art of exploratory testing

Ingo Philipp

CFTL

TRICENTIS

www.tricentis.com

TRICENTIS

E2E Test Automation	Load Testing	Test Management	Process Automation
TRICENTIS Tosca	TRICENTIS flood	TRICENTIS qTest	TRICENTIS RPA

Just as glasses do not see, **tools** do not test.
Glasses help us to see, tools help us to test

James Bach

Testing is no more about **test cases** than astronomy is about telescopes

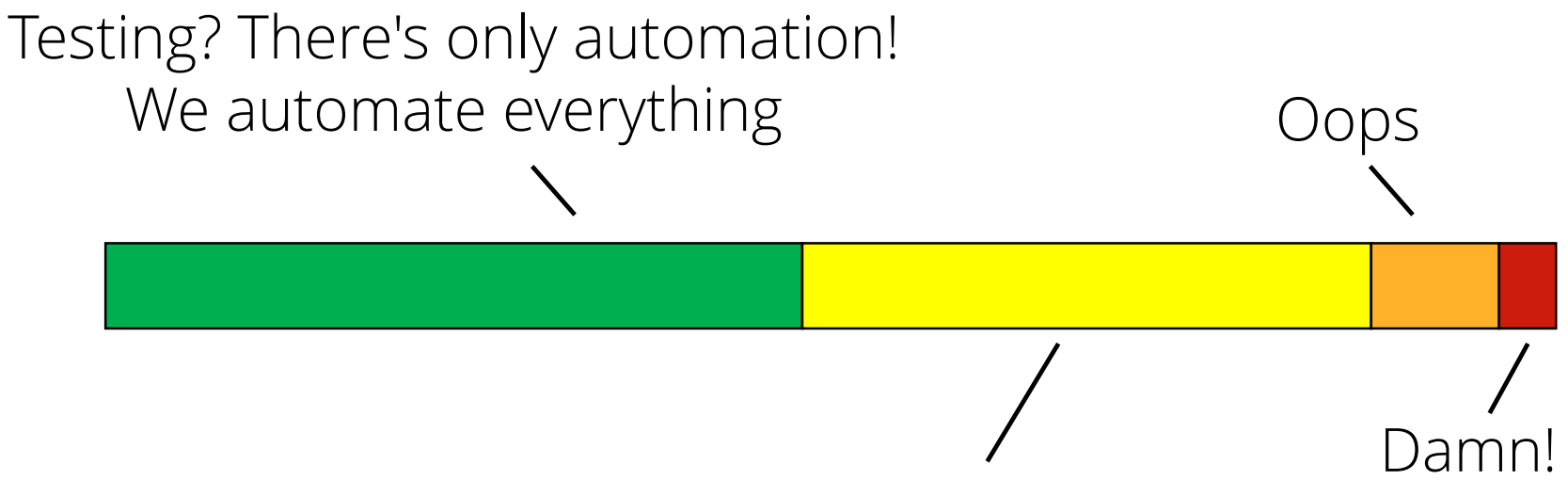
Ingo Philipp

:: Inspired by **Edsger Wybe Dijkstra** & **Michael Bolton**

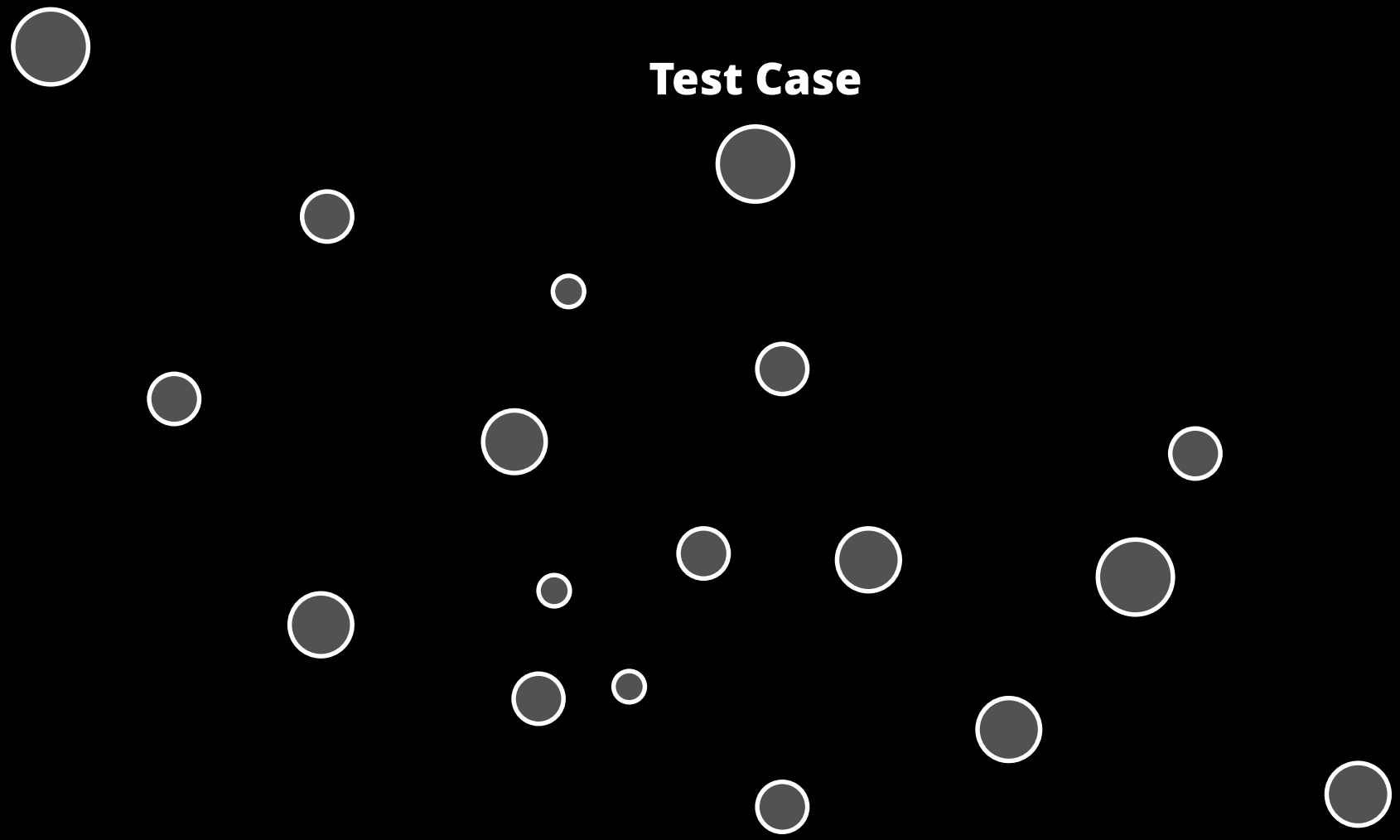
If we're testing badly, then **automation** can help us to test badly at an accelerated rate

