TRICENTIS

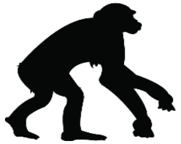
The Future of Testing

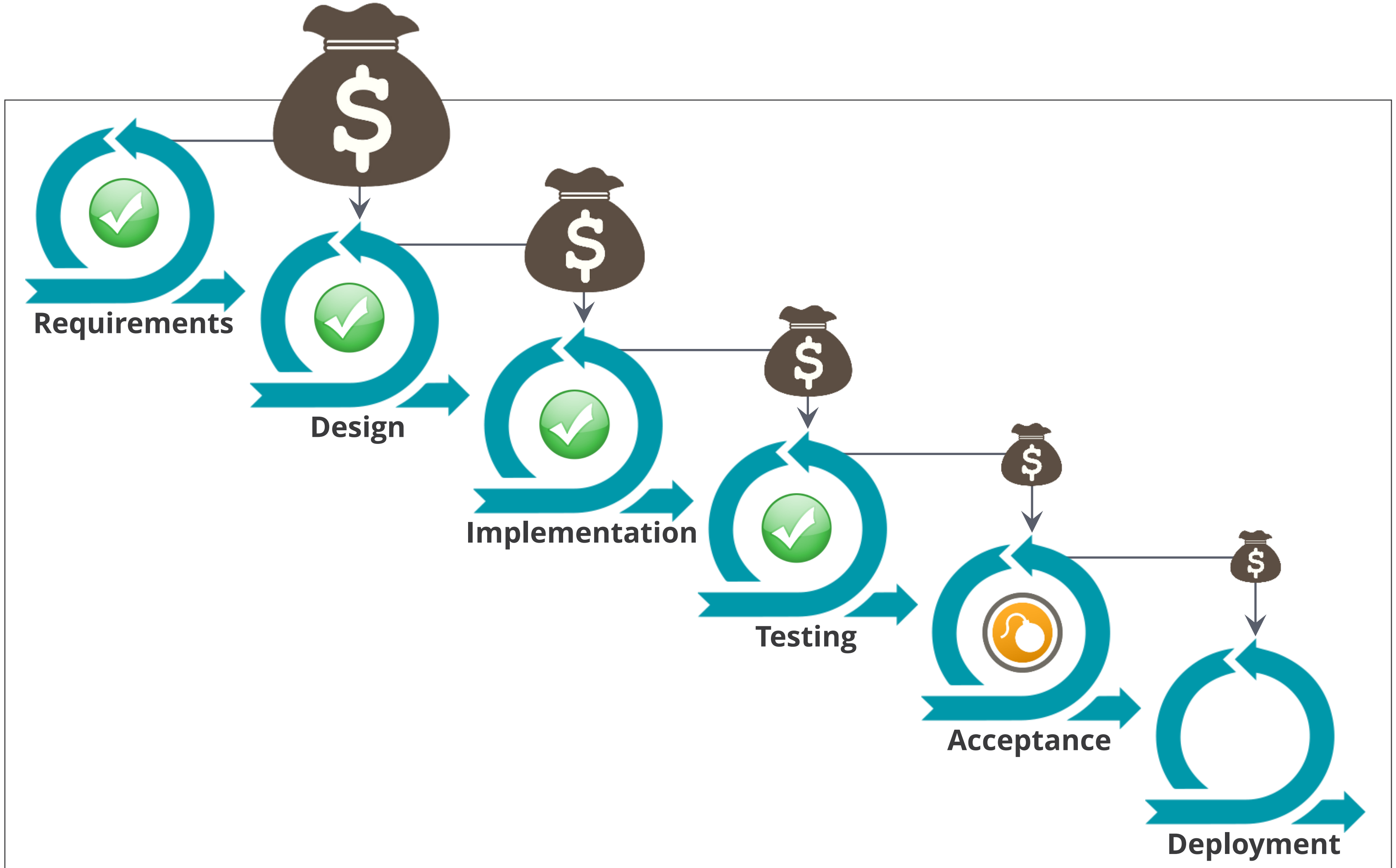
How to build in **quality** & **efficiency** right from the start?

Dr. Gerd **Weishaar**

Cycle Time

Years
Months





Requirements

Design

Implementation

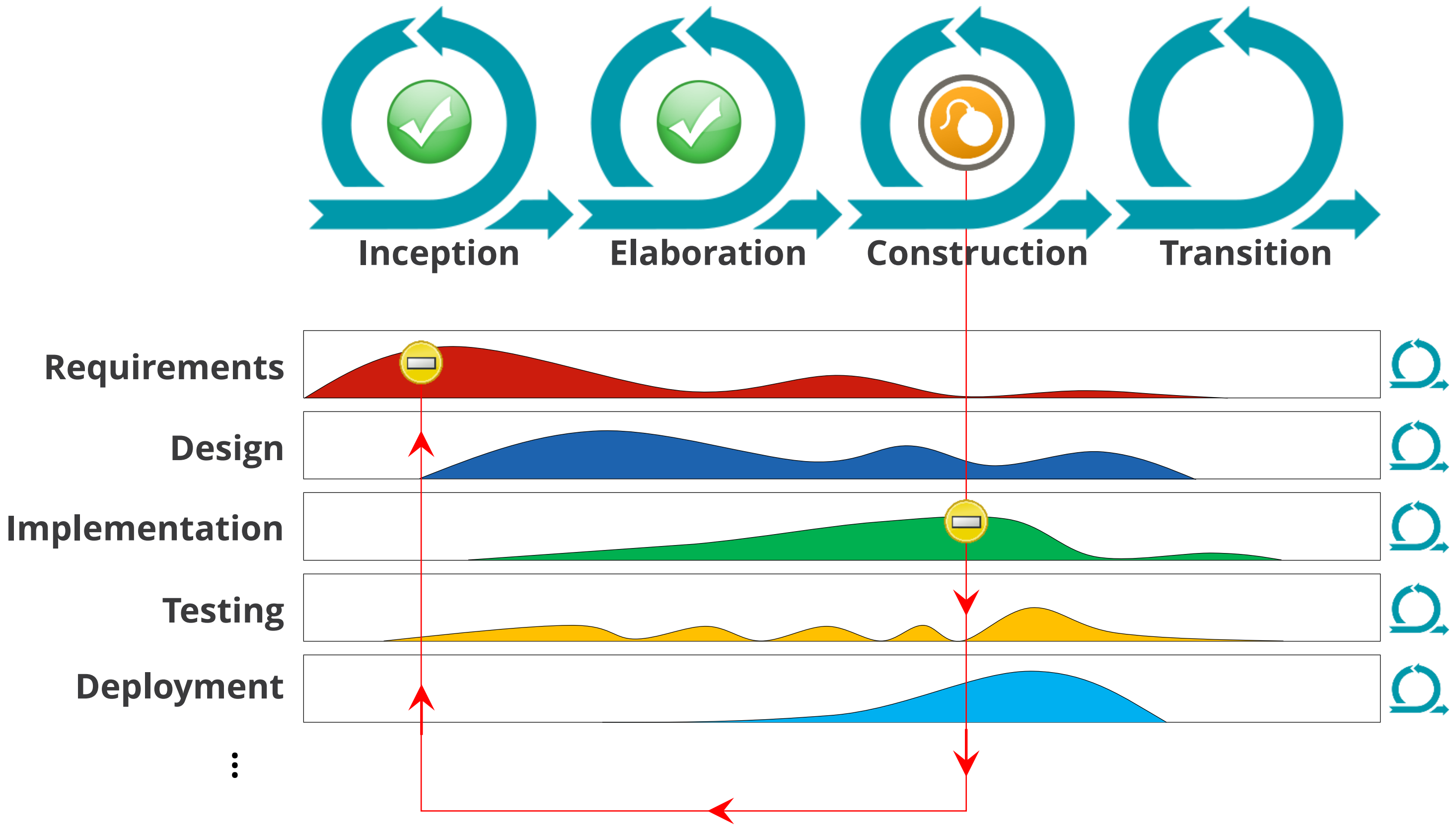
Testing

Acceptance

Deployment

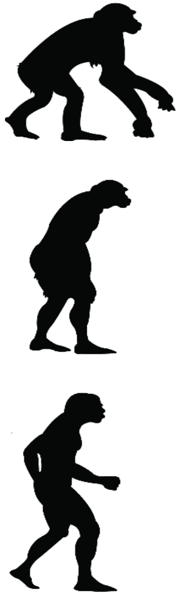
Waterfall

Cycle Time



Rational Unified Process

Cycle Time



Years
Months

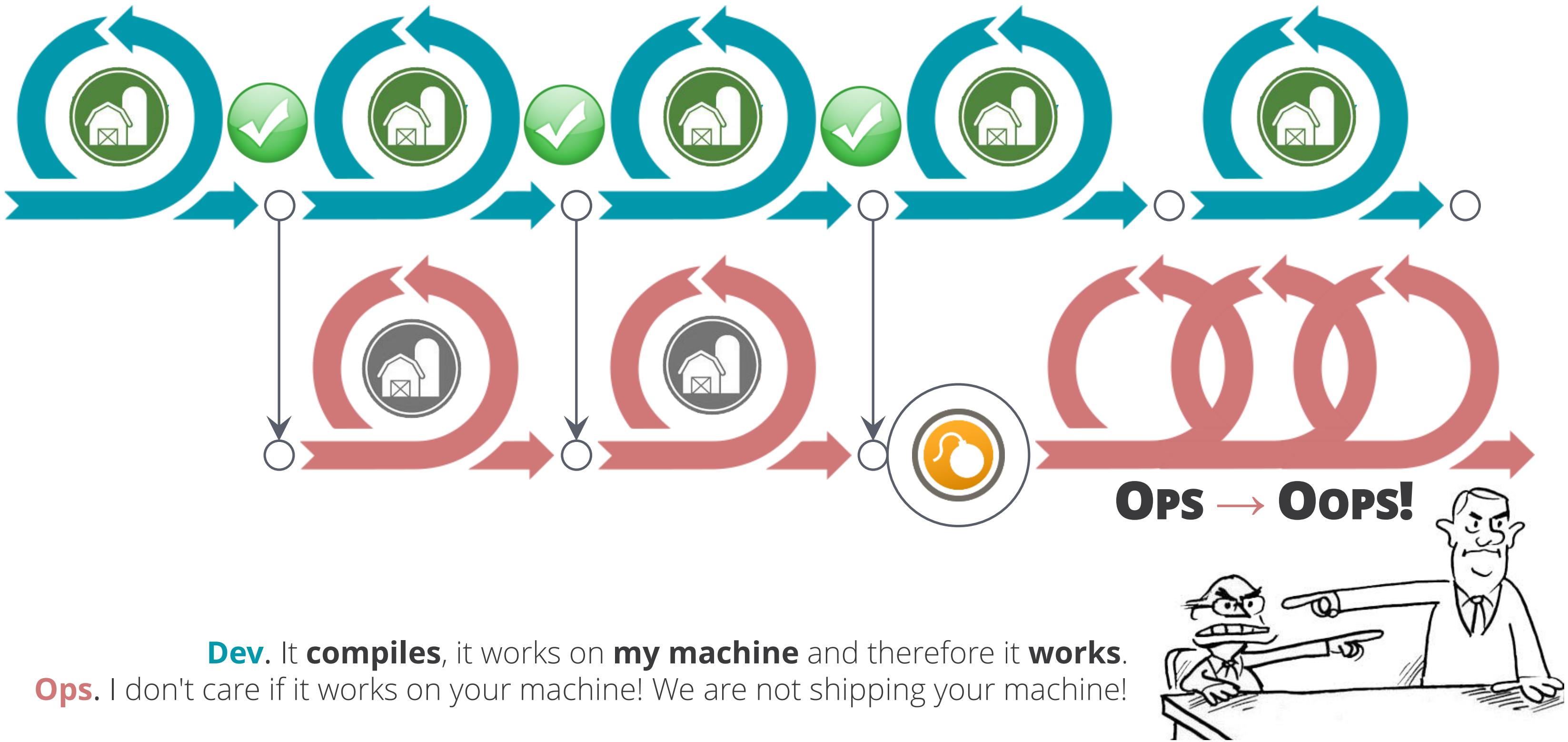
Months
Weeks

Weeks
Days

Siloisation

“Where we are right now **just sucks**.”

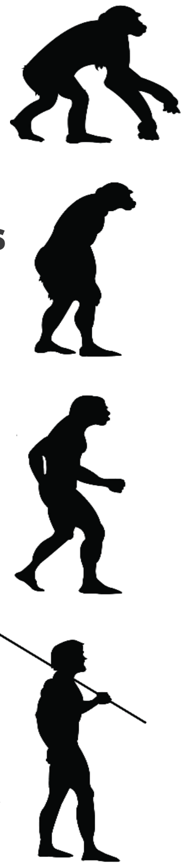
Patrick Debois, 2009



Dev. It **compiles**, it works on **my machine** and therefore it **works**.
Ops. I don't care if it works on your machine! We are not shipping your machine!

