



SmartQA

Masterclass for EM/SM



© 2020, STAG Software Pvt Ltd

www.stagsoftware.com

smartqa.org

TOPICS

Testing vs. Checking

Principle of orthogonality - Quality levels, Test types & Test techniques

The granularity of Entity Under Test (EUT)

Dev/QA test objectives

Test scenarios, cases

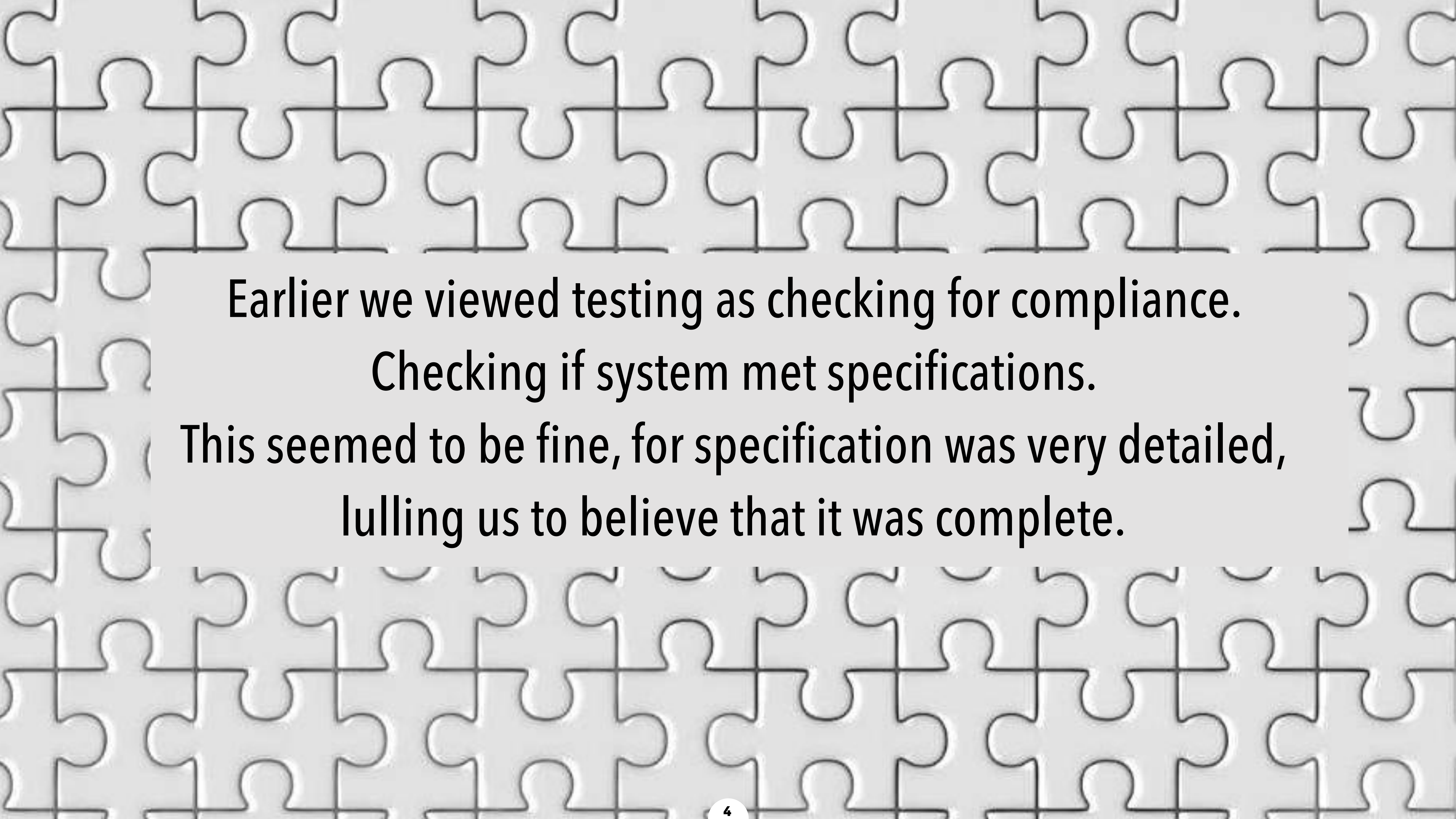
Test approaches (Scripted, Ad hoc, Exploratory, Automated...)

Role of QA - Executor, Automator, Designer, Questioner, Suggestor, Analyser

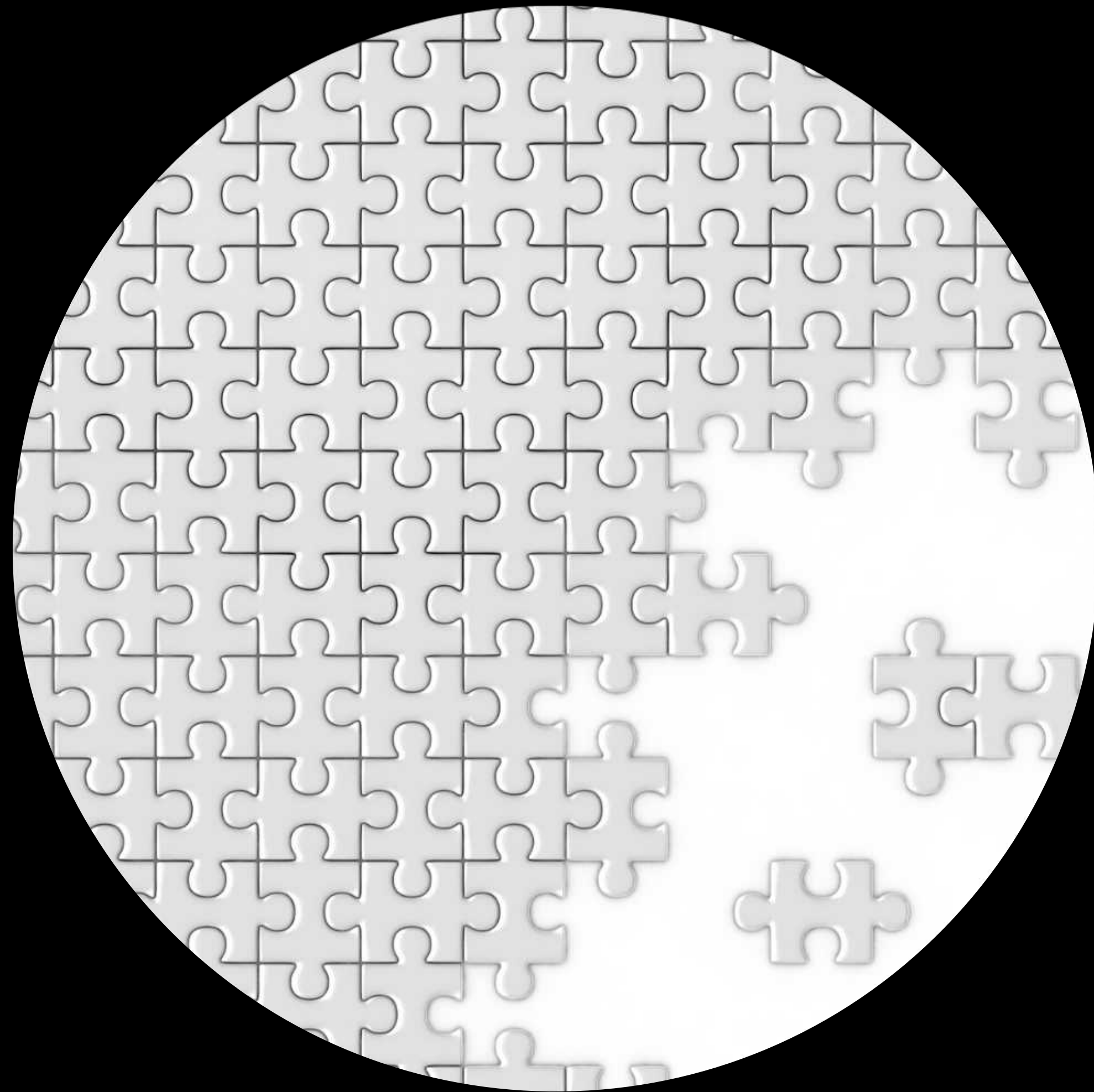
On Automation

Metrics of interest related to effectiveness, efficiency

TESTING vs. CHECKING



**Earlier we viewed testing as checking for compliance.
Checking if system met specifications.
This seemed to be fine, for specification was very detailed,
lulling us to believe that it was complete.**



**Truly, no specification can be deemed complete,
as there is no way to prove this.**

Well, specification now is not as detailed, evolves with time, with users revising when they see first cut implementation.

Well, specification now is not as detailed, evolves with time, with users revising when they see first cut implementation.



So we matured, revising our thinking.

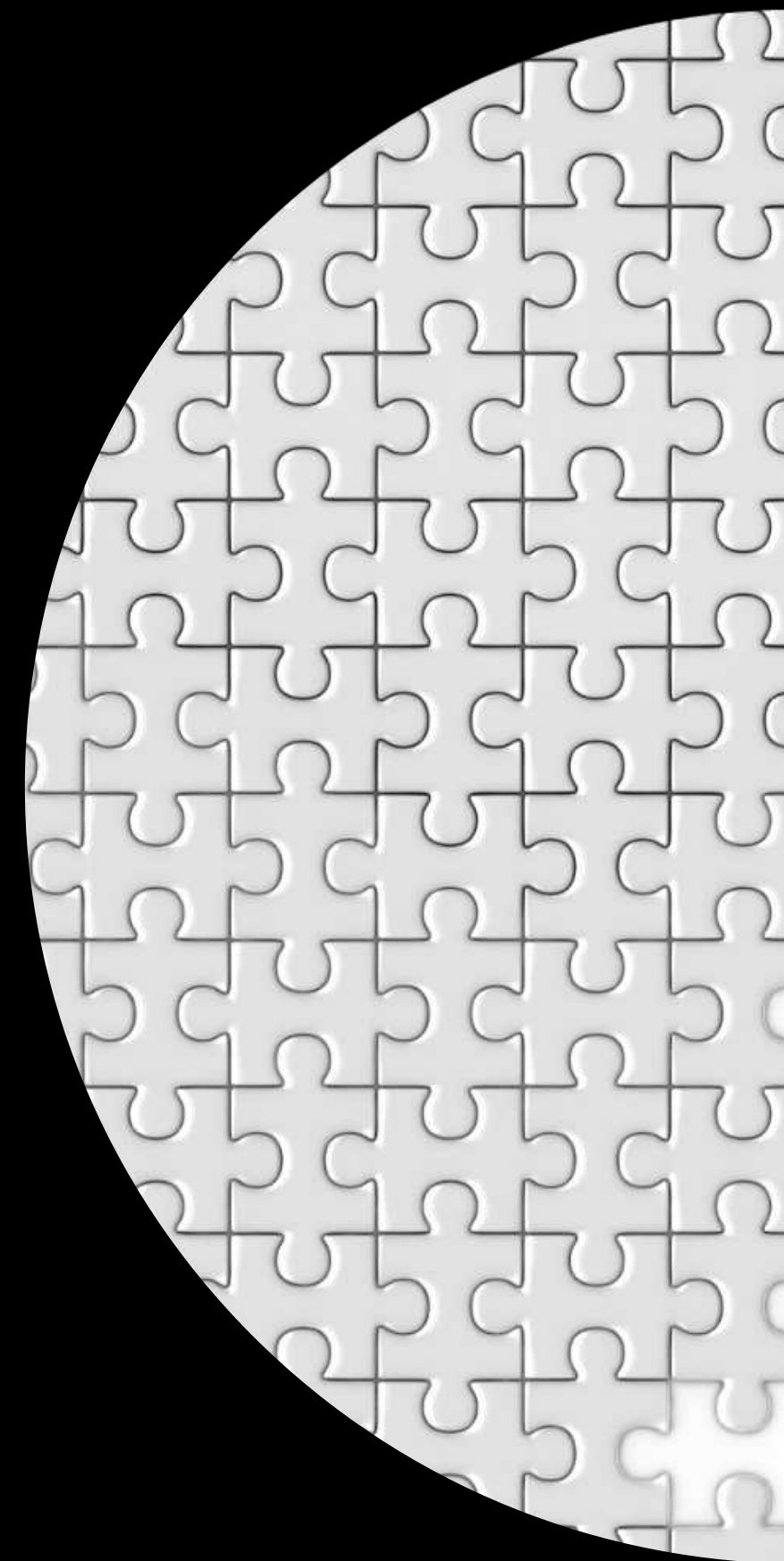
Testing was viewed as exploration.

To understand well, in the process spot anomalies, and suggest ideas that add value to end users/customers.

