TRICENTIS

AN

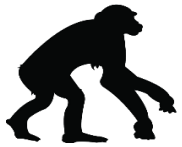
Agile Testing Survival Guide

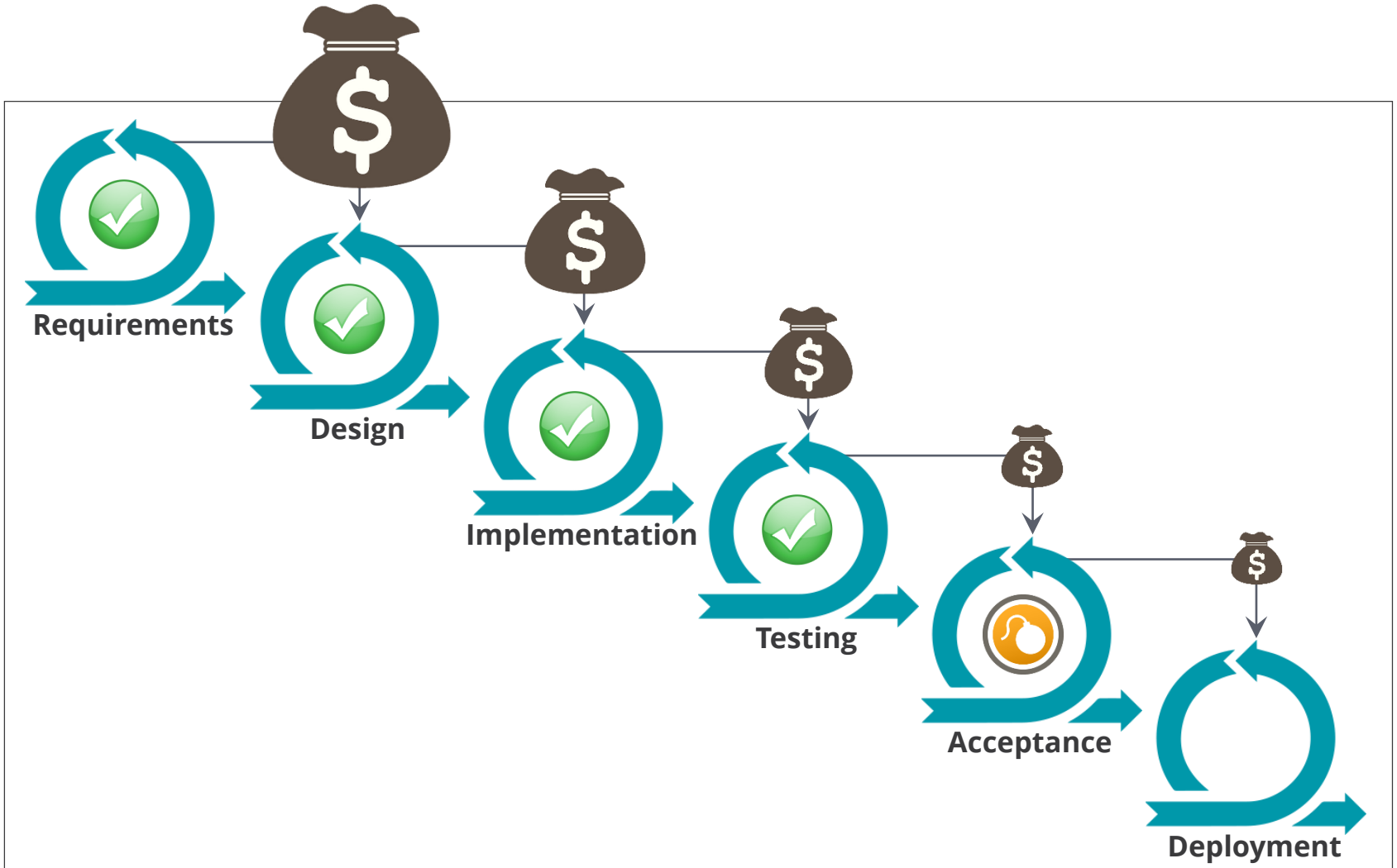
How to build in **quality** & **efficiency** right from the start?

Ingo **Philipp**

Cycle Time

Years
Months

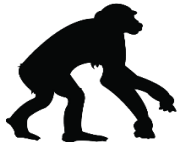




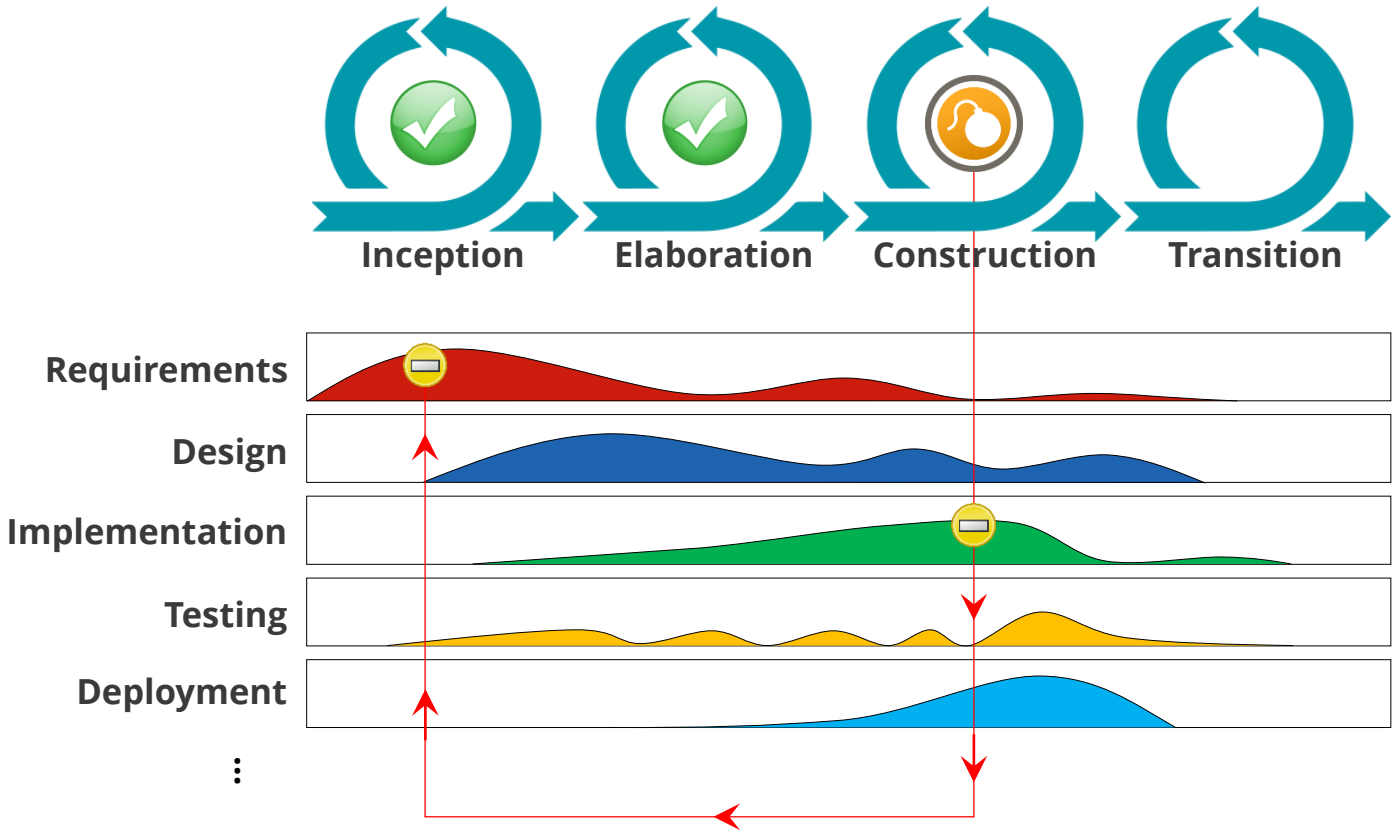
Waterfall

Cycle Time

Years
Months



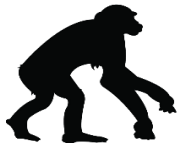
Months
Weeks



Rational Unified Process

Cycle Time

Years Months



Months Weeks

Weeks Days

Siloisation

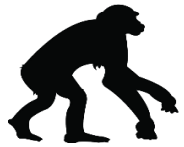
"Where we are right now **just sucks**."