Agile

Cycle Time



Years
Months

Months
Weeks

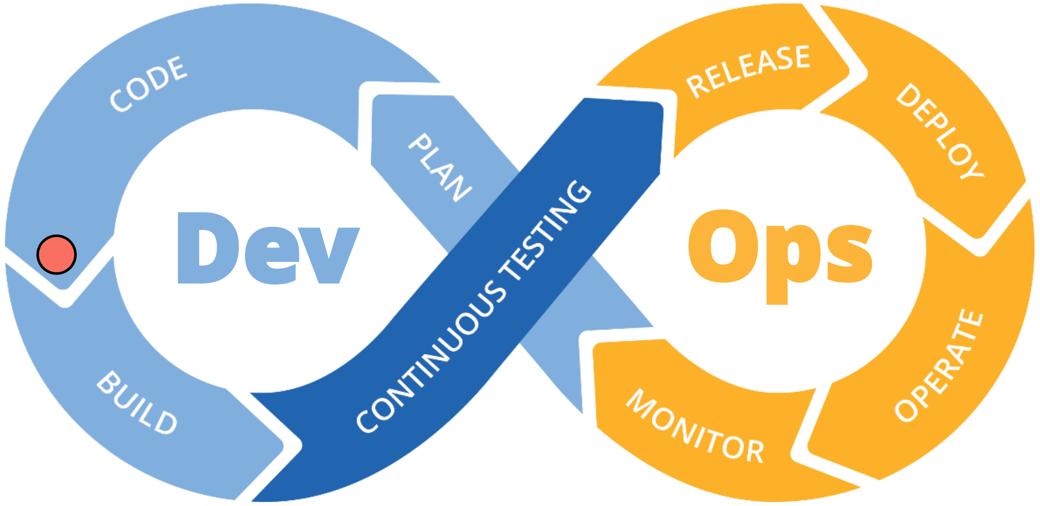
Weeks
Days

Days
Minutes

Collaborative
Development

Continuous
Release & Deployment

Continuous
Integration

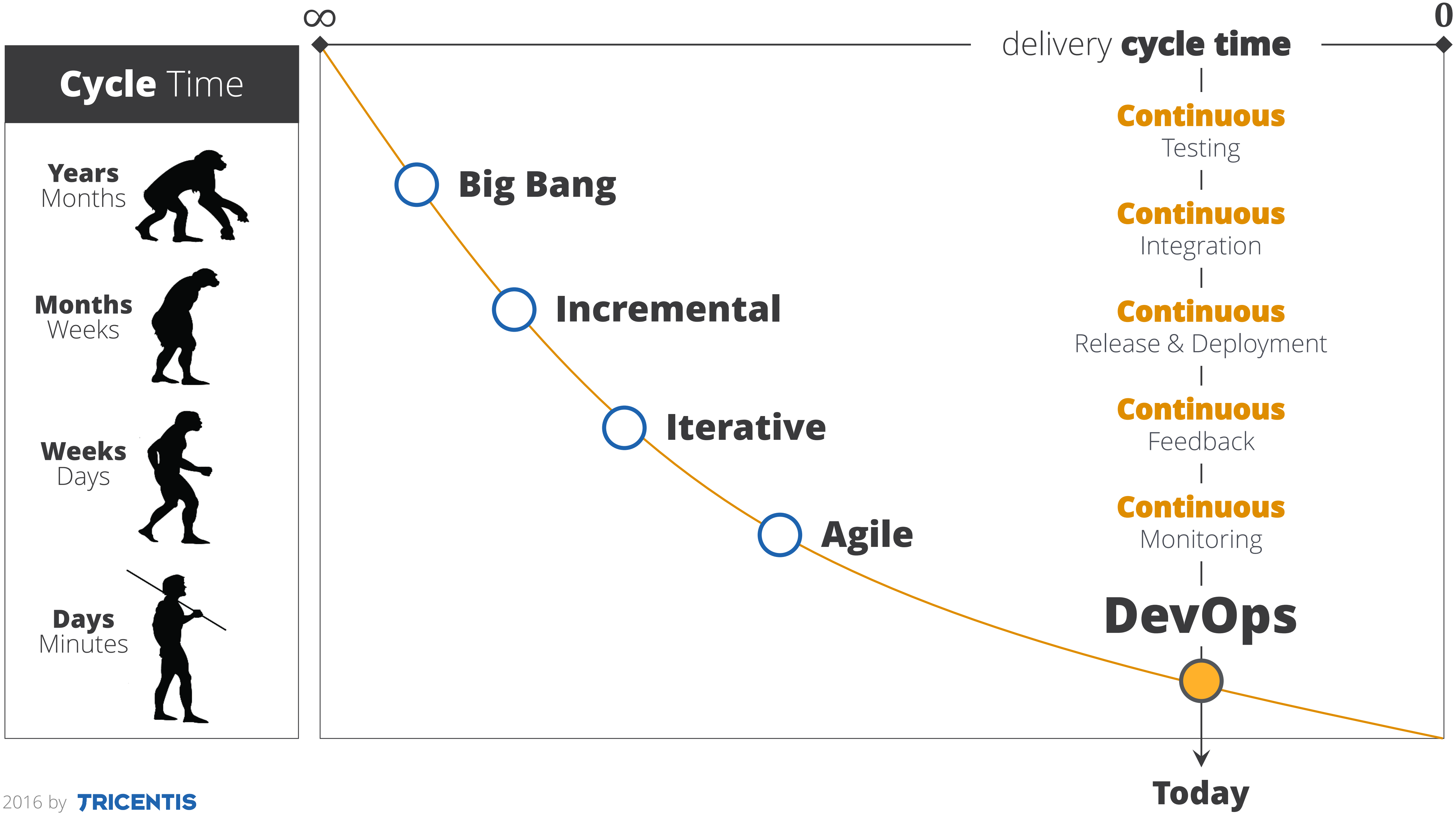


Continuous
Feedback

Continuous
Testing

Continuous
Monitoring

DevOps



Cycle Time

Years
Months

Months
Weeks

Weeks
Days

Days
Minutes

∞

delivery **cycle time**

0

Big Bang

Incremental

Iterative

Agile

Continuous
Testing

Continuous
Integration

Continuous
Release & Deployment

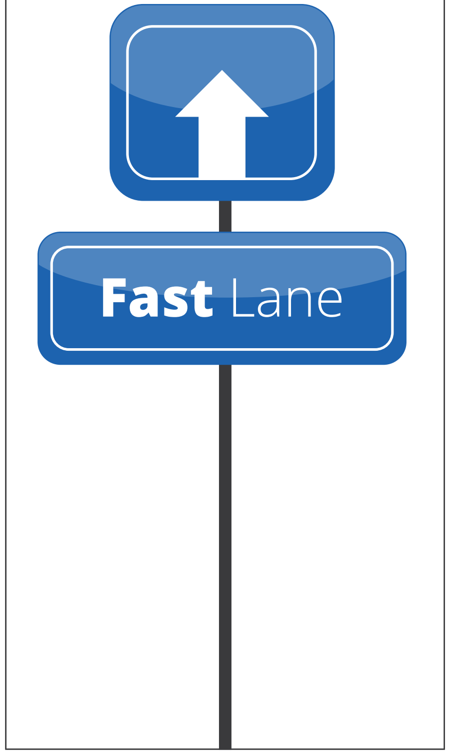
Continuous
Feedback

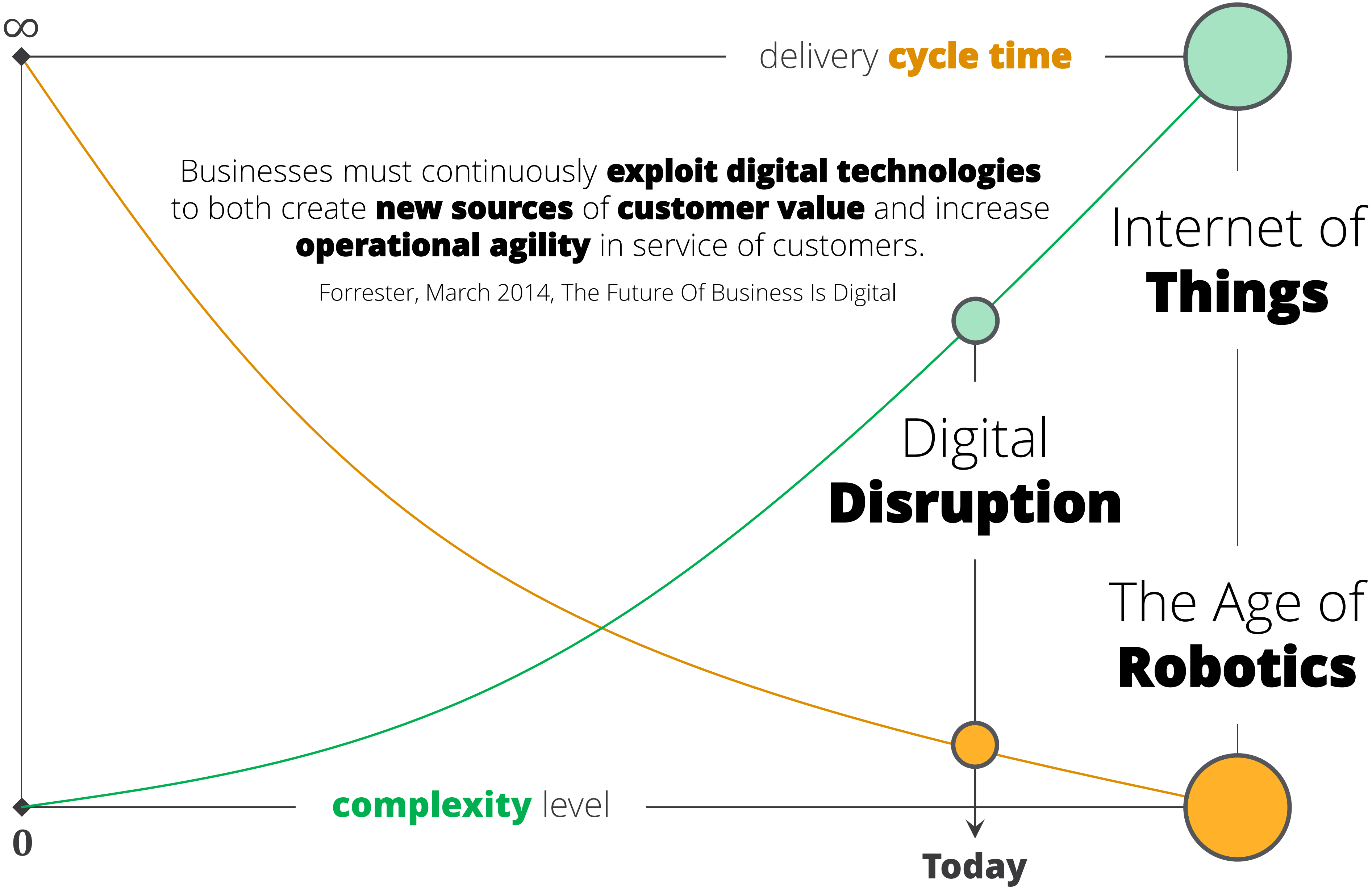
Continuous
Monitoring

DevOps

Today

Testing **Future**

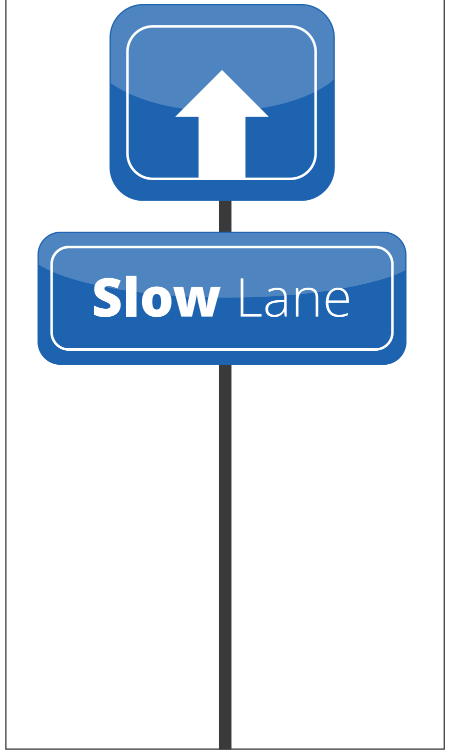




Businesses must continuously **exploit digital technologies** to both create **new sources** of **customer value** and increase **operational agility** in service of customers.

Forrester, March 2014, The Future Of Business Is Digital

Testing **Present**



Start — software **delivery cycle** — **End**

80%
manual testing

90%
of all test automation is **UI test automation**

67%
average level of **redundancy** in enterprise test portfolios

40%
average **risk coverage** achieved in enterprise test portfolios

It's me, a **problem**!

55%
of **systems** only partially **accessible** by Dev/Test

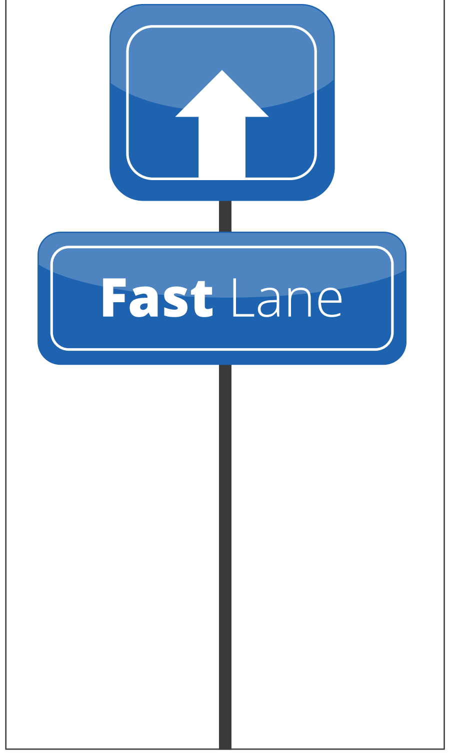
56%
of overall test effort goes into test case **maintenance**

30%
of **bugs** found in **acceptance** & **production** stage

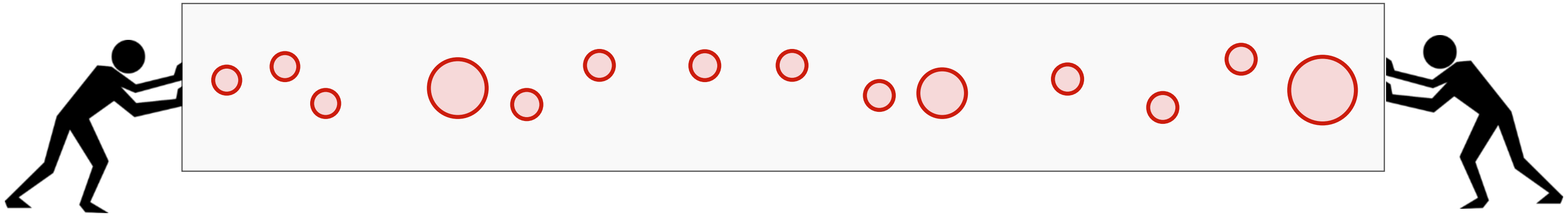
50%
of manual testing goes into **test data** preparation & organization

> "The first step to **solve** a problem is to **accept** that you have one."
>
> Albert Einstein, 1921

Testing **Future**

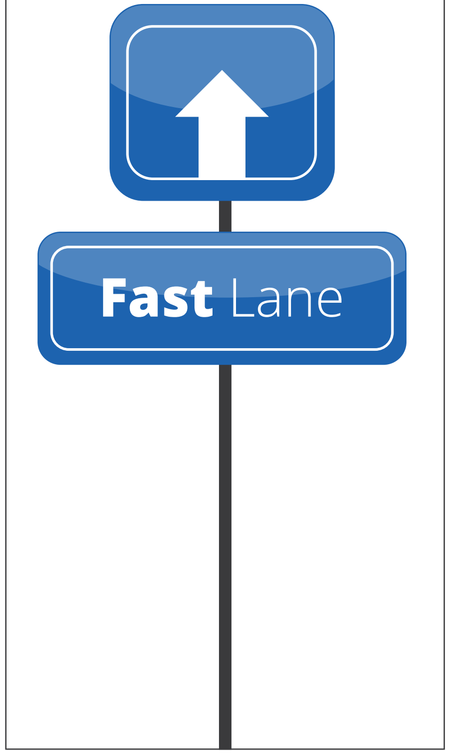


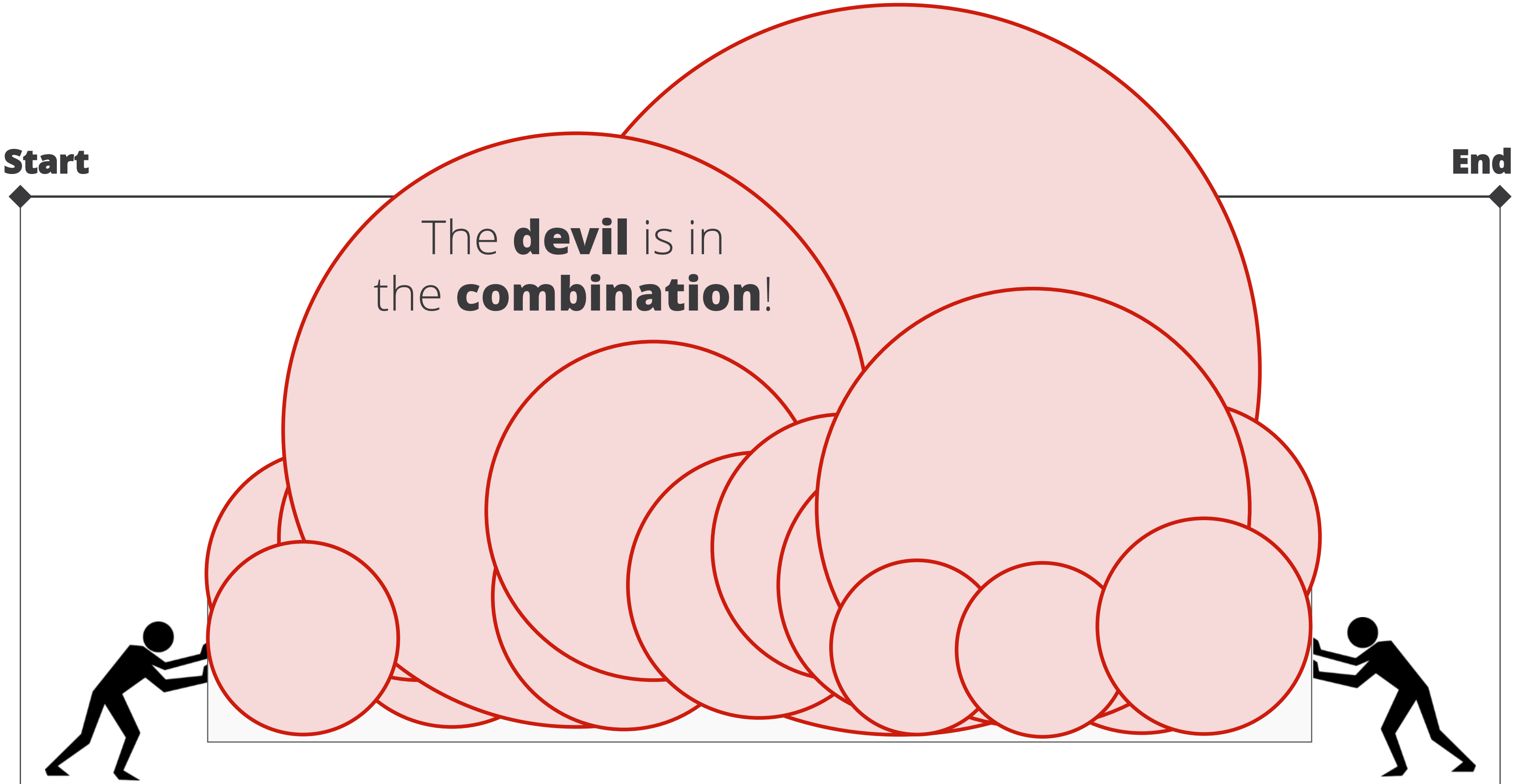
Start — software **delivery cycle** — **End**



By **compressing** the delivery cycle,
do problems just move closer to each other?