Michael Bolton

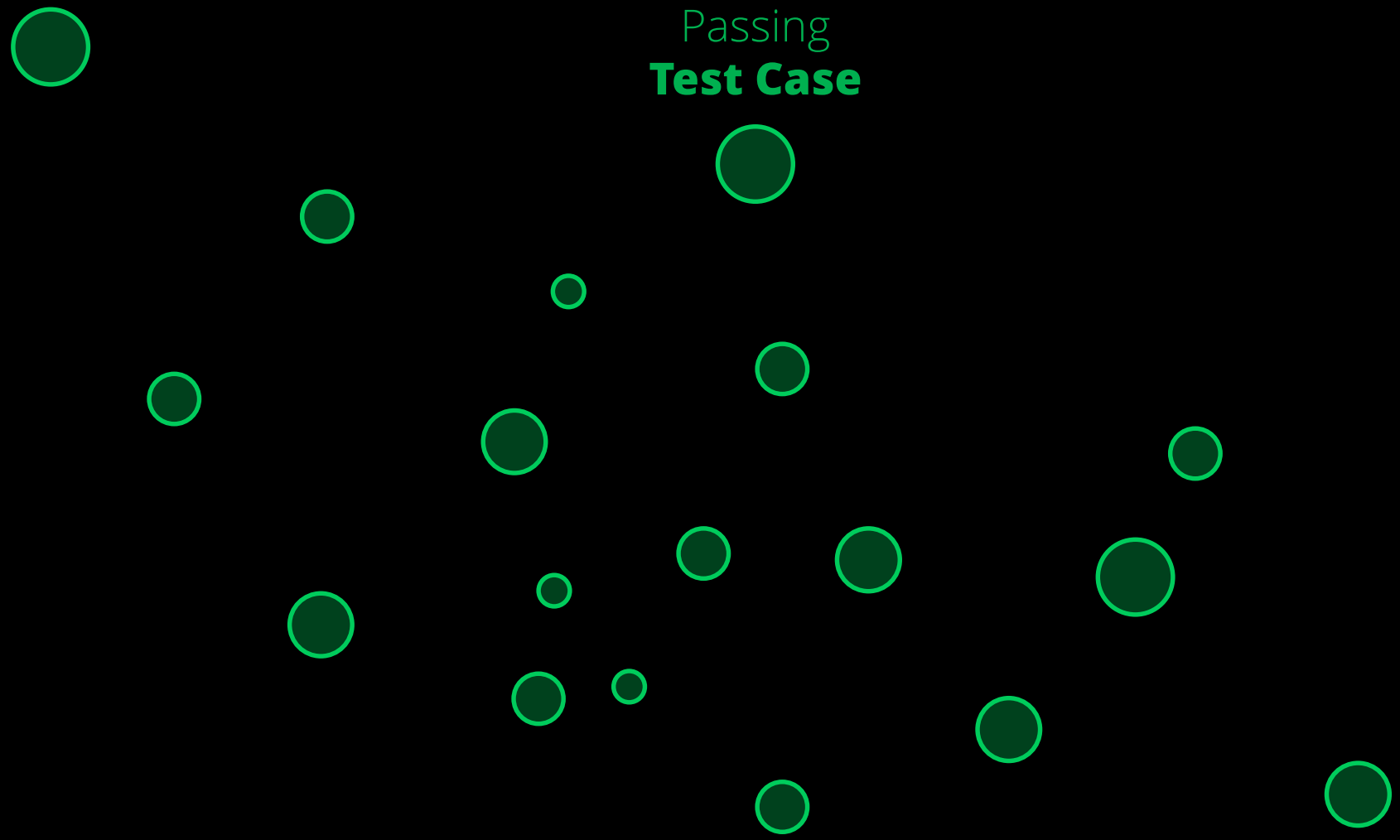
SOFTWARE TESTING TIMELINE



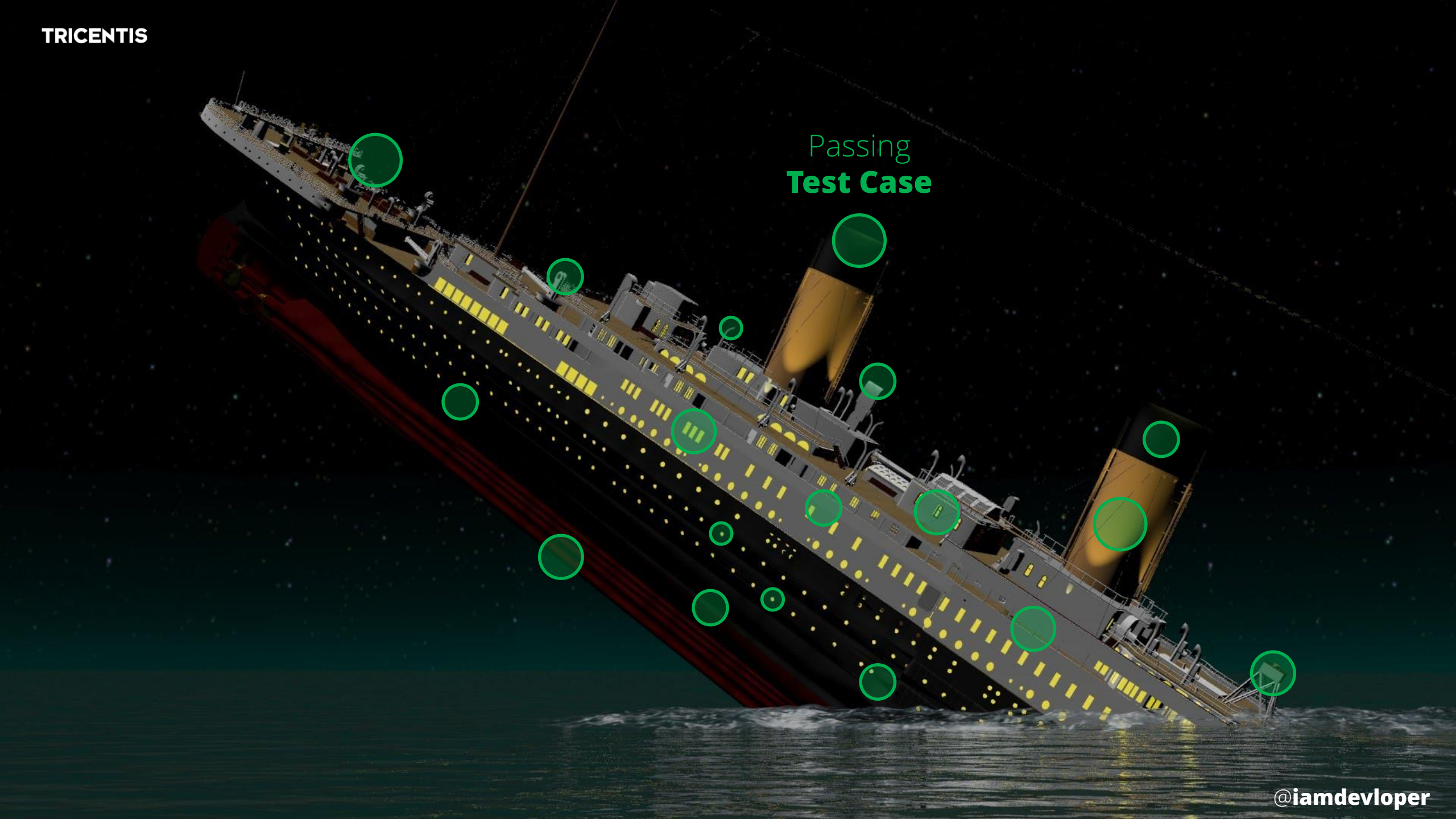
:: Inspired by **Brendan Leonard**


Test Case

Passing
Test Case



TRICENTIS



@iamdevloper


TRICENTIS
© BBC

Testing is exactly like **washing** a pig. Because it's messy. It has no rules. No clear beginning, middle, or end. It's kind of a pain in the ass, and when you're done you're not sure if the pig is really clean or even why you were washing a pig in the first place.