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

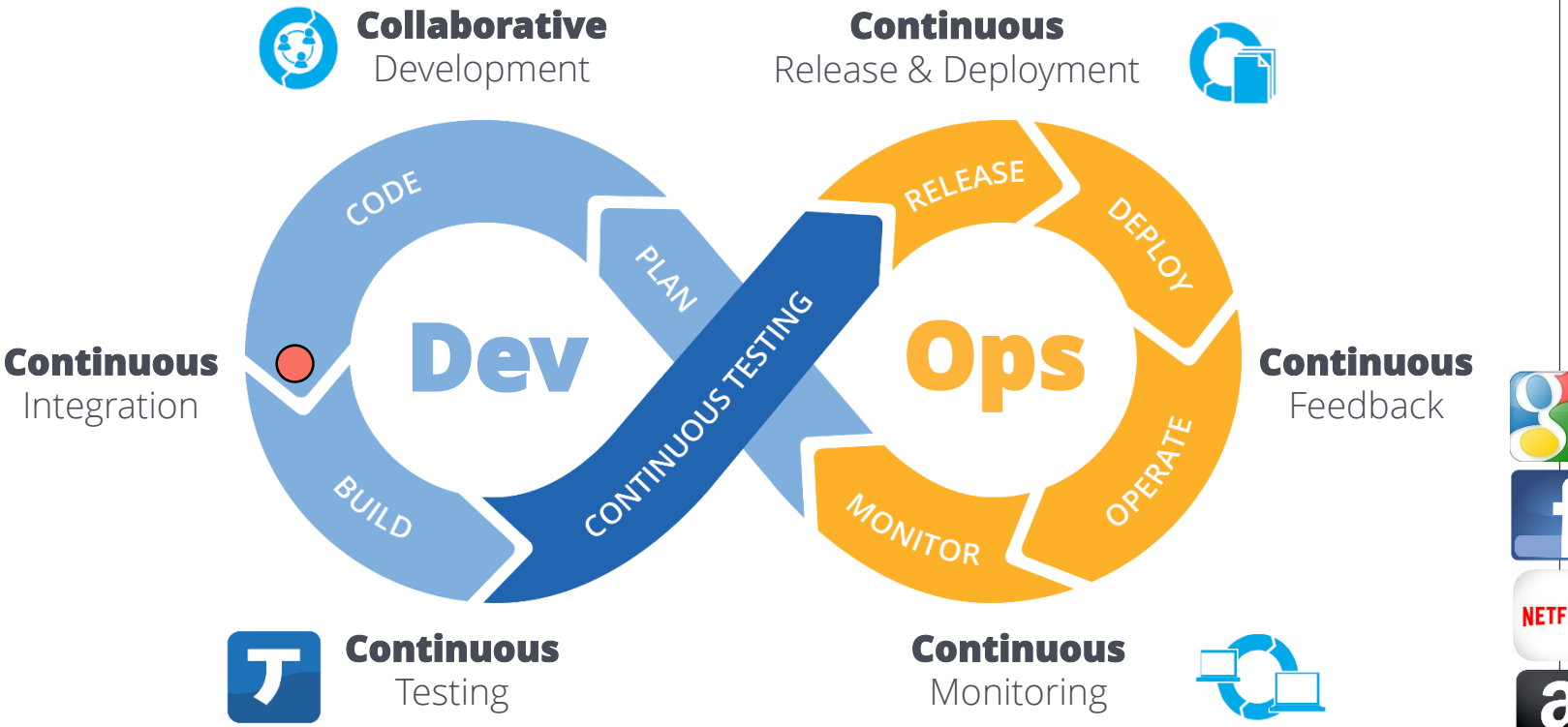


Years
Months

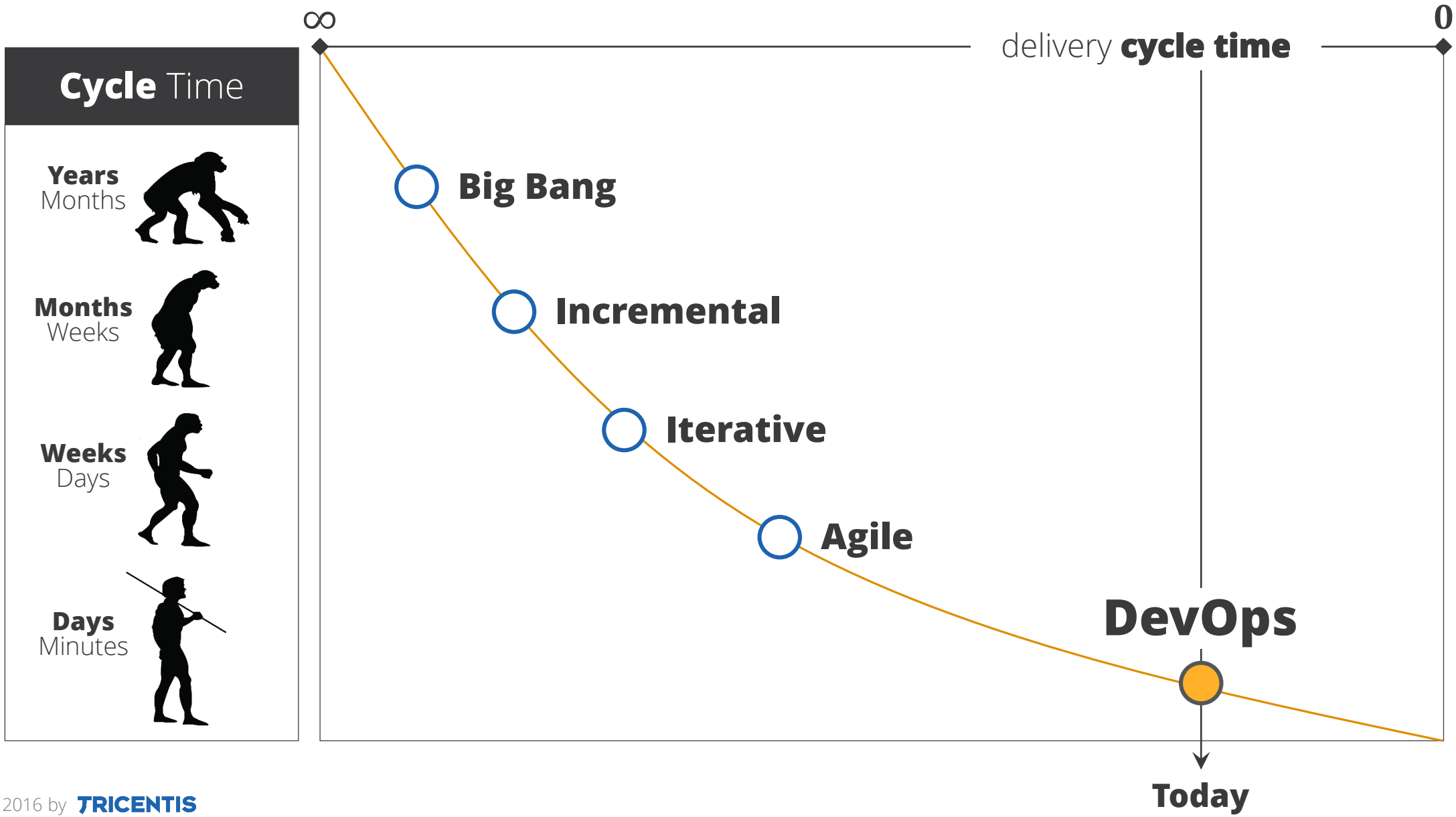
Months
Weeks

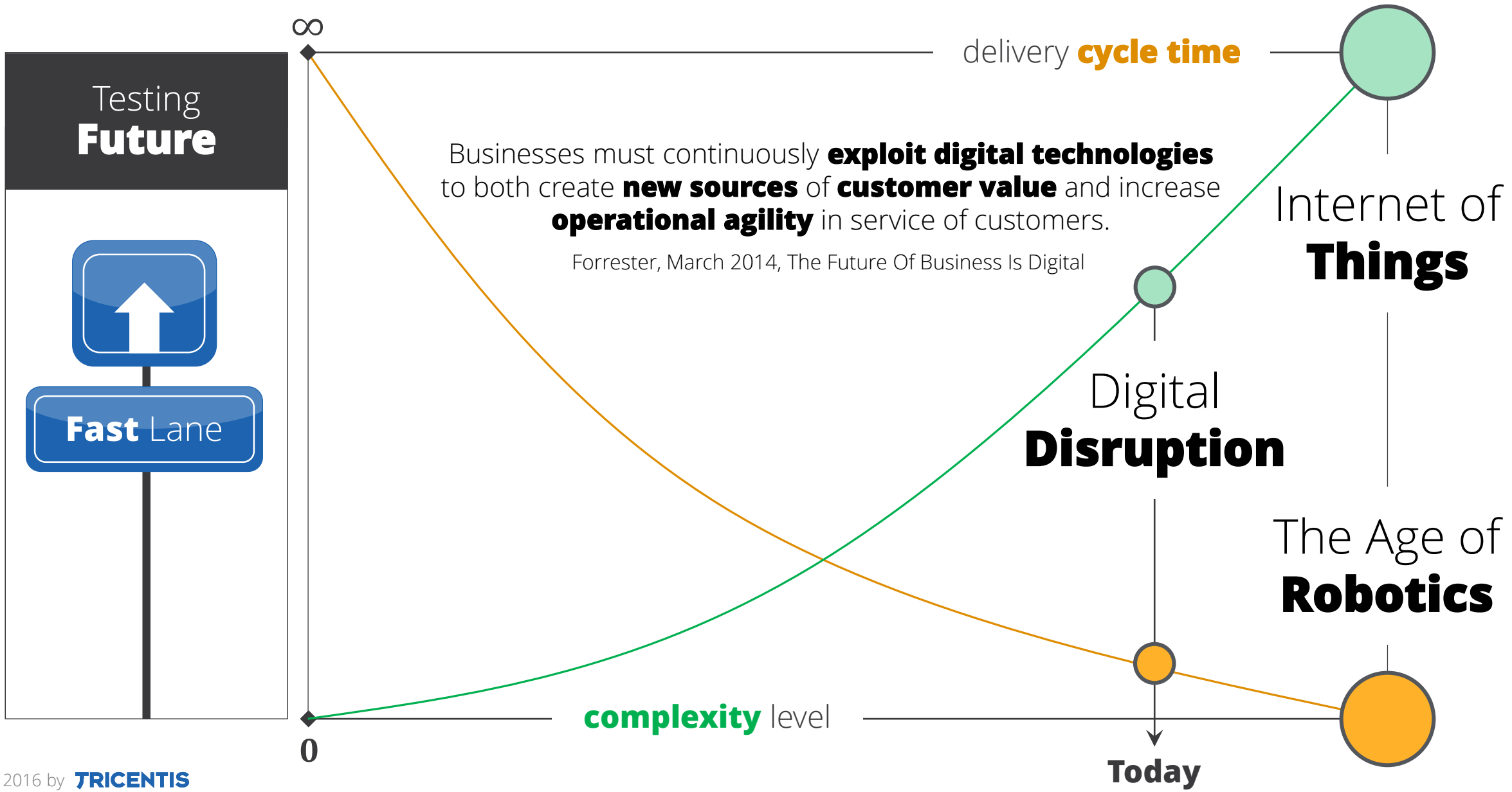
Weeks
Days

Days
Minutes

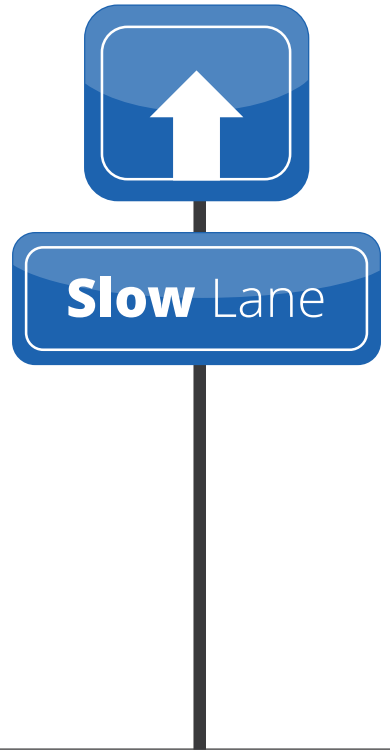


DevOps





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

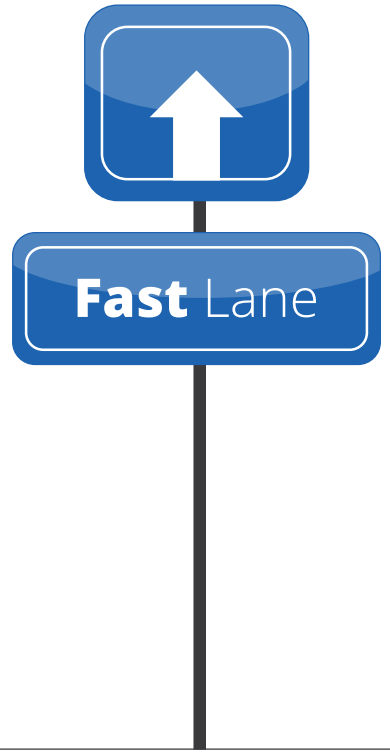
56% of overall test effort goes into test case **maintenance**

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

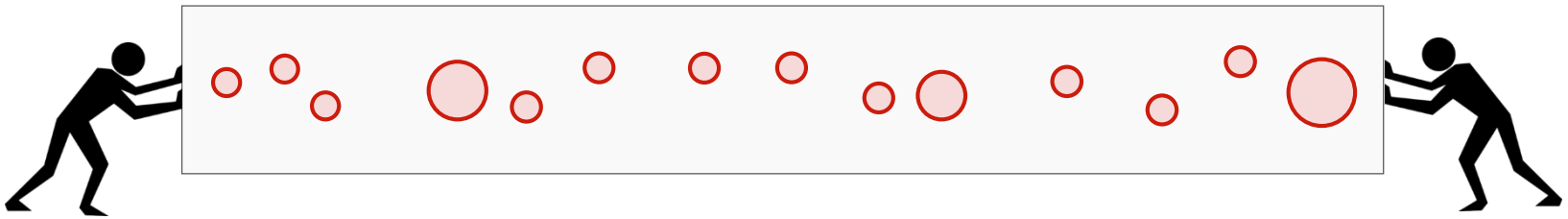
50% of manual testing goes into **test data** preparation & organization