Testing **Future**




Start

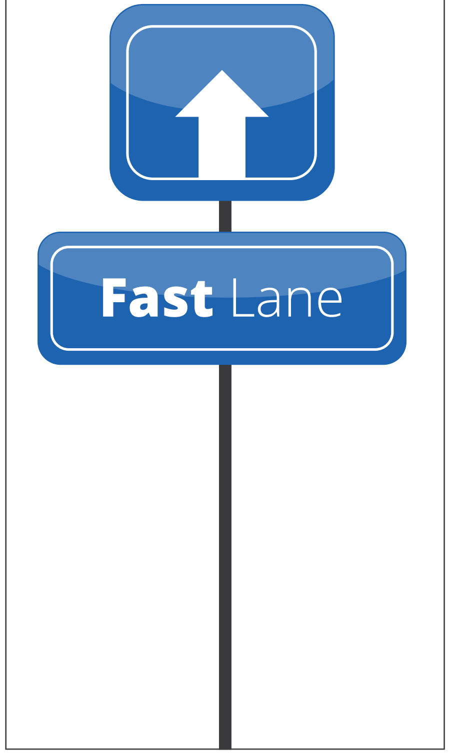
End

The **devil** is in the **combination**!

No, they **mutually reinforce** each other!

Hence, the biggest strength of DevOps is **not solving problems**, but rather **exposing buried problems**.

Testing
Future



"Testing **harder** isn't the answer, **testing smarter** is."

Wolfgang Platz, 2016

Automation

continuous testing is a **must**

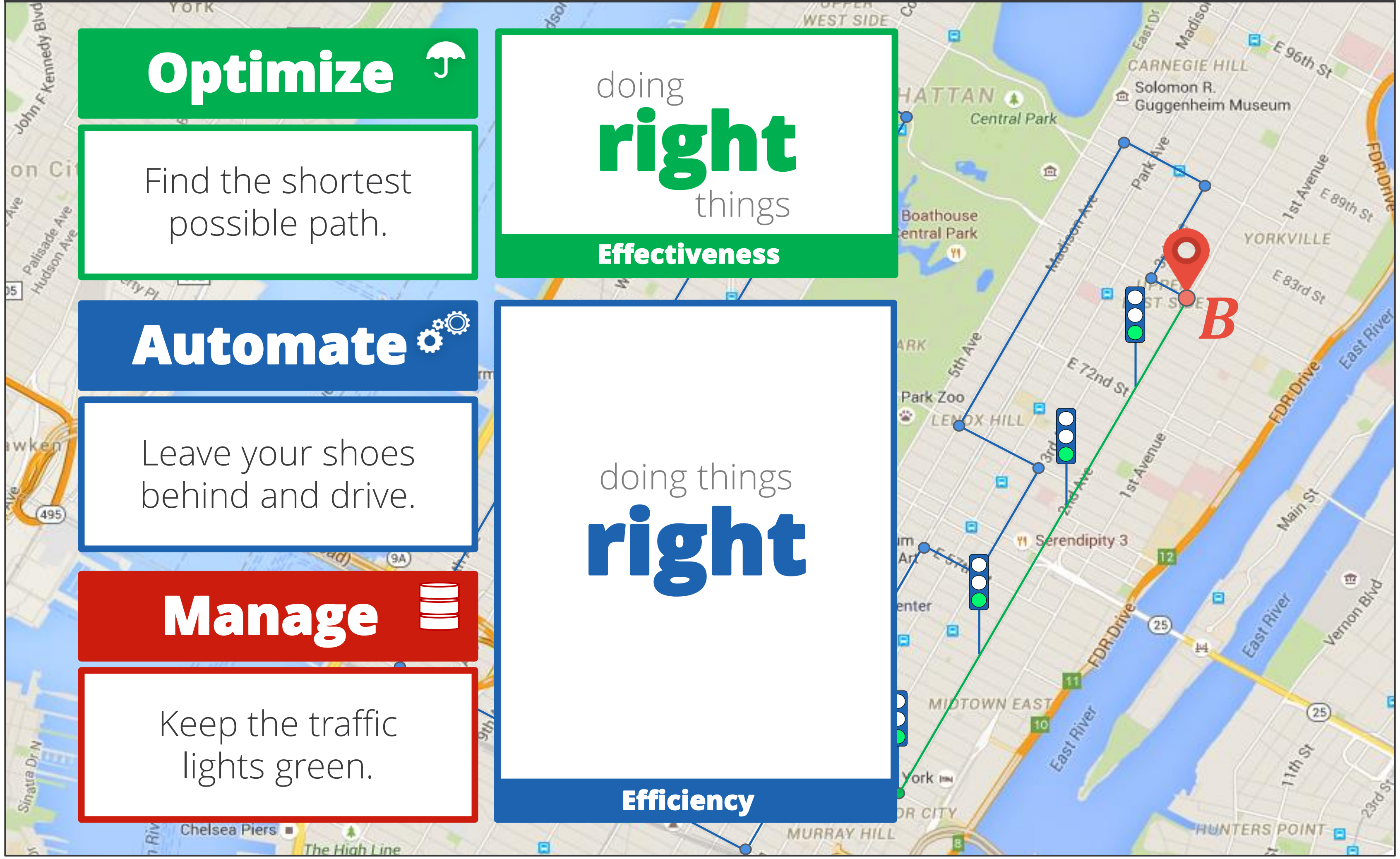
The **Silver Bullet**

Is it really the **only solution** to all these problems?

Testing **Future**

No, it's about being **efficient** & **effective**.

Is it just about **speed**?



Optimize

Find the shortest possible path.

Automate

Leave your shoes behind and drive.

Manage

Keep the traffic lights green.

doing **right** things

Effectiveness

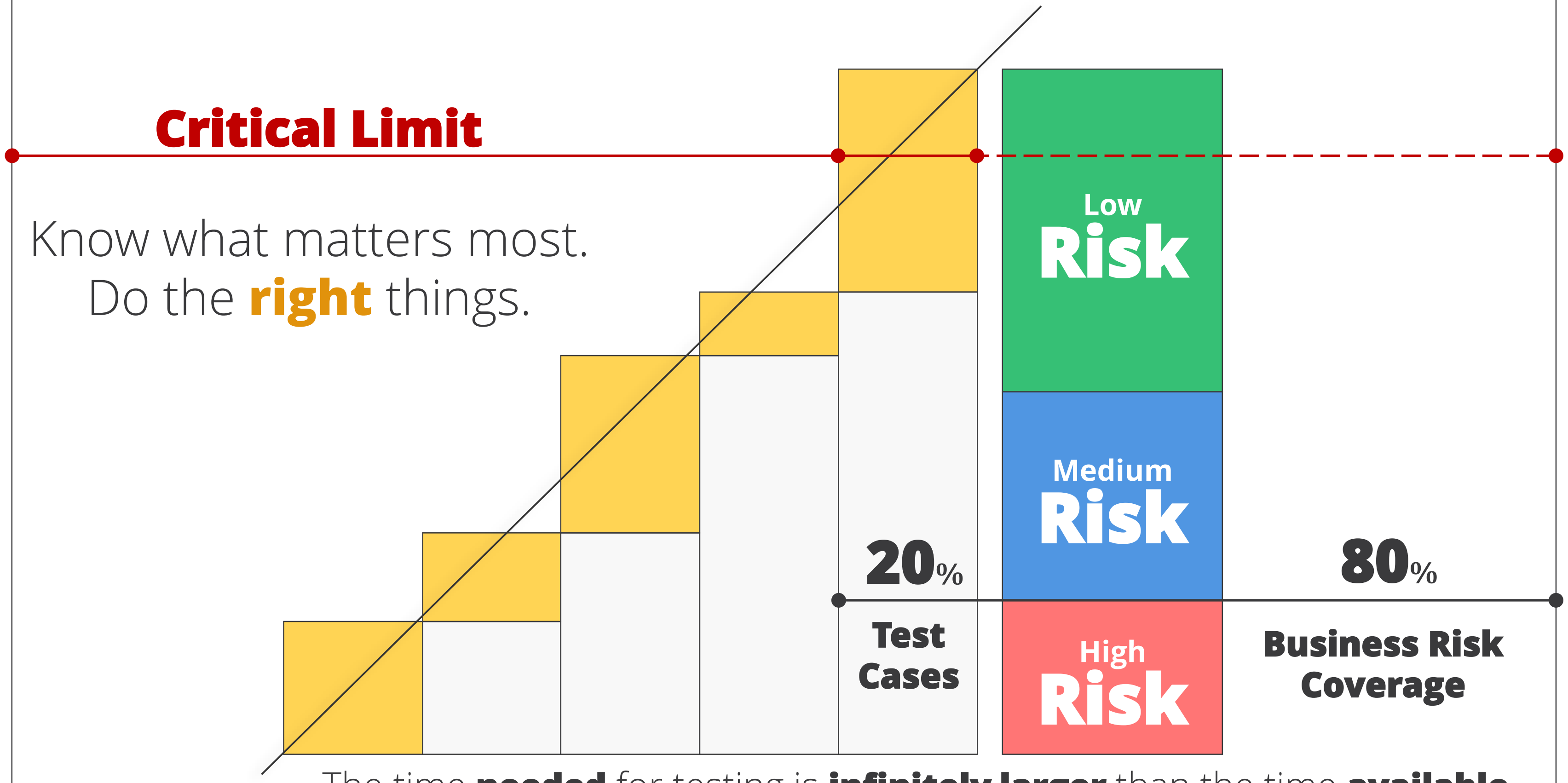
doing things **right**

Efficiency

> "Without **data** you're just another person with an opinion."
>
> W. Edwards Deming

40 % average **risk coverage** in enterprise test case portfolios

Tosca **Optimize**

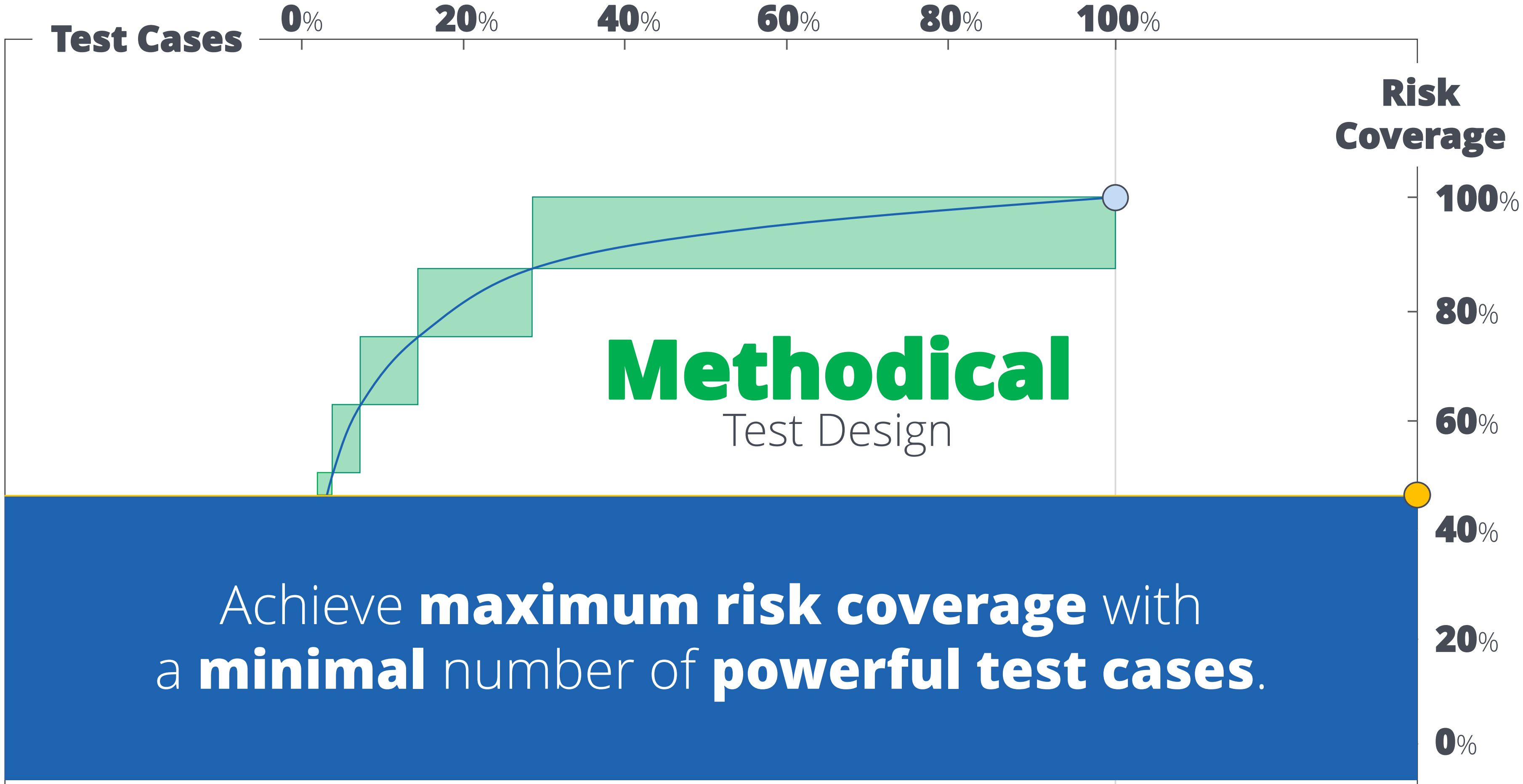


Know what matters most.
Do the **right** things.

The time **needed** for testing is **infinitely larger** than the time **available**.

Risk-Based Testing

Tosca
Optimize

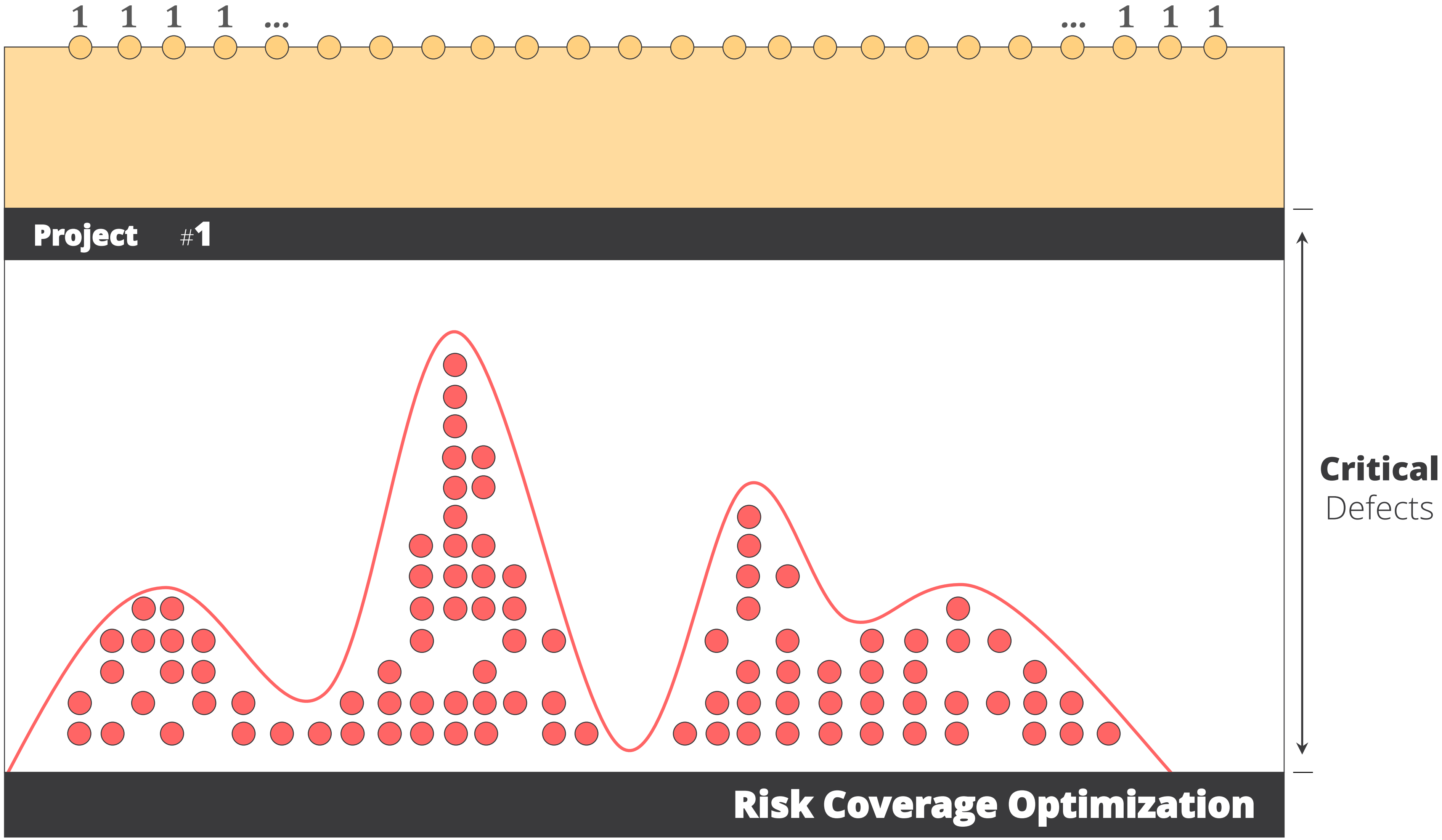


Achieve **maximum risk coverage** with a **minimal** number of **powerful test cases**.

