



# Reality Check

Does **manual testing** have a future in **DevOps**?

---

Ingo **Philipp**

# A



## Question

Why do we even ask this question?

# B



## Answer

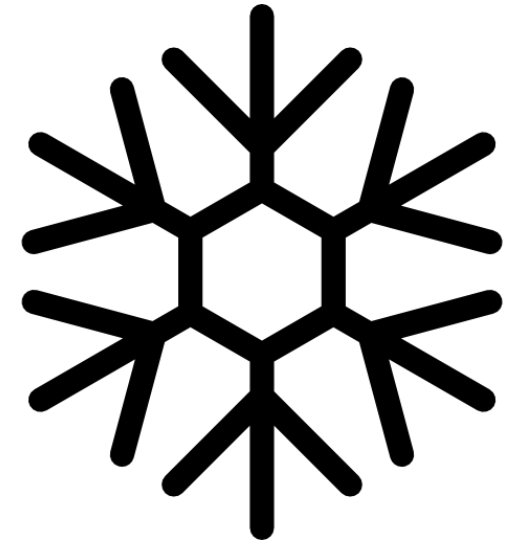
How to properly answer this question?

# C



## Discussion

Questions & Answers



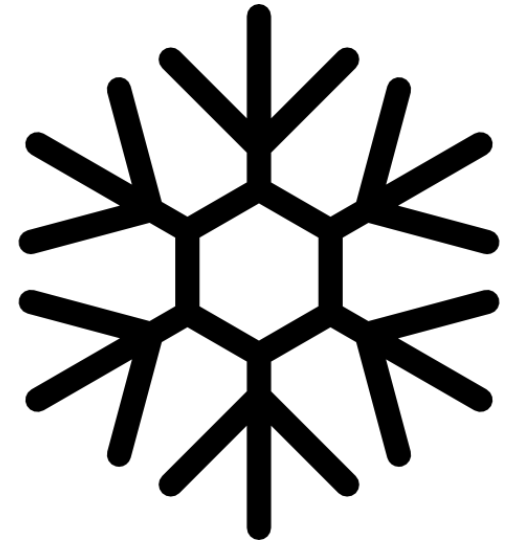
**Harvest**

**1.0**

TRICENTIS



► About **30 years** later



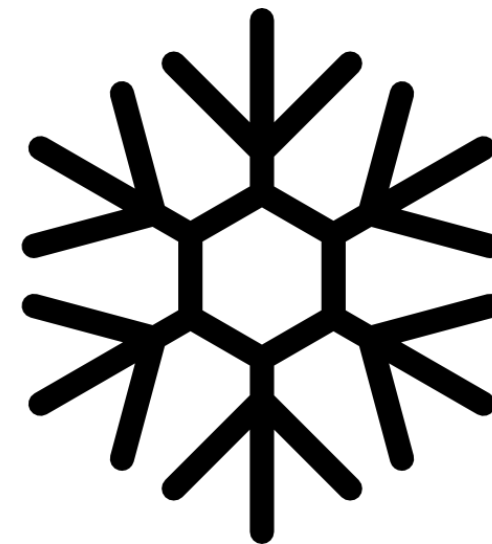
**Factory**  
**2.0**



TRICENTIS

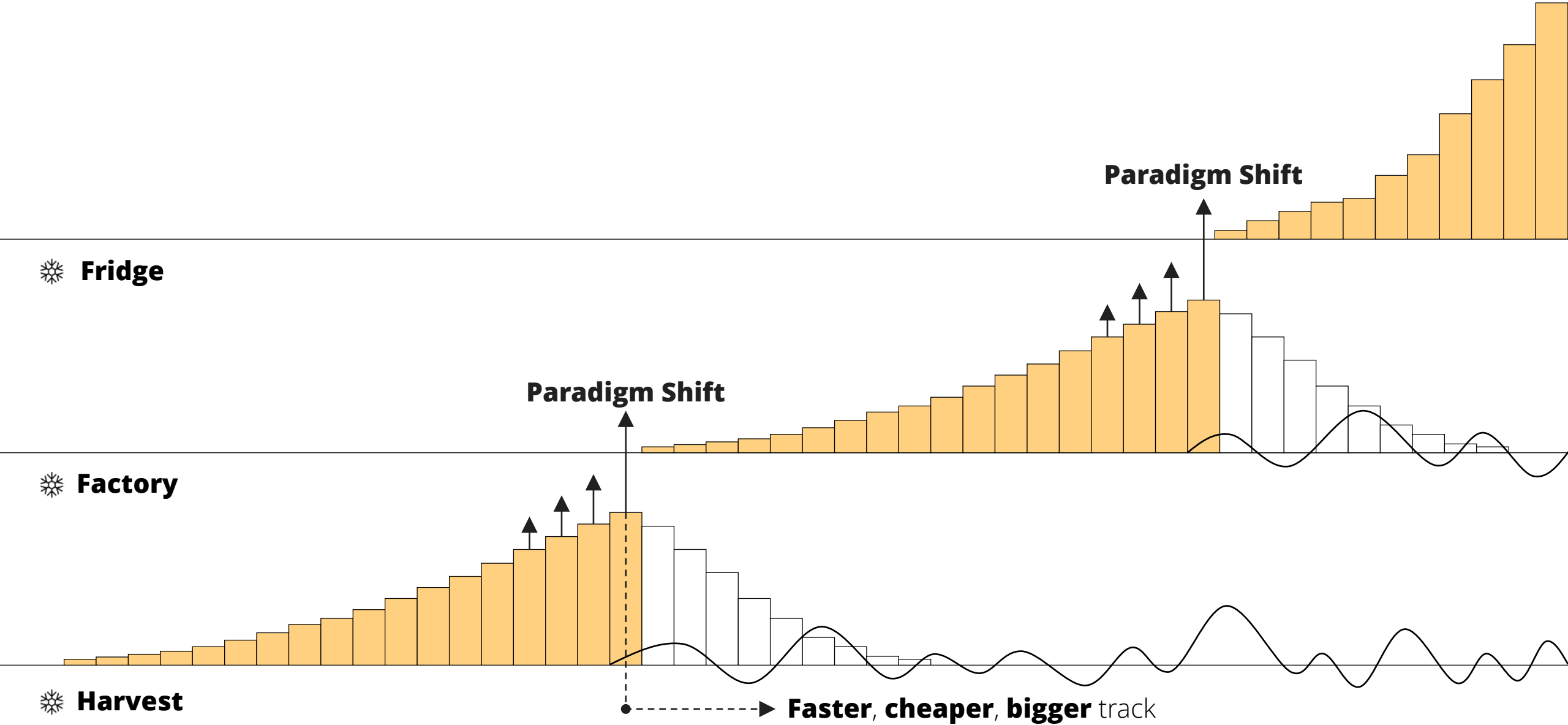


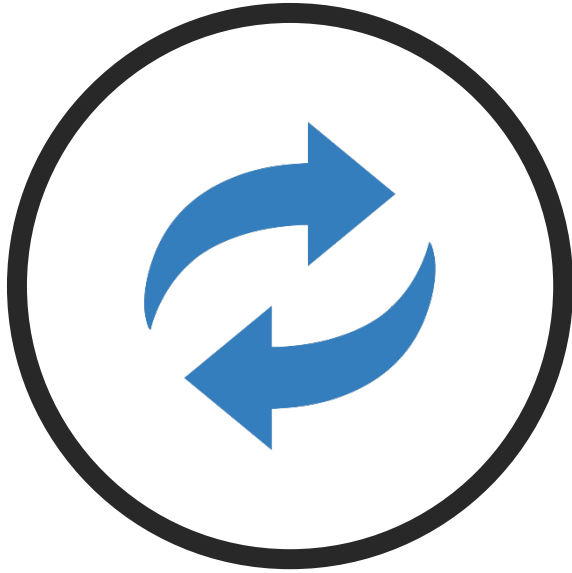