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

Testing
Future

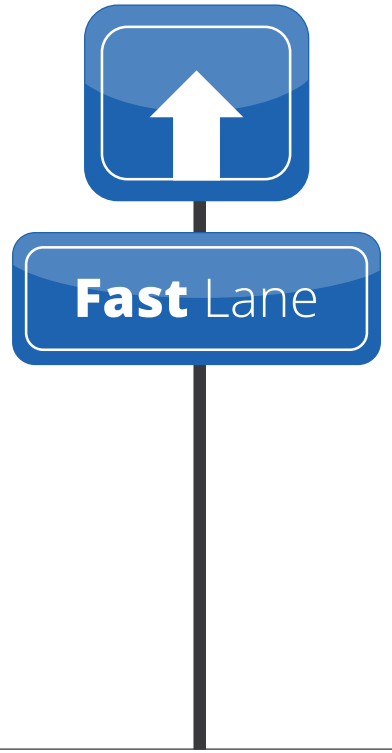


Start software **delivery cycle** **End**



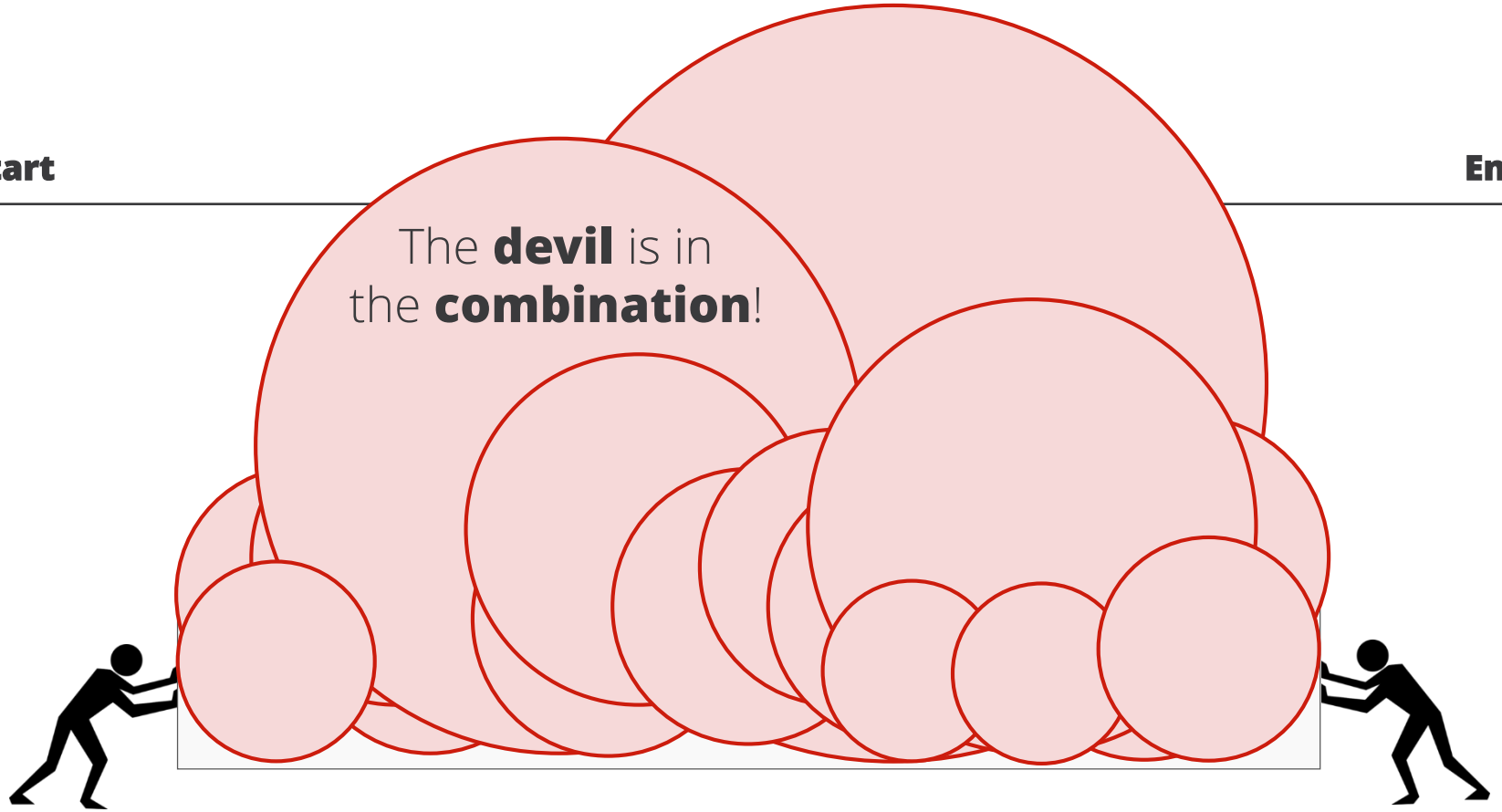
By **compressing** the delivery cycle,
do problems just move closer to each other?

Testing
Future



Start **End**

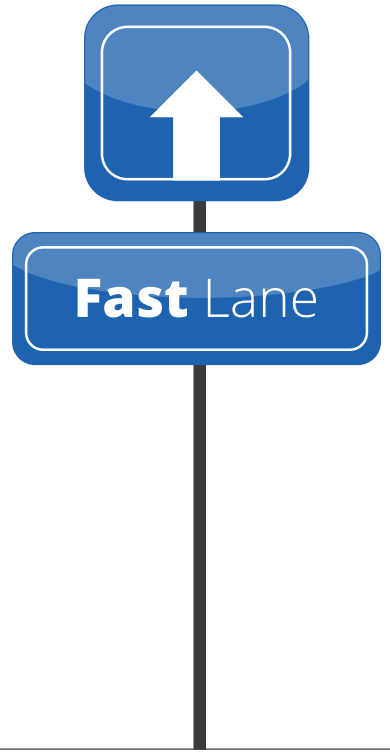
The **devil** is in
the **combination**!



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.

doing **right** things

Effectiveness

Automate

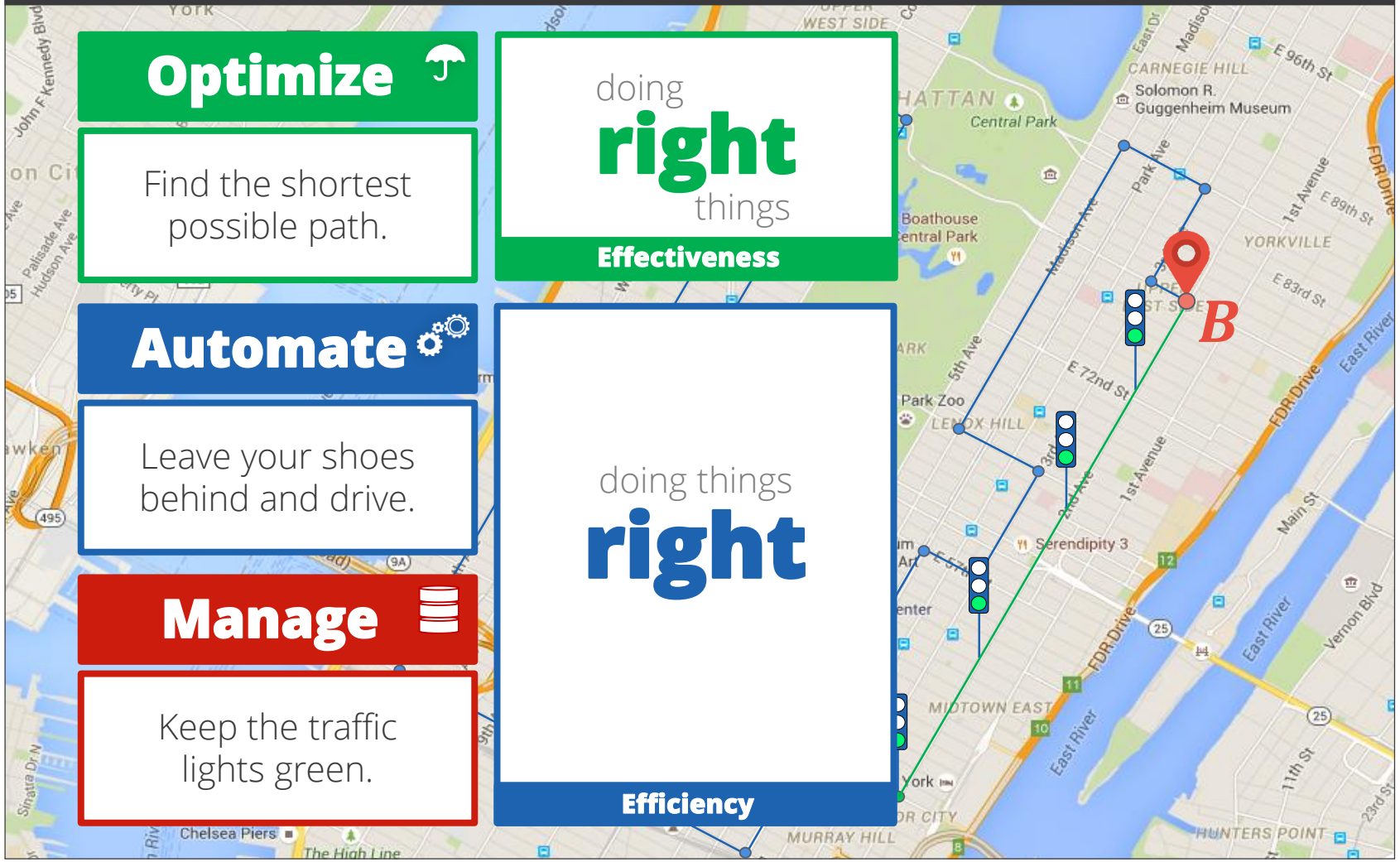
Leave your shoes behind and drive.

doing things **right**

Efficiency

Manage

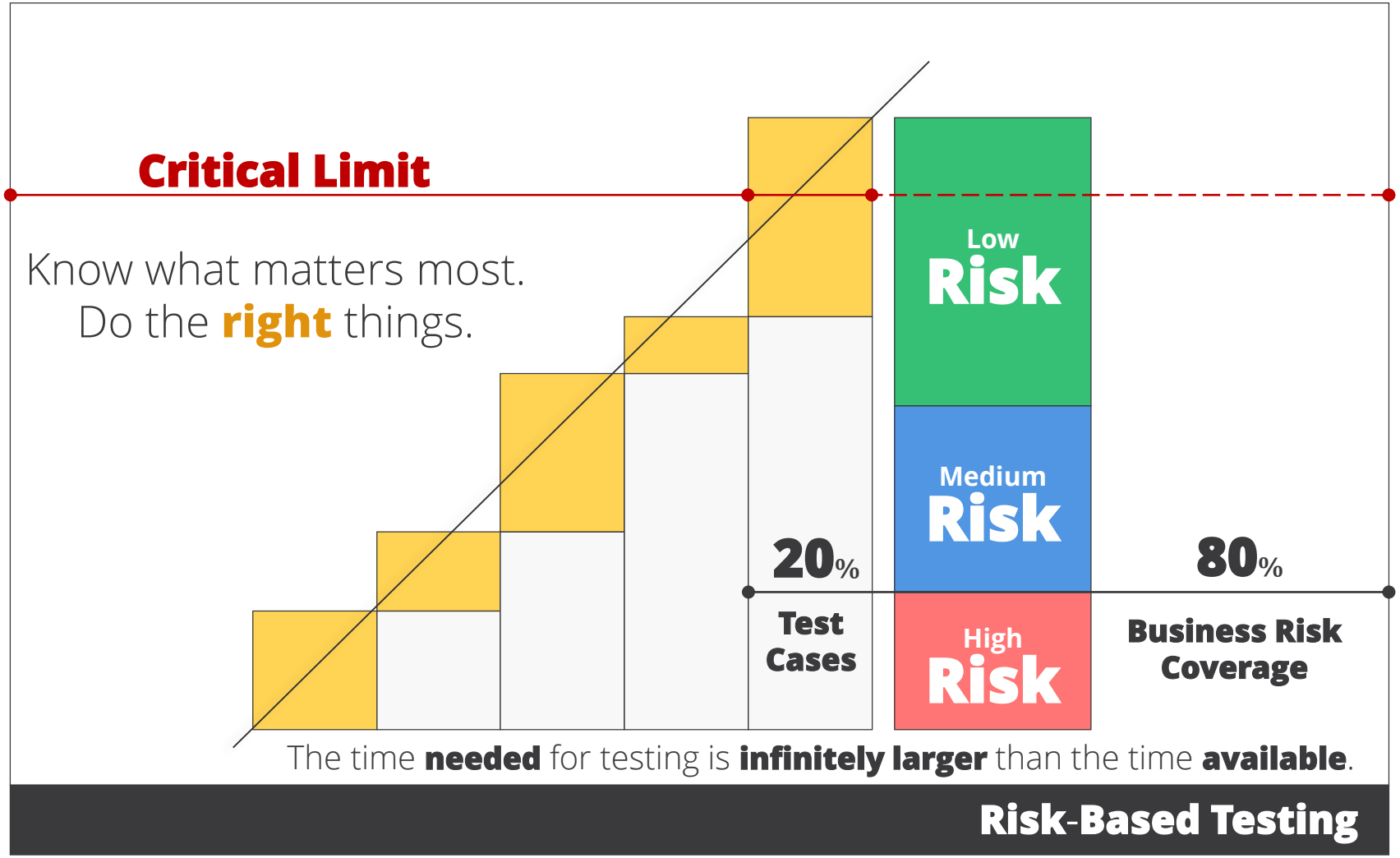
Keep the traffic lights green.