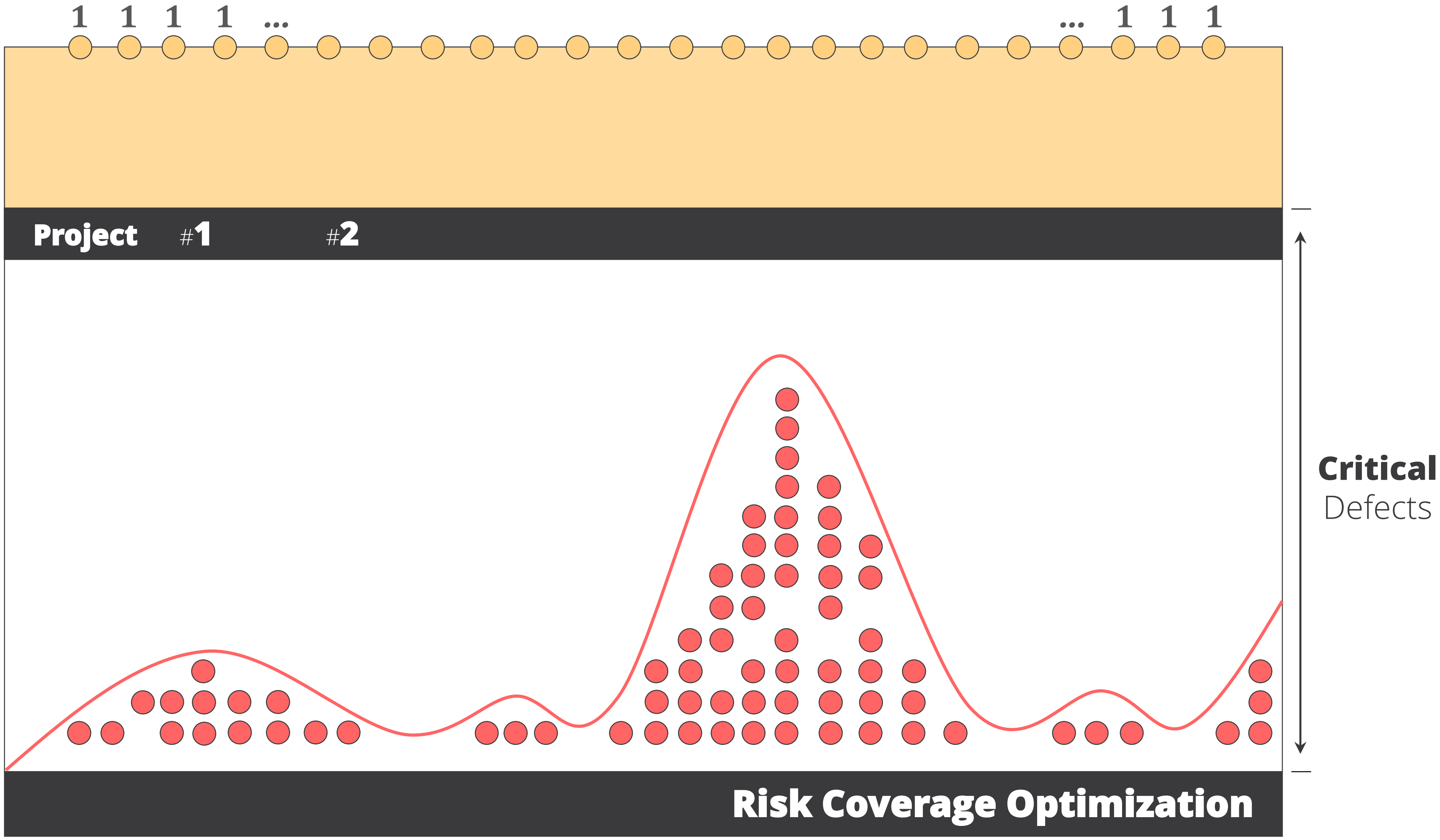
Risk Coverage Optimization

Tosca

Optimize

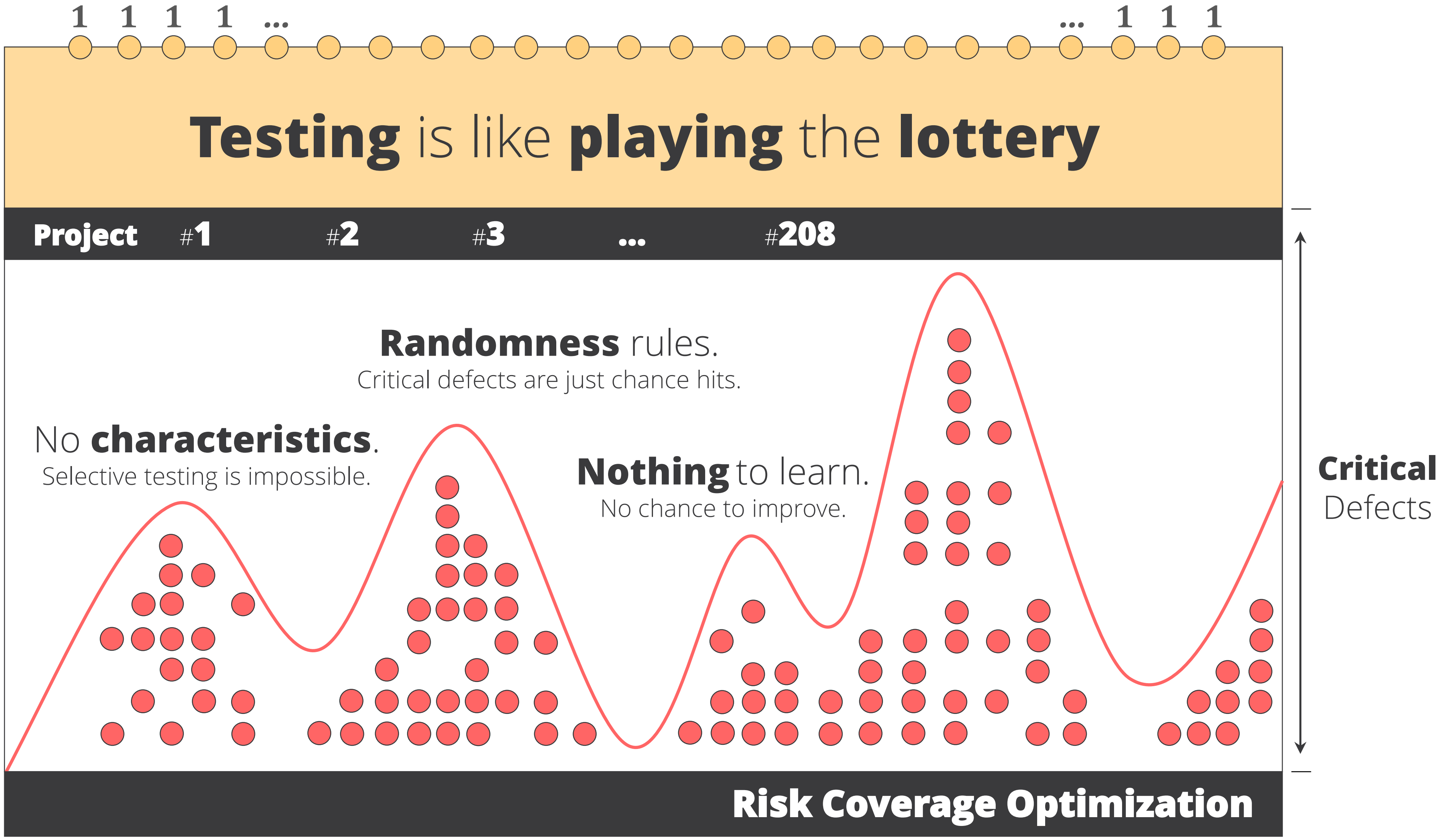


Tosca
Optimize



Tosca
Optimize

Testing is like **playing** the **lottery**



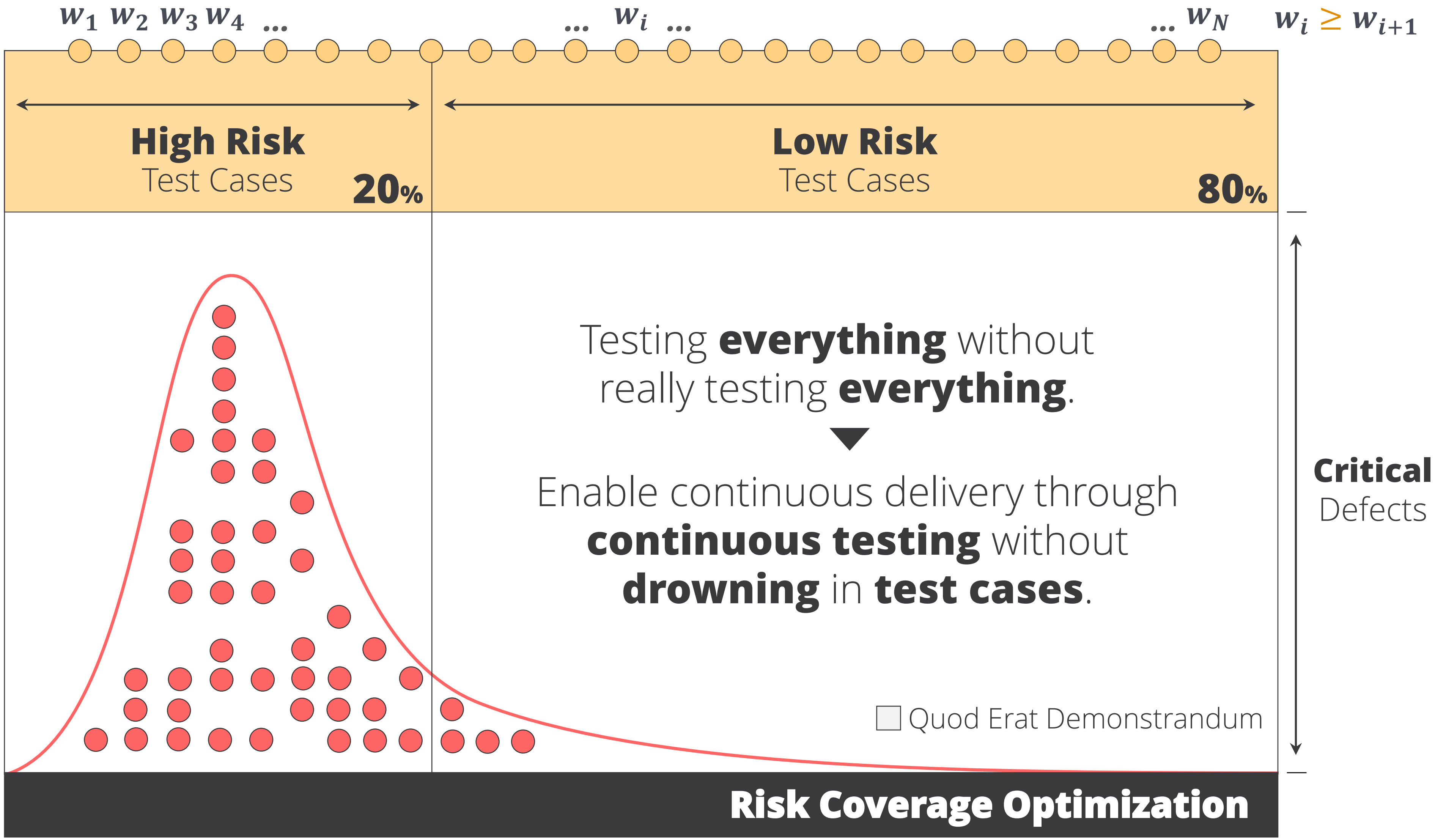
Randomness rules.
Critical defects are just chance hits.

No **characteristics**.
Selective testing is impossible.

Nothing to learn.
No chance to improve.

Risk Coverage Optimization

Tosca
Optimize



Testing **everything** without really testing **everything**.

▼

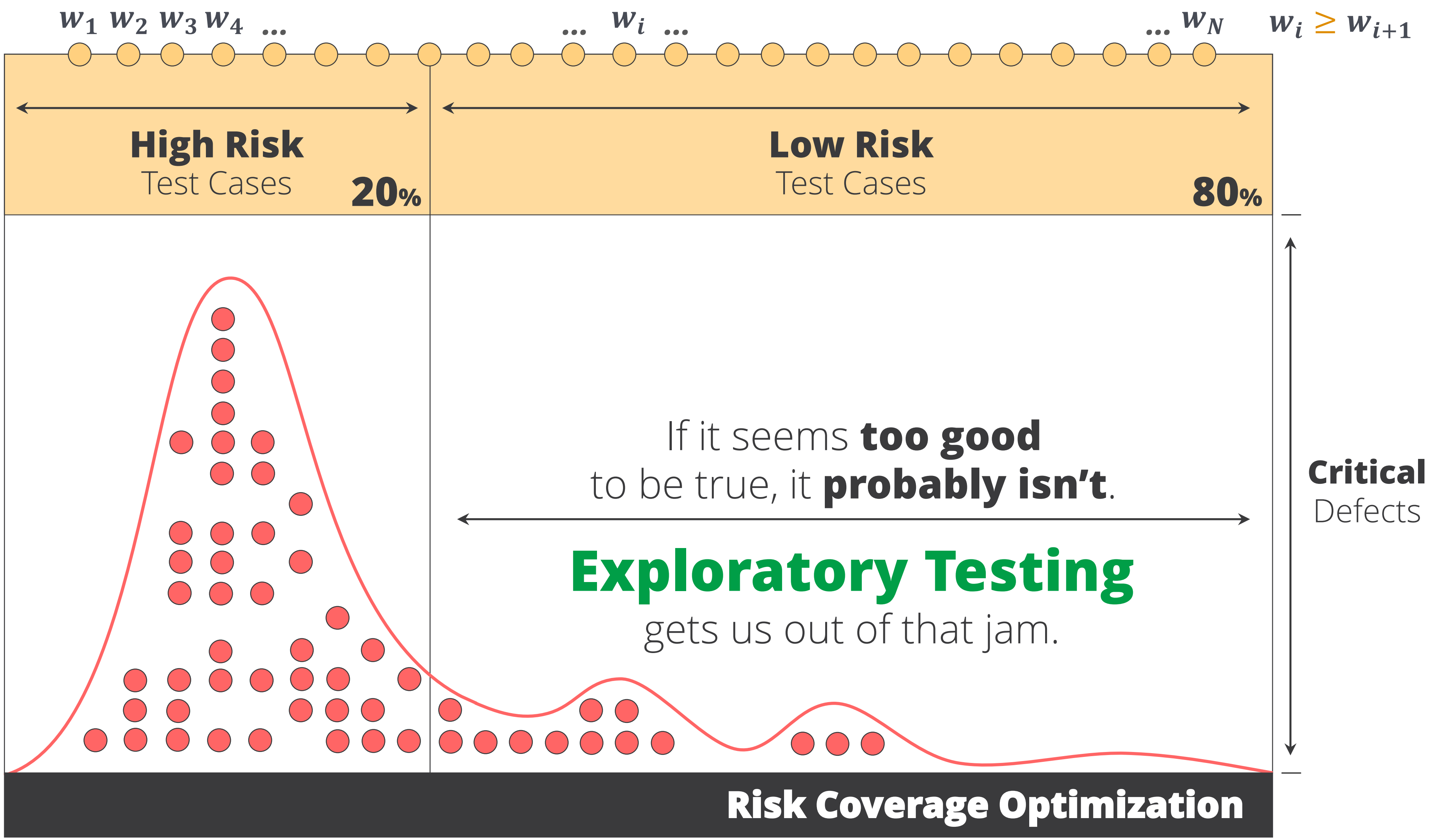
Enable continuous delivery through **continuous testing** without **drowning** in **test cases**.