▶ Another **30 years** later



**Fridge**

**3.0**





# Pursue **Change**

Our only thing constant



# Make **Meaning**

Basis to shift the paradigm



Democratize **Computers**



Democratize **Information**



Democratize **Commerce**

**TRICENTIS**

Democratize **Test Automation**



Make **Meaning**

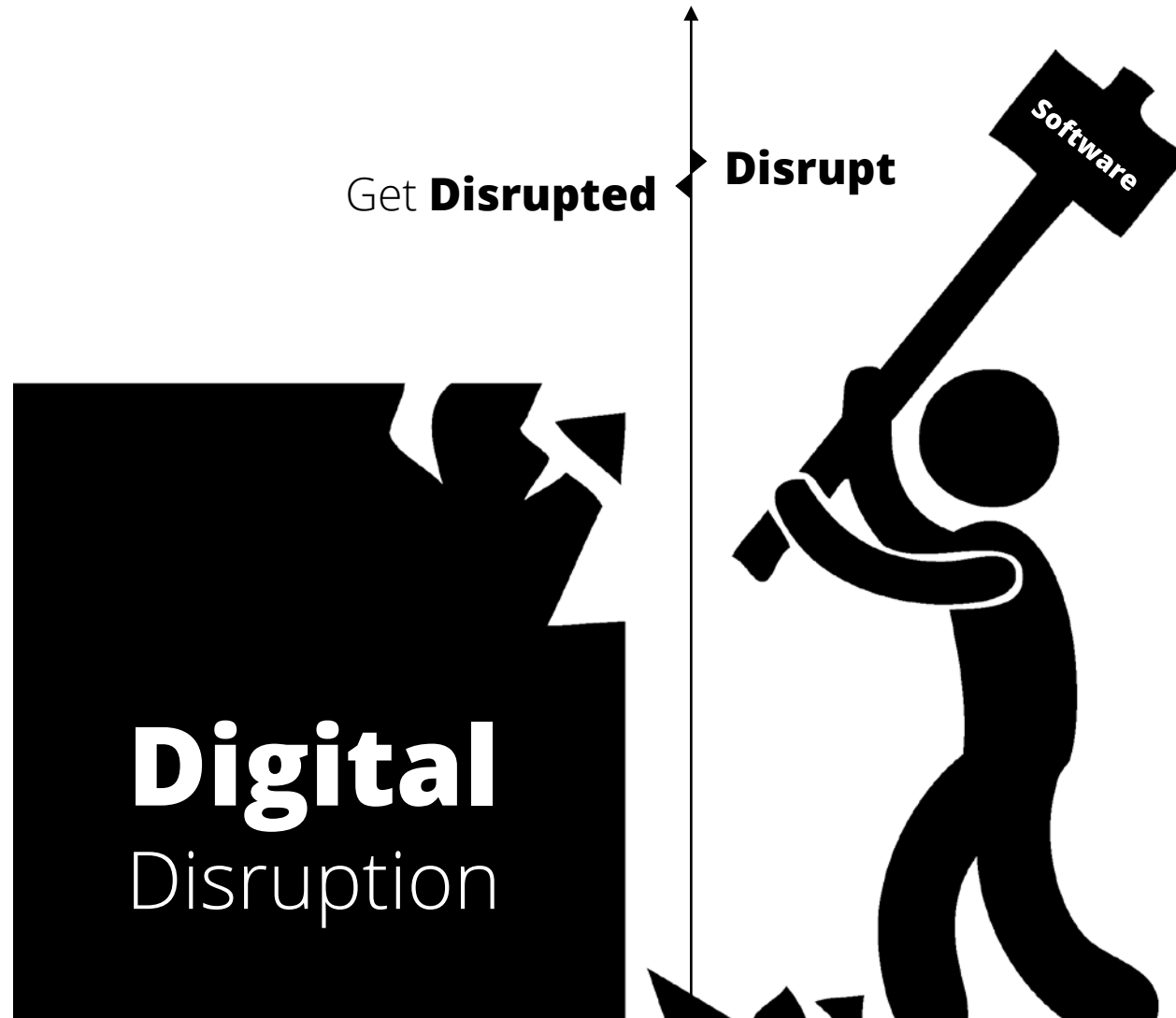
Basis to shift the paradigm





"Businesses must continuously exploit digital technologies to both create **new sources** of **customer value** and increase **operational agility** in service of customers."