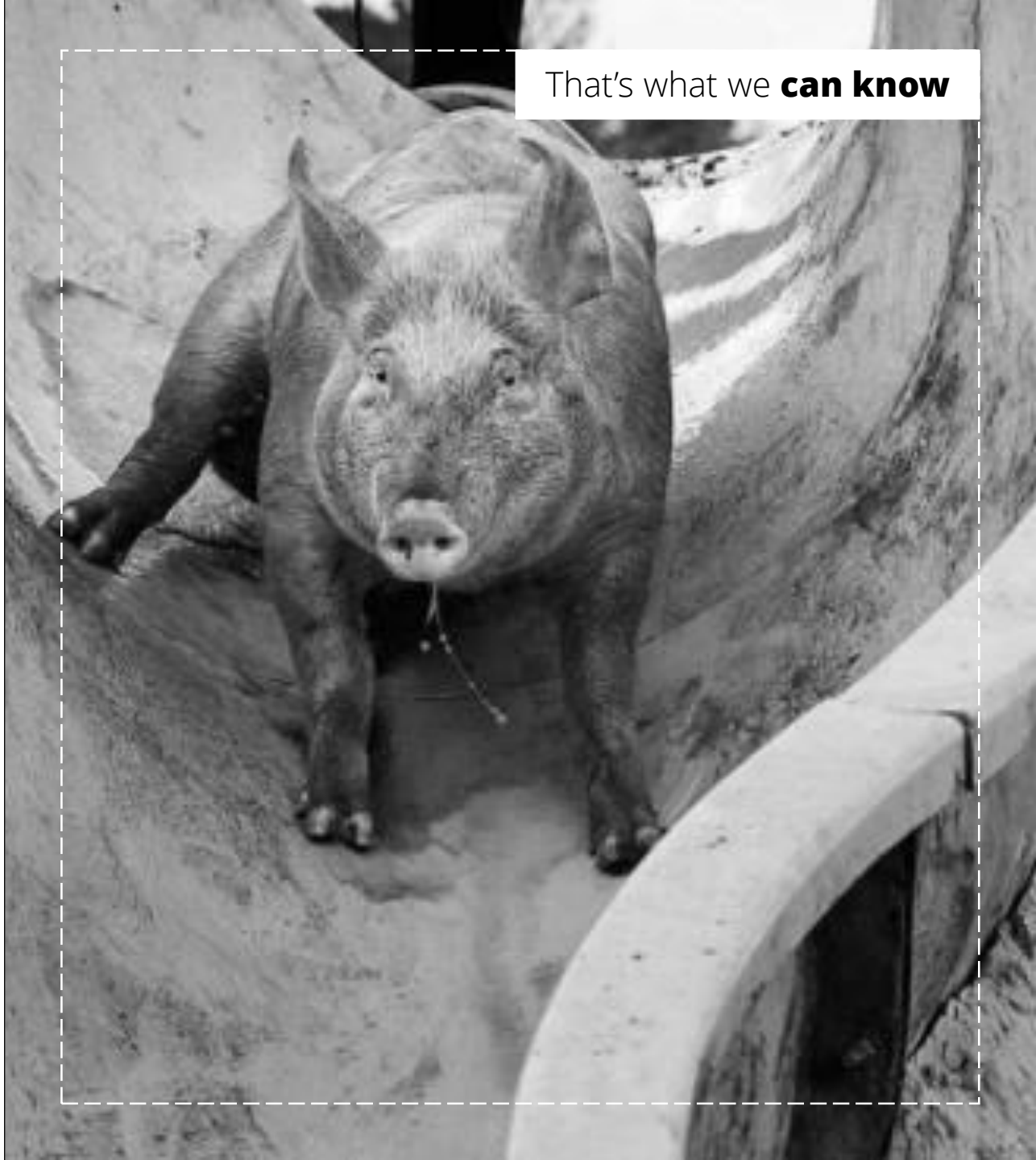
:: Inspired by **Luke Sullivan**

Testing is exactly like **washing** a pig. Because it's messy. It has no rules. No clear beginning, middle, or end. It's kind of a pain in the ass, and when you're done you're not sure if the pig is really clean or even why you were washing a pig in the first place.

:: Inspired by **Luke Sullivan**

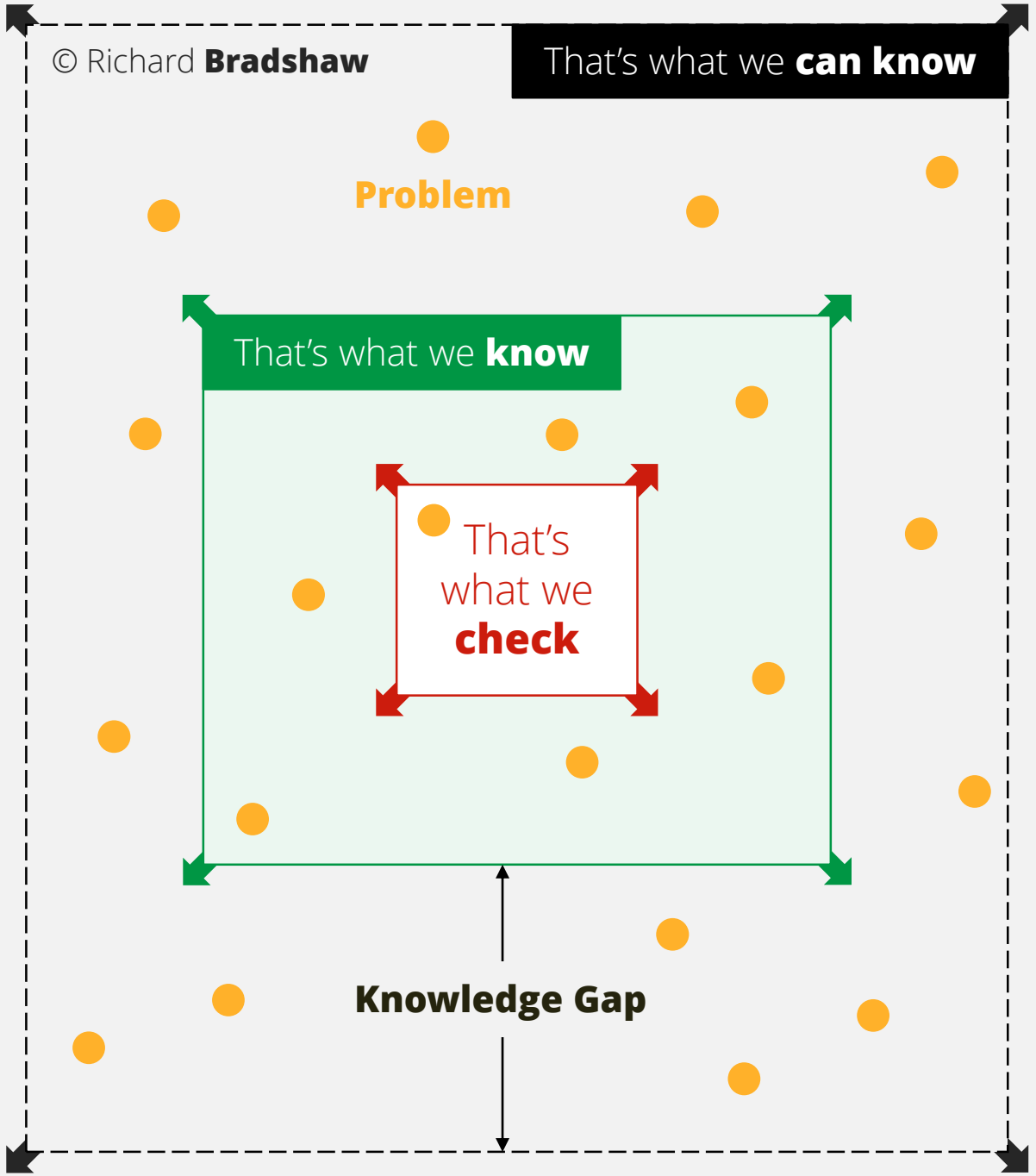
Testing is exactly like **washing** a pig. Because it's messy. It has no rules. No clear beginning, middle, or end. It's kind of a pain in the ass, and when you're done you're not sure if the pig is really clean or even why you were washing a pig in the first place.

:: Inspired by **Luke Sullivan**



Testing is exactly like **washing** a pig. Because it's messy. It has no rules. No clear beginning, middle, or end. It's kind of a pain in the ass, and when you're done you're not sure if the pig is really clean or even why you were washing a pig in the first place.