□ Quod Erat Demonstrandum

Risk Coverage Optimization

Tosca

Optimize



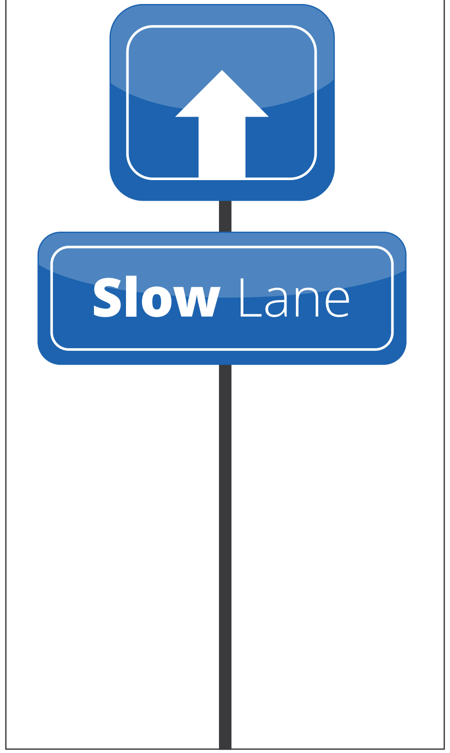
If it seems **too good**
to be true, it **probably isn't**.

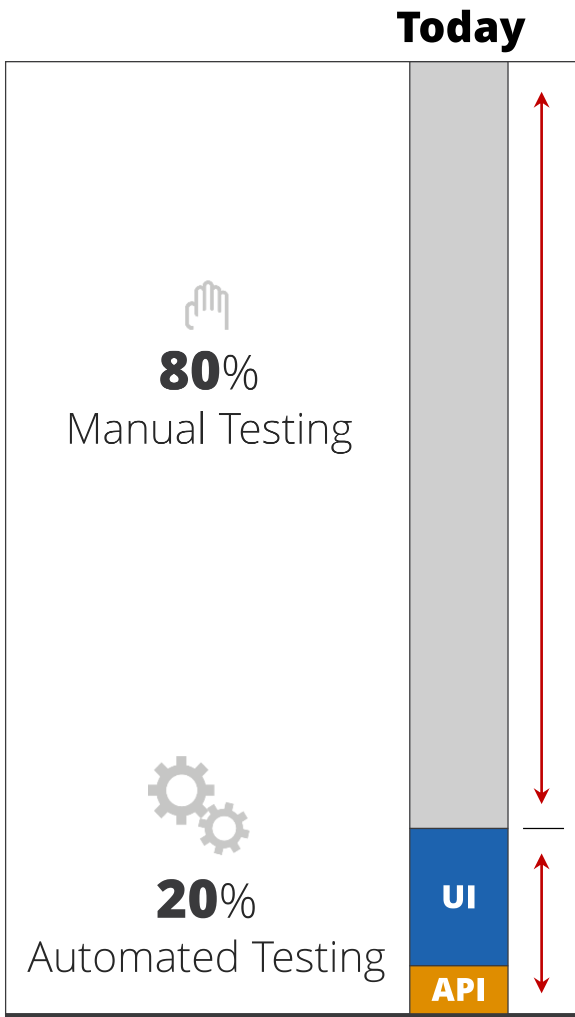
Exploratory Testing

gets us out of that jam.

Risk Coverage Optimization

Testing **Present**




Today

80%
Manual Testing

20%
Automated Testing

UI

API

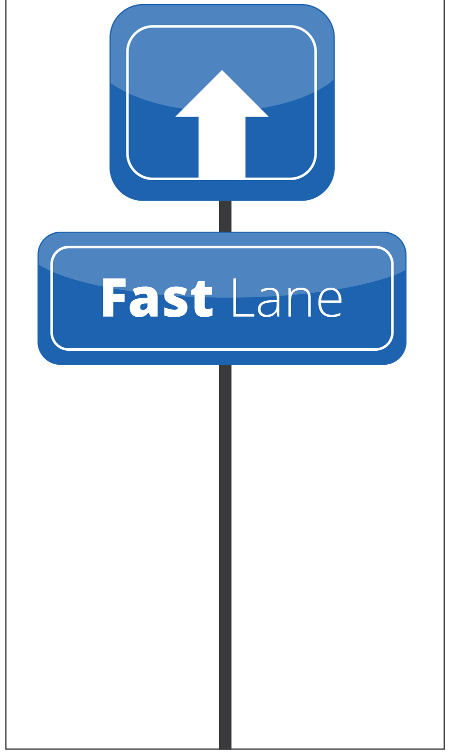
The reliance on **manual testing** is the **top technical challenge** in app development.

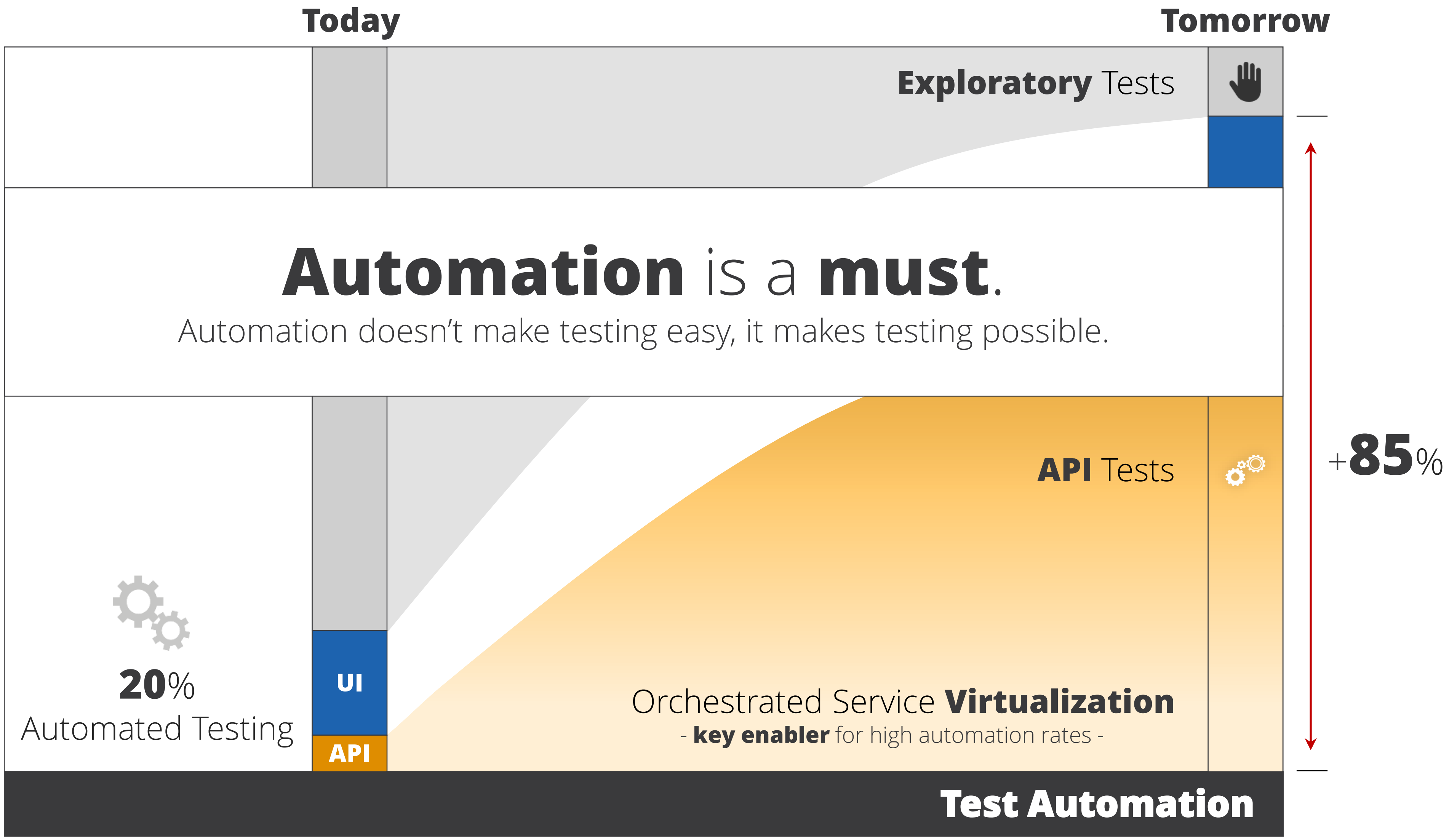
World Quality Report, 2015/16

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

Test Automation

Testing **Future**





Automation is a **must**.

Automation doesn't make testing easy, it makes testing possible.

Orchestrated Service **Virtualization**

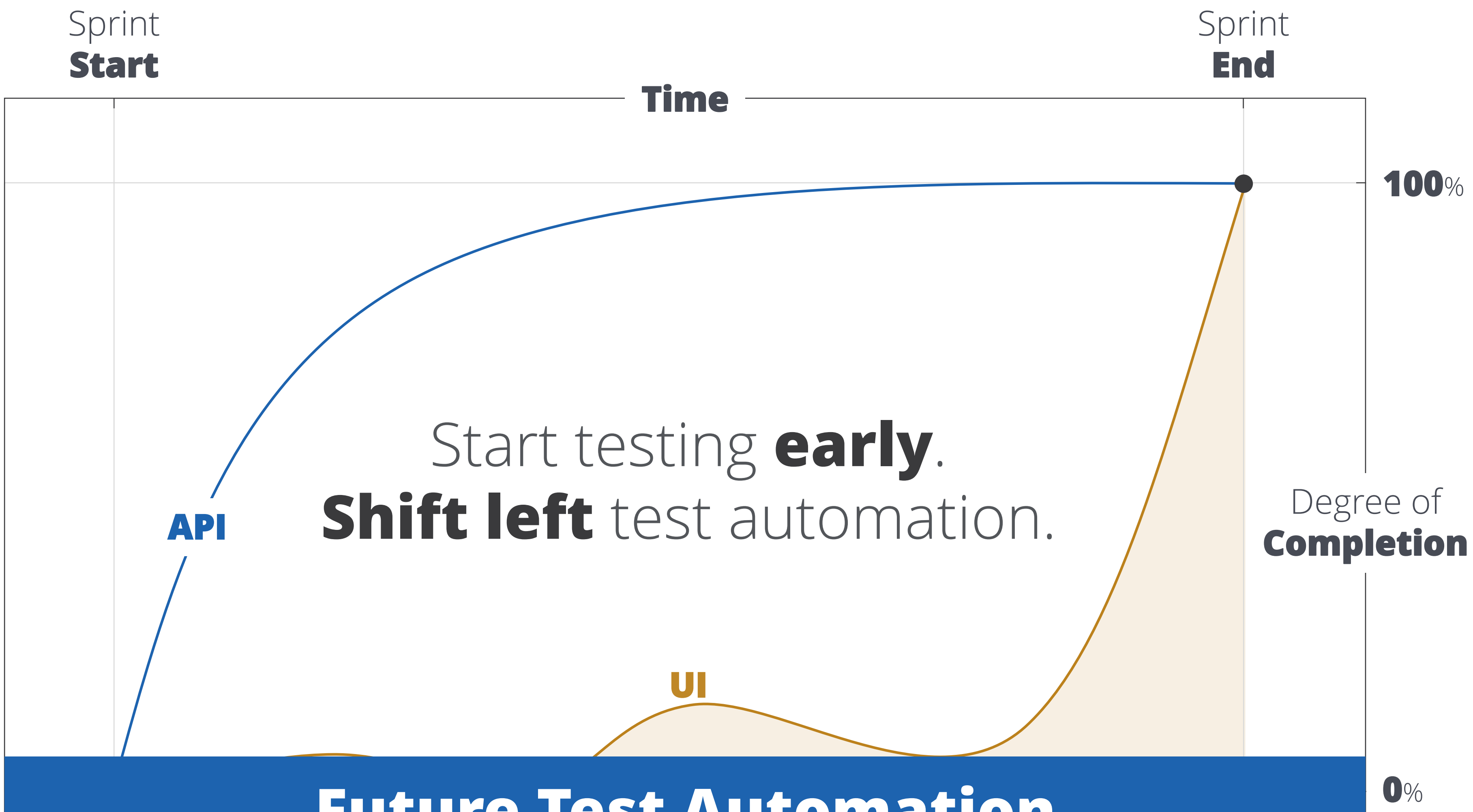
- **key enabler** for high automation rates -

Efficiency **Gain**

+4x
Creation

+6x
Maintenance

+20x
Execution



Start testing **early**.
Shift left test automation.

Future Test Automation
must focus on **API Testing**.

On average, organizations require access to **33** systems for development or testing.

Tosca **Orchestrate**

18

Average # of systems with **unrestricted access**

96

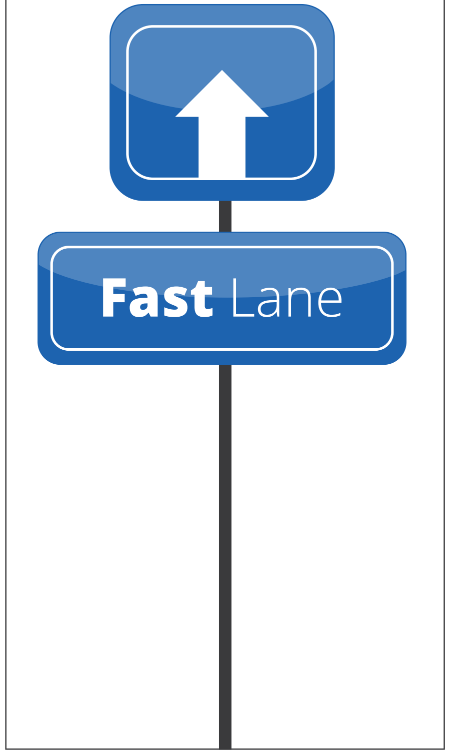
% of **testers** have restricted **test lab access**

Welcome to the **tester's hell**.

Orchestrated Service Virtualization

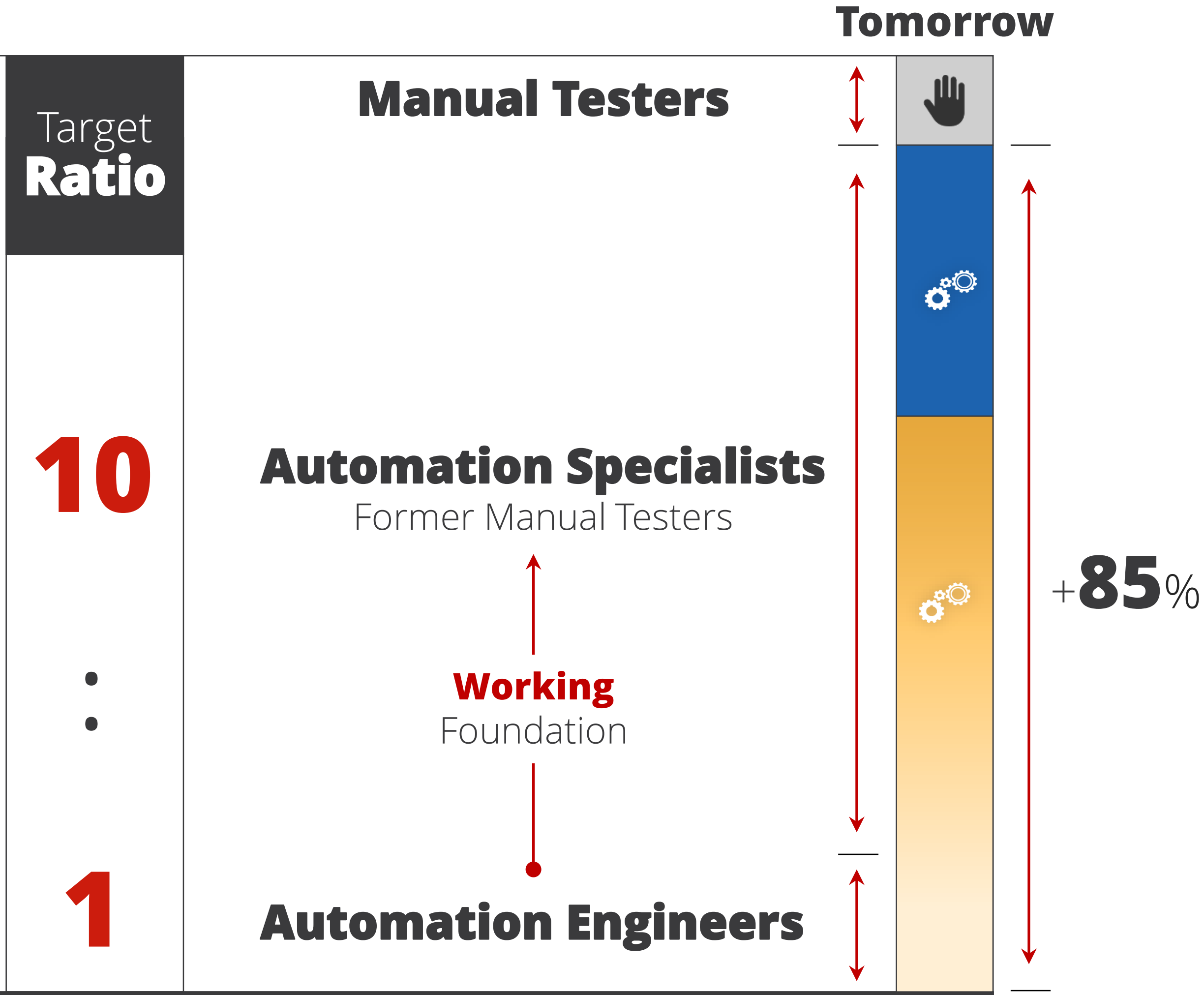
*voke, **market snapshot report** on service virtualization - 2012

Tosca **Automate**



It's **you**!