**Great quality is about
ensuring compliance and discovering potential gaps**

Great quality is about
ensuring compliance and discovering potential gaps

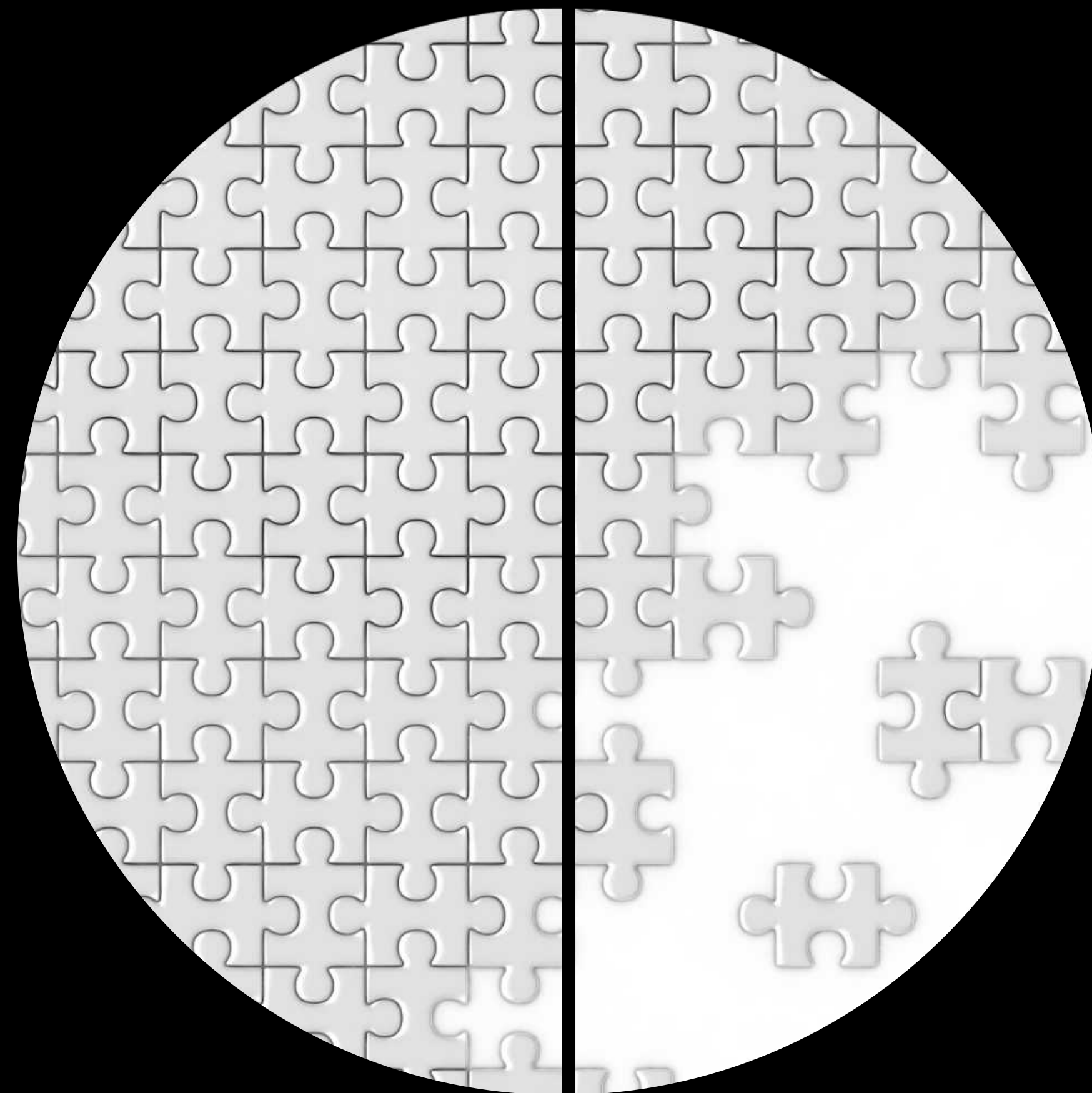
CHECKING



Great quality is about
ensuring compliance and **discovering potential gaps**

CHECKING

TESTING



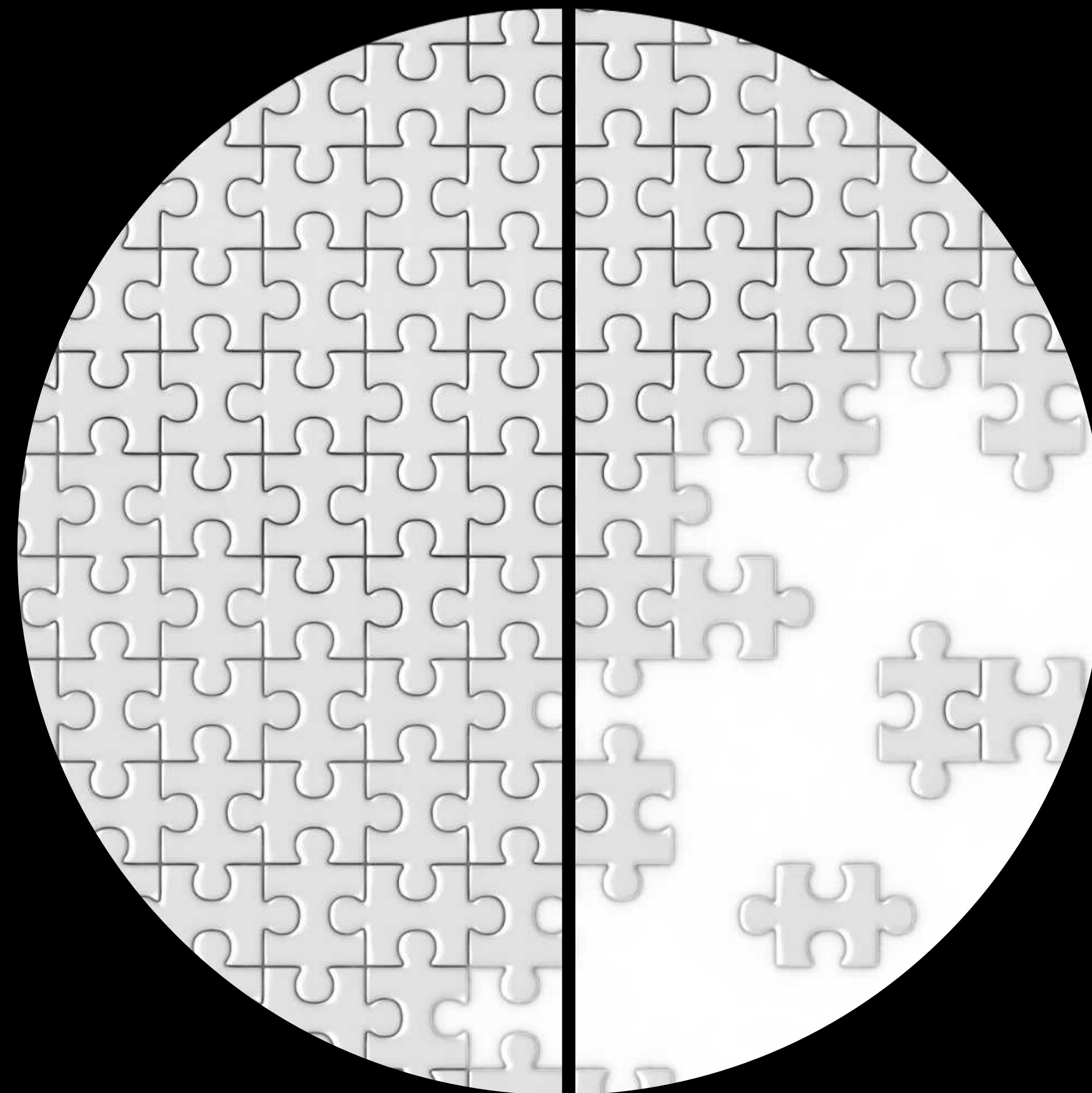
Great quality is about ensuring compliance and **discovering potential gaps**

CHECKING

is comparing
can be scripted
binary outcome Pass/Fail
design approach -
logical, experience

based on spec
wellness
AUTOMATED

TESTING

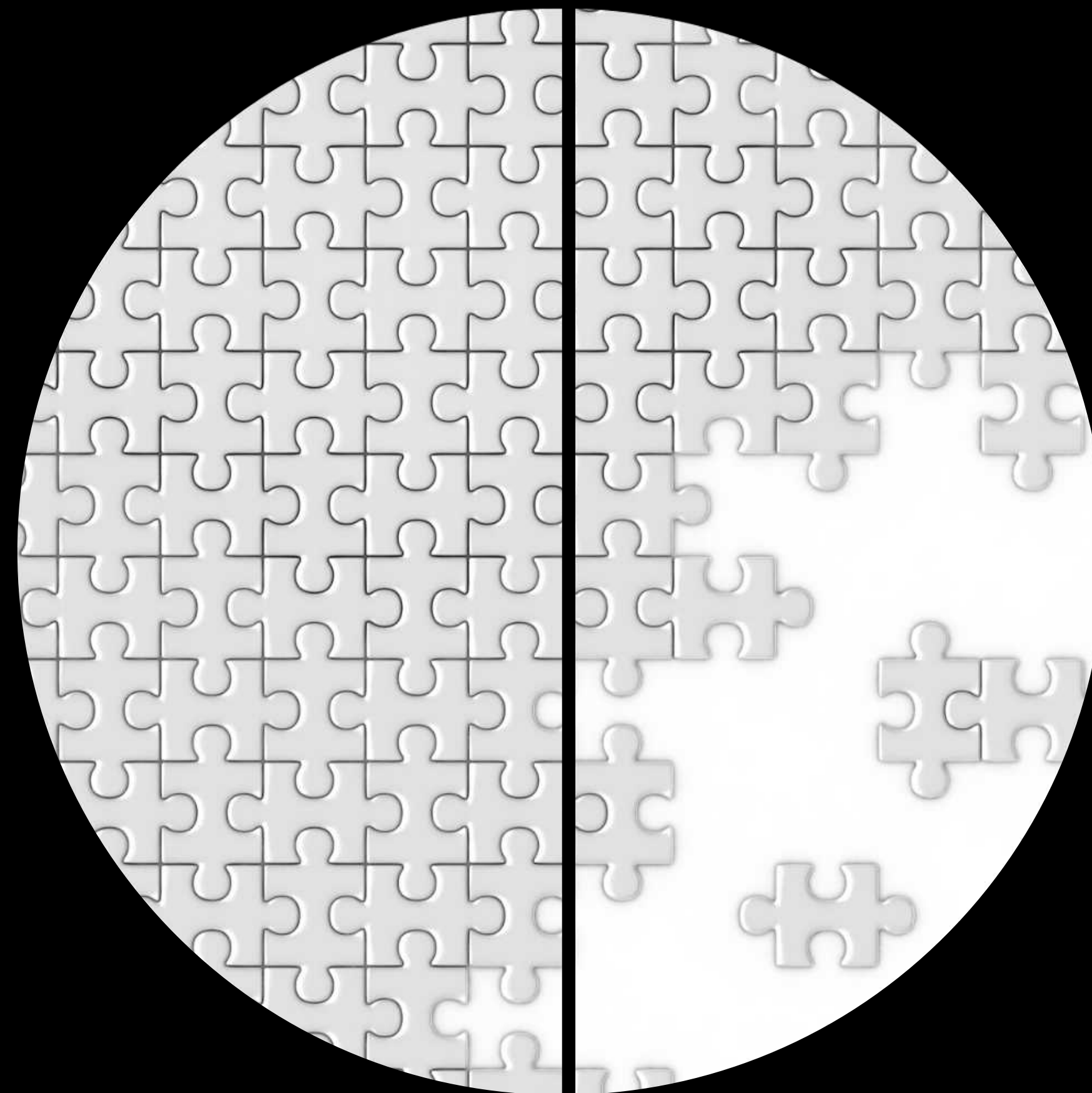


Great quality is about ensuring compliance and **discovering potential gaps**

CHECKING

is comparing
can be scripted
binary outcome Pass/Fail
design approach -
logical, experience