Across industries, companies face the challenge of **software-led transformation**



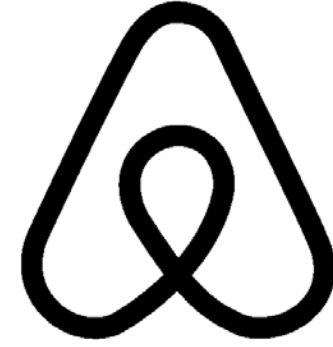


UBER



Owens **no vehicles**

Everyone can be a taxi driver



airbnb

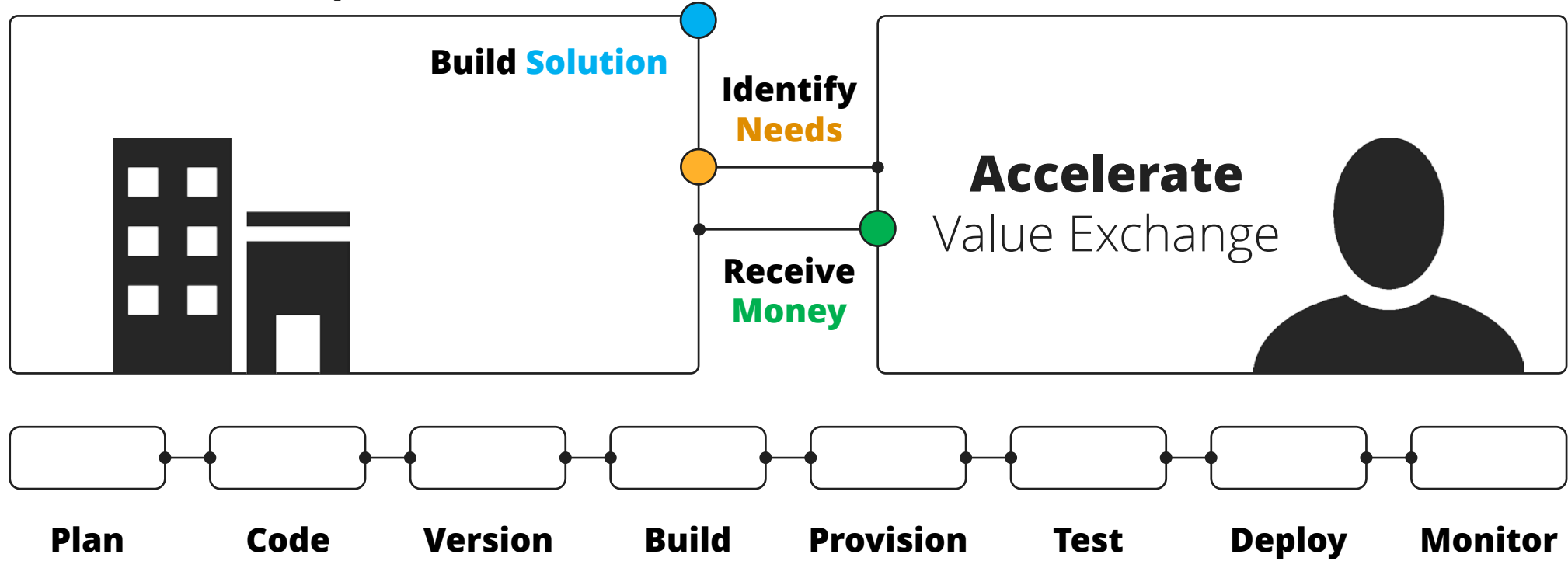


Owens **no real estate**

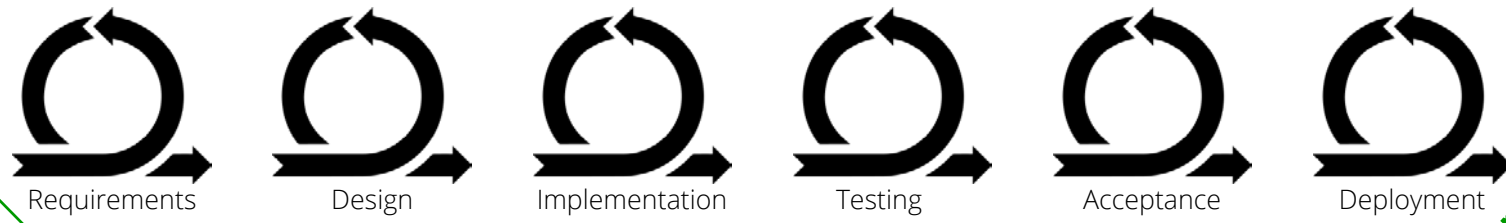
Everyone can be a landlord

Their value is all digital. **Digitalization** creates new ways of **business operations**, and **customer interactions**

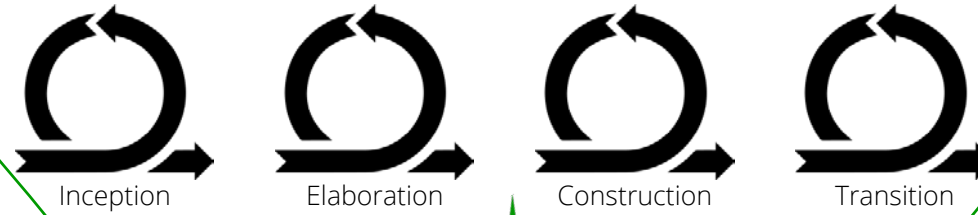
Fail **Fast** » Fail **Cheap** » Fail **Forward**



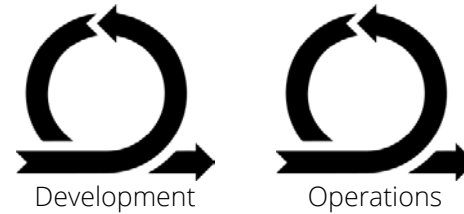
- 6+**  
**Big Bang**  
Waterfall
- 4**  
**Incremental**  
Rational Unified Process
- 2**  
**Agile**  
Scrum, Kanban
- 1**  
**Continuous**  
DevOps



**Revolution**



**Agile** Event Horizon



**Evolution**



Digital **Singularity**

**time to react on change**

Following a Plan  
Contract Negotiation  
Copious Documentation  
Processes & Tools

**Rigid Rules**

**Flexible Framework**

Individuals & Interactions  
Working Software  
Customer Collaboration  
Responding to Change