:: Inspired by **Luke Sullivan**



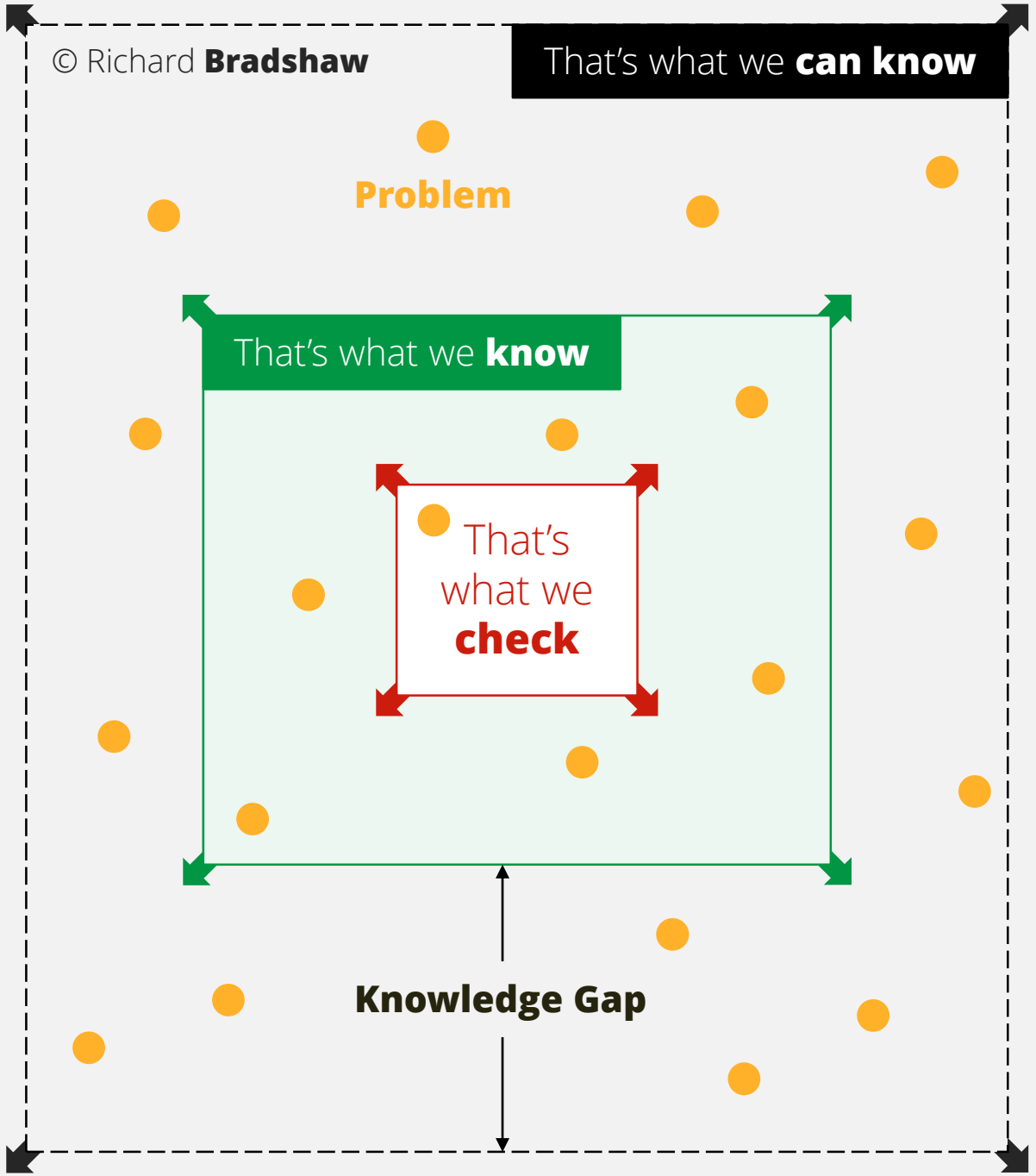
The purpose of testing is to close the **knowledge** gap

✸

The goal is **information**, not gratuitous automation

✸

Testing is and always will be a **search** for information



Under

Confirmation

« Demonstrate your **depth** of knowledge »

Over

Exploration

« Demonstrate your **breadth** of knowledge »

TRICENTIS

Change Detector

▲

Low Information Value

Here we check what we have already learned

Monitor Known Risks

Here we confirm what we already know

Mechanical Testing

Here we process predesigned data in predefined steps

▲

Confirmation

« Demonstrate your **depth** of knowledge »

Problem Detector

▲

High Information Value

Here we learn something new about software

Analyze Potential Risks

Here we focus on the things we don't know

Creative Testing

Here we create new test ideas

▲

Exploration

« Demonstrate your **breadth** of knowledge »

:: **Bolton** ‹› **Kaner** ‹› **Bach**

TRICENTIS

Change Detector

Does this assertion pass or fail?

Evaluate a product by applying **algorithmic** decision rules to specific observations of a product

▲

Checking

« Requires **Processing** »

Problem Detector

▲

High Information Value

Here we learn something new about software

Analyze Potential Risks

Here we focus on the things we don't know

Creative Testing

Here we create new test ideas

▲

Exploration

« Demonstrate your **breadth** of knowledge »

:: **Bolton** ‹› **Kaner** ‹› **Bach**

TRICENTIS

Change Detector

Does this assertion pass or fail?

Evaluate a product by applying **algorithmic** decision rules to specific observations of a product

▲

Checking

« Requires **Processing** »

Problem Detector

Is there a problem here?

Evaluate a product by learning about it through **exploration** and experimentation

▲

Exploring

« Requires **Thinking** »

:: **Bolton** ‹› **Kaner** ‹› **Bach**

TRICENTIS

Change Detectors ‹› **Monitor Known Risks**

Evaluate through **Instructions**	Pay attention to **Deviations**
Create **Test Cases**	Follow **Procedure**
Examine **Requirements**	Factory **Process**

Checking

« Requires **Processing** »

:: Rich **Rogers**

Problem Detector

Is there a problem here?

Evaluate a product by learning about it through **exploration** and experimentation

Exploring

« Requires **Thinking** »

TRICENTIS

Change Detectors ‹› **Monitor Known Risks**

Evaluate through **Instructions**	Pay attention to **Deviations**
Create **Test Cases**	Follow **Procedure**
Examine **Requirements**	Factory **Process**

Checking

« Requires **Processing** »

Problem Detectors ‹› **Analyze Potential Risks**

Investigate through **Experiments**	Pay attention to **Oracles**
Create **Test Ideas**	Follow **Clues**
Examine **Risks**	Adaptive **Investigation**

Exploring

« Requires **Thinking** »

:: Rich **Rogers**

TRICENTIS

Change Detectors ‹› **Monitor Known Risks**

Mechanical Process

Evaluate through **Instructions**	Pay attention to **Deviations**
Create **Test Cases**	Follow **Procedure**
Examine **Requirements**	Factory **Process**

Checking

« Requires **Processing** »

Robots

Problem Detectors ‹› **Analyze Potential Risks**

Creative Process

Investigate through **Experiments**	Pay attention to **Oracles**
Create **Test Ideas**	Follow **Clues**
Examine **Risks**	Adaptive **Investigation**

Exploring

« Requires **Thinking** »

Investigators

:: Rich **Rogers**

Behind a **test case**, there's a **test**. Behind a test, there's a **test idea**. Behind a test idea, there's a **human tester**

Ingo Philipp

:: Just as you don't blame gravity for falling flat on your face, you shouldn't blame your tools for your poor test cases.