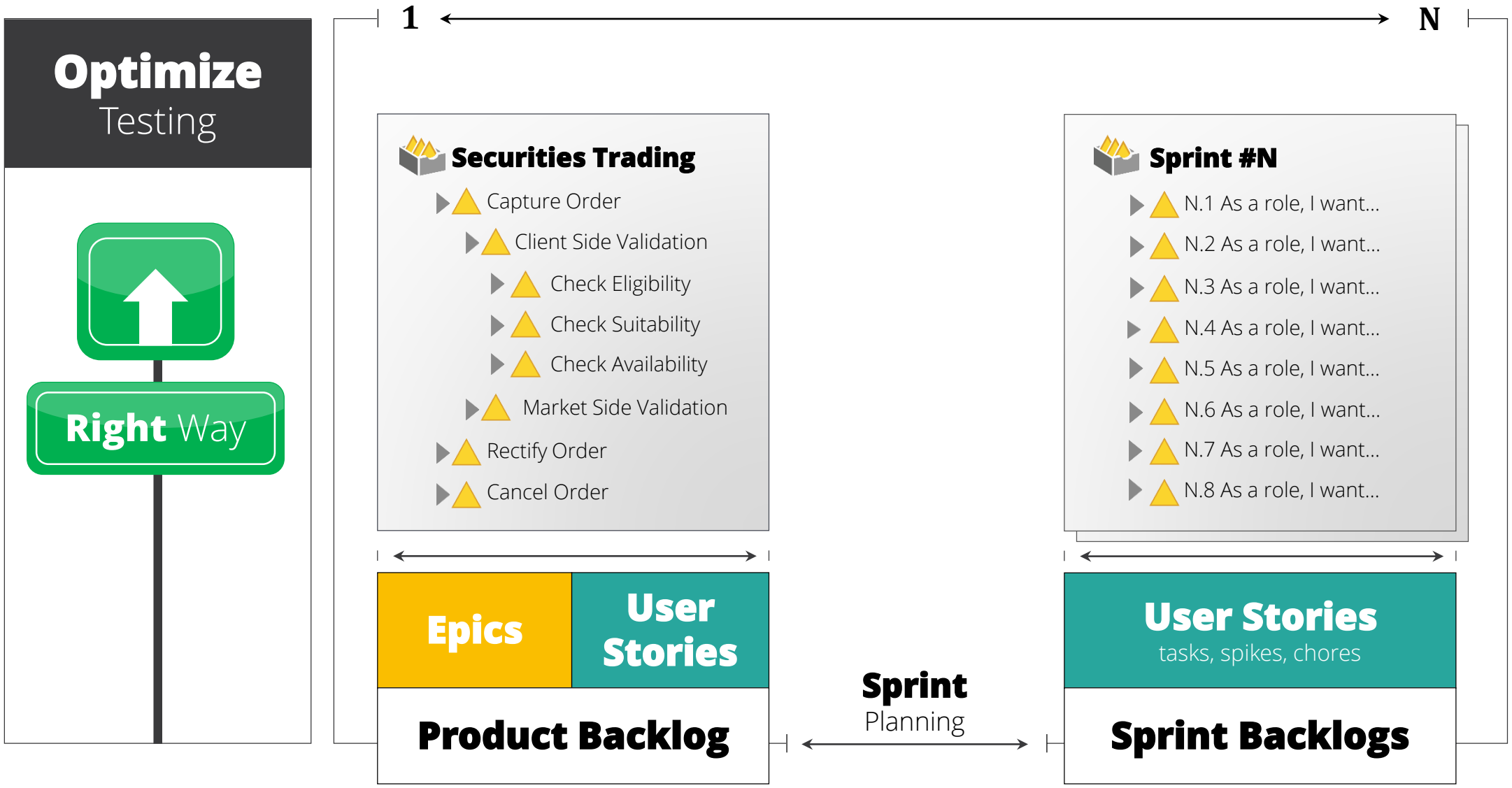


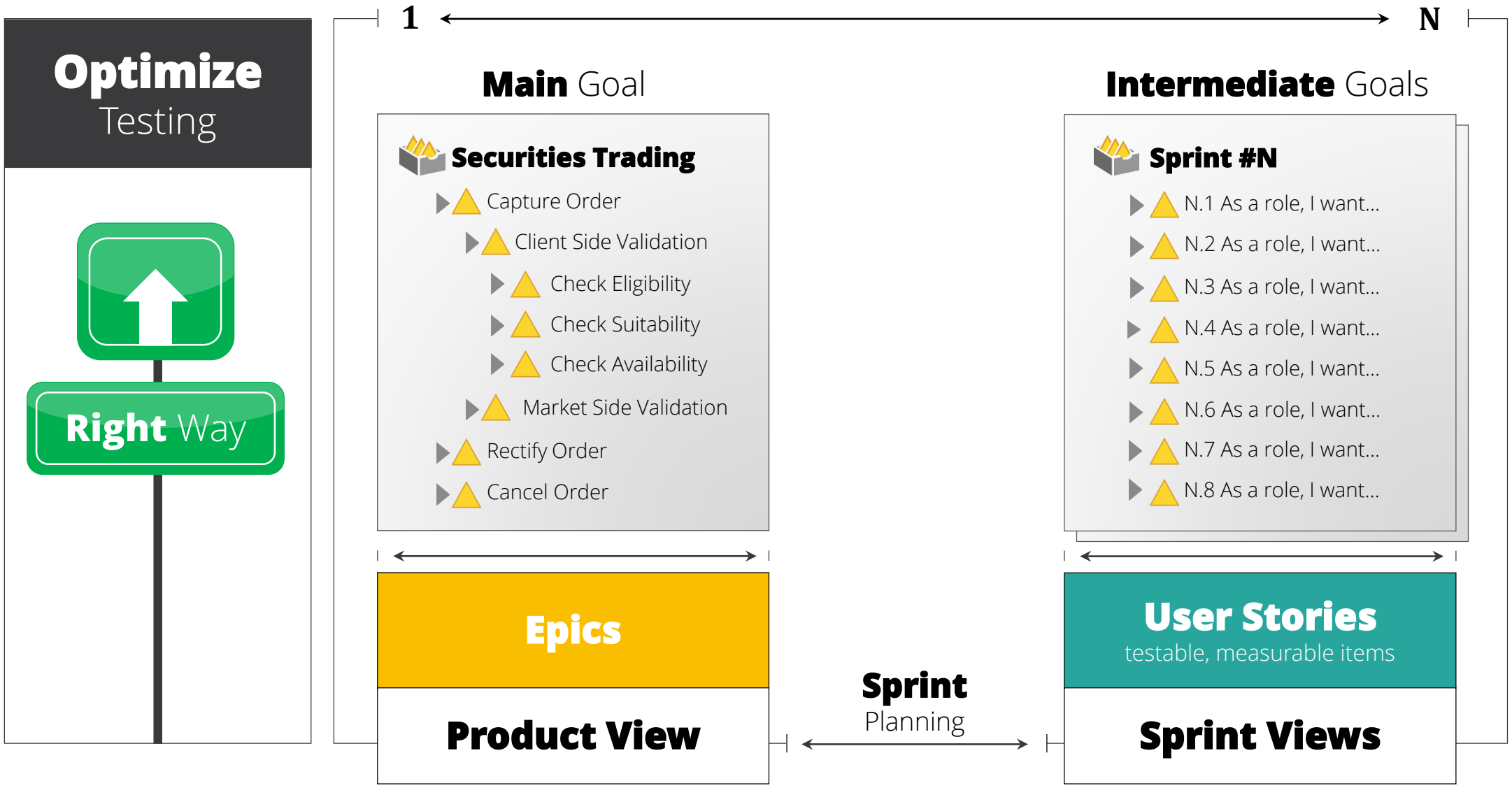
"Without **data** you're just another person with an opinion."
W. Edwards Deming

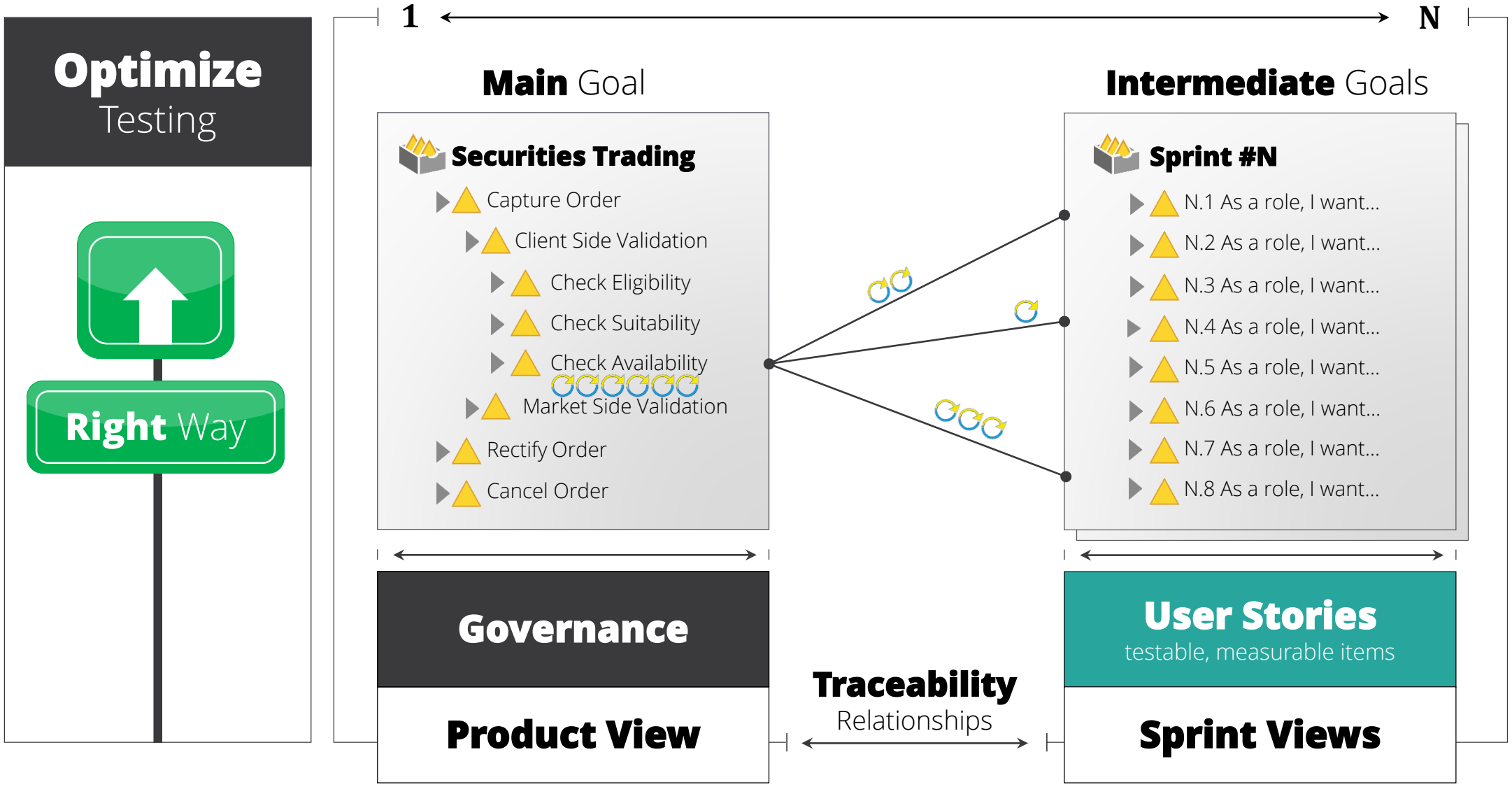
40 % average **risk coverage** in enterprise test case portfolios