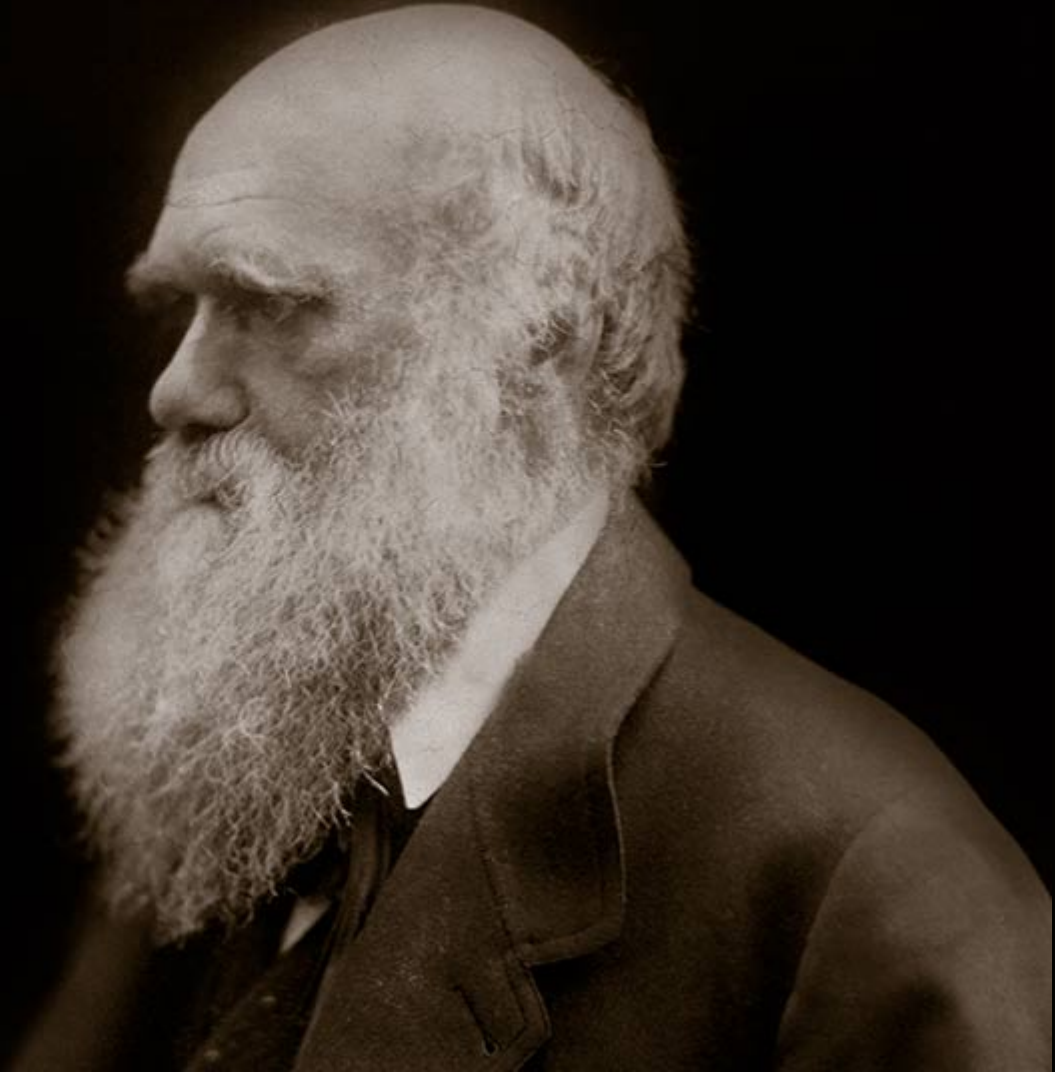


It's not the strongest that  
survive, nor the most  
intelligent, but the one **most**  
**responsive** to change