Who should bring that **glory**?



Tomorrow

Target **Ratio**

10

:

1

Manual Testers

Automation Specialists
Former Manual Testers

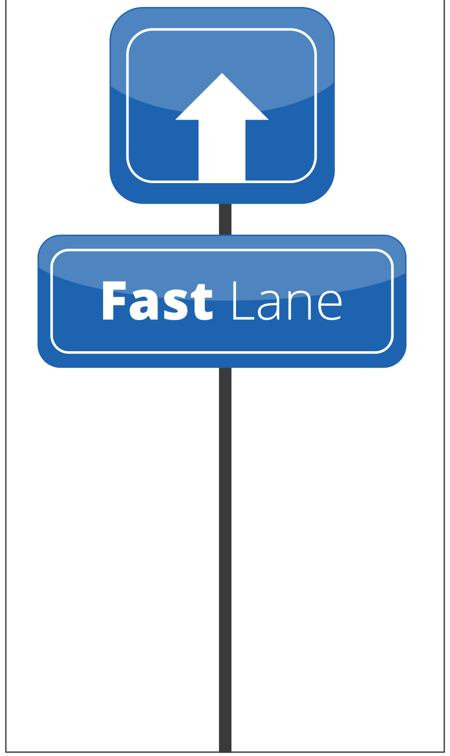
Working
Foundation

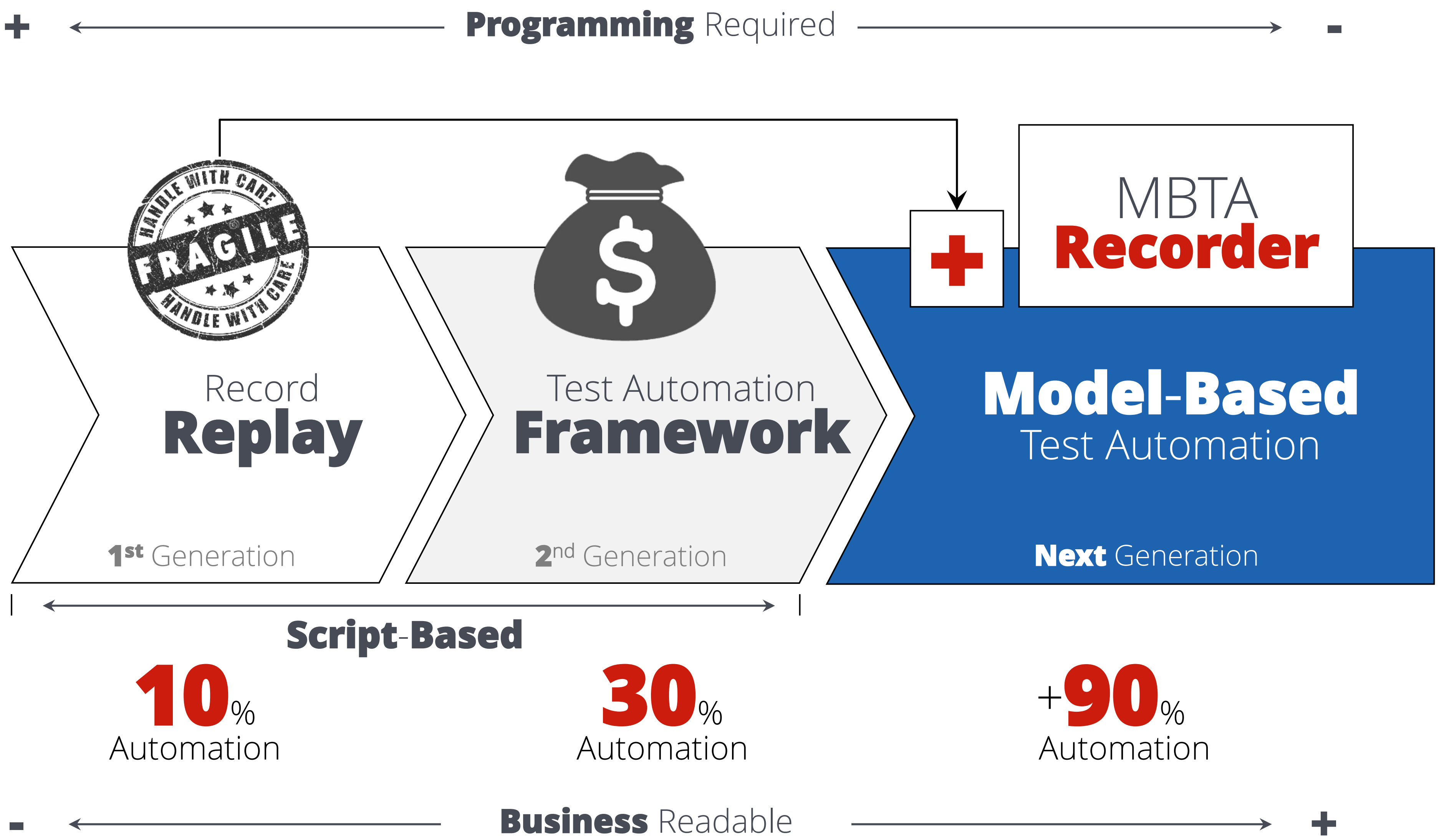
Automation Engineers

+**85**%

Manual Testers become **Automation Specialists**

Tosca
Automate





+ ← **Programming** Required → -

Record
Replay

1st Generation

Test Automation
Framework

2nd Generation

+ **MBTA**
Recorder

Model-Based
Test Automation

Next Generation

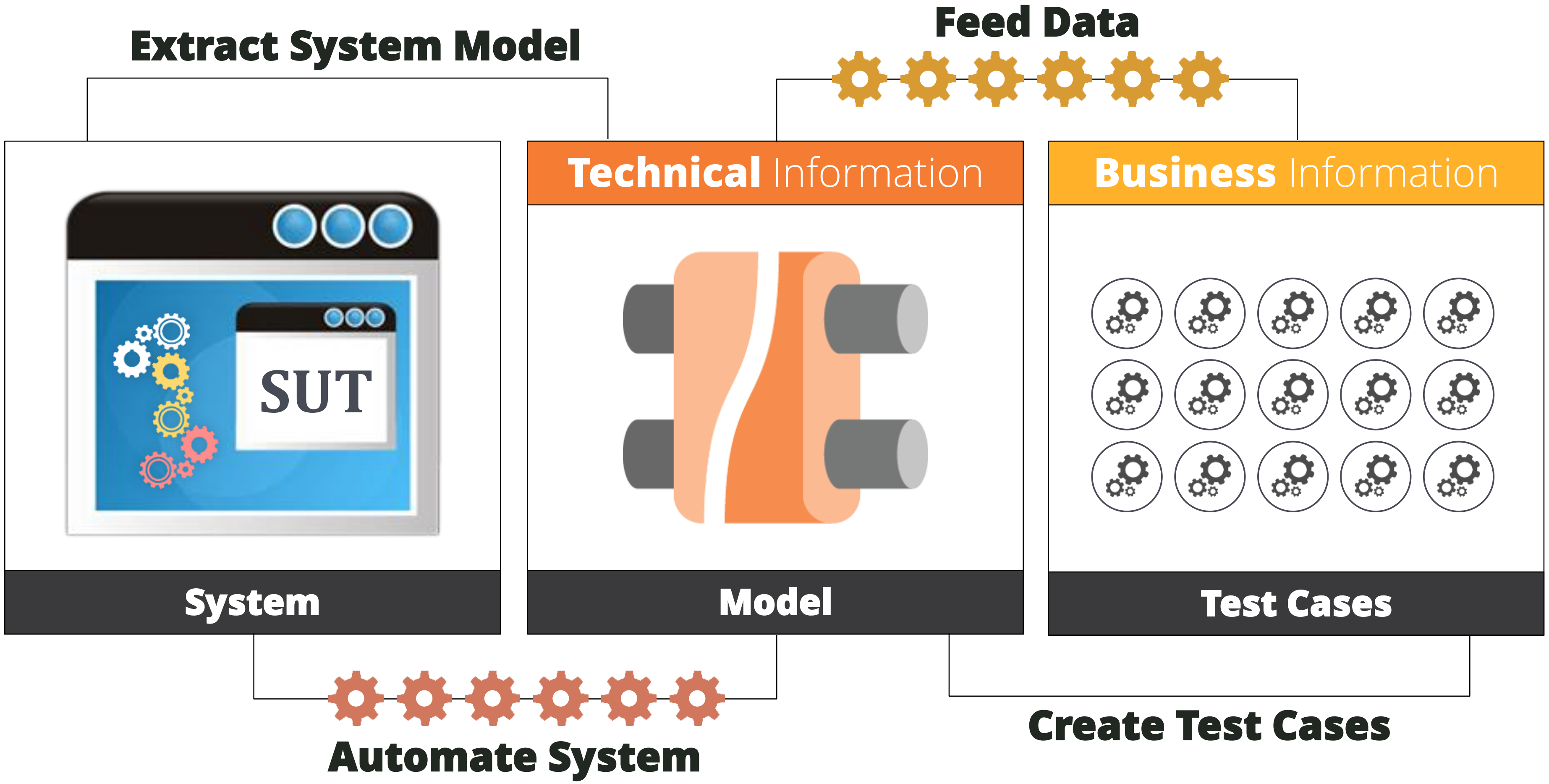
Script-Based

10%
Automation

30%
Automation

+**90**%
Automation

- ← **Business** Readable → +



Extract System Model

Feed Data

Technical Information

Business Information

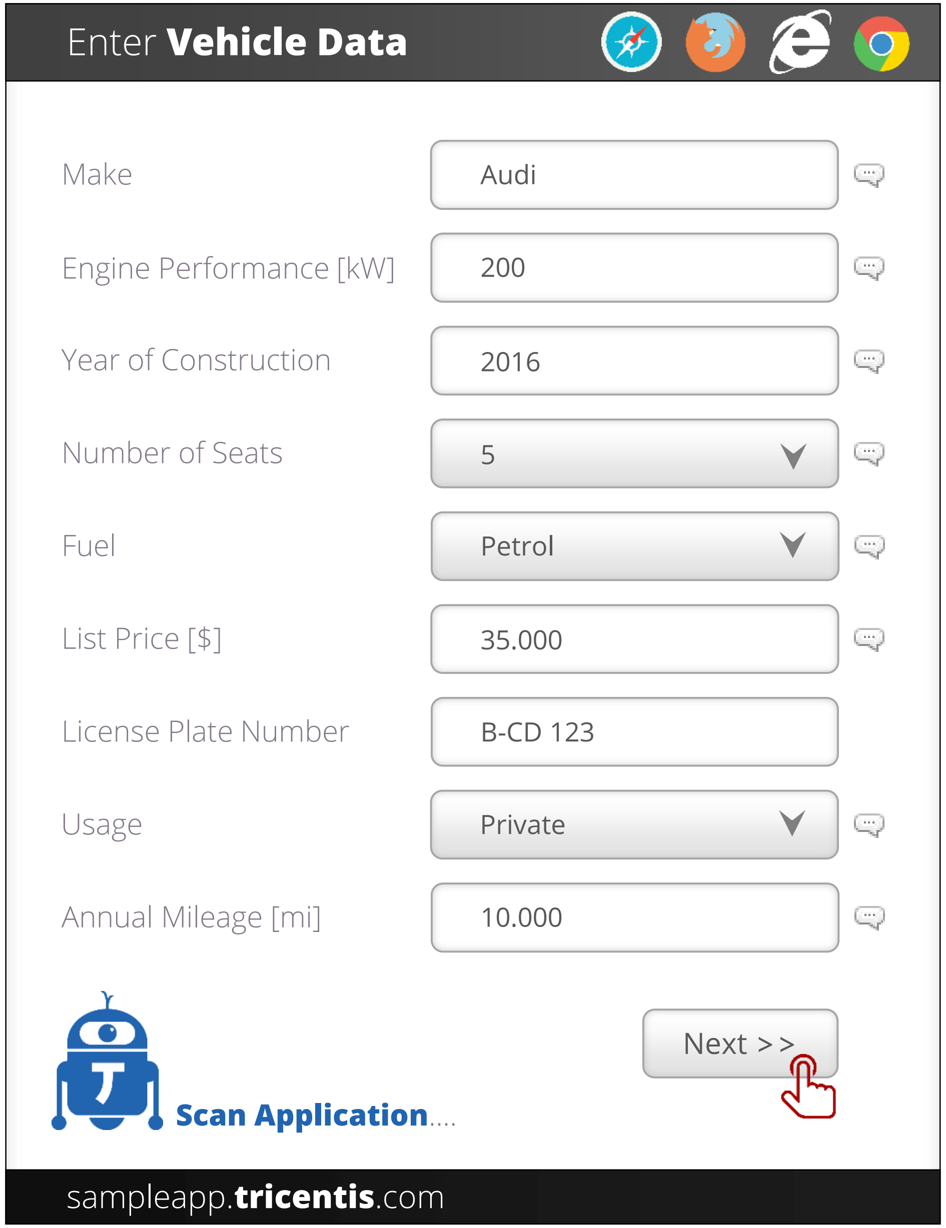
System

Model

Test Cases

Automate System

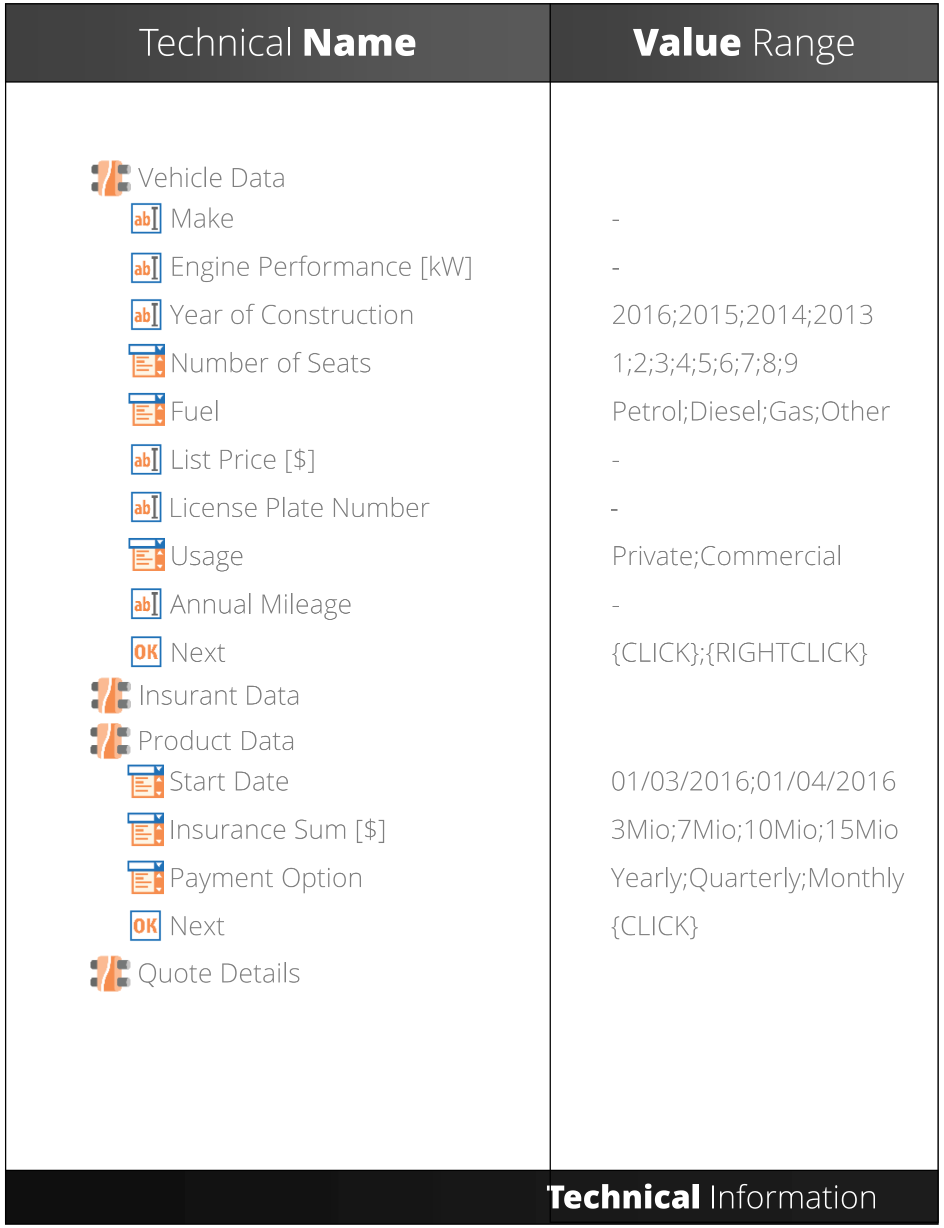
Create Test Cases



Enter **Vehicle Data**

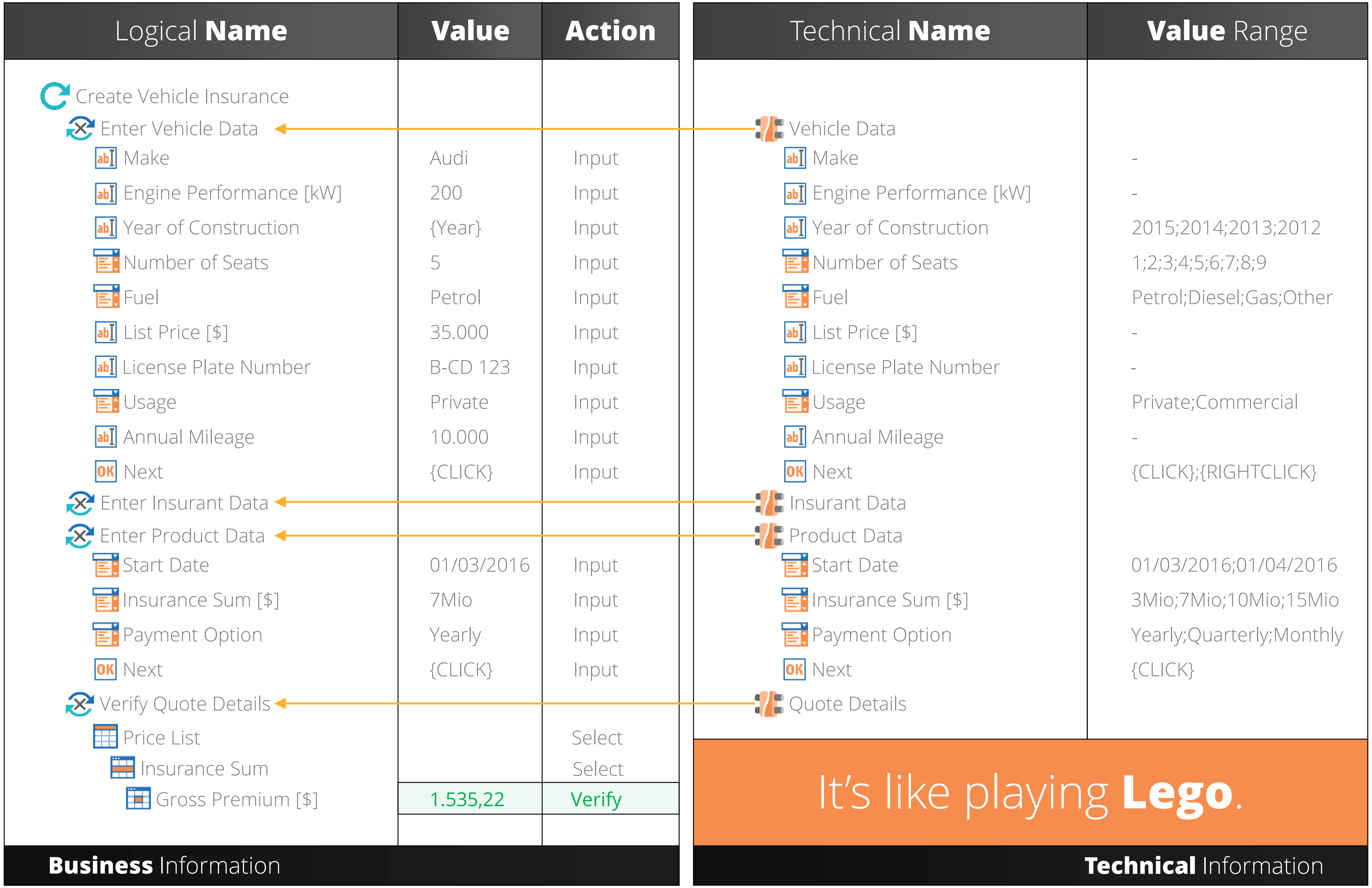
Make	Audi
Engine Performance [kW]	200
Year of Construction	2016
Number of Seats	5
Fuel	Petrol
List Price [$]	35.000
License Plate Number	B-CD 123
Usage	Private
Annual Mileage [mi]	10.000

sampleapp.**tricentis**.com



Technical **Name**	**Value** Range
Vehicle Data	
Make	-
Engine Performance [kW]	-
Year of Construction	2016;2015;2014;2013
Number of Seats	1;2;3;4;5;6;7;8;9
Fuel	Petrol;Diesel;Gas;Other
List Price [$]	-
License Plate Number	-
Usage	Private;Commercial