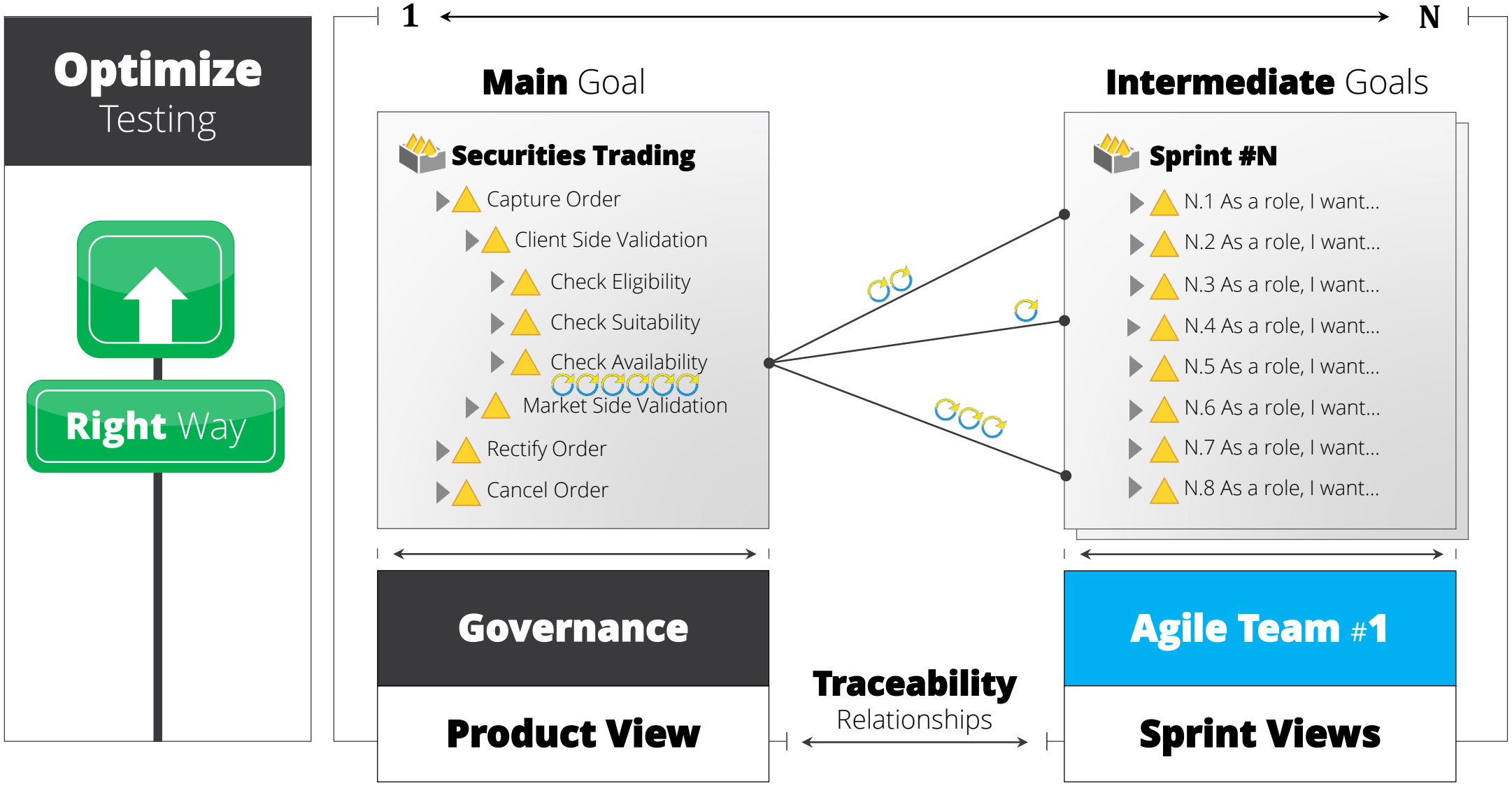
Optimize Testing

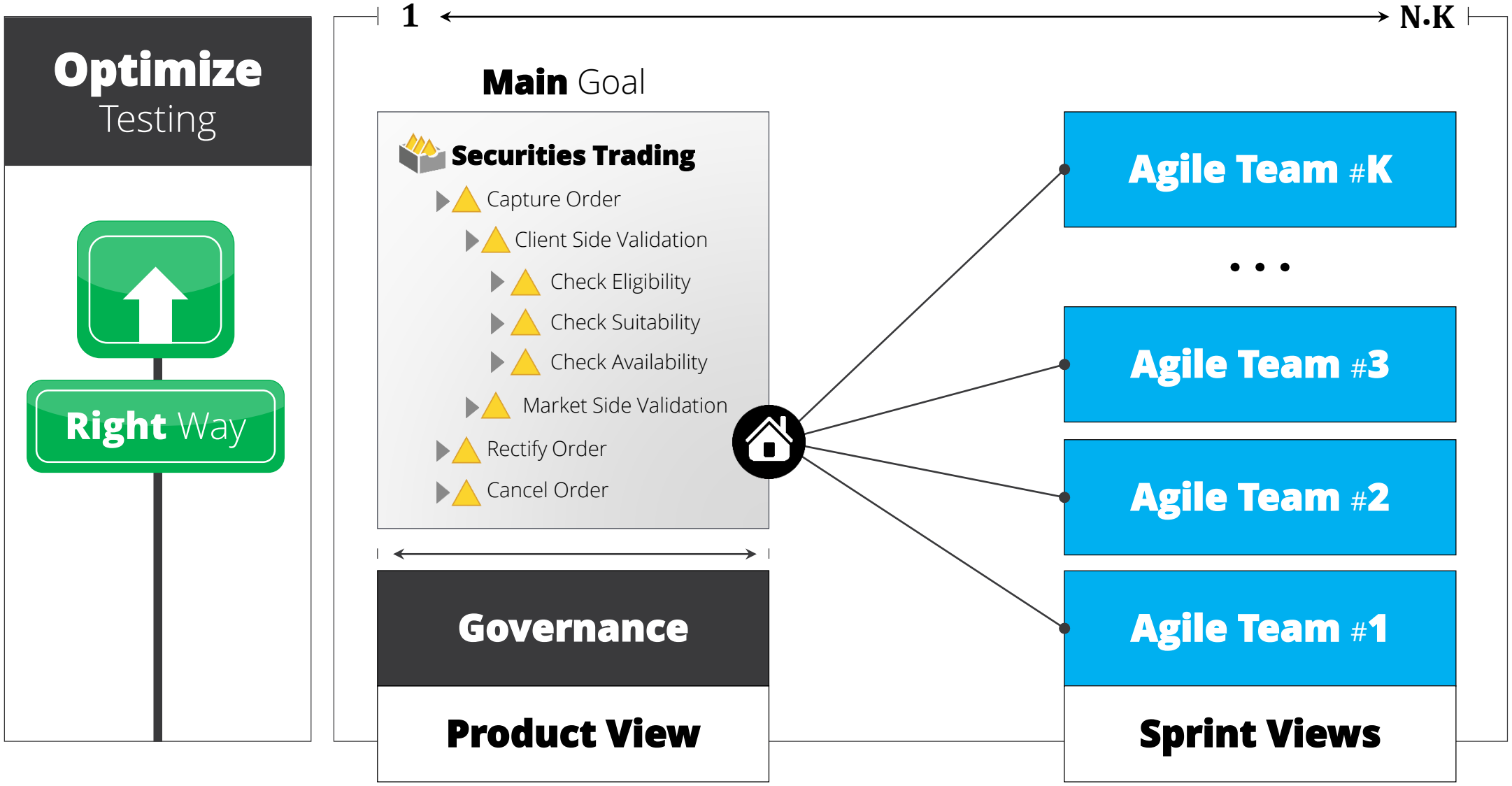


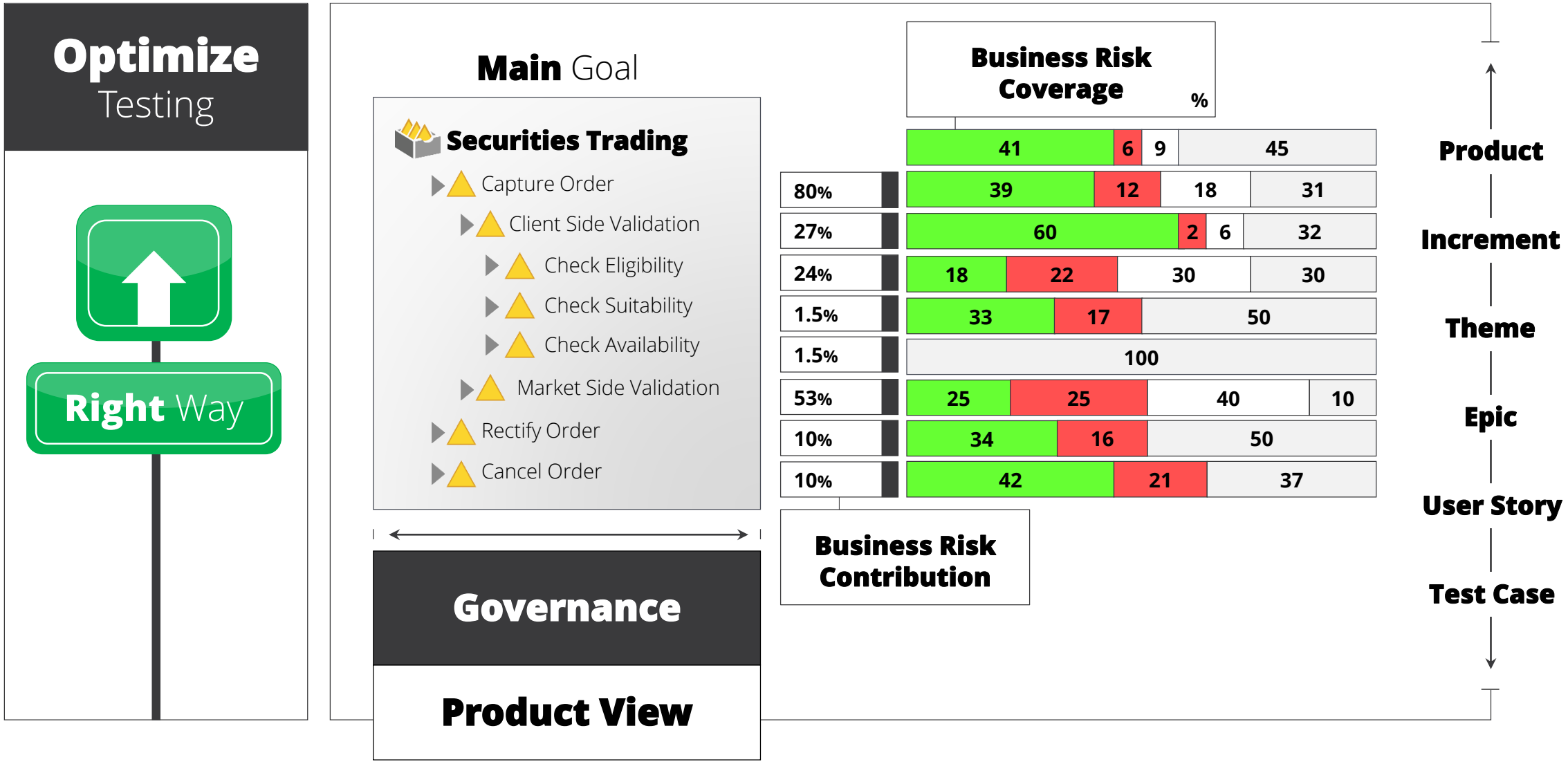






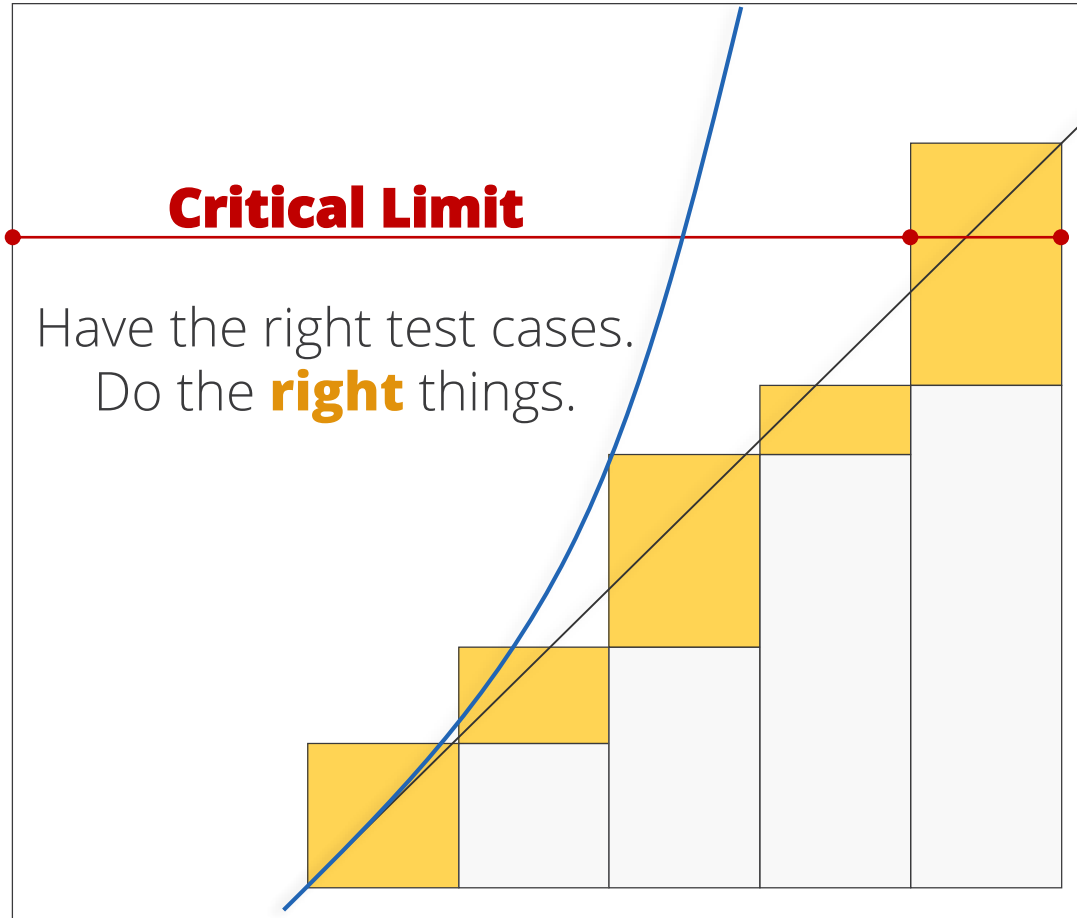






67 % average **redundancy** level in enterprise test case portfolios

Optimize
Testing



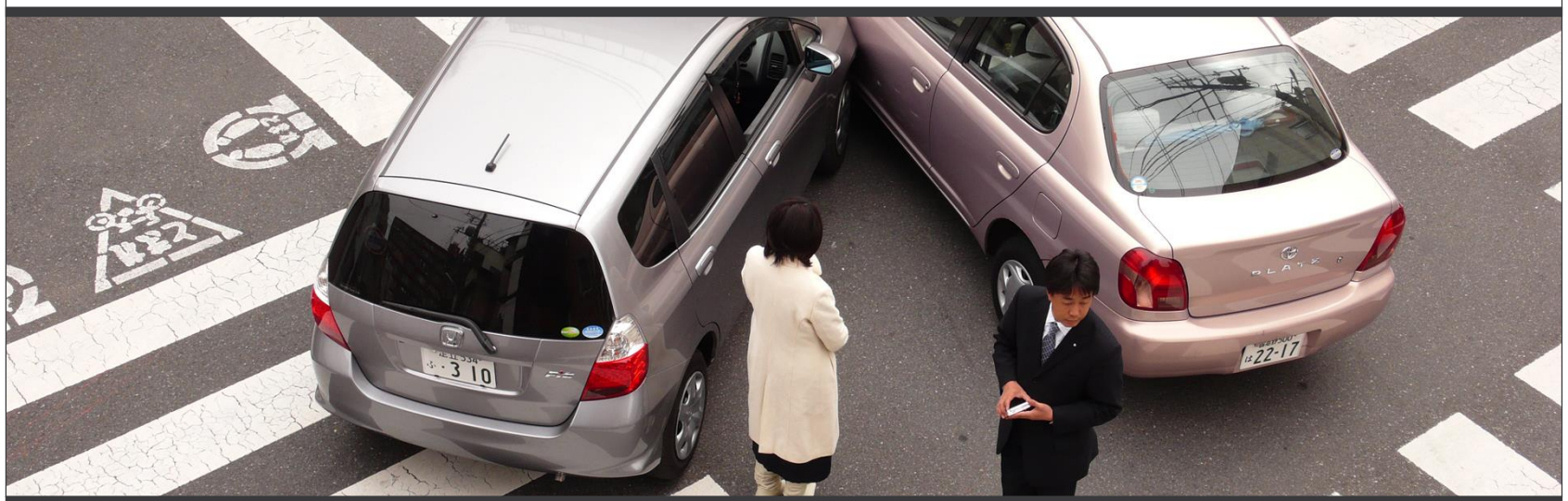
Have the right test cases.
Do the **right** things.

Don't **get drown** in the **number** of **test cases.**

Test Case Design

Optimize
Testing

It’s time to get insured.



Combinatorial Testing Approaches

Create Quote

1. Age
2. Gender
3. Payload [kg]
4. Vehicle Type
5. Country
6. Fuel Type
7. List Price
8. Mileage Per Year