based on spec
wellness
AUTOMATED



TESTING

is questioning
not always scripted
outcome Pass/Fail,??
design approach-
many

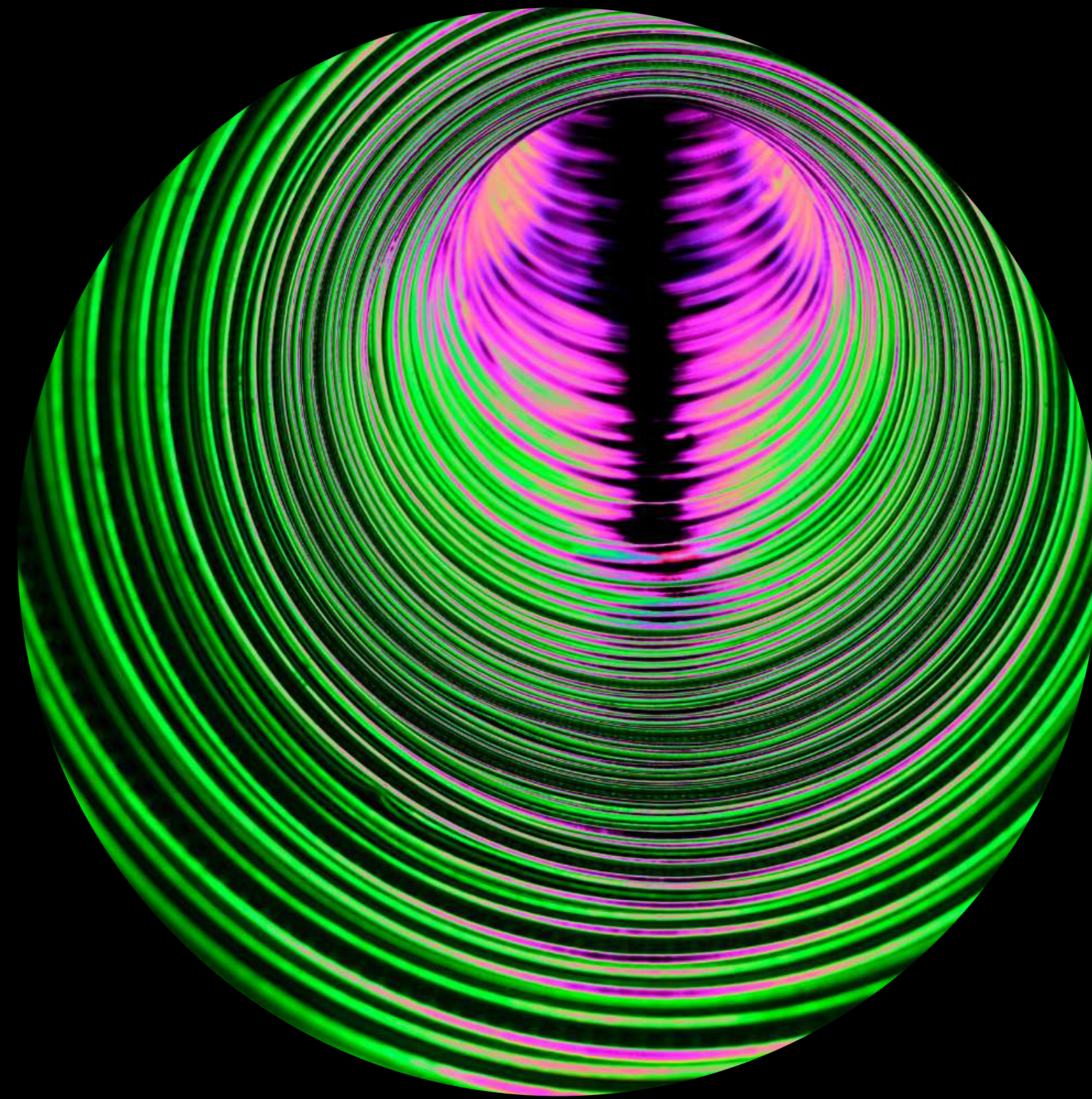
beyond spec
illness
HUMAN

A mature view is :

Testing is a brilliant combination of checking for expected, exploring the whole, looking for unexpected, uncovering issues, suggest needs not yet thought of, improving what is done, and sensitise to prevent issues.



“Testing” really is a gateway to doing good work
and produce systems of value.



PRINCIPLE OF ORTHOGONALITY

Quality Levels,

Test Types &

Test Techniques

Quality LEVEL

L9 End user value

L8 Deployment correctness

L7 Attribute correctness

L6 Environment correctness

L5 Flow correctness

L4 Behaviour correctness

L3 Structural correctness

L2 Interface correctness

L1 Input correctness

View user's expectation of quality as a series of **levels** to attain.

Quality LEVEL

L9 End user value

L8 Deployment correctness

L7 Attribute correctness

L6 Environment correctness

L5 Flow correctness

L4 Behaviour correctness

L3 Structural correctness

L2 Interface correctness

L1 Input correctness

Load test

Performance test

Test TYPE

View user's expectation of quality as a series of **levels** to attain.

To attain a level, defects that affect this level must not be present
=> we must conduct **specific** tests

Quality LEVEL

L9 End user value

L8 Deployment correctness

L7 Attribute correctness

L6 Environment correctness

L5 Flow correctness

L4 Behaviour correctness

L3 Structural correctness

L2 Interface correctness

L1 Input correctness

Test TECHNIQUES
Operational profiling
Code profiling

Load test

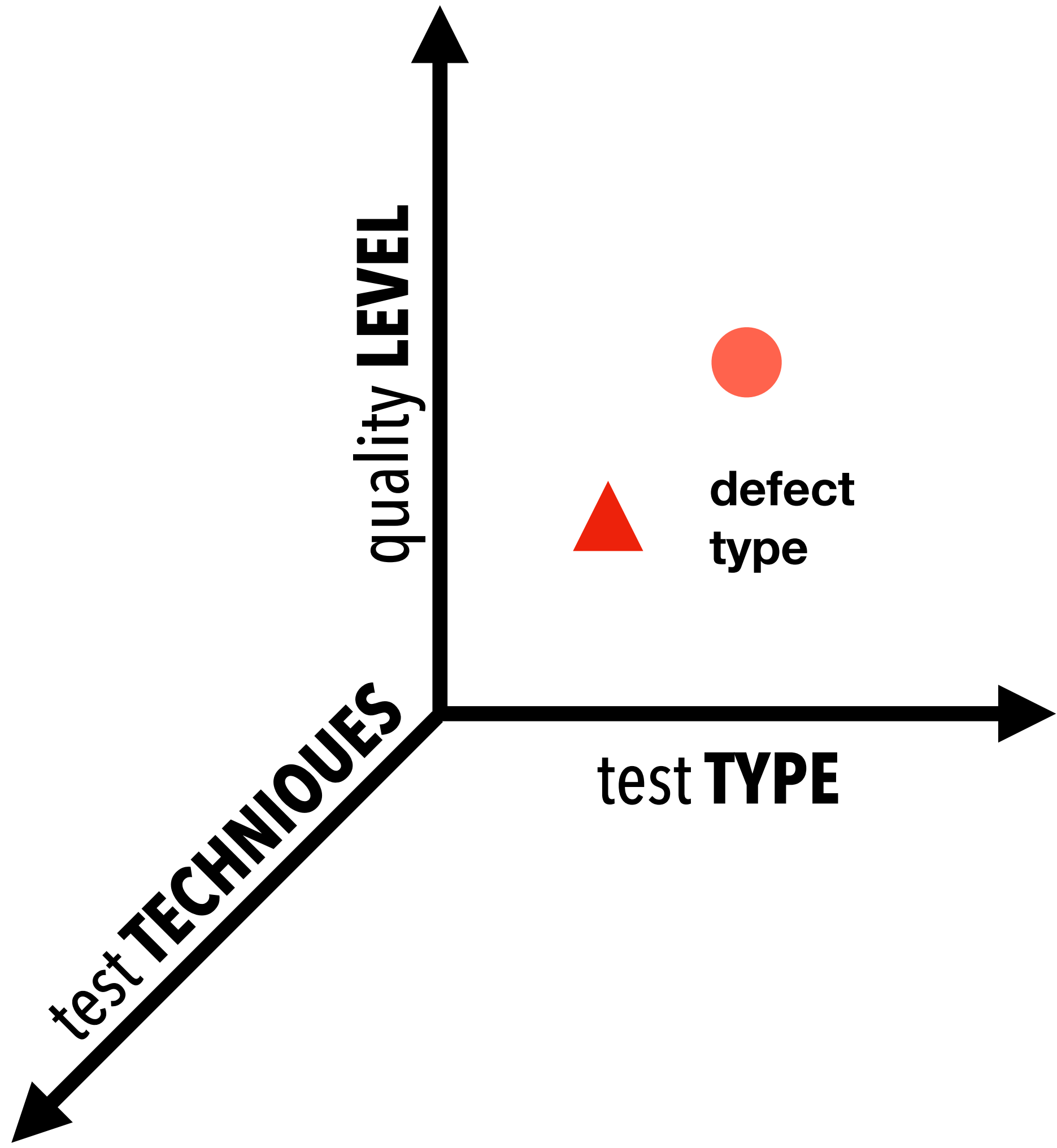
Performance test

Test TYPE

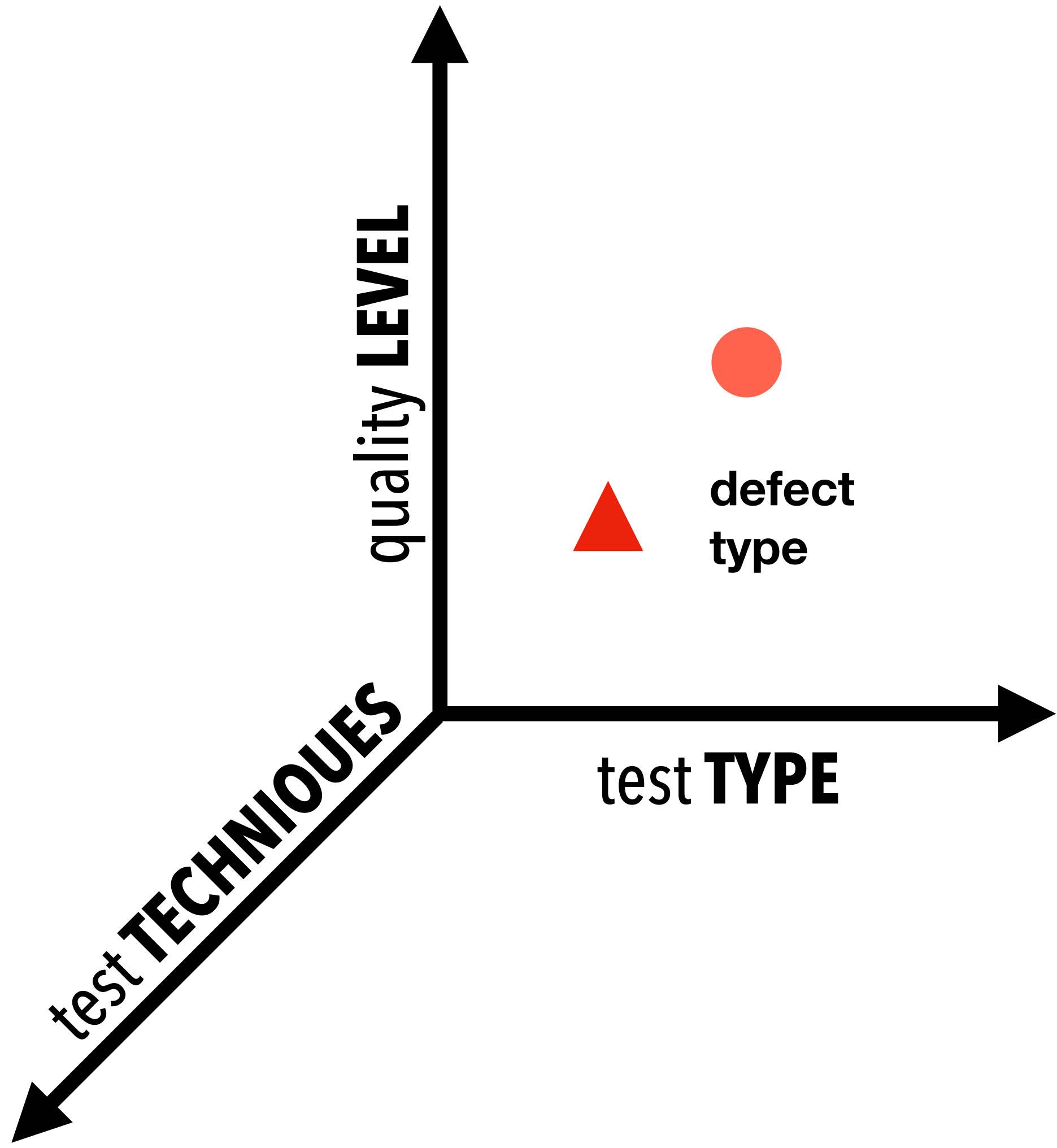
View user's expectation of quality as a series of **levels** to attain.