Annual Mileage	-
Next	{CLICK};{RIGHTCLICK}
Insurant Data	
Product Data	
Start Date	01/03/2016;01/04/2016
Insurance Sum [$]	3Mio;7Mio;10Mio;15Mio
Payment Option	Yearly;Quarterly;Monthly
Next	{CLICK}
Quote Details	

Technical Information



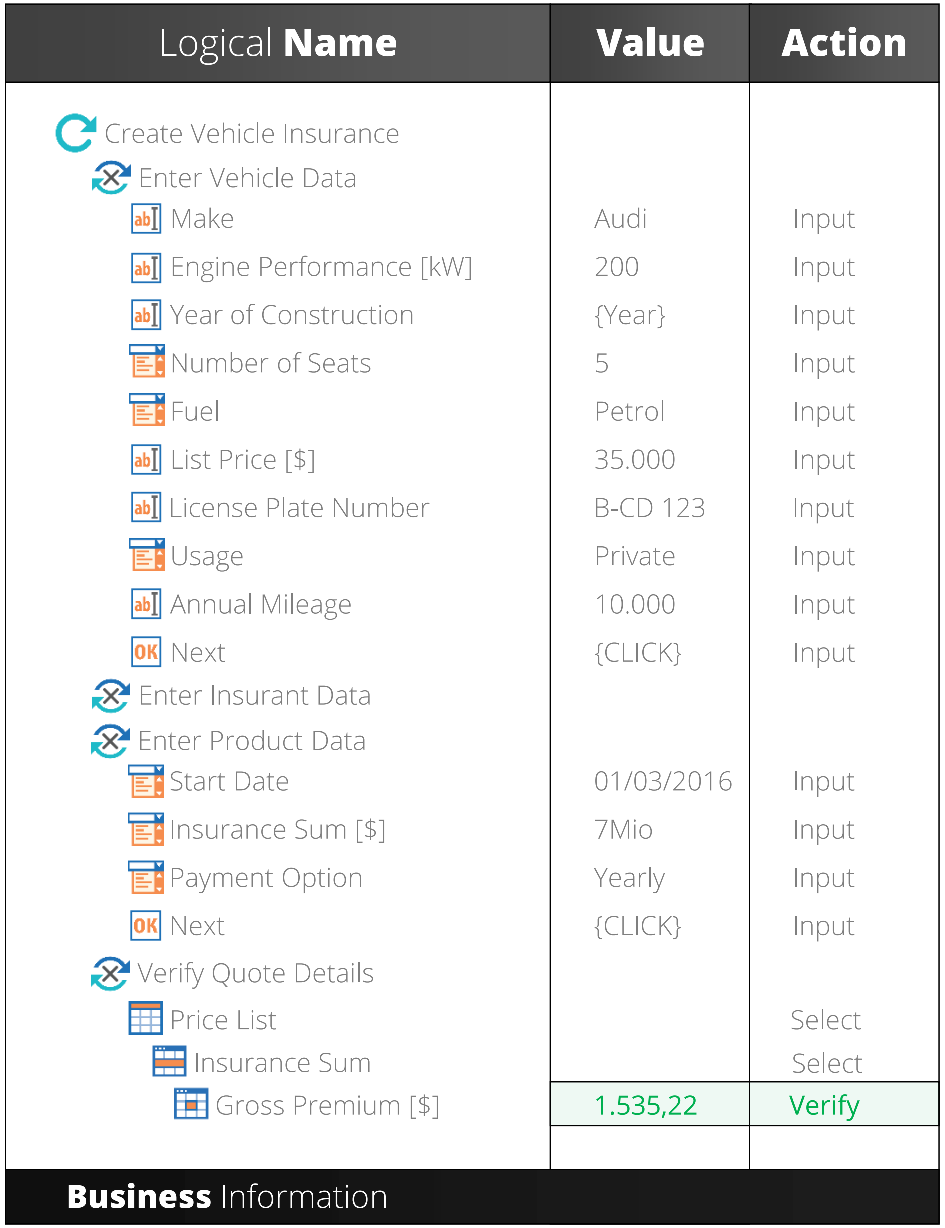
Logical Name	Value	Action
Create Vehicle Insurance		
Enter Vehicle Data		
Make	Audi	Input
Engine Performance [kW]	200	Input
Year of Construction	{Year}	Input
Number of Seats	5	Input
Fuel	Petrol	Input
List Price [$]	35.000	Input
License Plate Number	B-CD 123	Input
Usage	Private	Input
Annual Mileage	10.000	Input
Next	{CLICK}	Input
Enter Insurant Data		
Enter Product Data		
Start Date	01/03/2016	Input
Insurance Sum [$]	7Mio	Input
Payment Option	Yearly	Input
Next	{CLICK}	Input
Verify Quote Details		
Price List		Select
Insurance Sum		Select
Gross Premium [$]	1.535,22	Verify

Business Information

Technical Name	Value Range
Vehicle Data	
Make	-
Engine Performance [kW]	-
Year of Construction	2015;2014;2013;2012
Number of Seats	1;2;3;4;5;6;7;8;9
Fuel	Petrol;Diesel;Gas;Other
List Price [$]	-
License Plate Number	-
Usage	Private;Commercial
Annual Mileage	-
Next	{CLICK};{RIGHTCLICK}
Insurant Data	
Product Data	
Start Date	01/03/2016;01/04/2016
Insurance Sum [$]	3Mio;7Mio;10Mio;15Mio
Payment Option	Yearly;Quarterly;Monthly
Next	{CLICK}
Quote Details	

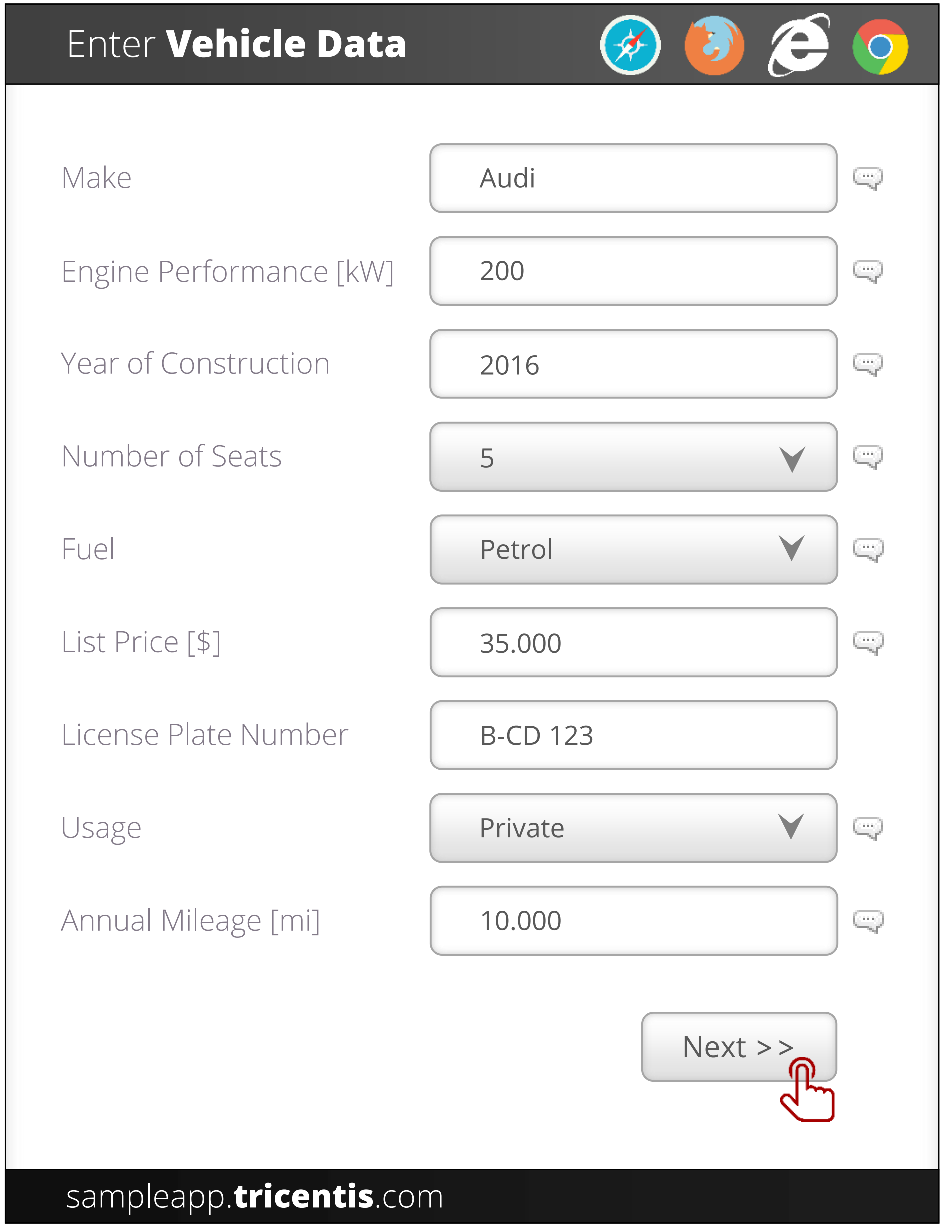
It's like playing **Lego**.