9. Engine Performance
10. Start Date
11. Insurance Sum
12. Payment Option
13. Damage Insurance
14. Euro Protection
15. Defense Insurance

Maximum risk coverage.
Equals exhaustive testing, and so leads to combinatorial explosion.

Good

Exponential growth in test cases.
Time, resource & cost intensive.

Bad

Antieconomical approach.
Test case count grows exponentially faster than risk coverage.

Bad

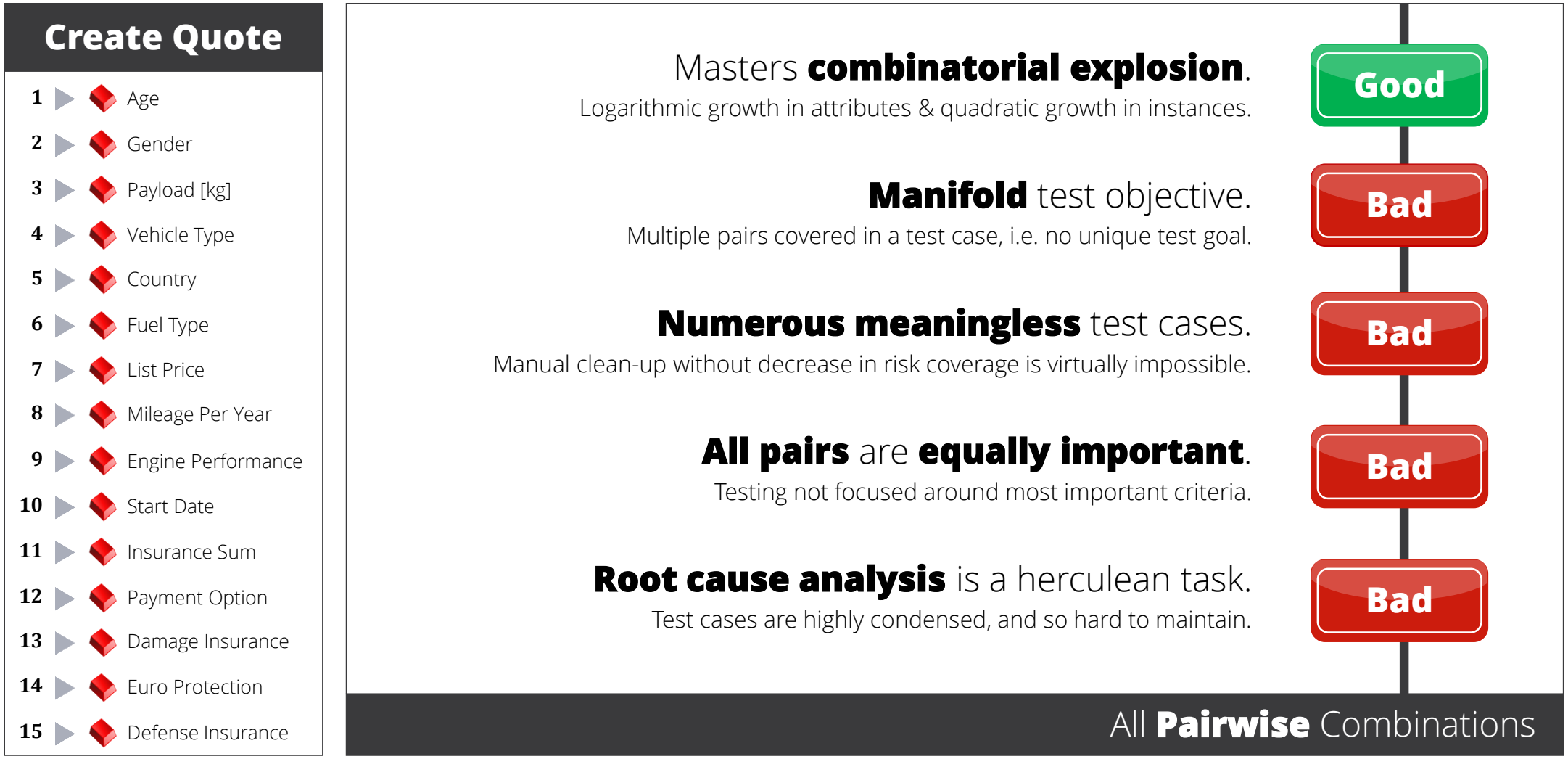
Test Objective is unknown.
Practical significance only for small scale testing missions.

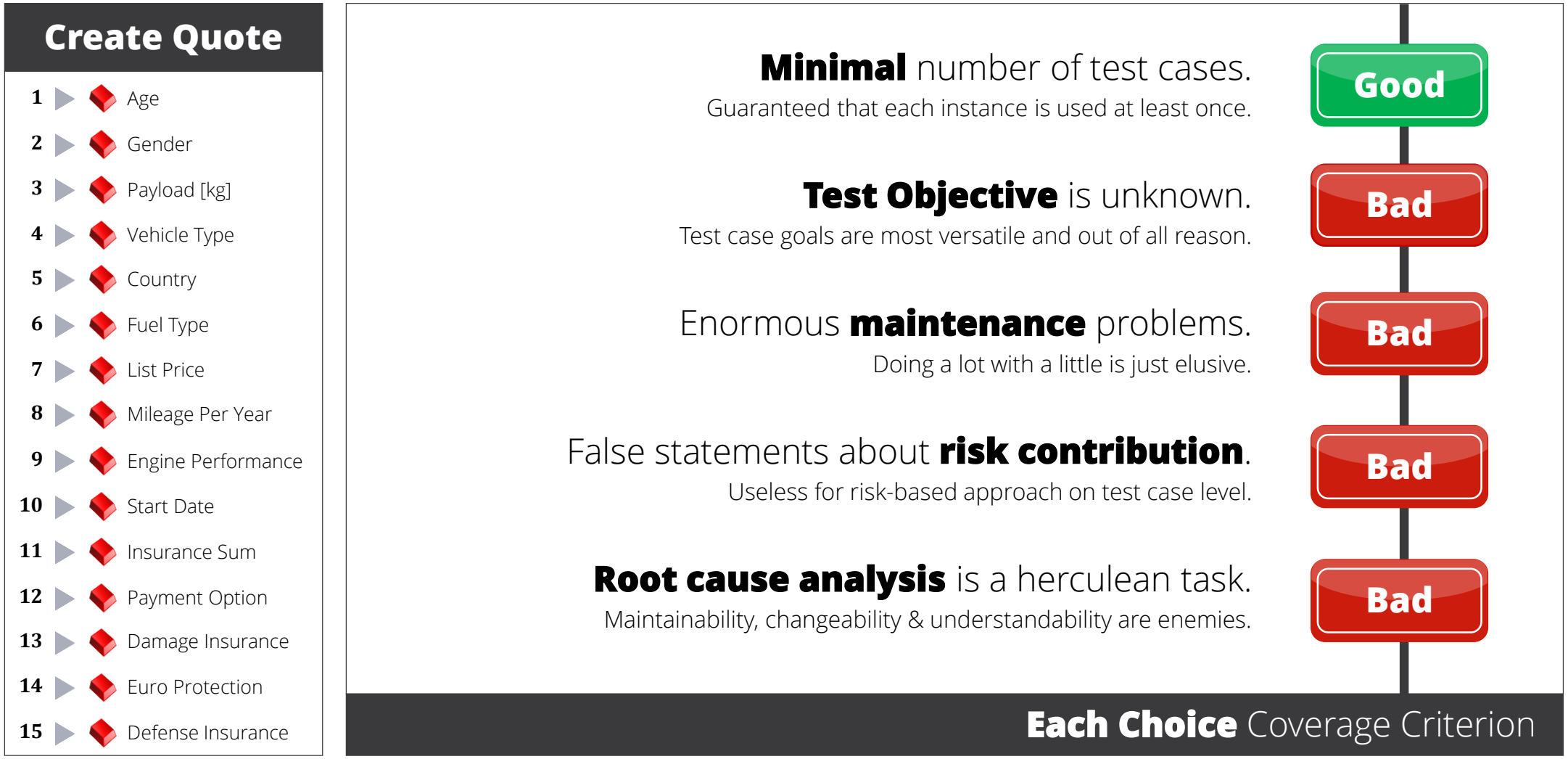
Bad

Root cause analysis is a herculean task.
Test objective is the entire test case portfolio.