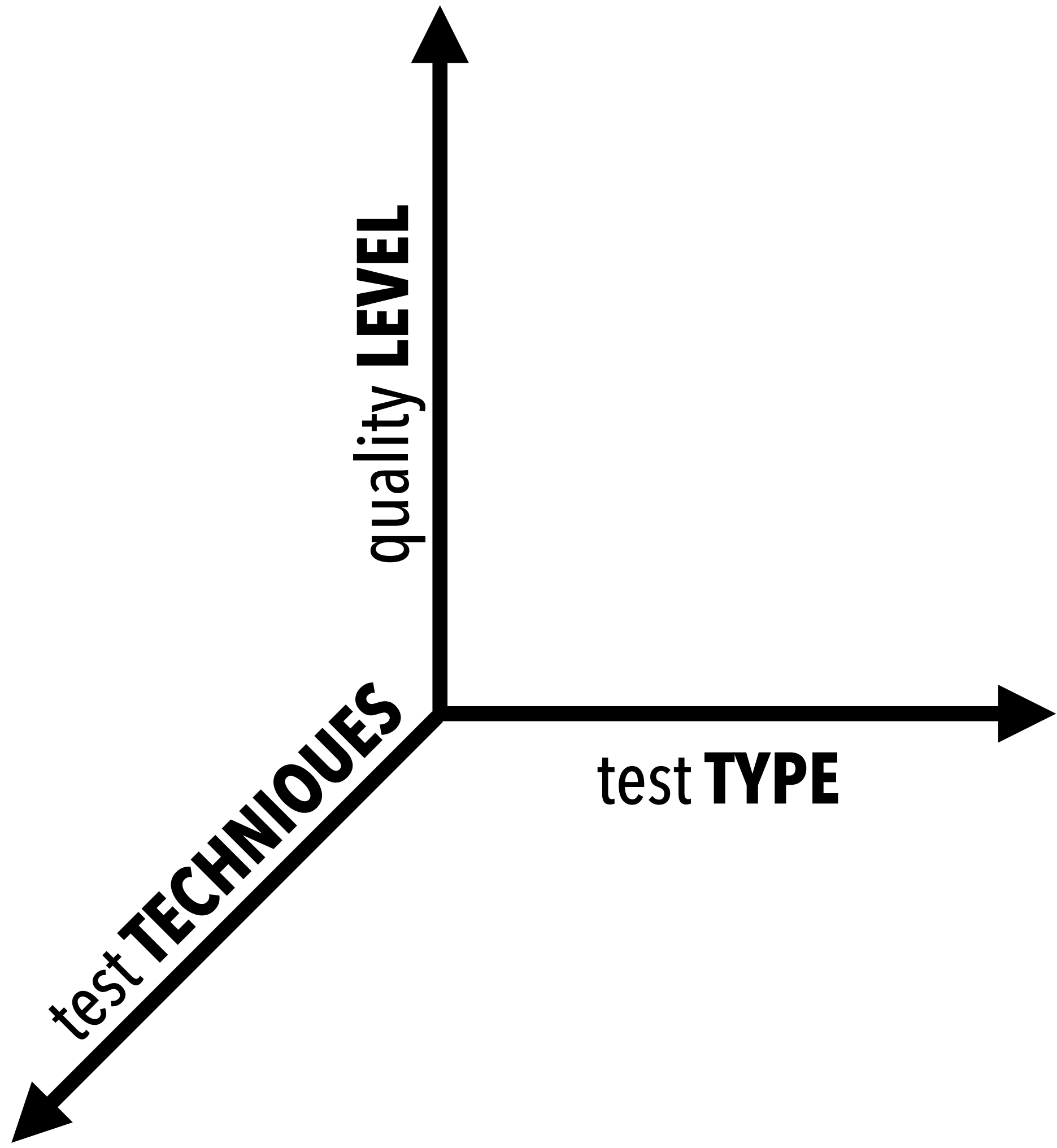
To attain a level, defects that affect this level must not be present
=> we must conduct **specific** tests

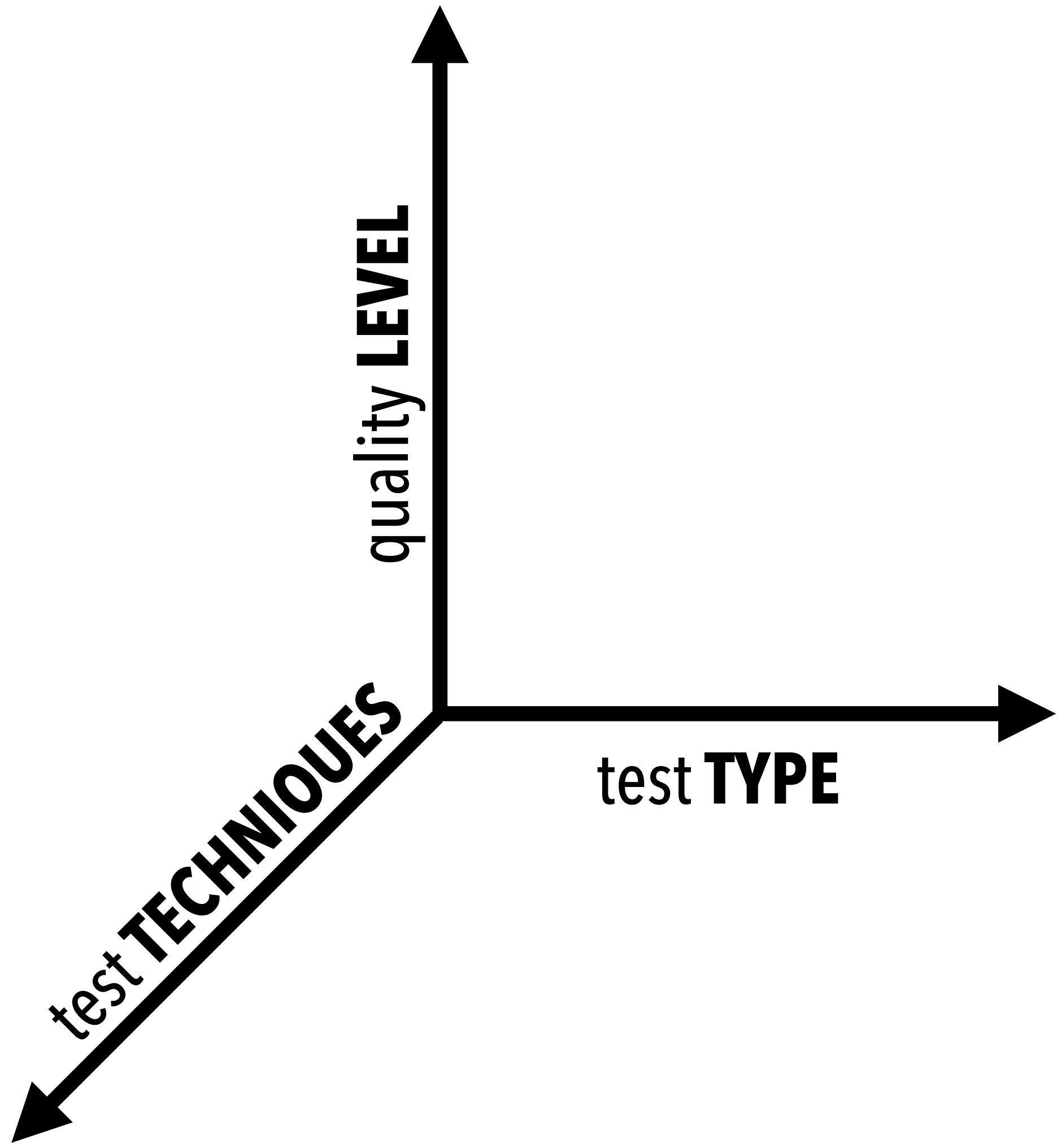
To do a test, we need to come up with test scenarios/cases
=> we need test **techniques**

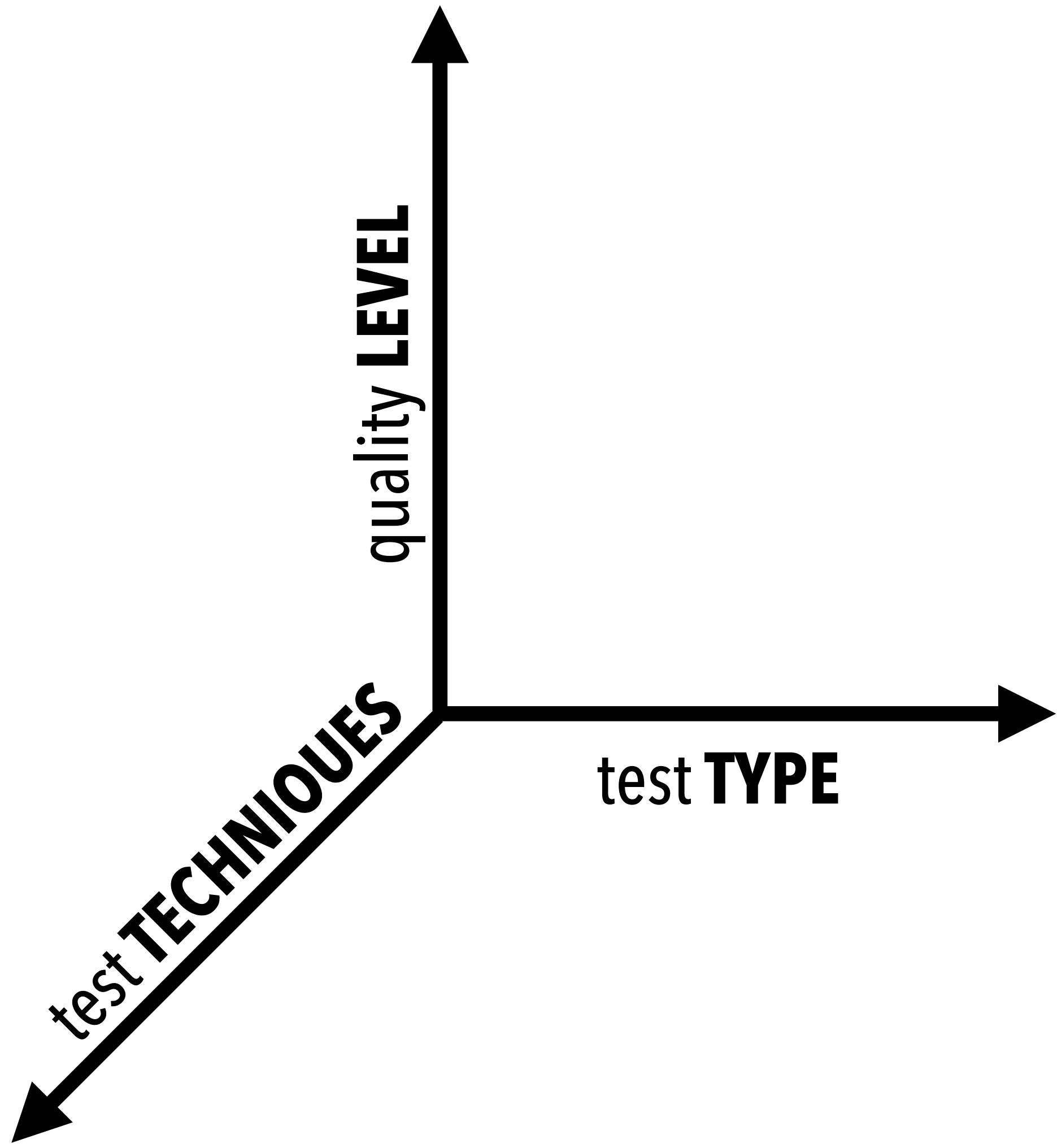


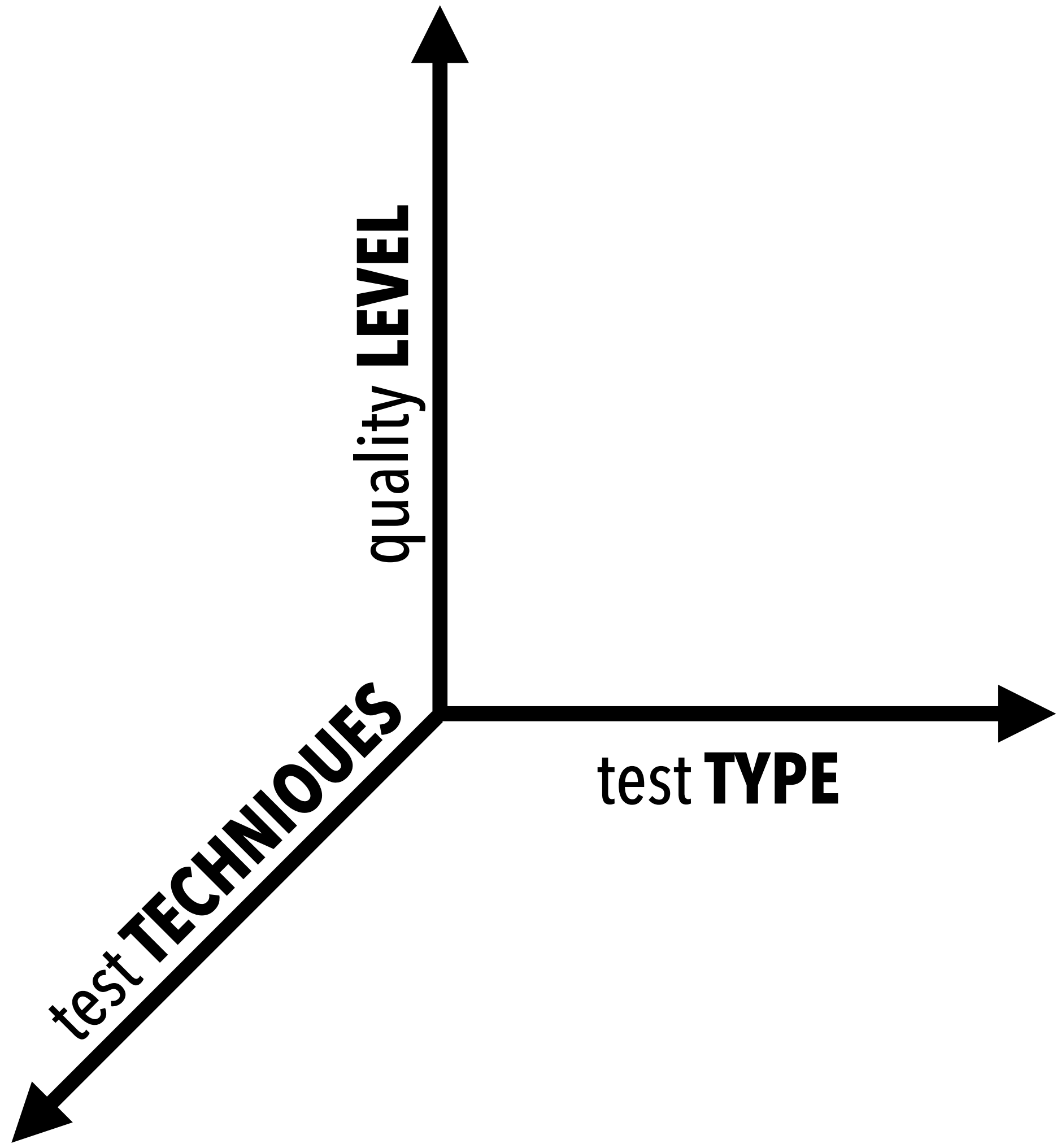
defect
type











The granularity of Entity Under Test (EUT)

WHAT TO TEST? Entity Under Test (EUT)

**structural
COMPONENT**

Basic building block

WHAT TO TEST? Entity Under Test (EUT)

**technical
FEATURE**

Basic offering from system

**structural
COMPONENT**

Basic building block

WHAT TO TEST? Entity Under Test (EUT)

**user
REQUIREMENT**

Enables an user to do a task

**technical
FEATURE**

Basic offering from system

**structural
COMPONENT**

Basic building block

WHAT TO TEST? Entity Under Test (EUT)

**business
FLOW**

A set of tasks by different users
to accomplish a business objective

**user
REQUIREMENT**

Enables an user to do a task

**technical
FEATURE**

Basic offering from system

**structural
COMPONENT**

Basic building block

WHAT TO TEST? Entity Under Test (EUT)

**business
FLOW**

A set of tasks by different users
to accomplish a business objective

**user
REQUIREMENT**

Enables an user to do a task

**technical
FEATURE**

Basic offering from system

**structural
COMPONENT**

Basic building block

TEST FOR WHAT?