Checking is a guest in the house of testing,
exploration is a permanent resident

Michael Bolton

Agile
Testing Equation

:: Elisabeth **Hendrickson**

Checked

Machine & Human Checking

+

Explored

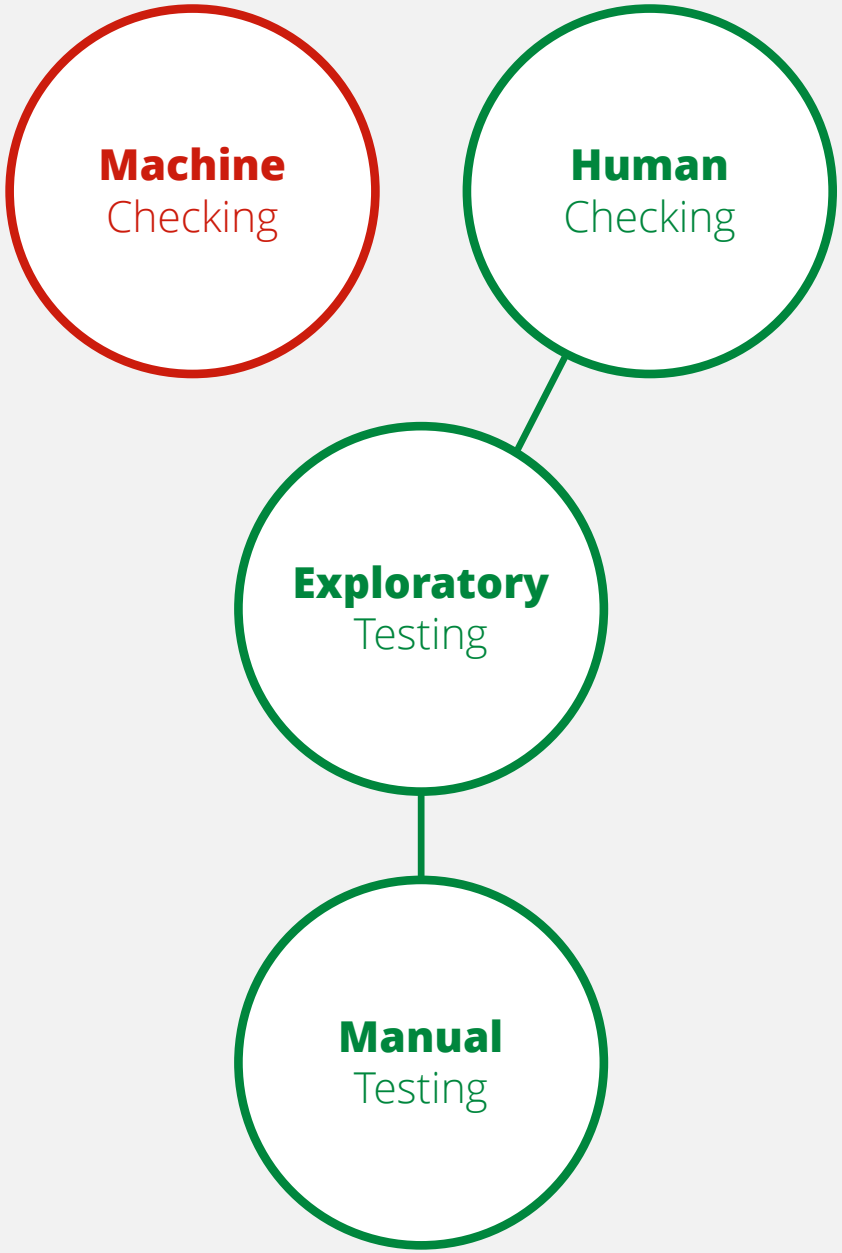
Human Exploration

=

Tested

Thorough Testing

TRICENTIS



Checked

Machine & Human Checking

+

Explored

Human Exploration

=

Tested

Thorough Testing

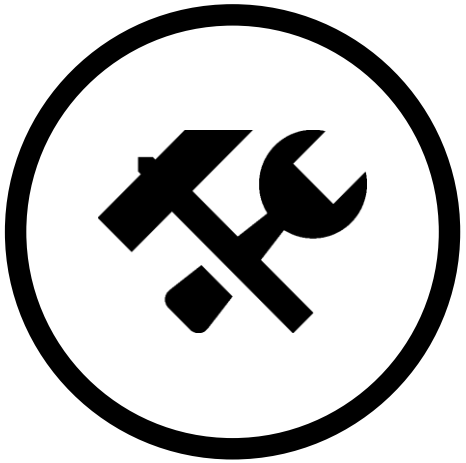
:: Picture by **Shelly Palmer**

Exploratory testing is not a talent,
it's a set of **skills** that can be learnt

Ingo Philipp

:: Inspired by **Edward de Bono**

TRICENTIS

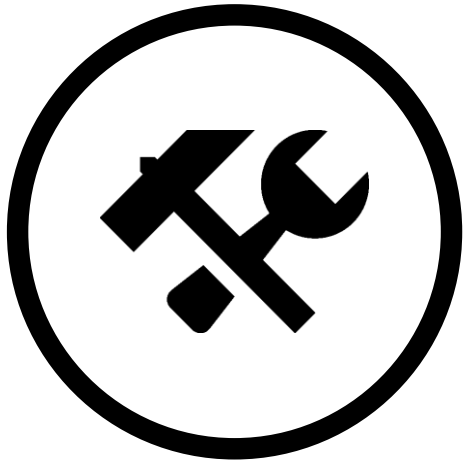


Technique

Provides Systematic Procedure

Approach

Provides Orientation



Technique

Provides Systematic Procedure

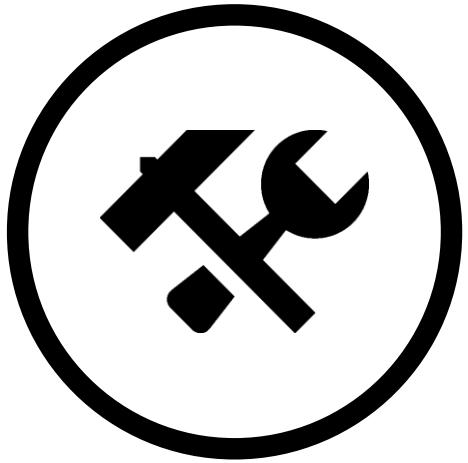
1

2

3

4

5



Technique

Provides Systematic Procedure

Session-Based Testing