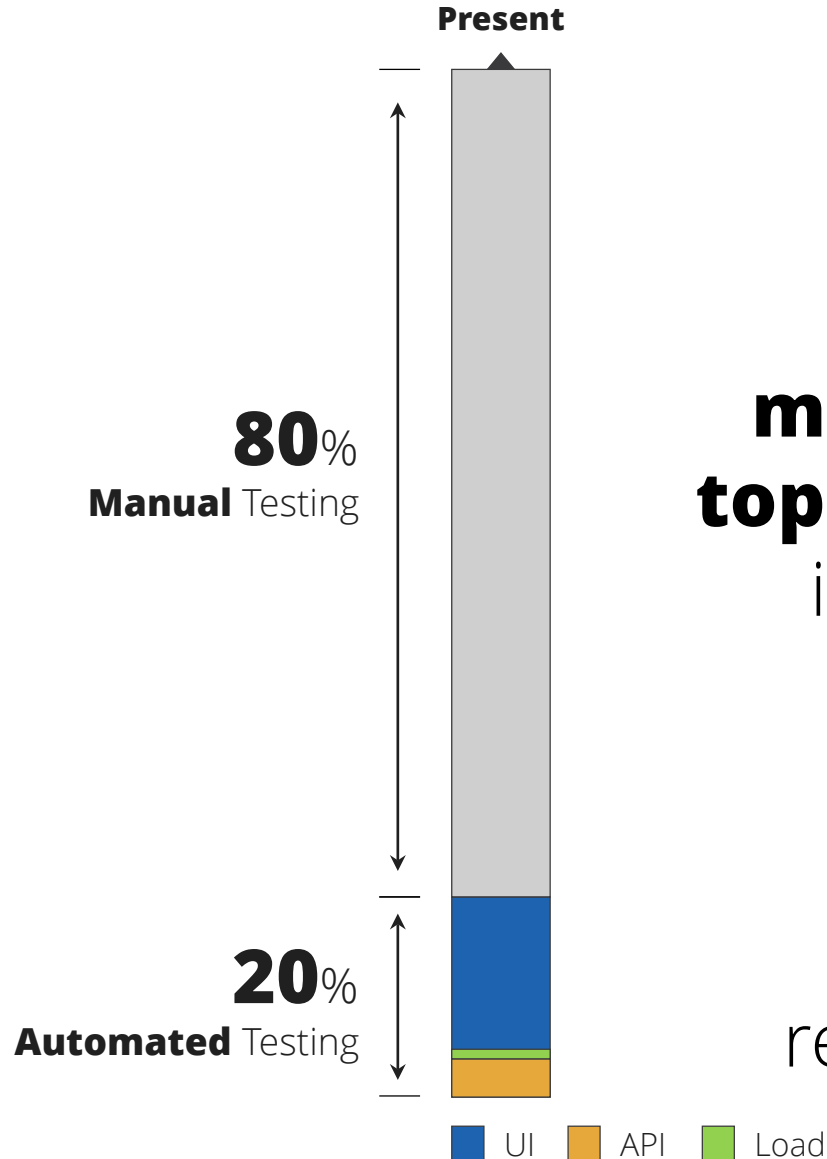
---

Charles **Darwin**

★ Because software is not written, it is **rewritten**







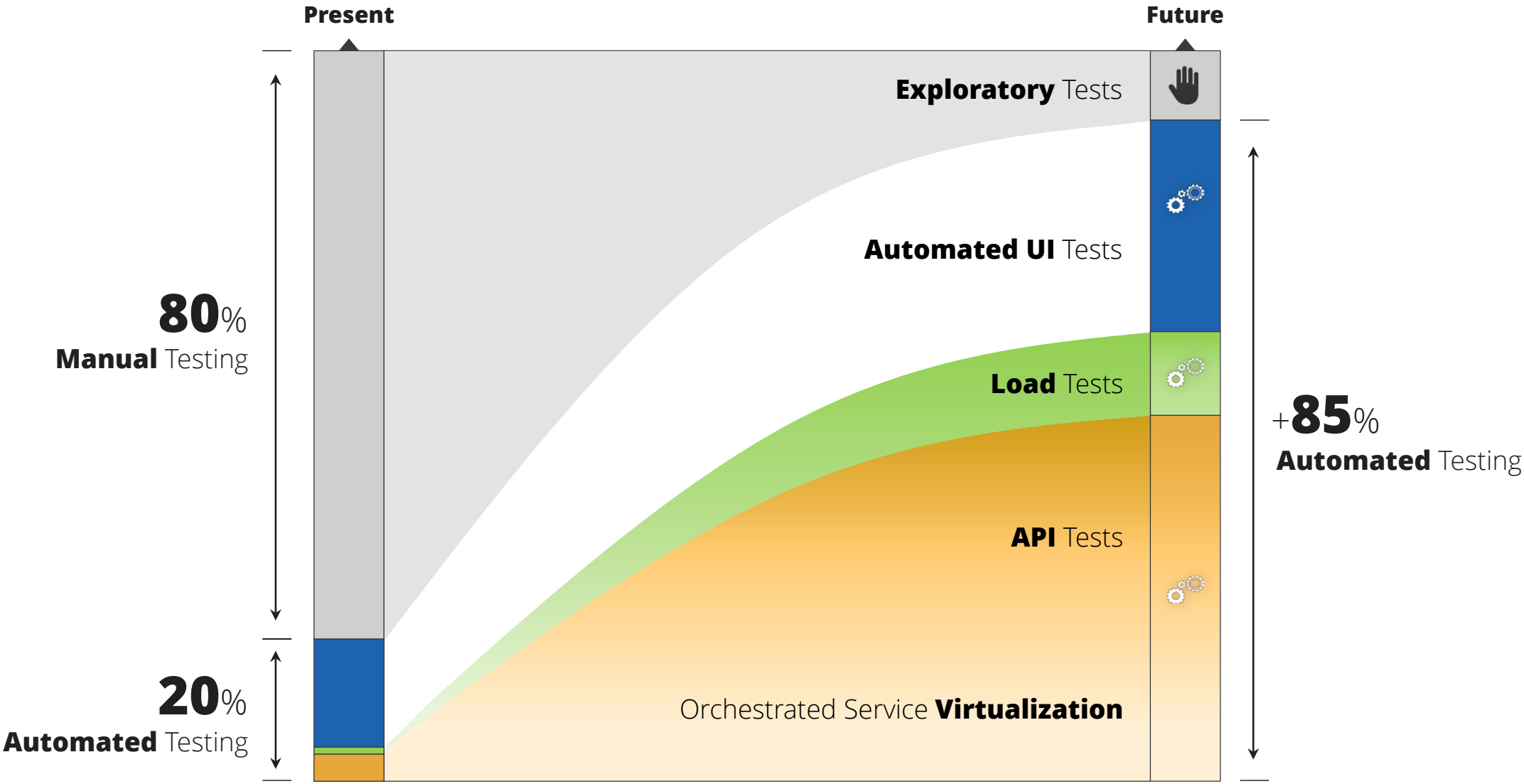
★

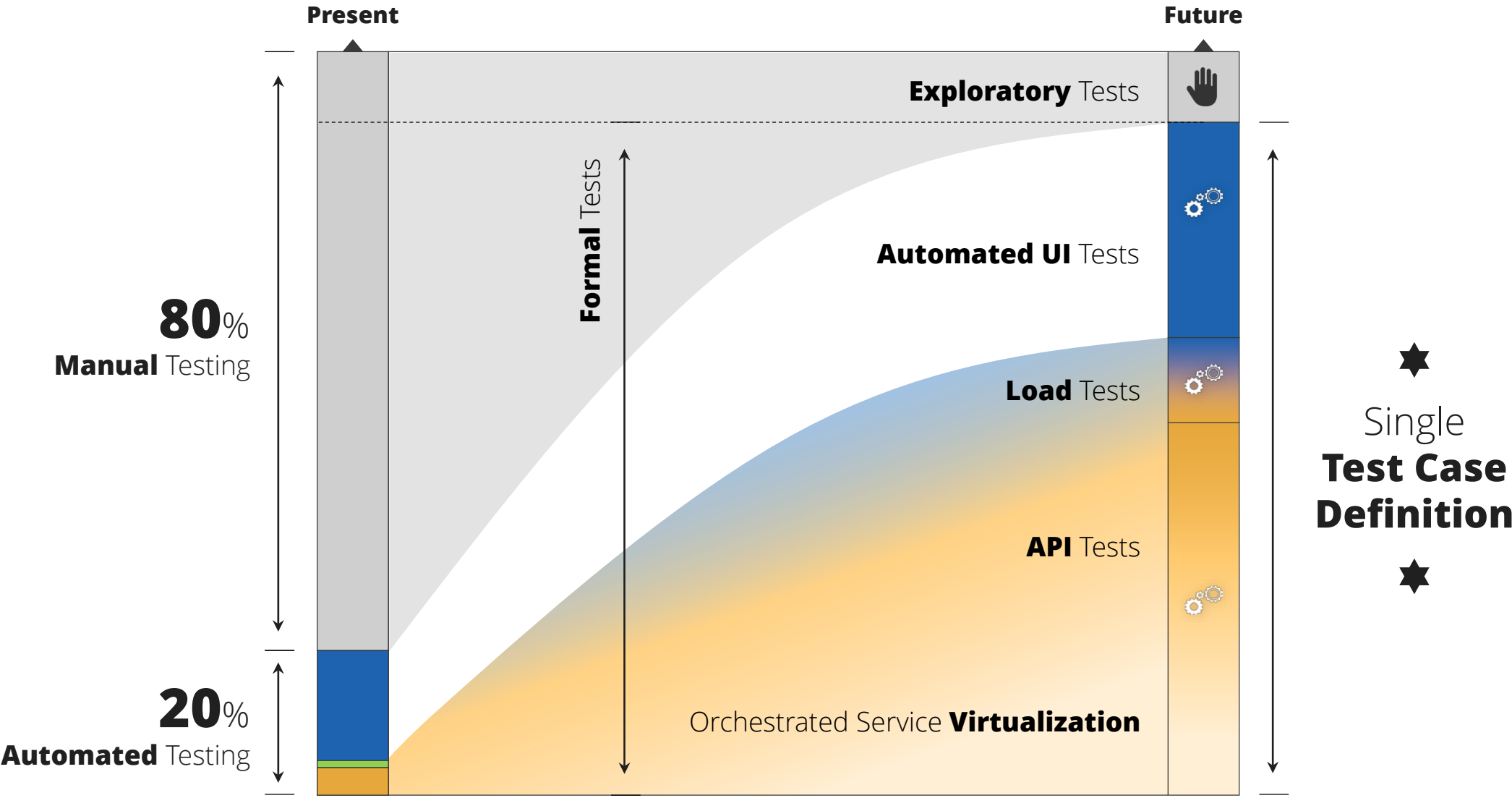
The reliance on  
**manual testing** is the  
**top technical challenge**  
in app development