L9 End user value

L8 Deployment correctness

L7 Attribute correctness

L6 Environment correctness

L5 Flow correctness

L4 Behaviour correctness

L3 Structural correctness

L2 Interface correctness

L1 Input correctness

WHAT TO TEST? Entity Under Test (EUT)

**business
FLOW**

A set of tasks by different users to accomplish a business objective

**user
REQUIREMENT**

Enables an user to do a task

**technical
FEATURE**

Basic offering from system

**structural
COMPONENT**

Basic building block

TEST FOR WHAT?

L9 End user value

L8 Deployment correctness

L7 Attribute correctness

L6 Environment correctness

L5 Flow correctness

L4 Behaviour correctness

L3 Structural correctness

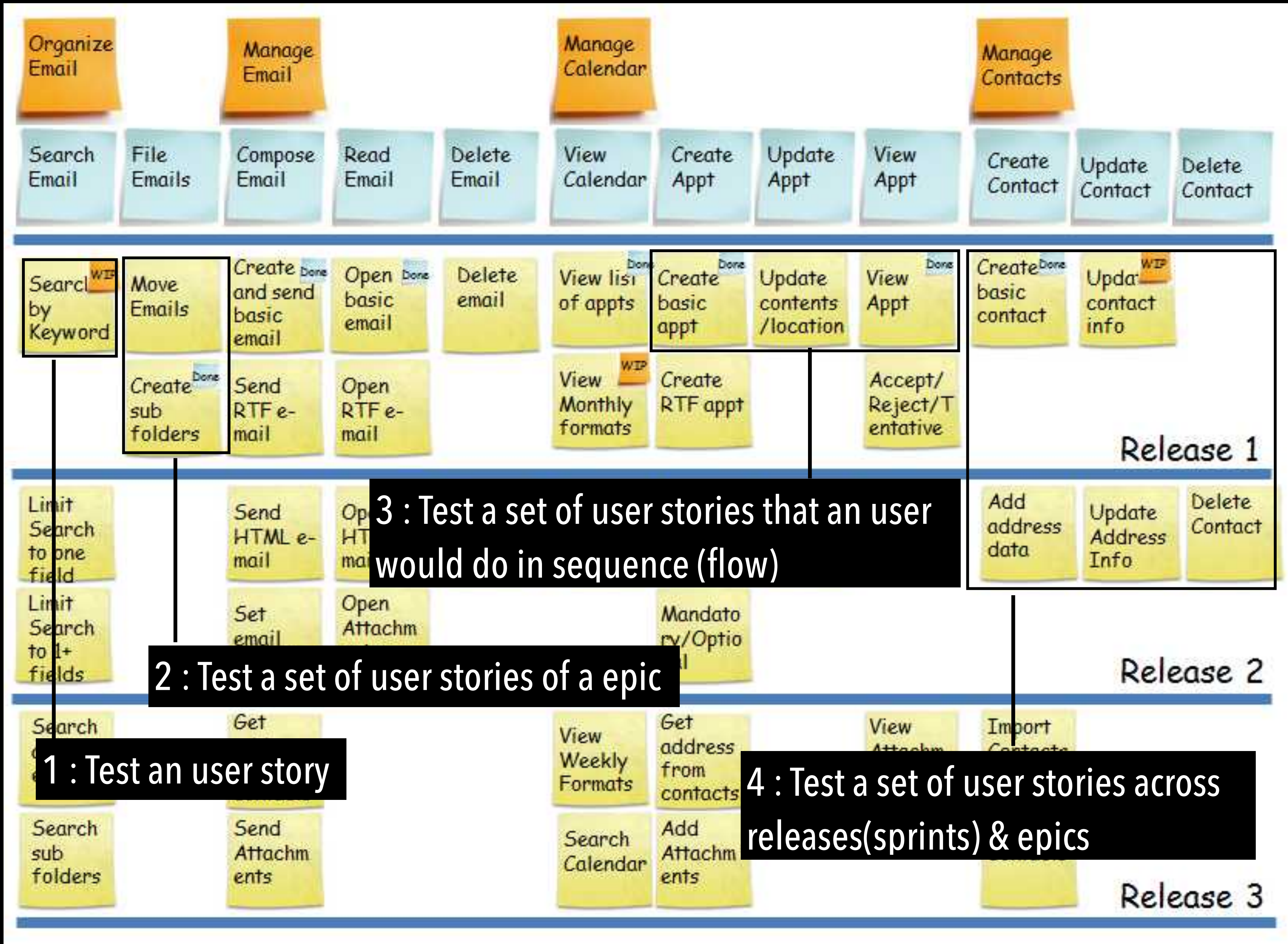
L2 Interface correctness

L1 Input correctness

Theme

Epic

User Story



1 : Test an user story

2 : Test a set of user stories of a epic

3 : Test a set of user stories that an user would do in sequence (flow)

4 : Test a set of user stories across releases(sprints) & epics

Dev/QA test objectives

WHAT TO TEST? Entity Under Test (EUT)

TEST FOR WHAT?

L9 End user value

L8 Deployment correctness

L7 Attribute correctness

L6 Environment correctness

L5 Flow correctness

L4 Behaviour correctness

L3 Structural correctness

L2 Interface correctness

L1 Input correctness

**technical
FEATURE**

Basic offering from system

Dev Test

**structural
COMPONENT**

Basic building block

WHAT TO TEST? Entity Under Test (EUT)

TEST FOR WHAT?

+
Shift
Left

L9 End user value

L8 Deployment correctness

L7 Attribute correctness

L6 Environment correctness

L5 Flow correctness

**technical
FEATURE**

Basic offering from system

Dev Test

L4 Behaviour correctness

L3 Structural correctness

L2 Interface correctness

**structural
COMPONENT**

Basic building block

L1 Input correctness

WHAT TO TEST? Entity Under Test (EUT)

TEST FOR WHAT?

**business
FLOW**

A set of tasks by different users to accomplish a business objective

**user
REQUIREMENT**

Enables an user to do a task

**technical
FEATURE**

Basic offering from system

**structural
COMPONENT**

Basic building block

**QA Test
(System)**

Dev Test

L9 End user value

L8 Deployment correctness

L7 Attribute correctness

L6 Environment correctness

L5 Flow correctness

L4 Behaviour correctness

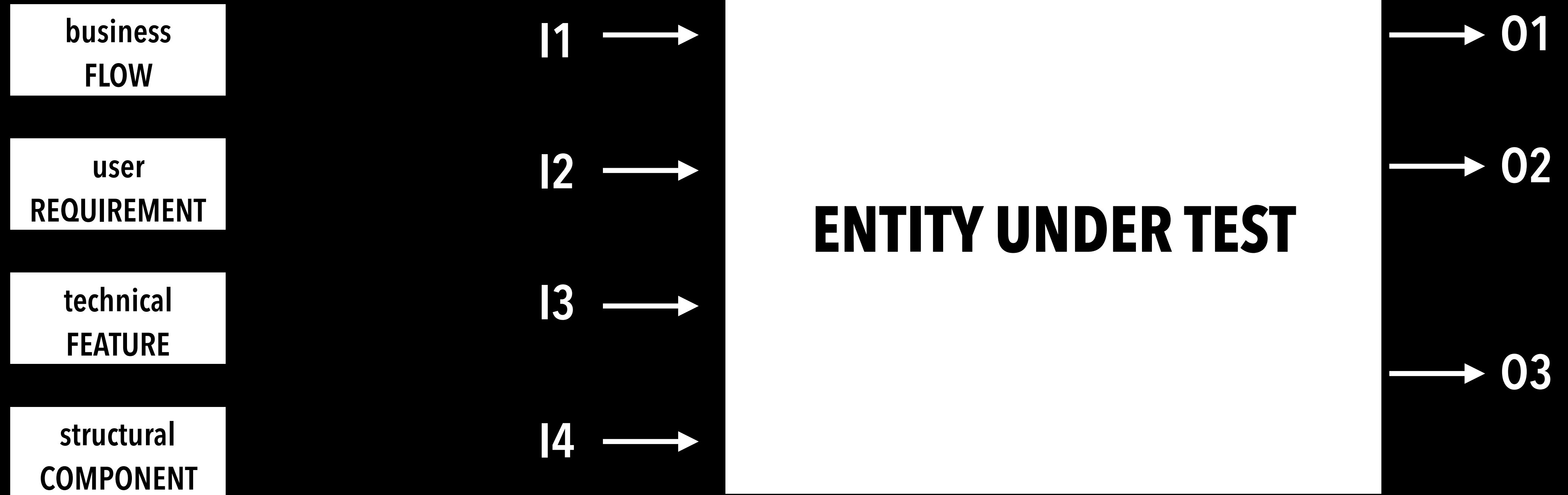
L3 Structural correctness

L2 Interface correctness

L1 Input correctness

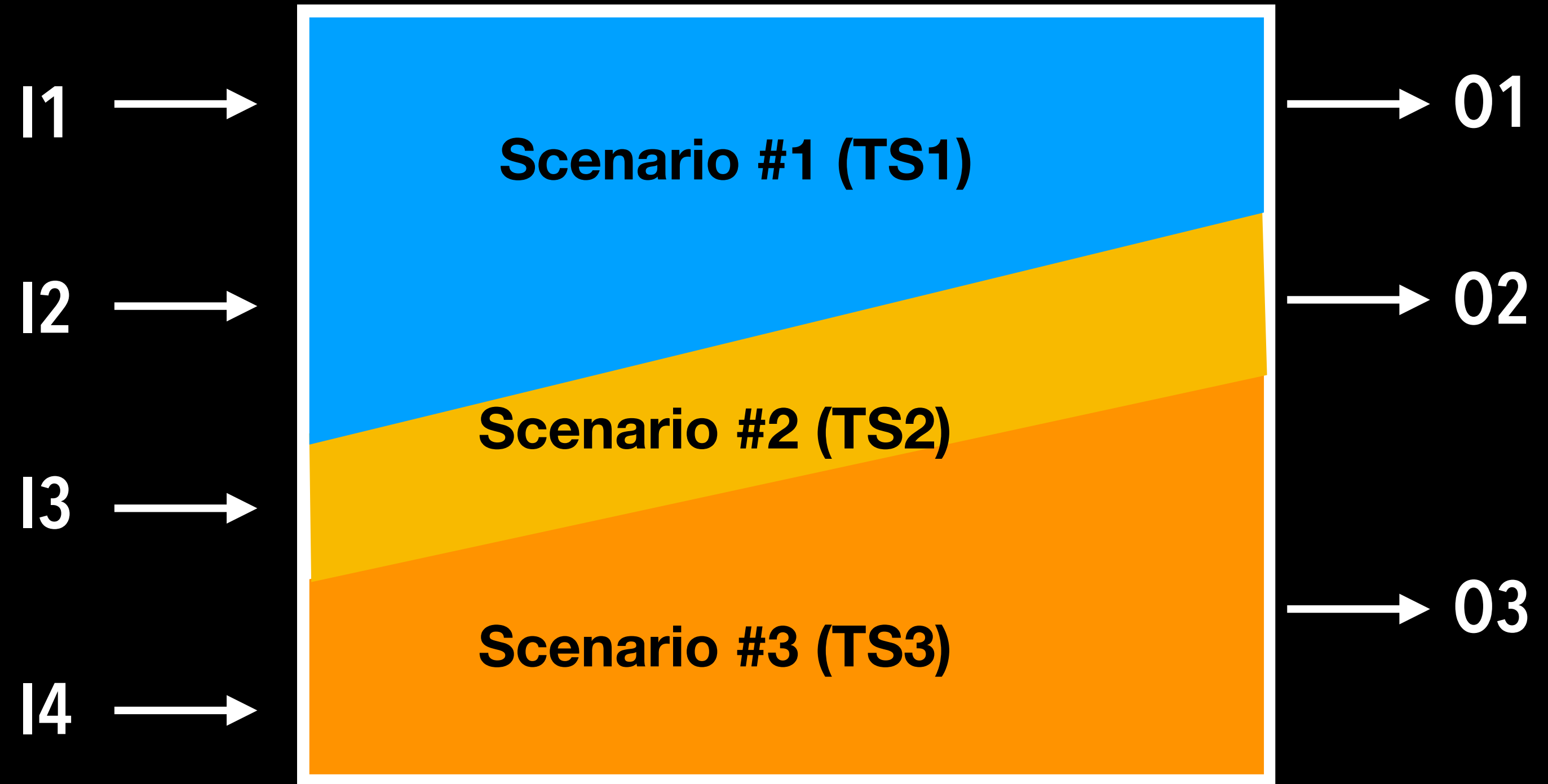
Test scenarios, cases

Here is a EUT that could be:



that has **FOUR** inputs & **THREE** outputs

ENTITY UNDER TEST



On analysis we see **THREE** distinct **behaviours**
i.e. **test SCENARIOS**

ENTITY UNDER TEST

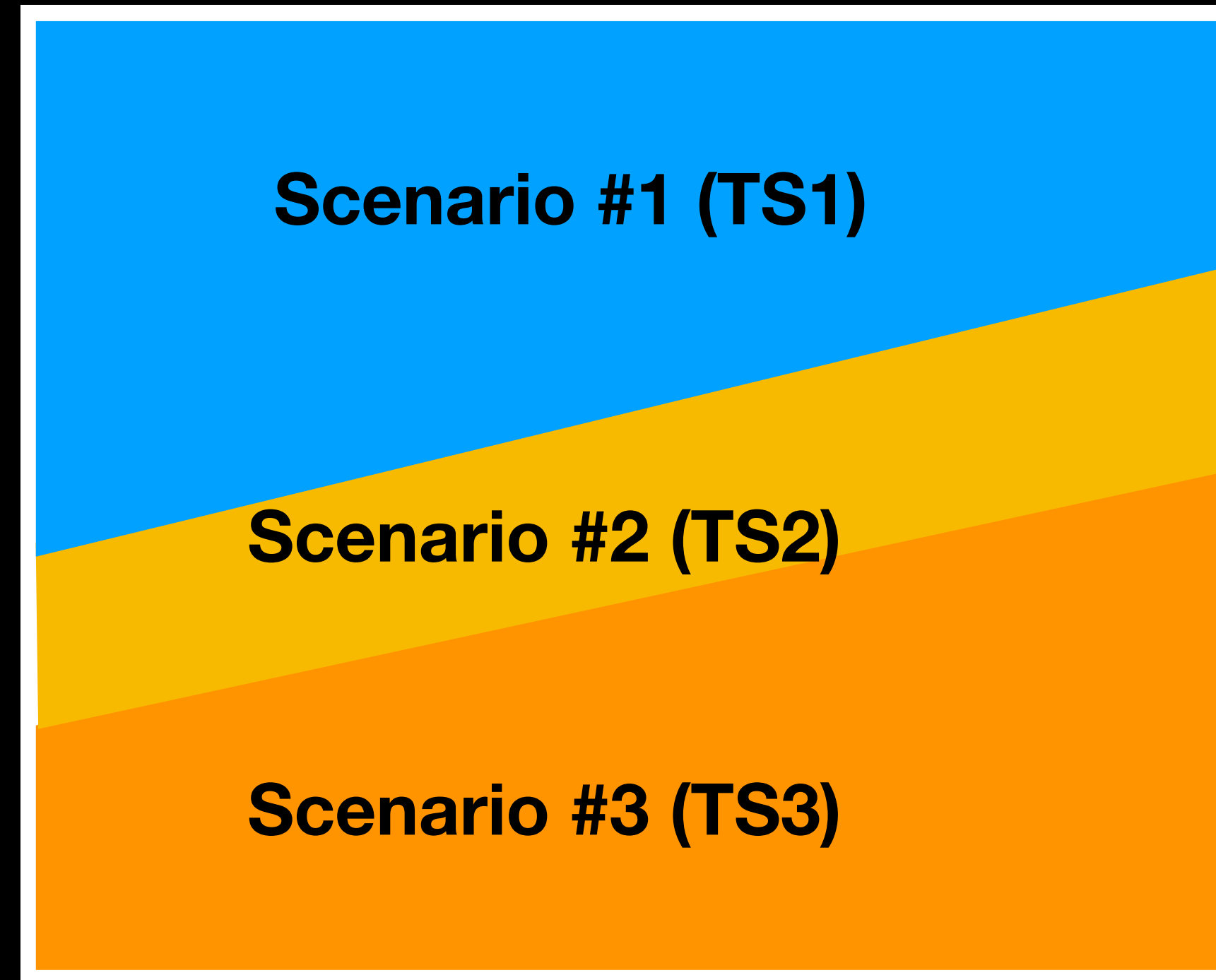
A1	A2	A3
B1	B2	B3
C1	C2	C3
D1	D2	D4

I1 →

I2 →

I3 →

I4 →



→ O1

→ O2

→ O3

To **stimulate** Scenario #1, it takes THREE sets of distinct combination of inputs i.e. **test CASES**

TEST CASES for

TS3

A6	A7
B6	B7
C6	C7
D6	D7

TS2

A4	A5
B4	B5
C4	C5
D4	D5

TS1

A1	A2	A3
B1	B2	B3
C1	C2	C3
D1	D2	D4

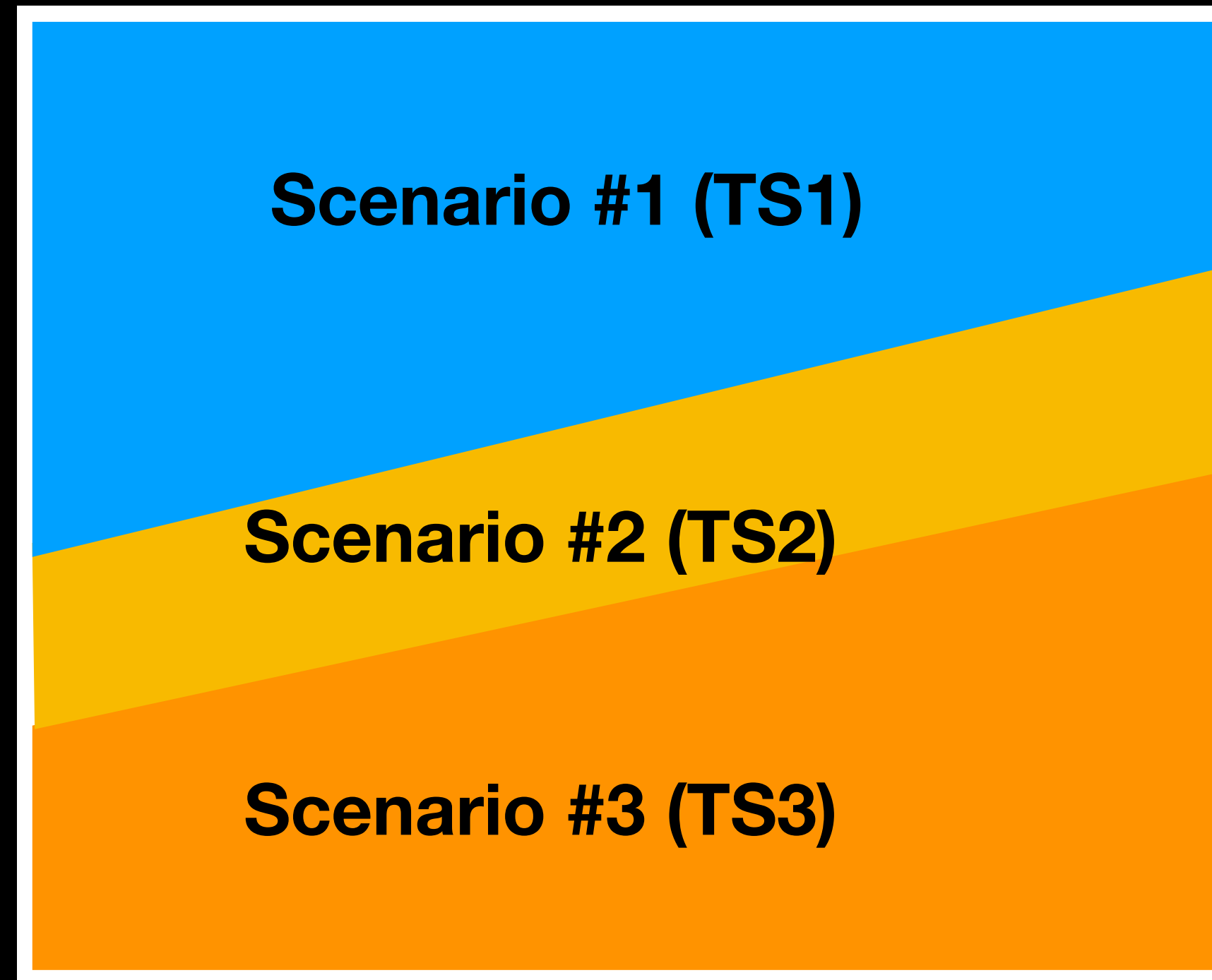
I1 →

I2 →

I3 →

I4 →

ENTITY UNDER TEST



→ O1

→ O2

→ O3

TEST CASES for

TS3

TS2

TS1

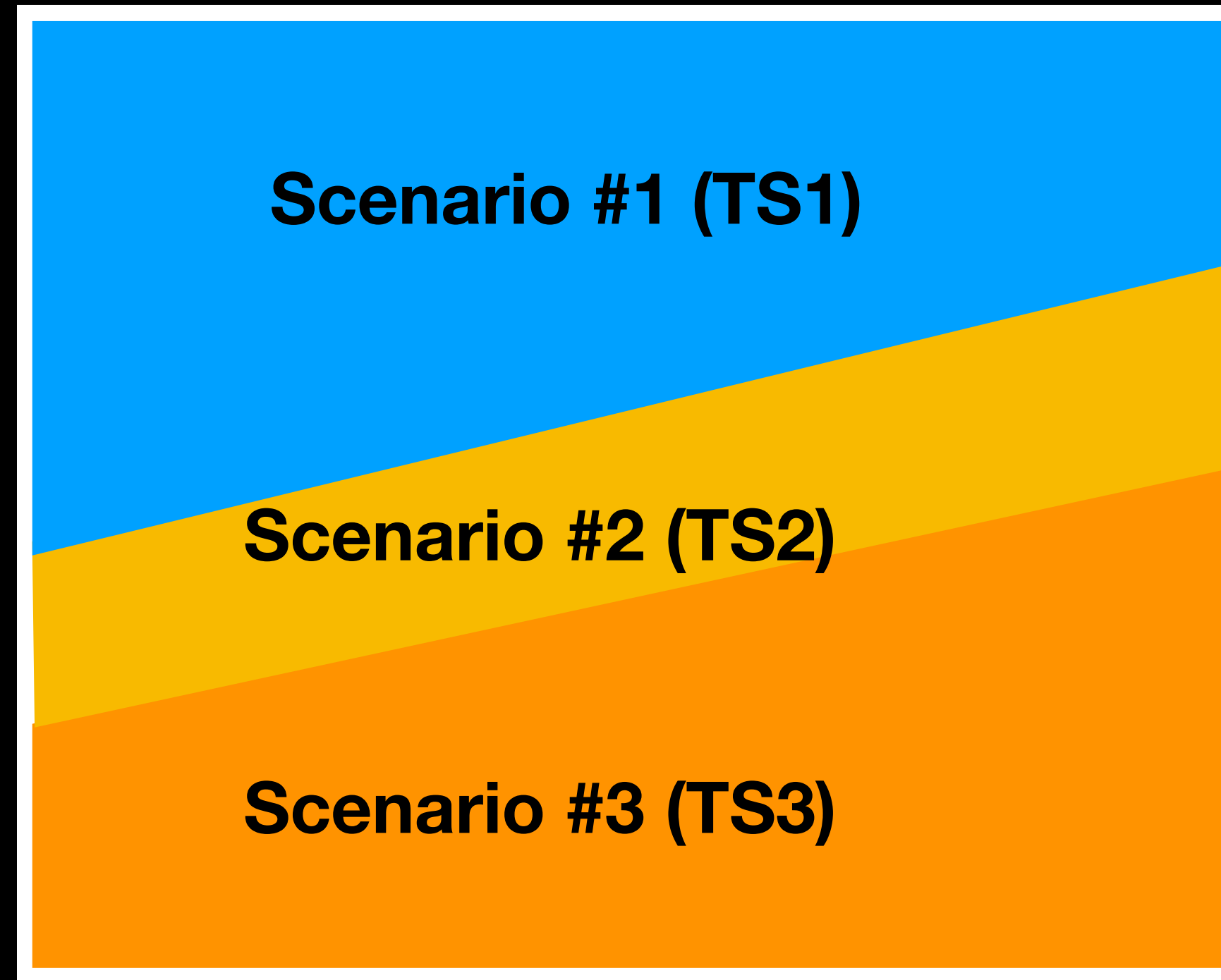
A6	A7
B6	B7
C6	C7
D6	D7

A4	A5
B4	B5
C4	C5
D4	D5

A1	A2	A3
B1	B2	B3
C1	C2	C3
D1	D2	D4

I1 →
I2 →
I3 →
I4 →

ENTITY UNDER TEST



→ O1
→ O2
→ O3

Test SCENARIOS represent behaviours
Test CASES are stimuli

Test approaches

(Scripted, Ad hoc, Exploratory, Automated...)