Structure exploratory testing to allow large-scale implementations

2

3

4

5

Straightjacketed **Imagination**

Session-Based Testing

Structure exploratory testing to allow large-scale implementations

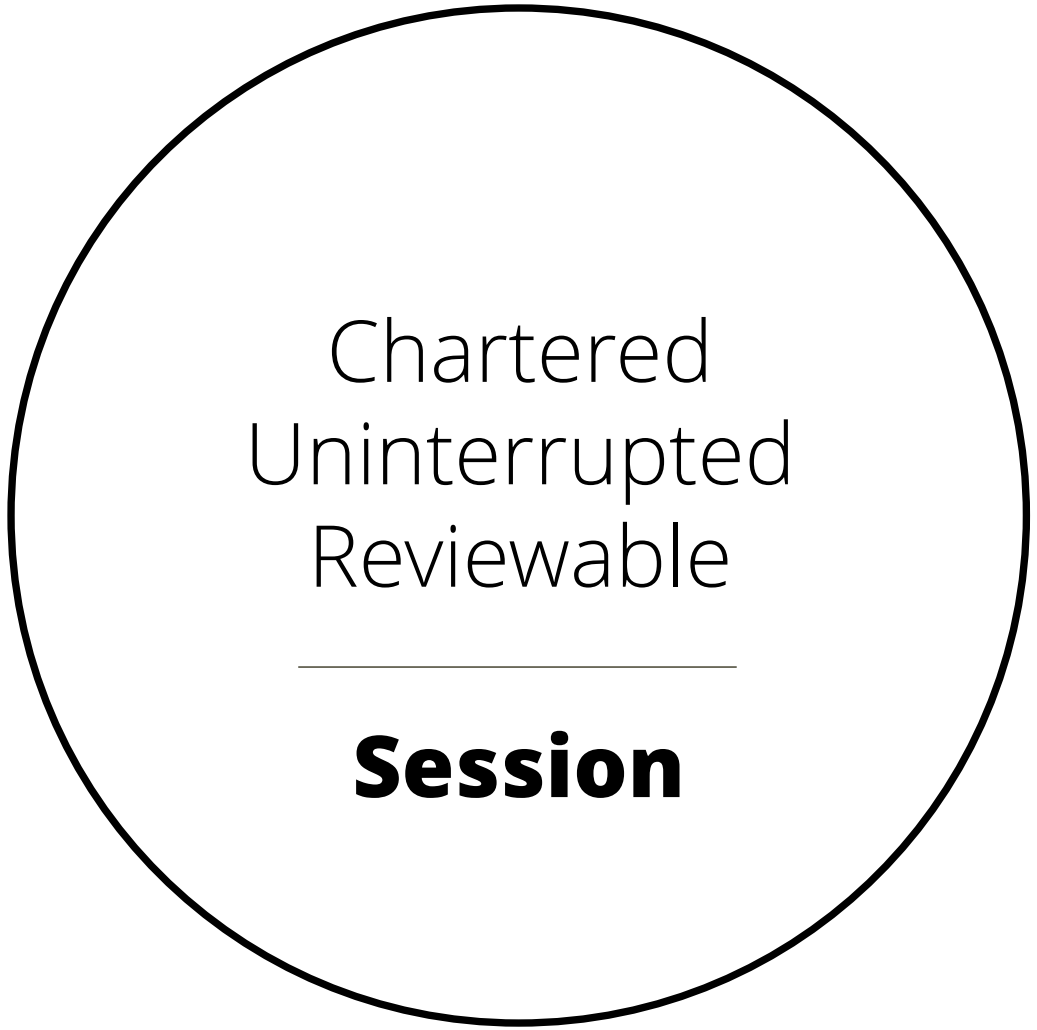
2

3

4

5

TRICENTIS



Session-Based Testing

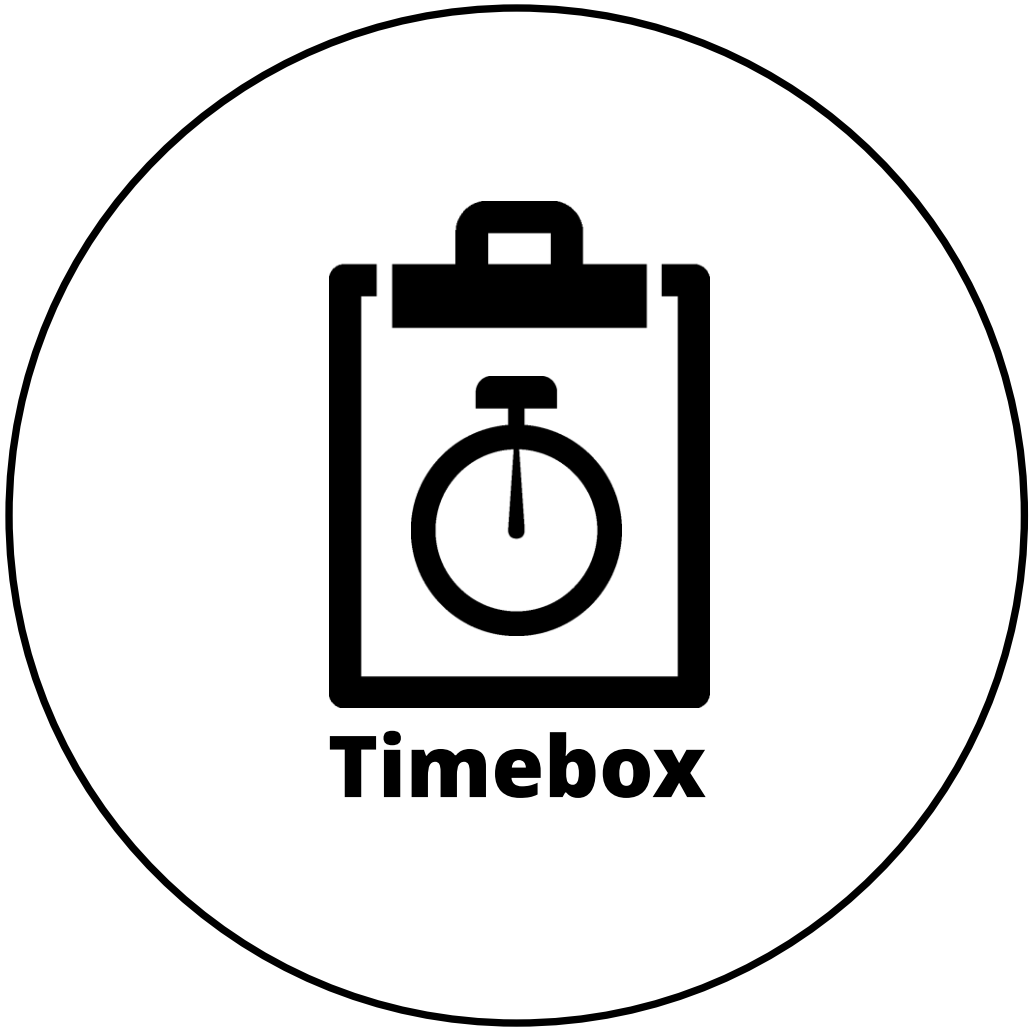
Structure exploratory testing to allow large-scale implementations

2

3

4

5



Session-Based Testing

Structure exploratory testing to allow large-scale implementations

2

3

4

5

Session-Based Testing

Structure exploratory testing to allow large-scale implementations

2

3

4

5

TRICENTIS

Session-Based Testing

Structure exploratory testing to allow large-scale implementations

Requirements-Based Testing

Limit the scope to make it manageable

3

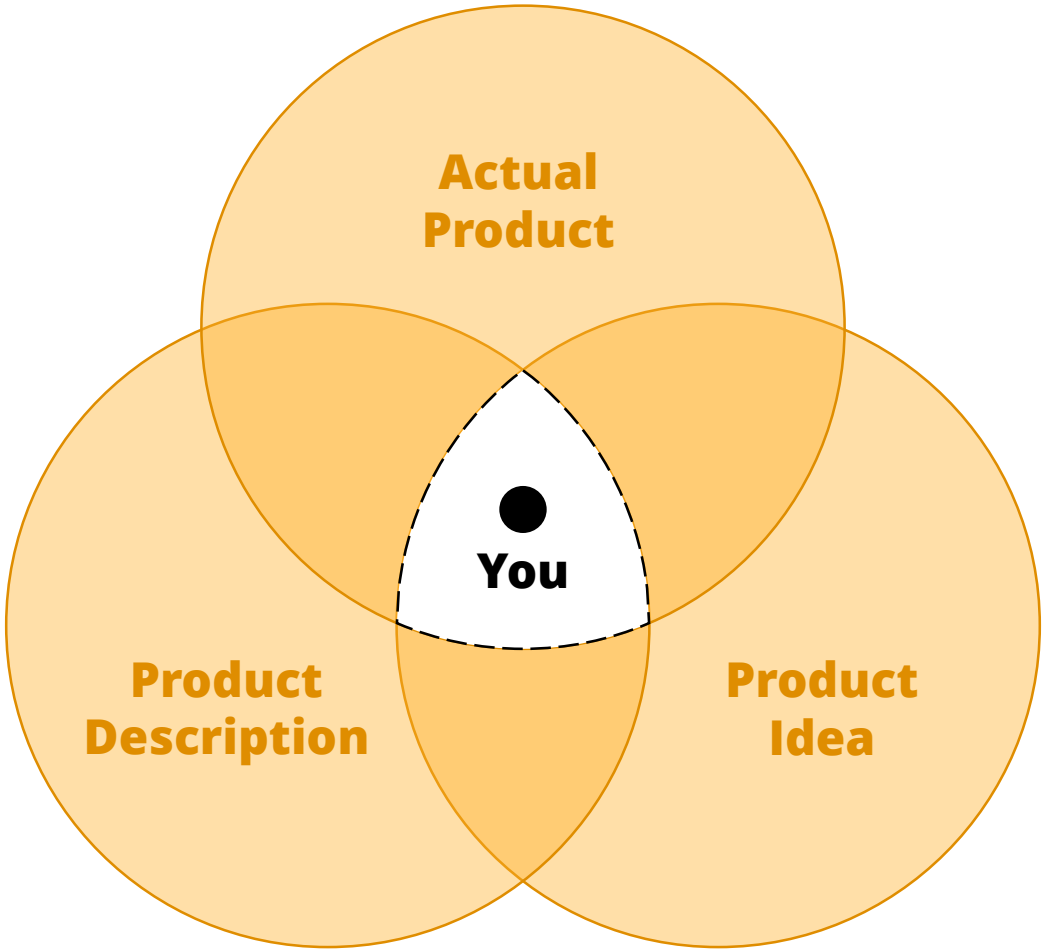
4

5

TRICENTIS

Validation

Does our software meets the user's needs?