World Quality Report, 2015/16

★

Test **automation**  
requires **developers**





# Change Detector



**High** Risk Coverage

Easy to scale because it's parallelizable

**Low** Information Value

Repeat what you have already learned

**Monitor** Known Risks

Confirm what you already know (measurable things)

**Mechanical** Testing

Process pre-defined data in pre-designed steps



# Formal Testing

« Demonstrate your **depth** of knowledge »

# Problem Detector



**Low** Risk Coverage

Hard to scale continuously because it relies on humans

**High** Information Value

Learn something new

**Analyze** Potential Risks

Focus on the things you don't know

**Intelligent** Testing

Create new test ideas based on what you have learned



# Exploratory Testing

« Demonstrate your **breadth** of knowledge »

# Change Detector



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

## Checking

« Requires **Processing** »

# Problem Detector

## Low Risk Coverage

Hard to scale continuously because it relies on humans

## High Information Value

Learn something new

## Analyze Potential Risks

Focus on the things you don't know

## Intelligent Testing

Create new test ideas based on what you have learned

# Exploratory Testing

« Demonstrate your **breadth** of knowledge »

# Change Detector

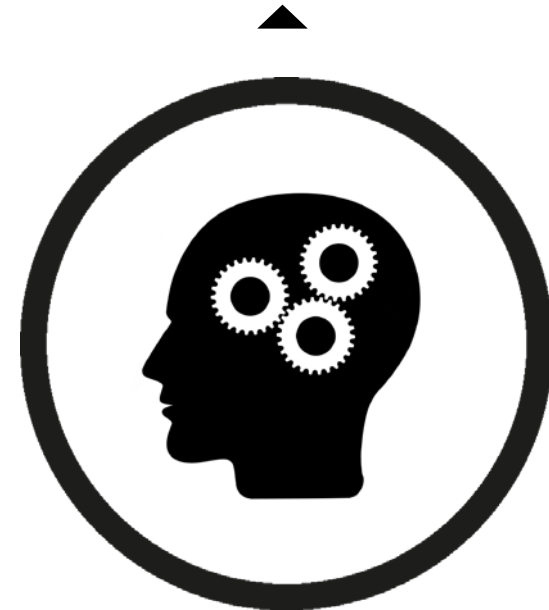


Evaluate a product by applying **algorithmic** decision rules to specific observations of a product

## Checking

« Requires **Processing** »

# Problem Detector



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

## Exploring

« Requires **Thinking** »