doing **STYLE**

activity **SEQUENCE**

predominantly **USES**

A

completed SCRIPTED

U

D

E



Validation Approaches

U=understand | D=design | E=evaluate

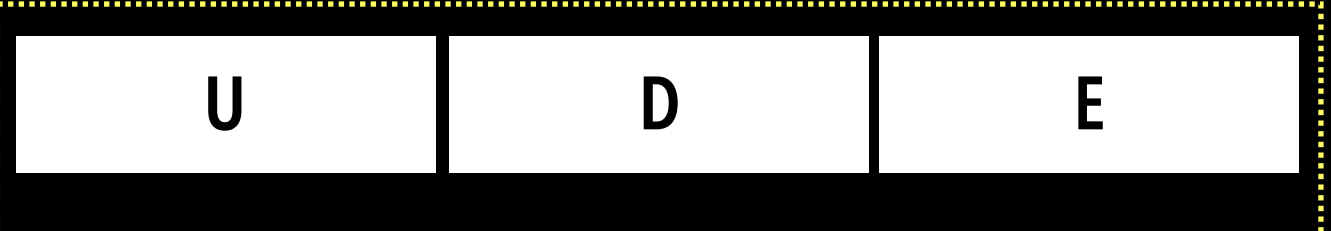
Brain picture from <https://images.app.goo.gl/uvuqi95q7cDupGXS9>

doing STYLE

activity SEQUENCE

predominantly USES

A completed SCRIPTED



**B completely UNSCRIPTED
(ad hoc)**



Validation Approaches

U=understand | D=design | E=evaluate

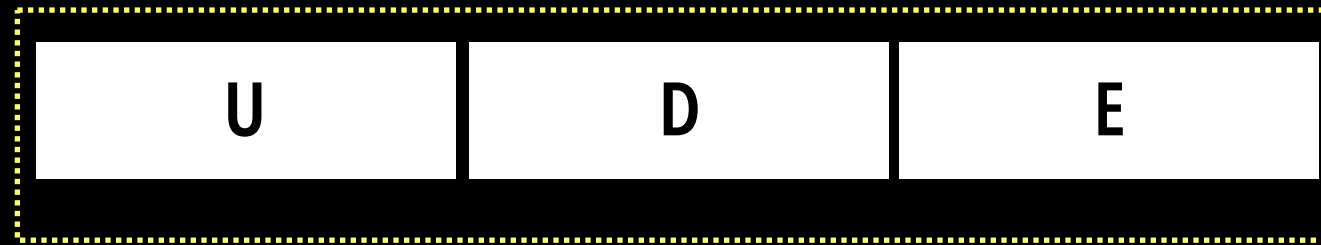
Brain picture from <https://images.app.goo.gl/uvuqi95q7cDupGXS9>

doing STYLE

activity SEQUENCE

predominantly USES

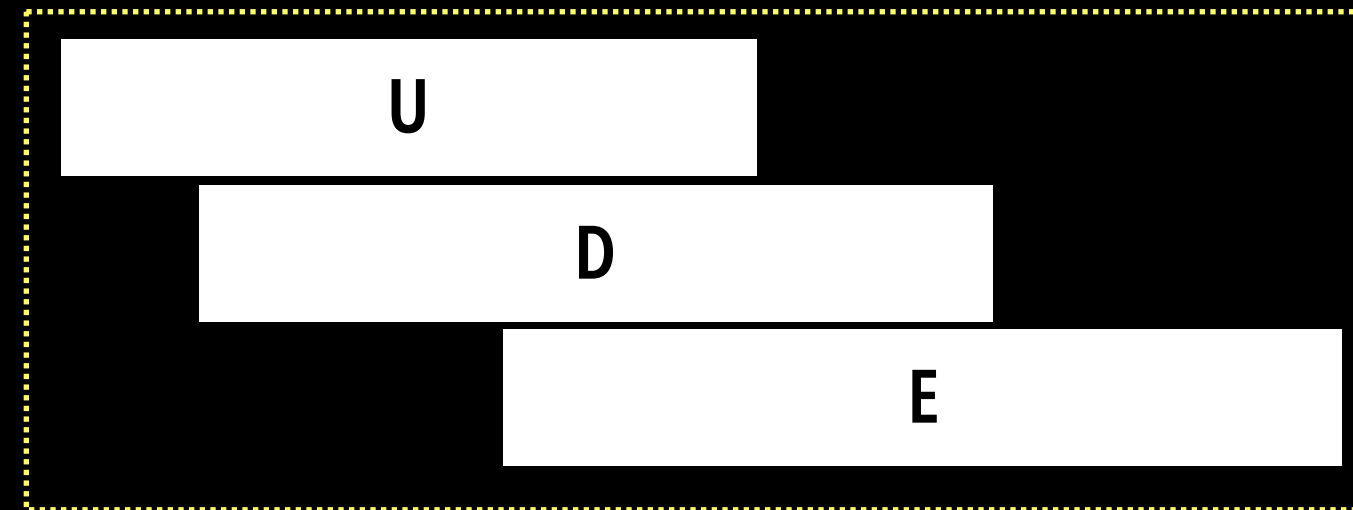
A completed SCRIPTED



B completely UNSCRIPTED
(ad hoc)



C EVOLVING script
(exploratory)

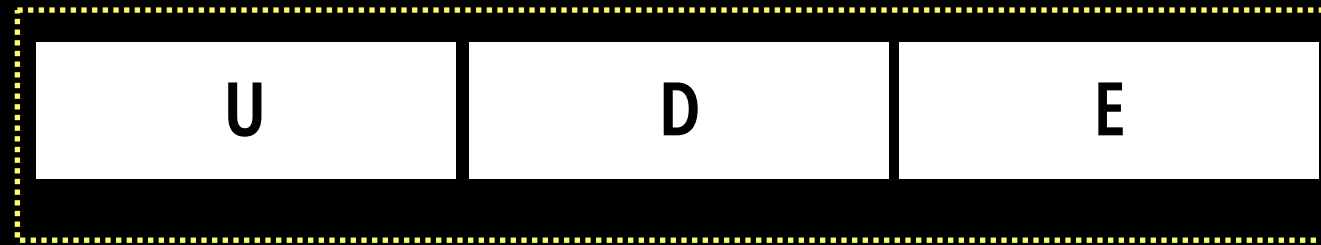


doing STYLE

activity SEQUENCE

predominantly USES

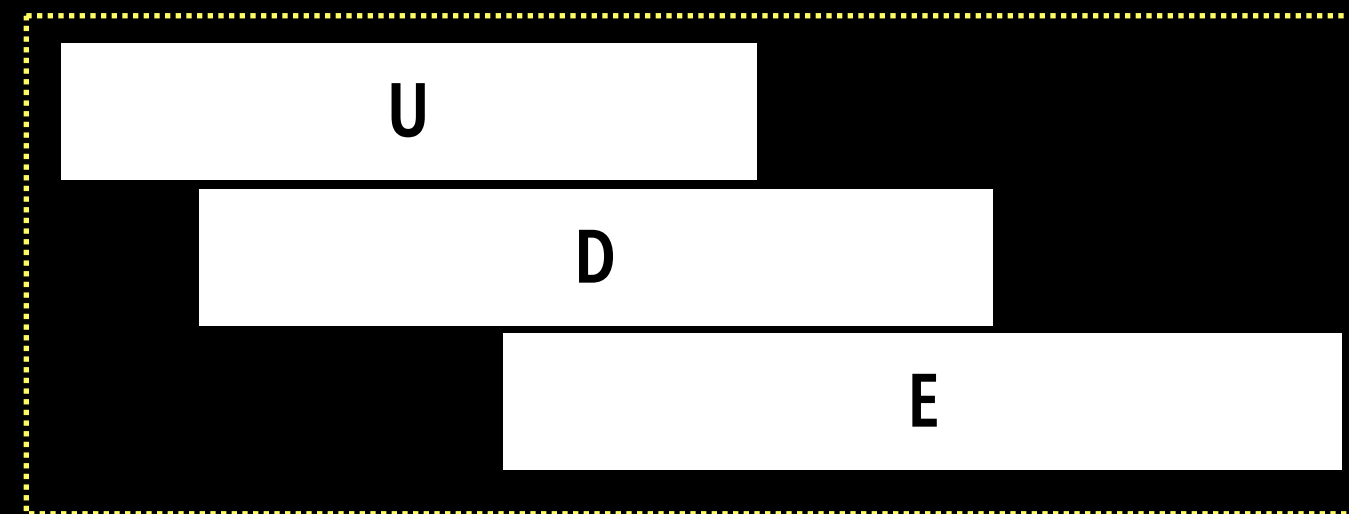
A completed **SCRIPTED**



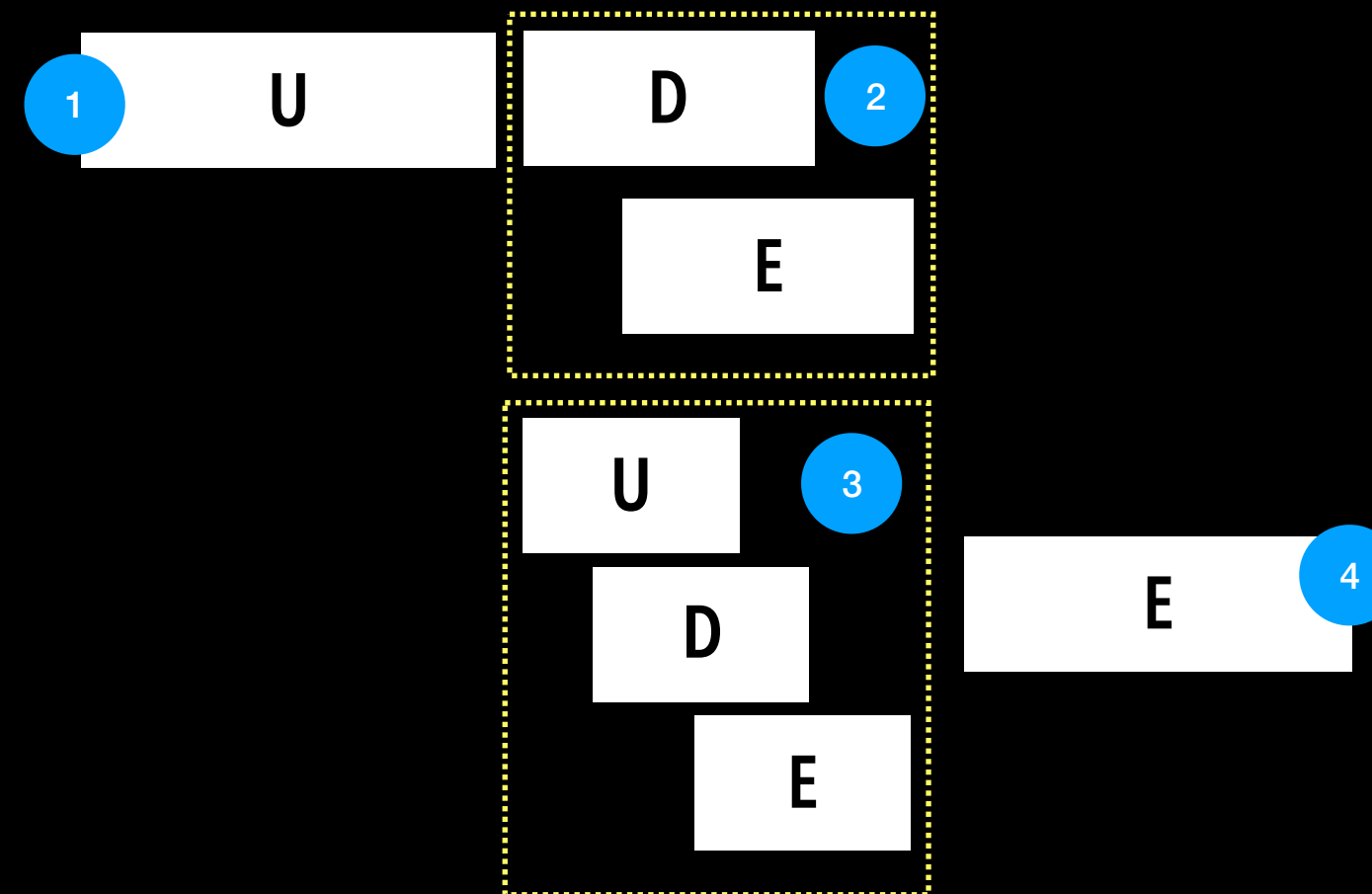
B completely **UNSCRIPTED**
(ad hoc)



C **EVOLVING** script
(exploratory)



D **EVOLVING** script
(SESSION BASED)