Verification

Does our software meet the specification?

Session-Based Testing

Structure exploratory testing to allow large-scale implementations

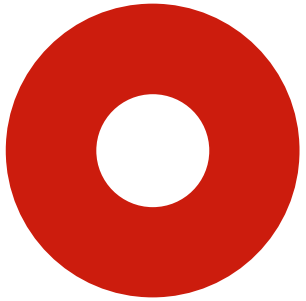
Requirements-Based Testing

Limit the scope to make it manageable

3

4

5



BDD is the art of using examples in conversations to illustrate behavior

Liz Keogh

Session-Based Testing

Structure exploratory testing to allow large-scale implementations

Requirements-Based Testing

Limit the scope to make it manageable

3

4

5

Having
Conversations

is more important than

Capturing
Conversations

is more important than

Automating
Conversations

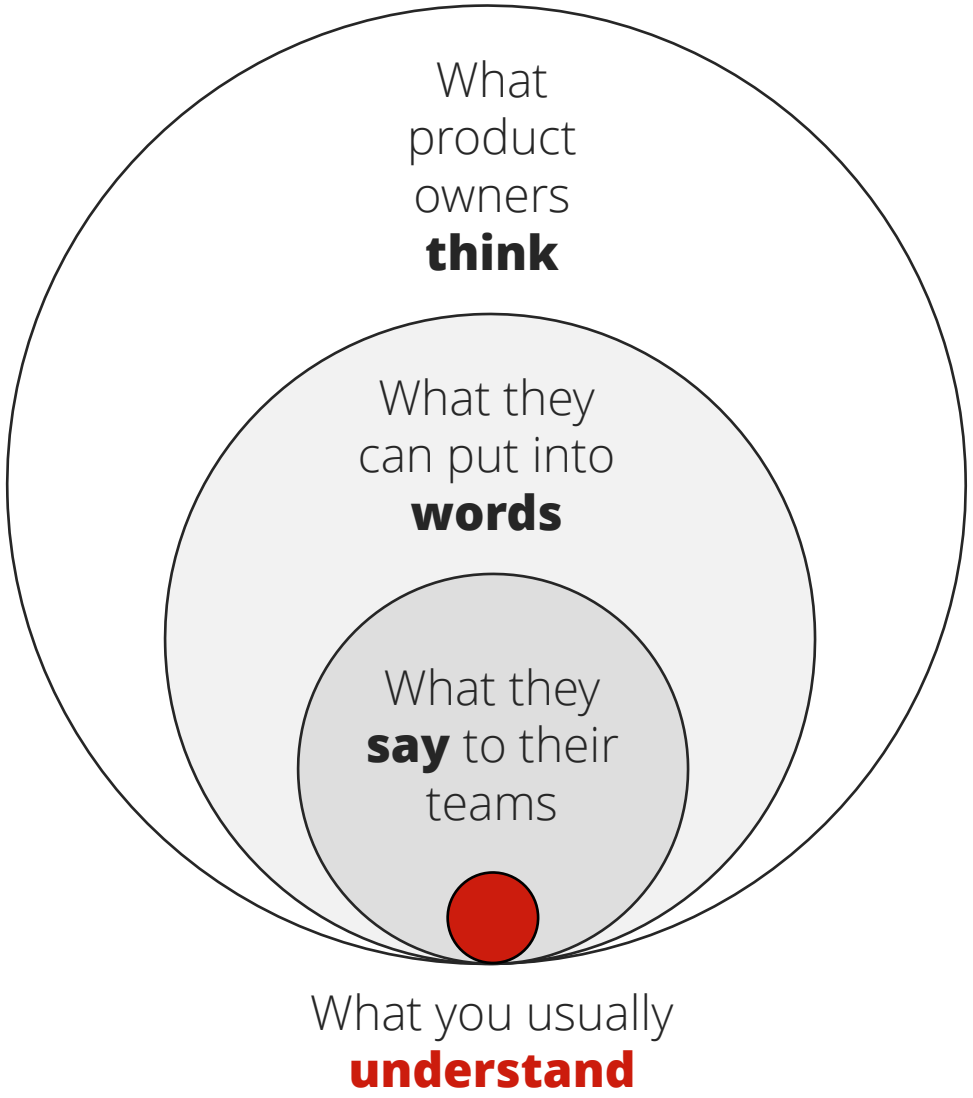
Session-Based Testing
Structure exploratory testing to allow large-scale implementations

Requirements-Based Testing
Limit the scope to make it manageable

3

4

5



Session-Based Testing

Structure exploratory testing to allow large-scale implementations

Requirements-Based Testing

Limit the scope to make it manageable

3

4

5

TRICENTIS

R » Recent
What parts of the product changed recently?

C » Core
What critical parts of the product must continue to work?

R » Risky
What parts of the product are inherently risky?

C » Configuration
What parts of the product depend on environment settings?

R » Repaired
What parts of the product changed to address defects?

C » Chronic
What parts of the product chronically break?

:: Karen **Johnson**

Session-Based Testing

Structure exploratory testing to allow large-scale implementations

Requirements-Based Testing

Limit the scope to make it manageable

3

4

5

TRICENTIS

S » Structure
Test what the product is made of.

F » Function
Test what the product does.

D » Data
Test what the product processes.

P » Platform
Test what the product depends upon.

O » Operations
Test how the product is used.

T » Time
Test how the product is affected by time.

:: James **Bach**

Session-Based Testing
Structure exploratory testing to allow large-scale implementations

Requirements-Based Testing
Limit the scope to make it manageable

3

4

5

TRICENTIS

:: James **Whittaker**

Session-Based Testing

Structure exploratory testing to allow large-scale implementations

Requirements-Based Testing

Limit the scope to make it manageable

Tour-Based Testing

Set concrete goals to provide a clear focus

4

5

@speed Quality

Quality is value to some person

:: Jerry **Weinberg**

Session-Based Testing

Structure exploratory testing to allow large-scale implementations

Requirements-Based Testing

Limit the scope to make it manageable

Tour-Based Testing

Set concrete goals to provide a clear focus

4

5

Quality is inherently **subjective**

Different stakeholders
will perceive the same product as having different levels of quality

We must look for **different things** for different stakeholders

We must **diversify** testing

:: Jerry **Weinberg**

Session-Based Testing
Structure exploratory testing to allow large-scale implementations