# Agile

## Testing Equation

# Checking

Efficient **Formal Testing**

+

# Exploring

Effective **Exploratory Testing**

=

# Testing

**Thorough** Testing

# Agile

## Testing Equation

# Checking

Efficient **Formal Testing**

=

# Machine

Machine **Checking**

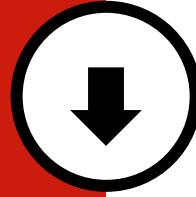
+

# Human

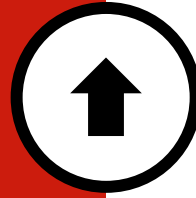
Human **Checking**

# Agile

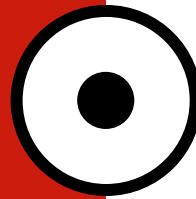
## Testing Equation



**Chêcking**  
Human Checking



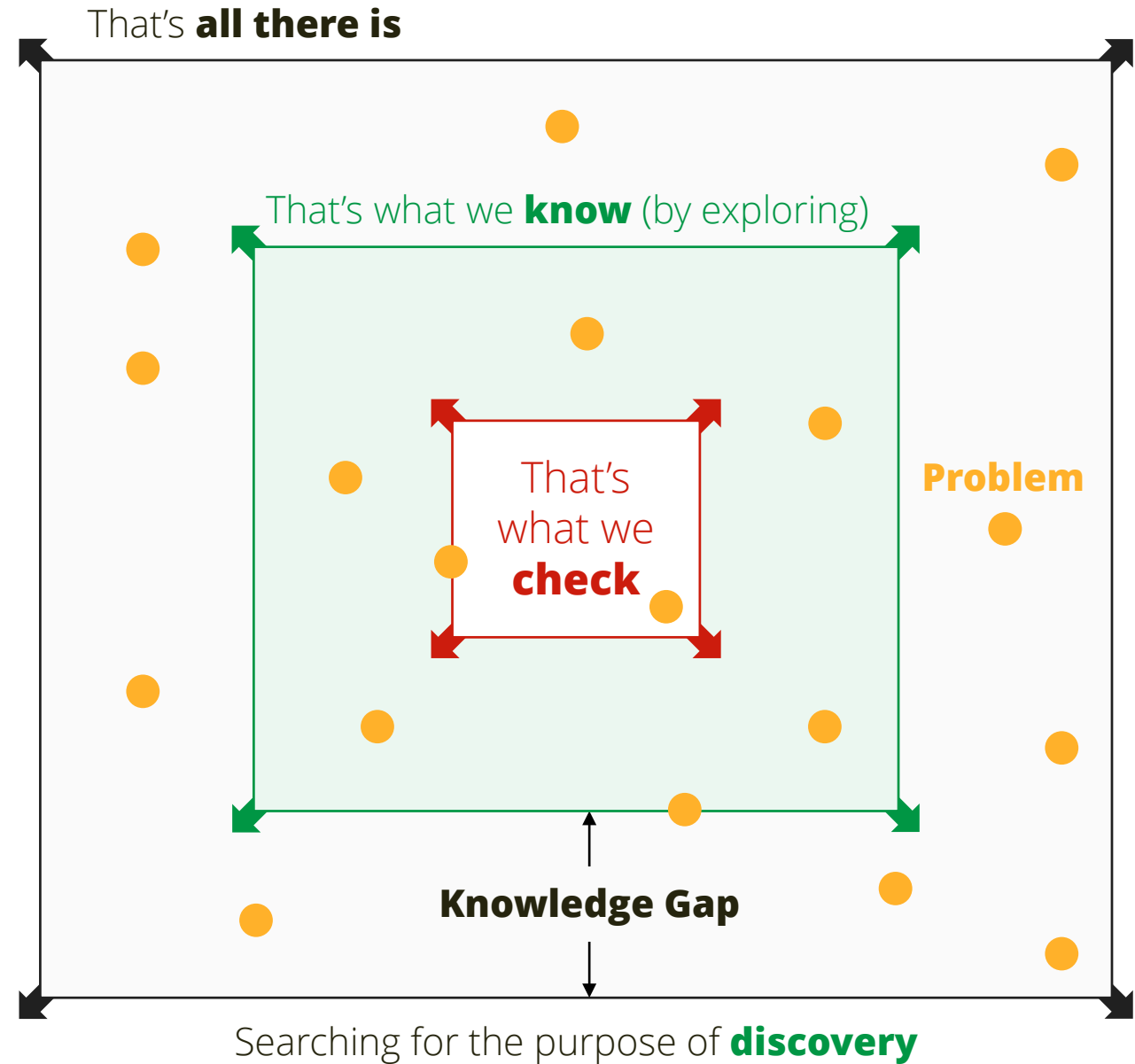
**Êxploring**  
Human Exploration



**Testing**  
Manual Testing

Closing the  
gap between what  
we **know** and what  
we **don't know**

**Testing** Purpose



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.



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.



# Approach

---

Provides **orientation**





# Technique

---

Provides a **systematic procedure**



# Approach

---

Provides **orientation**



# Technique

---

Provides a **systematic procedure**

1

2

3

4

5



# Technique

---

Provides a **systematic procedure**

## Session-Based Testing

Structure exploratory testing to allow large-scale implementations

2

3

4

5



**Imagination** in a **straightjacket**

## **Session**-Based Testing

Structure exploratory testing to allow large-scale implementations

**2**

**3**

**4**

**5**

Chartered  
Uninterrupted  
Reviewable

---

**Session**

## **Session**-Based Testing

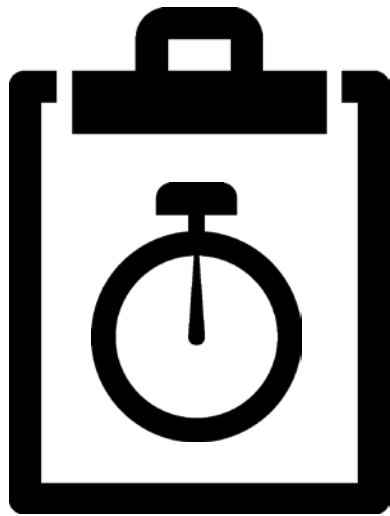
Structure exploratory testing to allow large-scale implementations