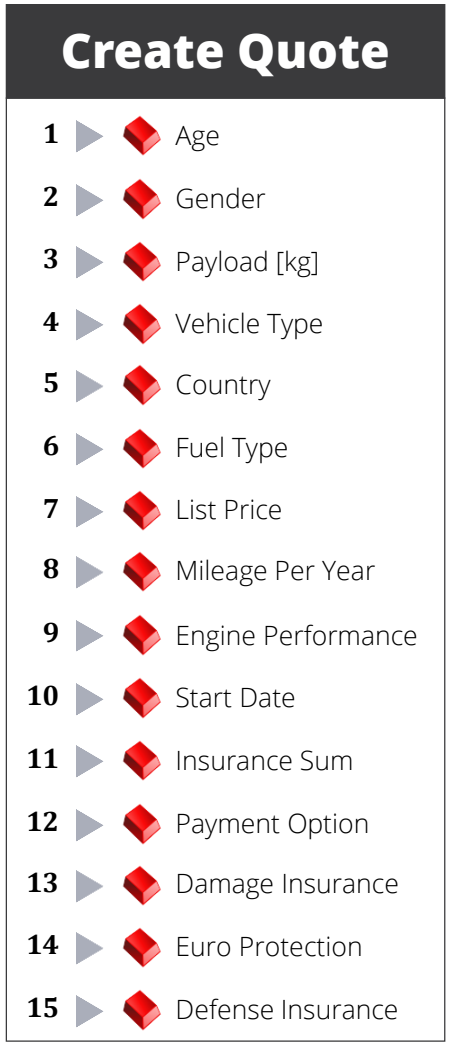
Bad

All **Possible** Combinations





"Nothing is perfect, life is messy, outcomes are uncertain, people are irrational, **relationships** are complex"



Creates only **slightly more** test cases.
Linear increase in test case count up to about 95% risk coverage. — Good

Assigns a **unique** & **well-defined** test objective.
Strongly supports changeability, maintainability & understandability. — Good

Enables to derive **risk contribution**.
Best to apply risk-based approach on test case level. — Good

Makes **root cause analysis** an easy task.
Test smarter, not harder and keep it simplistic instead of complex. — Good

Assumes attribute **independence**.
About 20% interdependent attributes, and so the logic partly breaks down. — Bad

Linear Expansion

"Nothing is perfect, life is messy, outcomes are uncertain, people are irrational, **relationships** are complex"

Create Quote

1. Age
2. Gender
3. Payload [kg]
4. Vehicle Type
5. Country
6. Fuel Type
7. List Price
8. Mileage Per Year
9. Engine Performance
10. Start Date
11. Insurance Sum
12. Payment Option
13. Damage Insurance
14. Euro Protection
15. Defense Insurance

Case #1	Case #2	Case #3	Case #4			Case #7
(24,59)	>59	(24,59)	(24,59)	(24,59)	(24,59)	(24,59)
Male	Male	Female	Male	Male	Male	Male
[1,1000]	[1,1000]	[1,1000]	>1000	[1,1000]	[1,1000]	[1,1000]
Truck	Truck	Truck	Truck	Car	Motorbike	Trailer

Test Case Design

Relations

Meaningless
instance combinations

Missing
instance combinations

Business Rules
unlinked from test cases

Customized Linear Expansion

Optimize
Testing

Would you **believe** in what
I have just presented?

Risk Coverage Optimization

Optimize
Testing

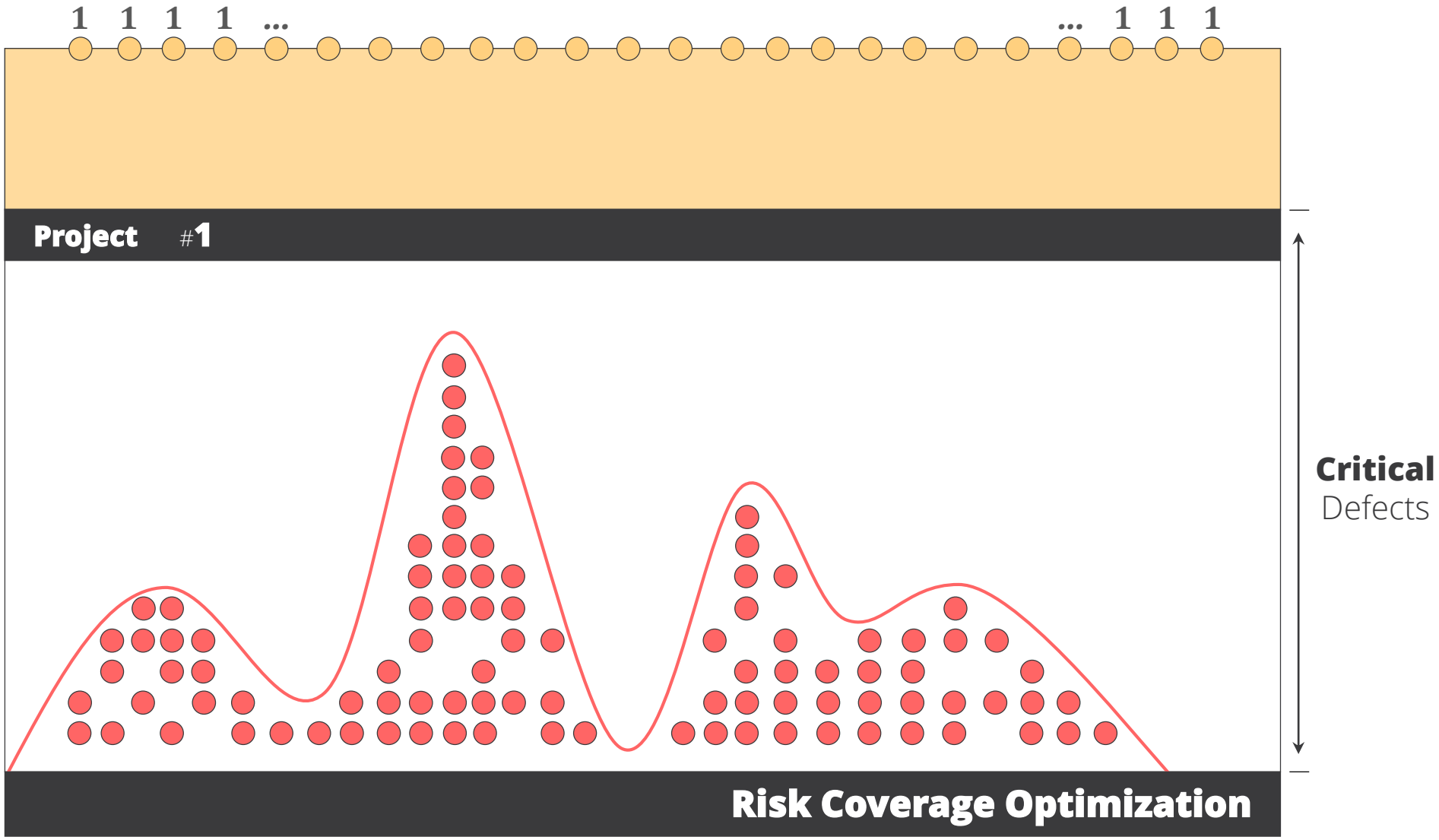
Projects Analyzed	**208**
Project Lifetime On Average	**8** Months
Test Cases On Average	**1712**
Automation Level On Average	**86%**
Distinct **Sectors**	**7**

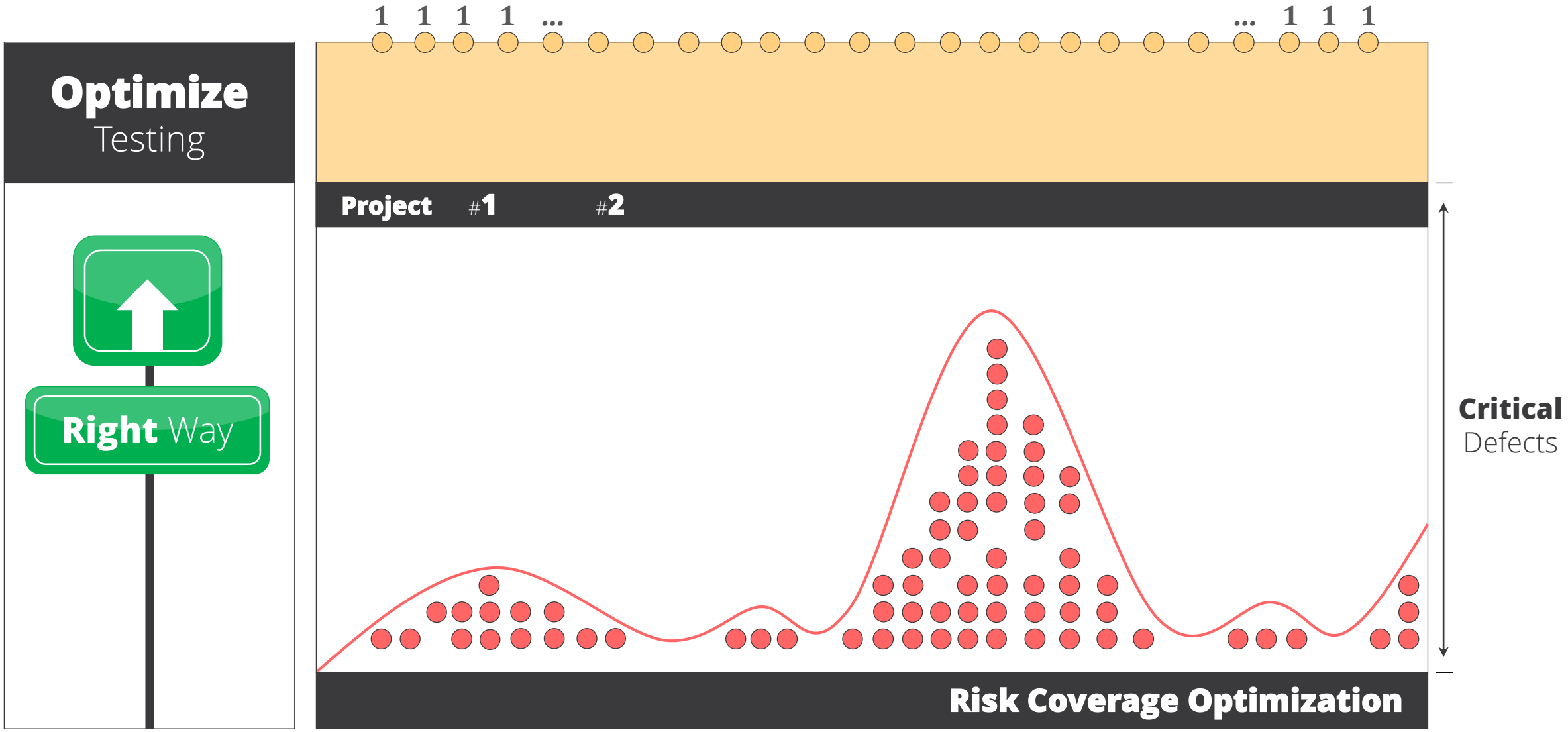
Financial
Consumer
Energy
Telecommunications
Industrials
Healthcare
▶ **Materials**