Requirements-Based Testing
Limit the scope to make it manageable

Tour-Based Testing
Set concrete goals to provide a clear focus

4

5

Session-Based Testing

Structure exploratory testing to allow large-scale implementations

Requirements-Based Testing

Limit the scope to make it manageable

Tour-Based Testing

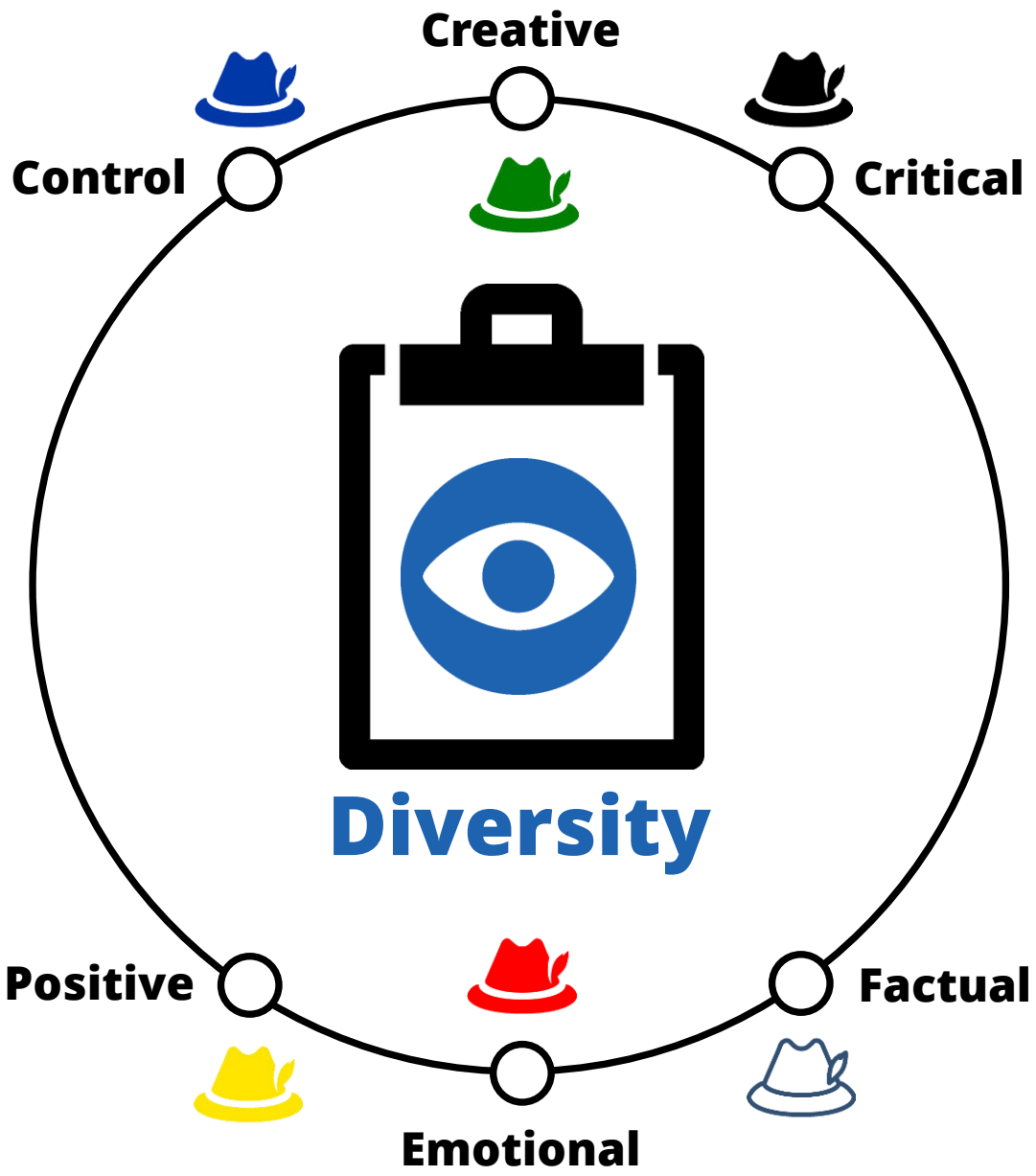
Set concrete goals to provide a clear focus

Polychrome Testing

Explore the product from different viewpoints to diversify testing

5

TRICENTIS



© Edward **De Bono**

Session-Based Testing

Structure exploratory testing to allow large-scale implementations

Requirements-Based Testing

Limit the scope to make it manageable

Tour-Based Testing

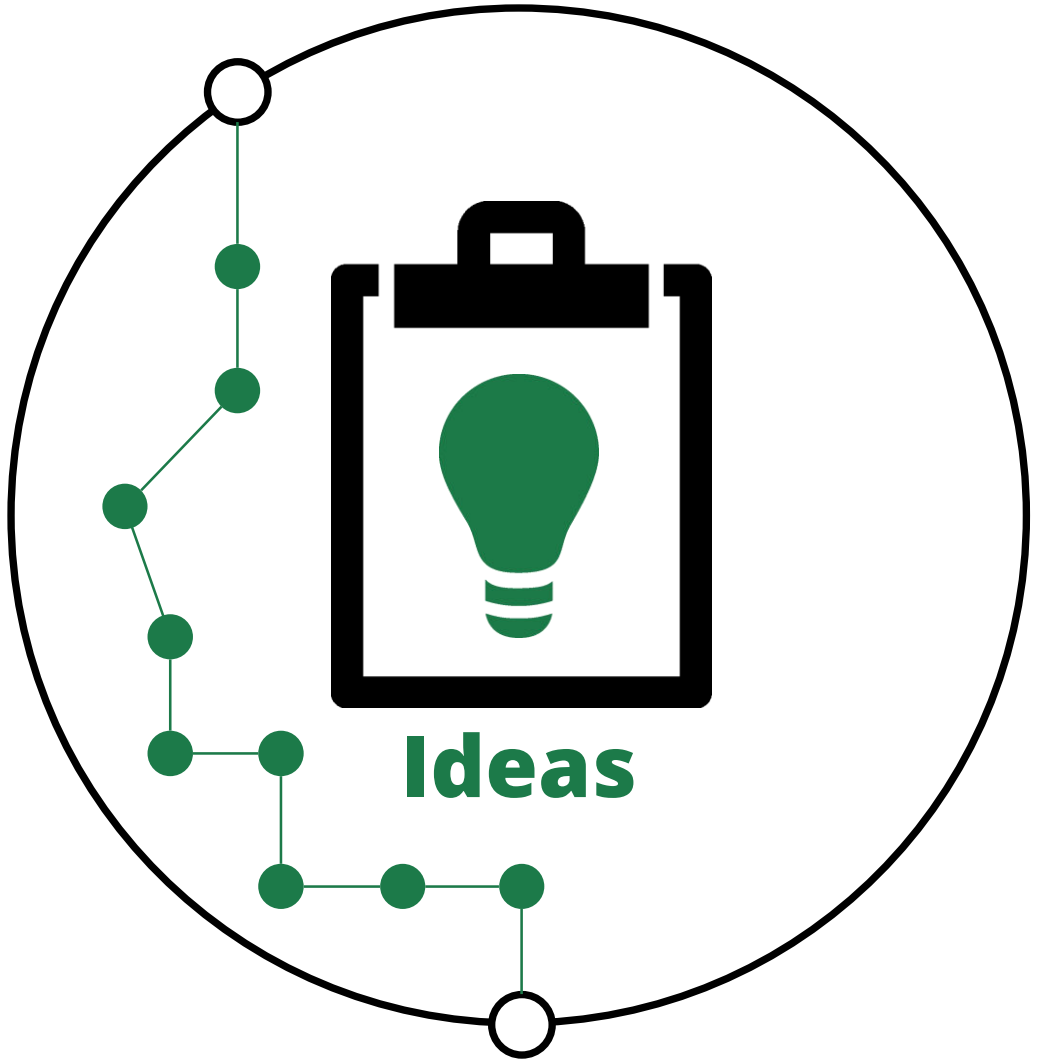
Set concrete goals to provide a clear focus

Polychrome Testing

Explore the product from different viewpoints to diversify testing

5

TRICENTIS



Session-Based Testing

Structure exploratory testing to allow large-scale implementations

Requirements-Based Testing

Limit the scope to make it manageable

Tour-Based Testing

Set concrete goals to provide a clear focus

Polychrome Testing

Explore the product from different viewpoints to diversify testing

Scenario-Based Testing

Capture each test idea to make it reviewable

Testing is not so much a thing you **do**,
it's far more a way you **think**

Michael Bolton

Questions

The show is **over**. It's your turn.