Technical Information



Logical **Name**	**Value**	**Action**
Create Vehicle Insurance		
Enter Vehicle Data		
Make	Audi	Input
Engine Performance [kW]	200	Input
Year of Construction	{Year}	Input
Number of Seats	5	Input
Fuel	Petrol	Input
List Price [$]	35.000	Input
License Plate Number	B-CD 123	Input
Usage	Private	Input
Annual Mileage	10.000	Input
Next	{CLICK}	Input
Enter Insurant Data		
Enter Product Data		
Start Date	01/03/2016	Input
Insurance Sum [$]	7Mio	Input
Payment Option	Yearly	Input
Next	{CLICK}	Input
Verify Quote Details		
Price List		Select
Insurance Sum		Select
Gross Premium [$]	1.535,22	Verify

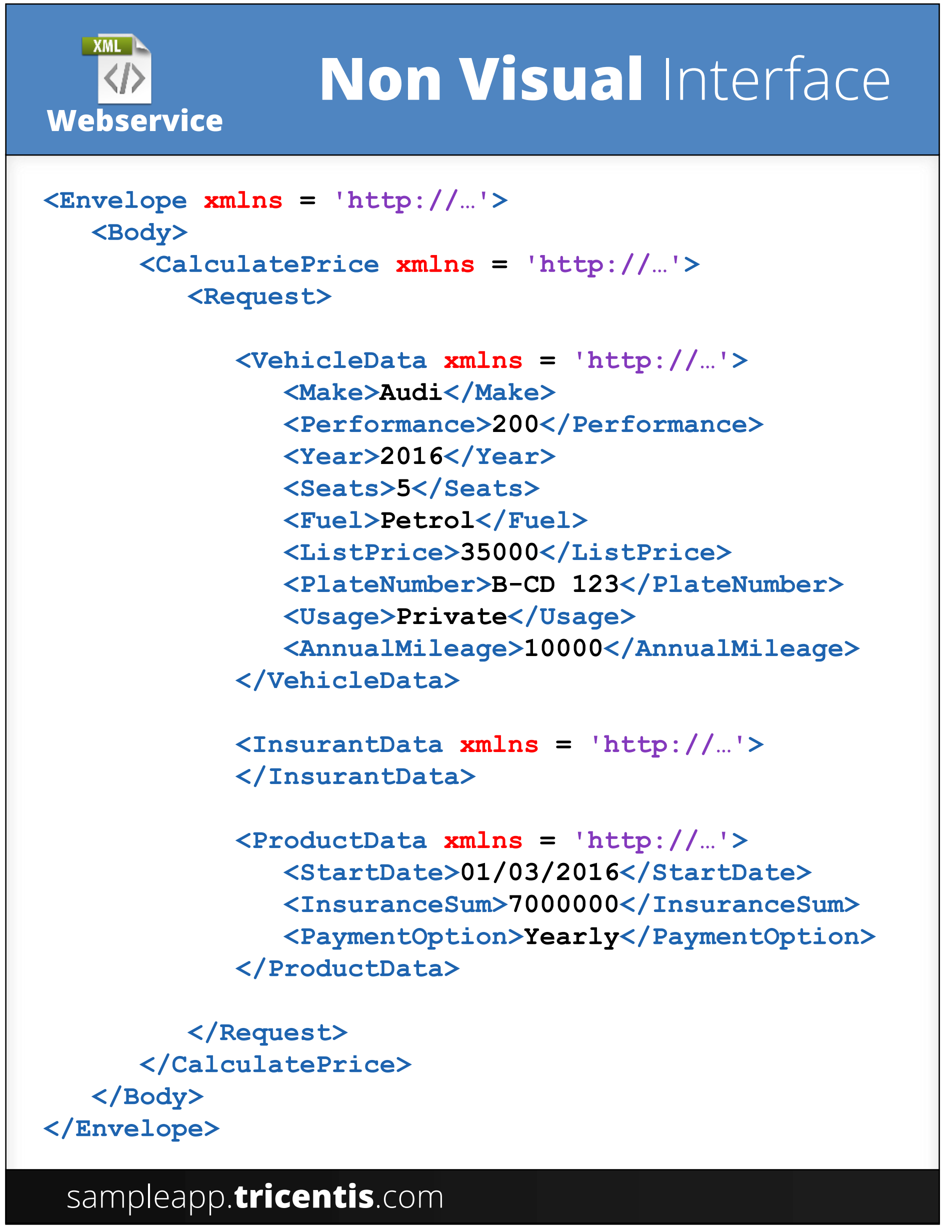
Business Information



Enter **Vehicle Data**

Make	Audi
Engine Performance [kW]	200
Year of Construction	2016
Number of Seats	5
Fuel	Petrol
List Price [$]	35.000
License Plate Number	B-CD 123
Usage	Private
Annual Mileage [mi]	10.000

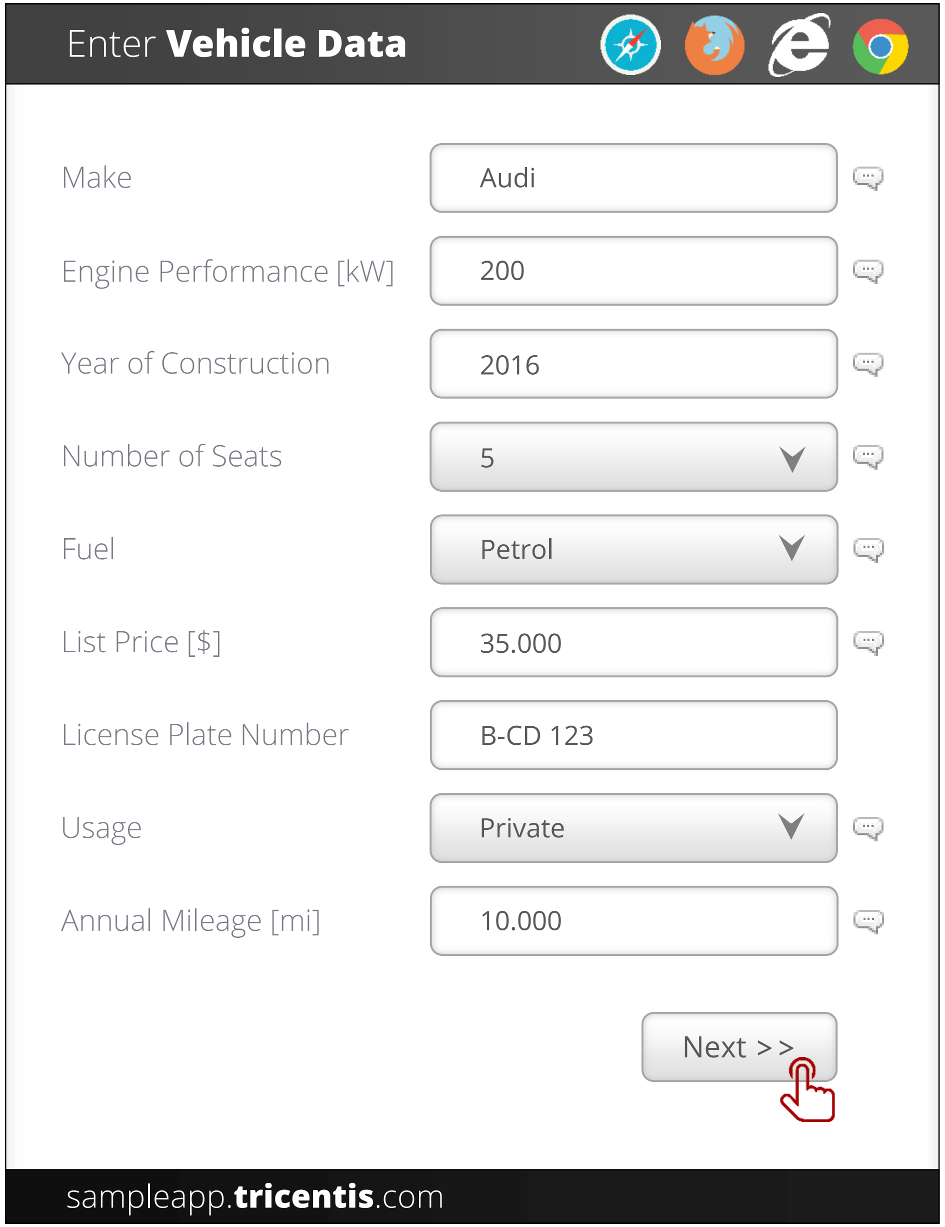
sampleapp.**tricentis**.com



Non Visual Interface

```
<Envelope xmlns = 'http://…'>
   <Body>
      <CalculatePrice xmlns = 'http://…'>
         <Request>

            <VehicleData xmlns = 'http://…'>
               <Make>Audi</Make>
               <Performance>200</Performance>
               <Year>2016</Year>
               <Seats>5</Seats>
               <Fuel>Petrol</Fuel>
               <ListPrice>35000</ListPrice>
               <PlateNumber>B-CD 123</PlateNumber>
               <Usage>Private</Usage>
               <AnnualMileage>10000</AnnualMileage>
            </VehicleData>

            <InsurantData xmlns = 'http://…'>
            </InsurantData>

            <ProductData xmlns = 'http://…'>
               <StartDate>01/03/2016</StartDate>
               <InsuranceSum>7000000</InsuranceSum>
               <PaymentOption>Yearly</PaymentOption>
            </ProductData>

         </Request>
      </CalculatePrice>
   </Body>
</Envelope>
```

sampleapp.**tricentis**.com



Enter **Vehicle Data**

Make	Audi
Engine Performance [kW]	200
Year of Construction	2016
Number of Seats	5
Fuel	Petrol
List Price [$]	35.000
License Plate Number	B-CD 123
Usage	Private
Annual Mileage [mi]	10.000

sampleapp.**tricentis**.com

Webservice **Non Visual** Interface

```
<Envelope xmlns = 'http://…'>
   <Body>
      <CalculatePrice xmlns = 'http://…'>
         <Request>

            <VehicleData xmlns = 'http://…'>
               <Make>Audi</Make>
               <Performance>200</Performance>
               <Year>2016</Year>
               <Seats>5</Seats>
               <Fuel>Petrol</Fuel>
               <ListPrice>35000</ListPrice>
               <PlateNumber>B-CD 123</PlateNumber>
               <Usage>Private</Usage>
               <AnnualMileage>10000</AnnualMileage>
            </VehicleData>

            <InsurantData xmlns = 'http://…'>
            </InsurantData>

            <ProductData xmlns = 'http://…'>
               <StartDate>01/03/2016</StartDate>
               <InsuranceSum>7000000</InsuranceSum>
               <PaymentOption>Yearly</PaymentOption>
            </ProductData>

         </Request>
      </CalculatePrice>
   </Body>
</Envelope>
```

sampleapp.**tricentis**.com

API Test Case Webservice

Enter Vehicle Data	
Make	Audi
Engine Performance [kW]	200
Year of Construction	{Year}
Number of Seats	5
Fuel	Petrol
List Price [$]	35.000
License Plate Number	B-CD 123
Usage	Private
Annual Mileage	10.000
Enter Insurant Data	
Enter Product Data	
Start Date	01/03/2016
Insurance Sum [$]	7Mio
Payment Option	Yearly
Verify Quote Details	
Price List	
Insurance Sum	
Gross Premium [$]	1.535,22

Webservice **Request**

Webservice **Response**

Business Information

UI Test Case

HTML 5 **Browser**

Enter Vehicle Data		
Make	Audi	Input
Engine Performance [kW]	200	Input
Year of Construction	{Year}	Input
Number of Seats	5	Input
Fuel	Petrol	Input
List Price [$]	35.000	Input
License Plate Number	B-CD 123	Input
Usage	Private	Input
Insurance Sum [$]	7Mio	Input
Payment Option	Yearly	Input
Next	{CLICK}	Input
Verify Quote Details		
Price List		Select
Insurance Sum		Select
Gross Premium [$]	1.535,22	Verify

Business Information

API Test Case

XML **Webservice**

Enter Vehicle Data	
Make	Audi
Engine Performance [kW]	200
Year of Construction	{Year}
Number of Seats	5
Fuel	Petrol
List Price [$]	35.000
License Plate Number	B-CD 123
Usage	Private
Insurance Sum [$]	7Mio
Payment Option	Yearly
Verify Quote Details	
Price List	
Insurance Sum	
Gross Premium [$]	1.535,22

Business Information

Manual Testers become **Automation Specialists**

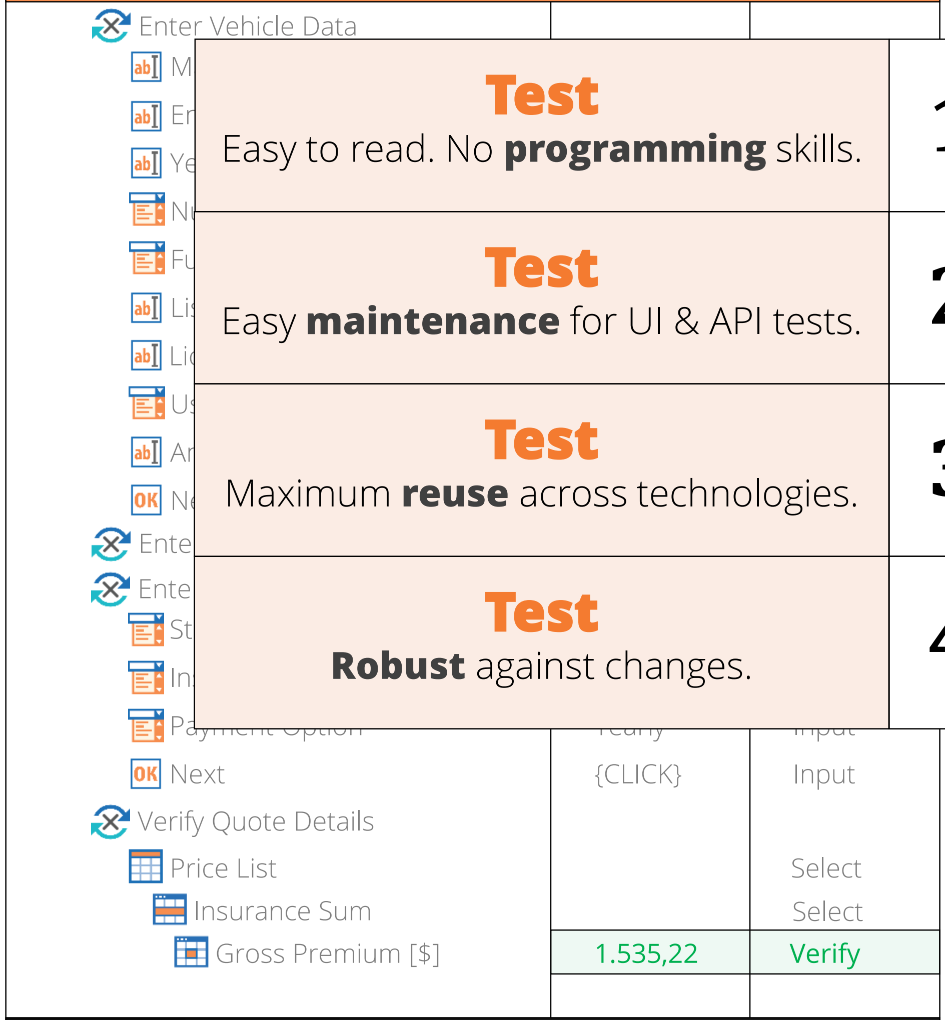
Model-Based

HTML 5 Browser

Script-Based

HTML 5 Browser

Test Easy to read. No **programming** skills.	1	**Code** **Synchronization** Behavior
Test Easy **maintenance** for UI & API tests.	2	**Code** **Recovery** Handling
Test Maximum **reuse** across technologies.	3	**Code** **Data Driven** Framework
Test **Robust** against changes.	4	**Code** **Keyword Driven** Framework



Enter Vehicle Data		
M		
Er		
Ye		
N		
Fu		
Li		
Lic		
Us		
Ar		
Ne		
Ente		
Ente		
St		
In		
Payment Option	Yearly	Input
Next	{CLICK}	Input
Verify Quote Details		
Price List		Select
Insurance Sum		Select
Gross Premium [$]	1.535,22	Verify

```
Assert.AreEqual("1.535,22",
driver.FindElement(By.XPath("//table[@id='PriceList']/tbody/tr[10]/td
[2]/b"))
.GetAttribute("value"));
```

Business Information

Do you see the **business** information?

Testers

11

Tests

4755

Optimize

Value Based
Test Cases

1193

Redundancy 75%

Automate

Model Based
Automation

Automation Level

Integrate

Continuous
Integration

Distributed Smoke Testing

Automation Level Smoke Testing

Remodeling **traditional approaches** for greater **agility**.

10
Weeks

Execution

2.5
Weeks

Test Data
Management

48
Hours

Single Agent Regression

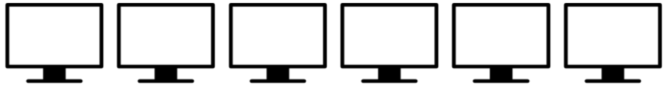
8
Hours

Distributed Regression

Test Case Count 7%

53%

Risk Coverage Smoke Testing



Tosca
Universe

Continuous Testing is a must.

Do the **right things** – and **do things right**.

Manual Testers become **Automation Specialists**.

Our **Mantra**