Risk Coverage Optimization

*last **status** update: **January 2016**

Optimize
Testing





Optimize
Testing

1 1 1 1 ... ... 1 1 1

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

Project #**1** #**2** #**3** ... #**208**

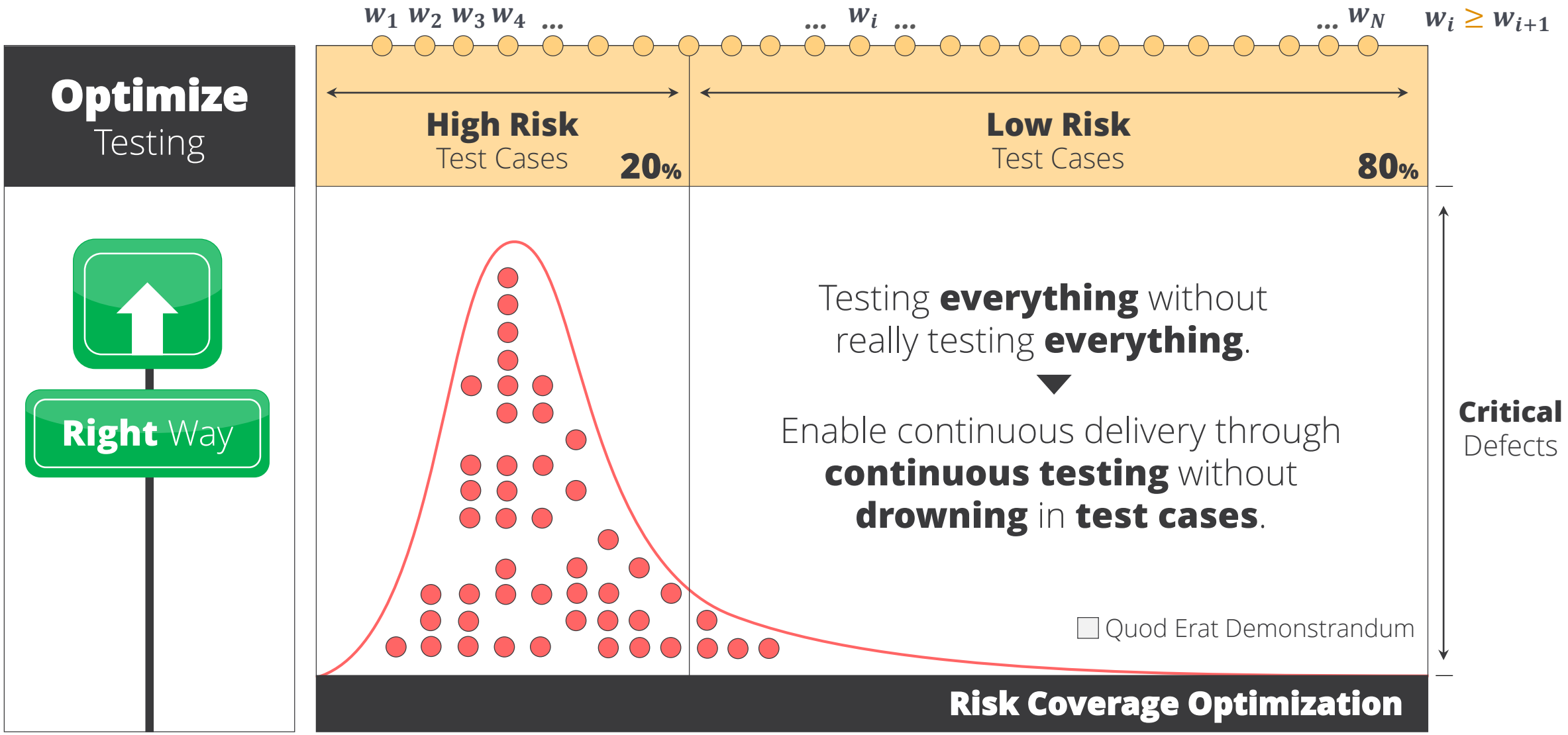
Randomness rules.
Critical defects are just chance hits.

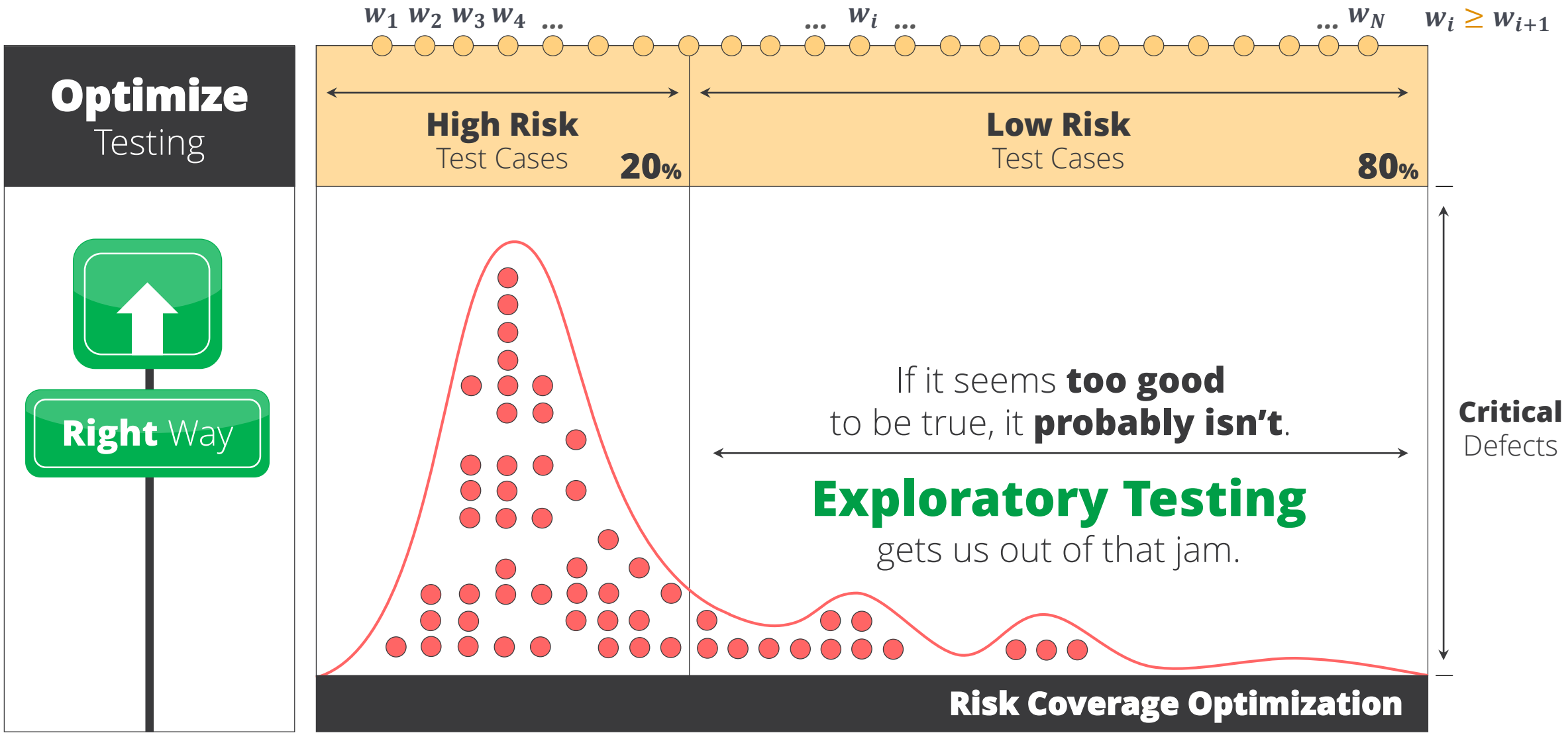
No **characteristics**.
Selective testing is impossible.

Nothing to learn.
No chance to improve.

Critical
Defects

Risk Coverage Optimization

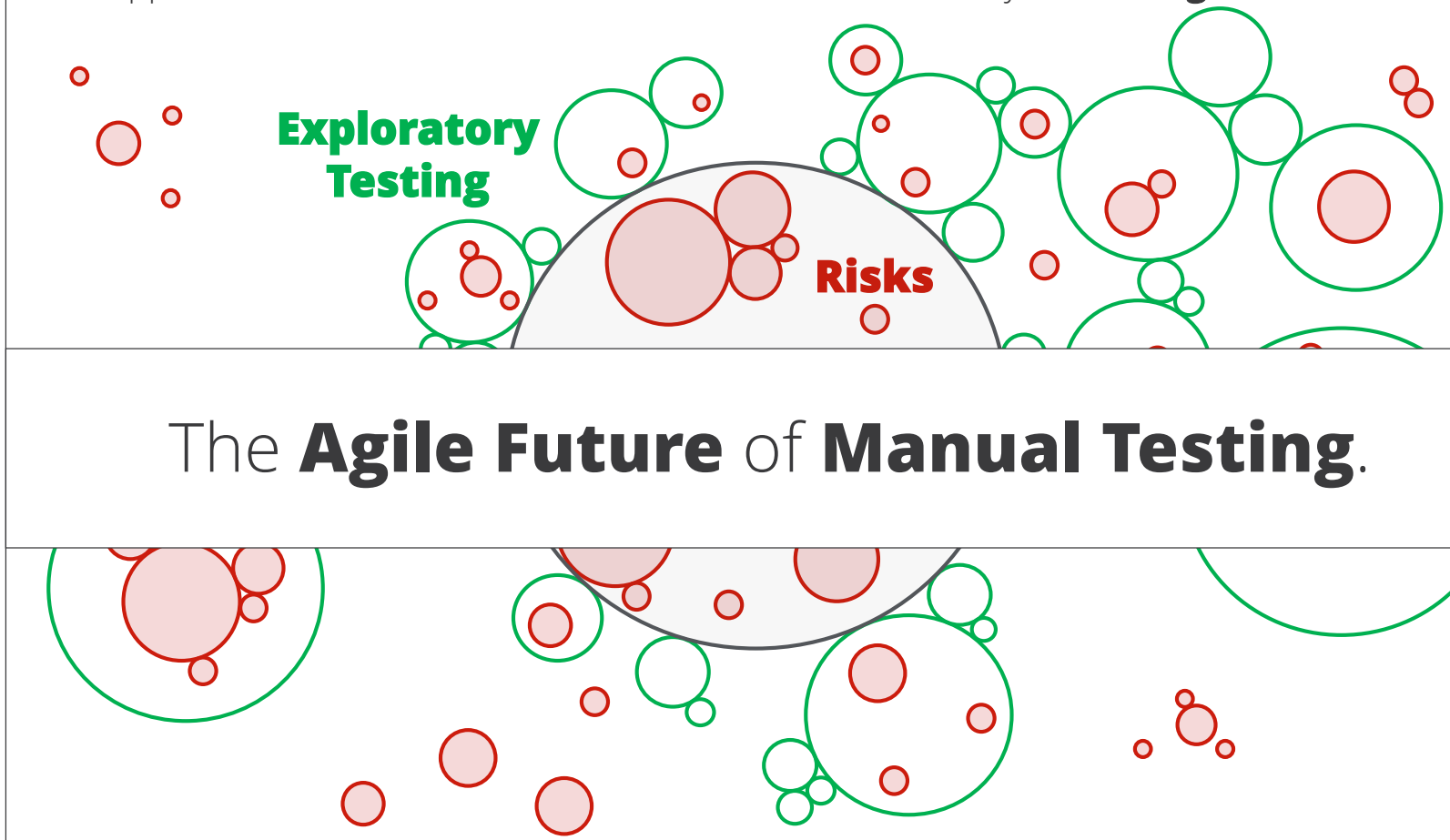




Optimize
Testing

application's **universe**

increase your **testing cross section**



The **Agile Future** of **Manual Testing**.

Exploratory Testing

Optimize
Testing

Does this approach **pay off**?

Risk Coverage Optimization

Testers	Optimize	Automate	Integrate	
11	Value Based **Test Cases**	Model Based **Automation**	Continuous **Integration**	
Tests				
4755	1193 Redundancy 75%	**Automation Level**	**Distributed Smoke Testing**	**Automation Level Smoke Testing**

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

				Test Case Count 7%
10 Weeks	2.5 Weeks	48 Hours	8 Hours	53%
Execution	**Test Data** Management	**Single Agent Regression**	**Distributed Regression**	**Risk Coverage Smoke Testing**