**2**

**3**

**4**

**5**



**Timebox**

## **Session**-Based Testing

Structure exploratory testing to allow large-scale implementations

**2**

**3**

**4**

**5**





**Mission**

## **Session**-Based Testing

Structure exploratory testing to allow large-scale implementations

**2**

**3**

**4**

**5**



**Scope**

*Straightjacket*

## **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

## **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?

## **Session**-Based Testing

Structure exploratory testing to allow large-scale implementations

## **Requirements**-Based Testing

Limit the scope to make it manageable

3

4

5

## **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

## **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

## Tour-Based Testing

Set concrete goals to provide a clear focus

4

5



@speed  
**Quality**

It's just **some value** to some person

## 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

## **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

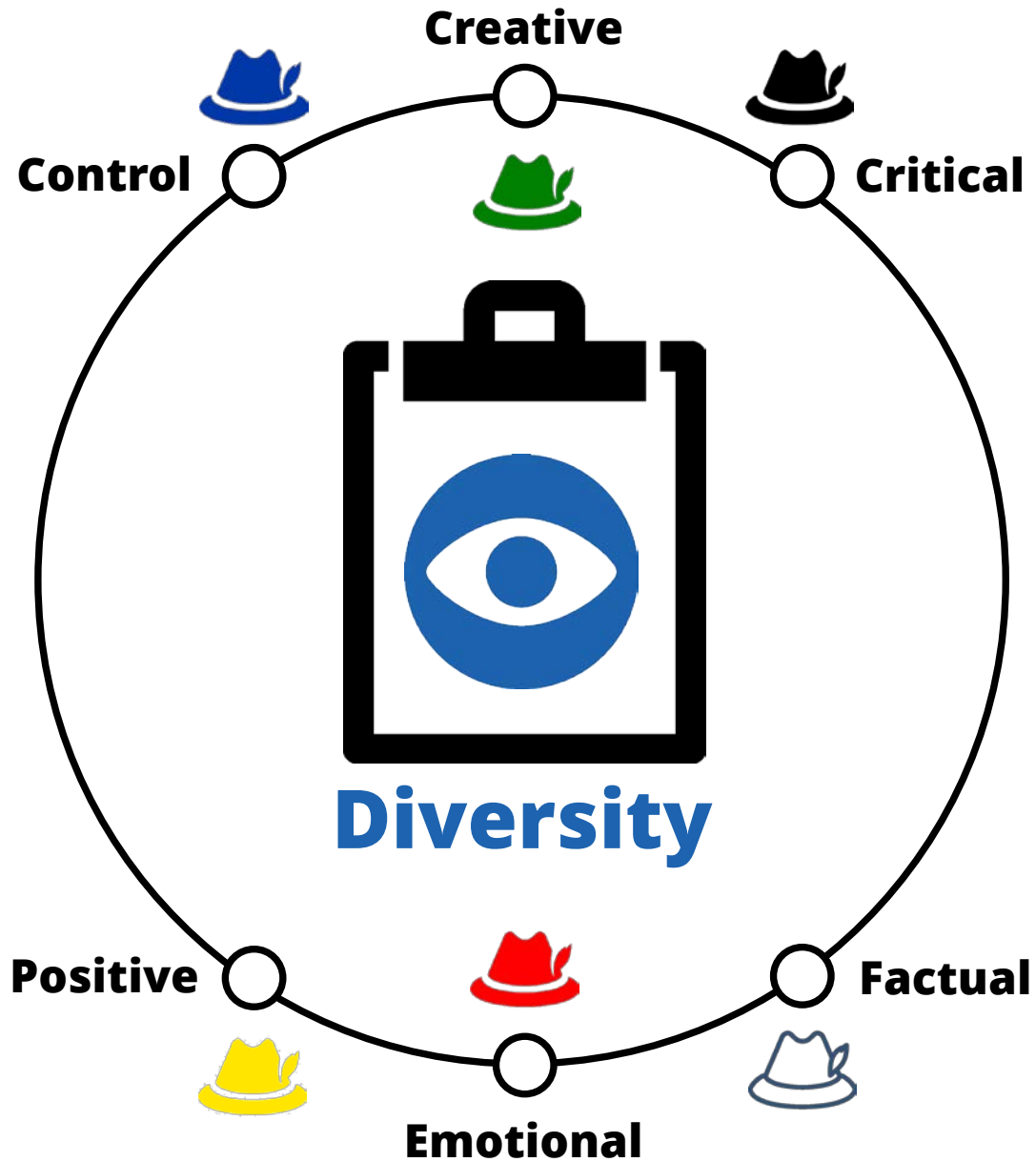
Limit the scope to make it manageable

## **Tour**-Based Testing

Set concrete goals to provide a clear focus

## **Polychrome** Testing

Explore the product from different viewpoints to diversify testing



## **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

## Structure exploratory testing to allow large-scale implementations

## Structure exploratory testing to allow large-scale implementations

Limit the scope to make it manageable

Limit the scope to make it manageable

## Set concrete goals to provide a clear focus

## Set concrete goals to provide a clear focus

Explore the product from different viewpoints to diversify testing

Explore the product from different viewpoints to diversify testing

## Capture each test idea to make it reviewable

## Capture each test idea to make it reviewable



Testing is not about creating test cases,  
it's about performing **experiments**

---

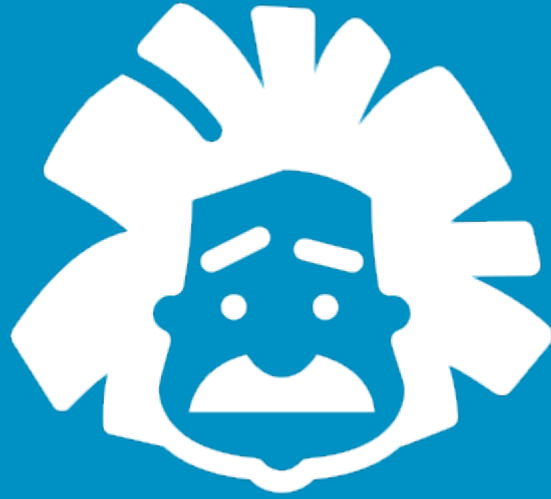
Lesson **Learned**



The test doesn't find the bug. A **human** finds the bug, and the test plays a role in helping the human find it

---

Lesson **Learned**

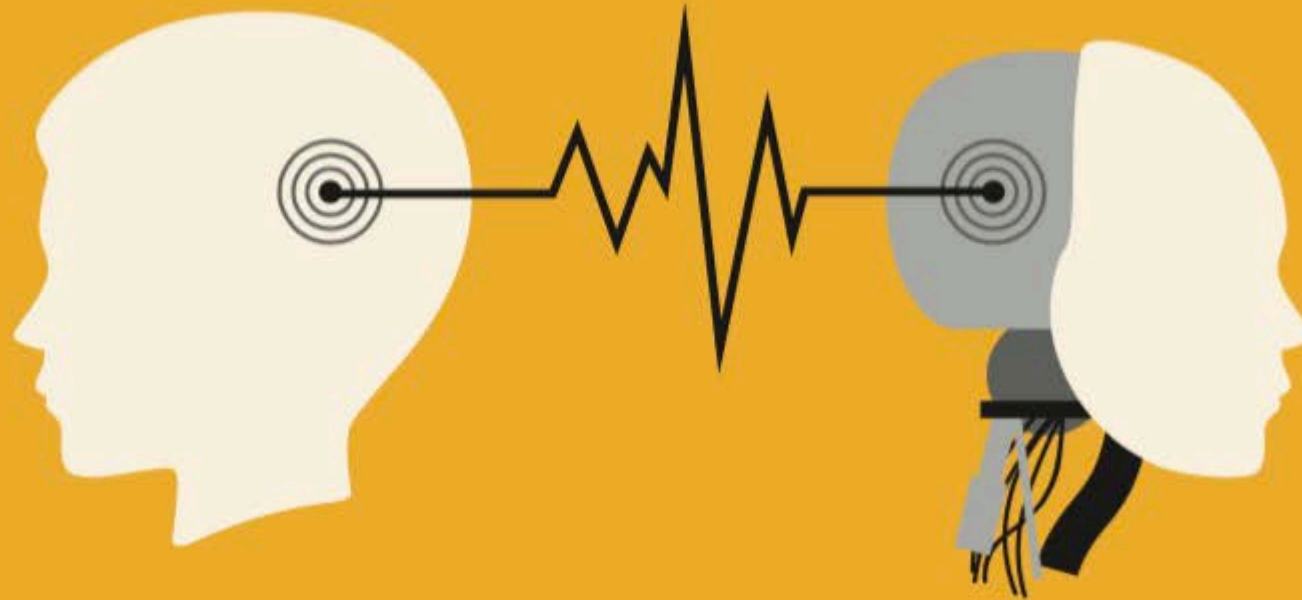


**Insanity** is doing the same thing over and over again and expecting different results

---

Lesson **Learned**





We don't need humans doing something that a **machine** can do,  
we want the human testers doing **exploratory testing**

---

Lesson **Learned**





Better to have a human tester who can look at a requirement and work out what needs to be **tested**

---


Lesson **Learned**



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

---

Lesson **Learned**

A man in a dark suit, white shirt, and red tie stands against a light-colored stone wall, looking upwards with a concerned expression. A young girl with brown hair, wearing a white t-shirt with colorful heart patterns, stands next to him, looking down sadly. They are holding a large, brown cardboard sign that reads: "ROBOT TOOK MY JOB  
WILL WORK  
FOR FOOD".

ROBOT TOOK MY JOB  
WILL WORK  
FOR FOOD

# Questions

---

Because **answers** exist only to questions...