Validation Approaches

U=understand | D=design | E=evaluate

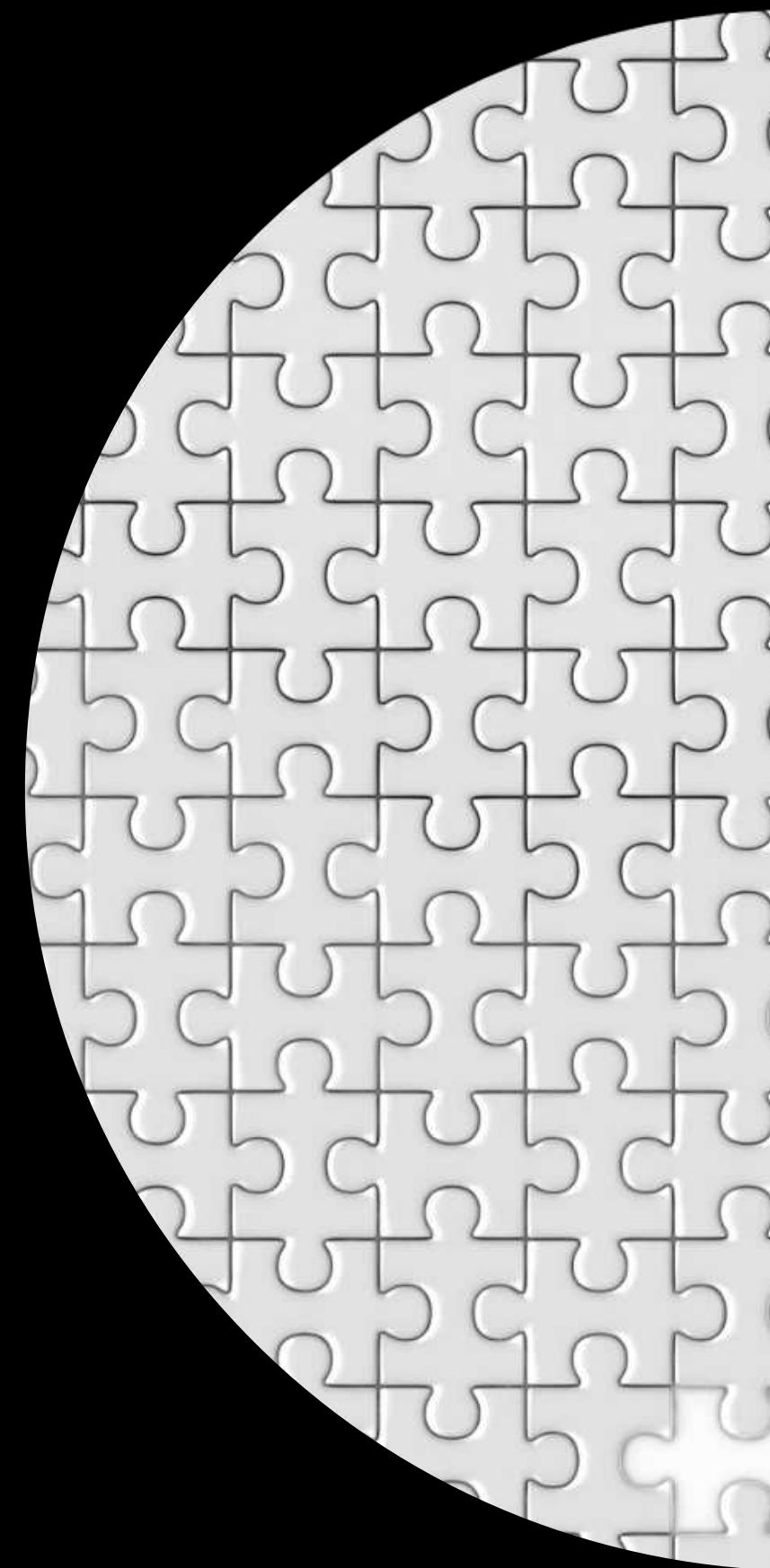
Brain picture from <https://images.app.goo.gl/uvuqi95q7cDupGXS9>

Once we discover SCENARIOS
we can choose to **AUTOMATE** it, this becomes

CHECKING

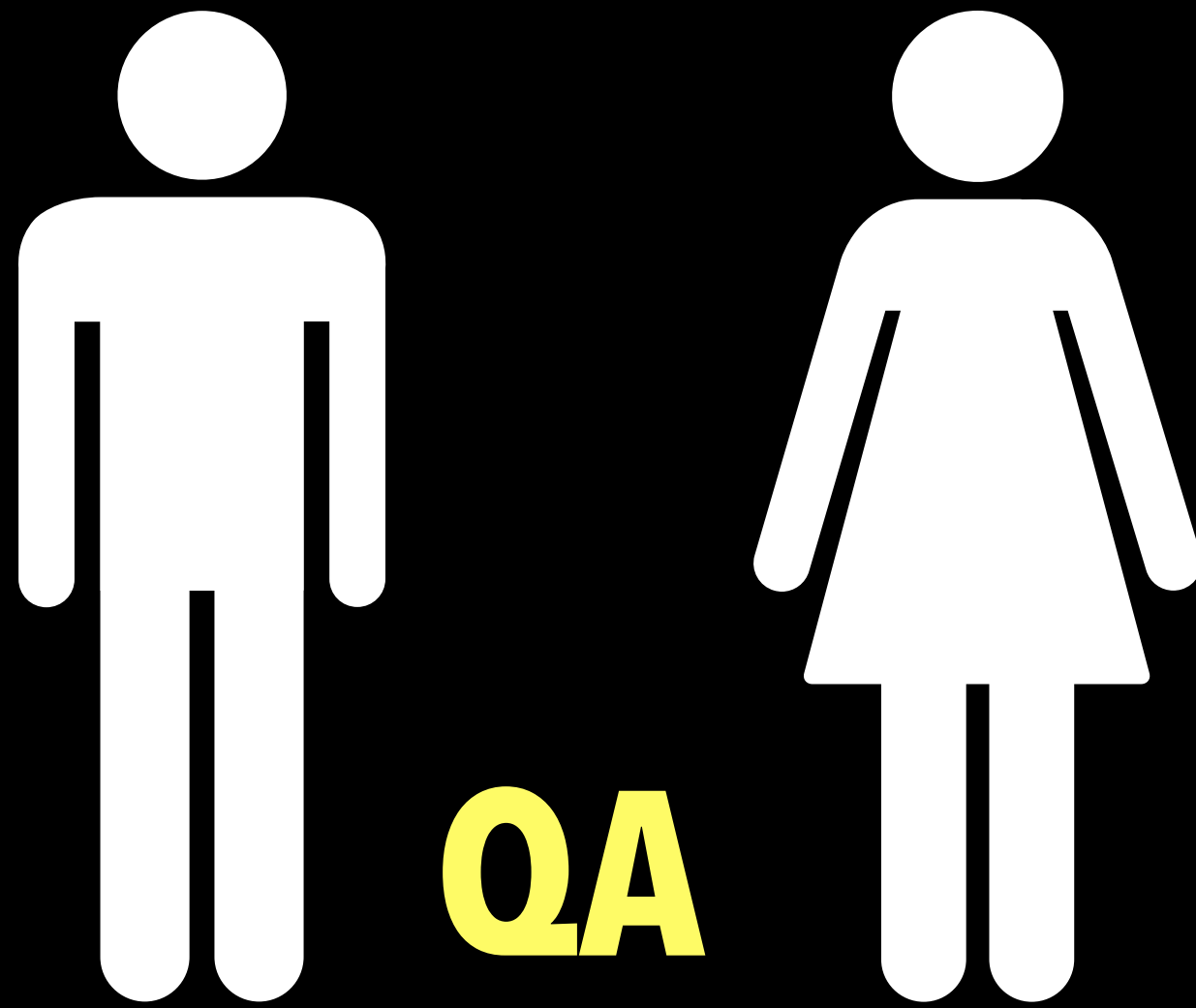
is comparing
can be scripted
binary outcome Pass/Fail
design approach -
logical, experience

based on spec
wellness
AUTOMATED



Role of QA

**Executor, Automator, Designer,
Questioner, Suggestor, Analyser**



What is their role?

Designer

Executor



What is their role?

Designer

Automator

Executor



What is their role?

Suggestor

Questioner

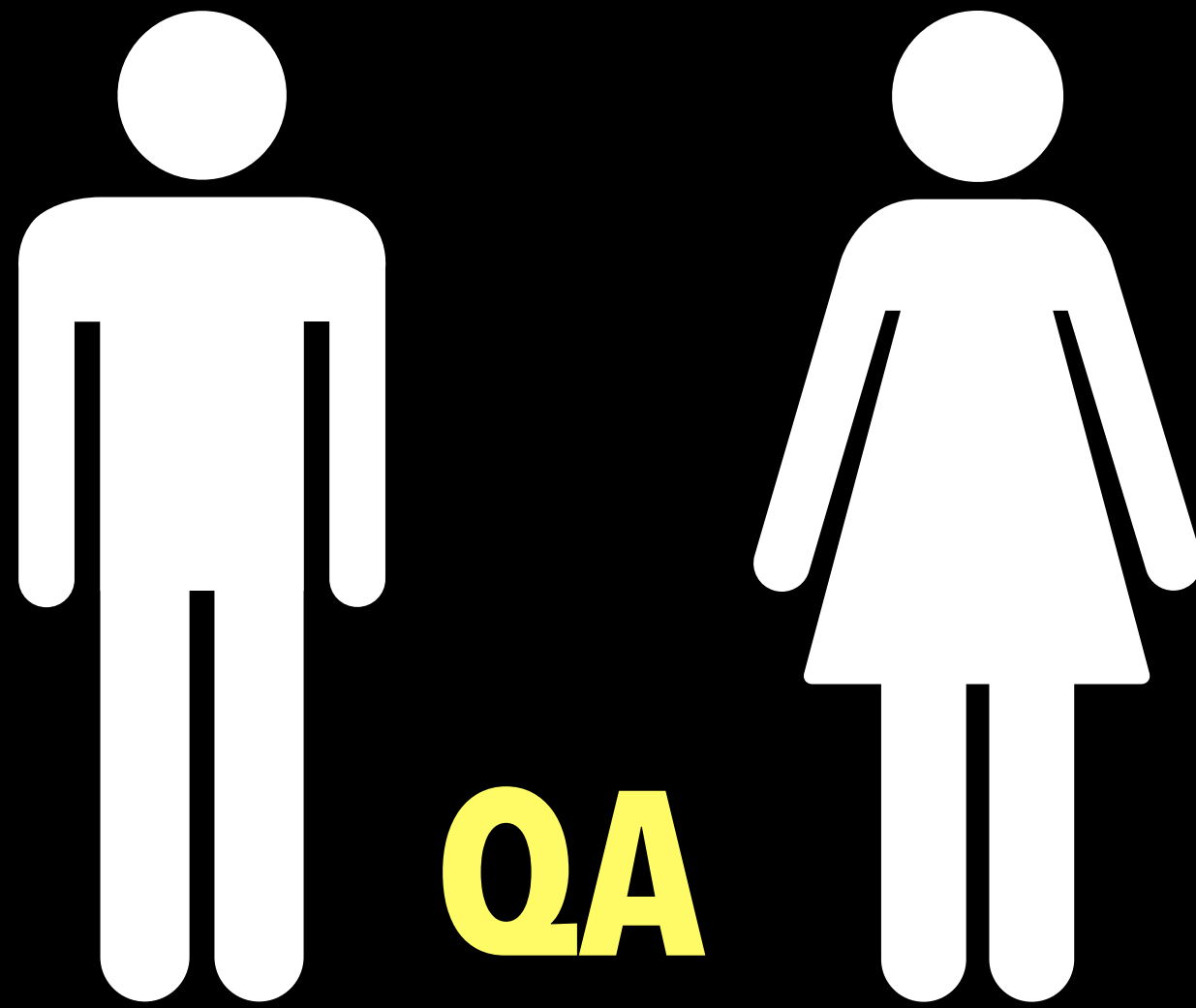
Designer

Automator

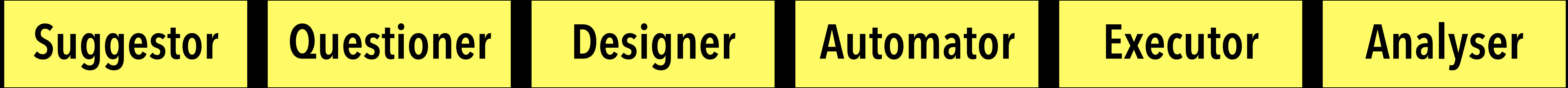
Executor

**Shift LEFT
to do less**

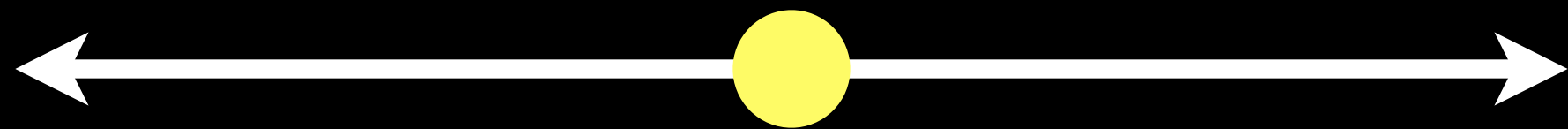




What is their role?



Shift LEFT
to do less



Shift RIGHT
to add value

On Automation

What & When - Making choices

Implementation

Dev test

Validation

System test

- Regression
- Functional
- Non-functional tests

What & When - Making choices

Implementation

Dev test

UI-less/API

Validation

System test

- Regression
- Functional
- Non-functional tests

UI, API

What & When - Making choices

Implementation

Dev test

UI-less/API

Incorporate testability

Validation

System test

- Regression
- Functional
- Non-functional tests

UI, API

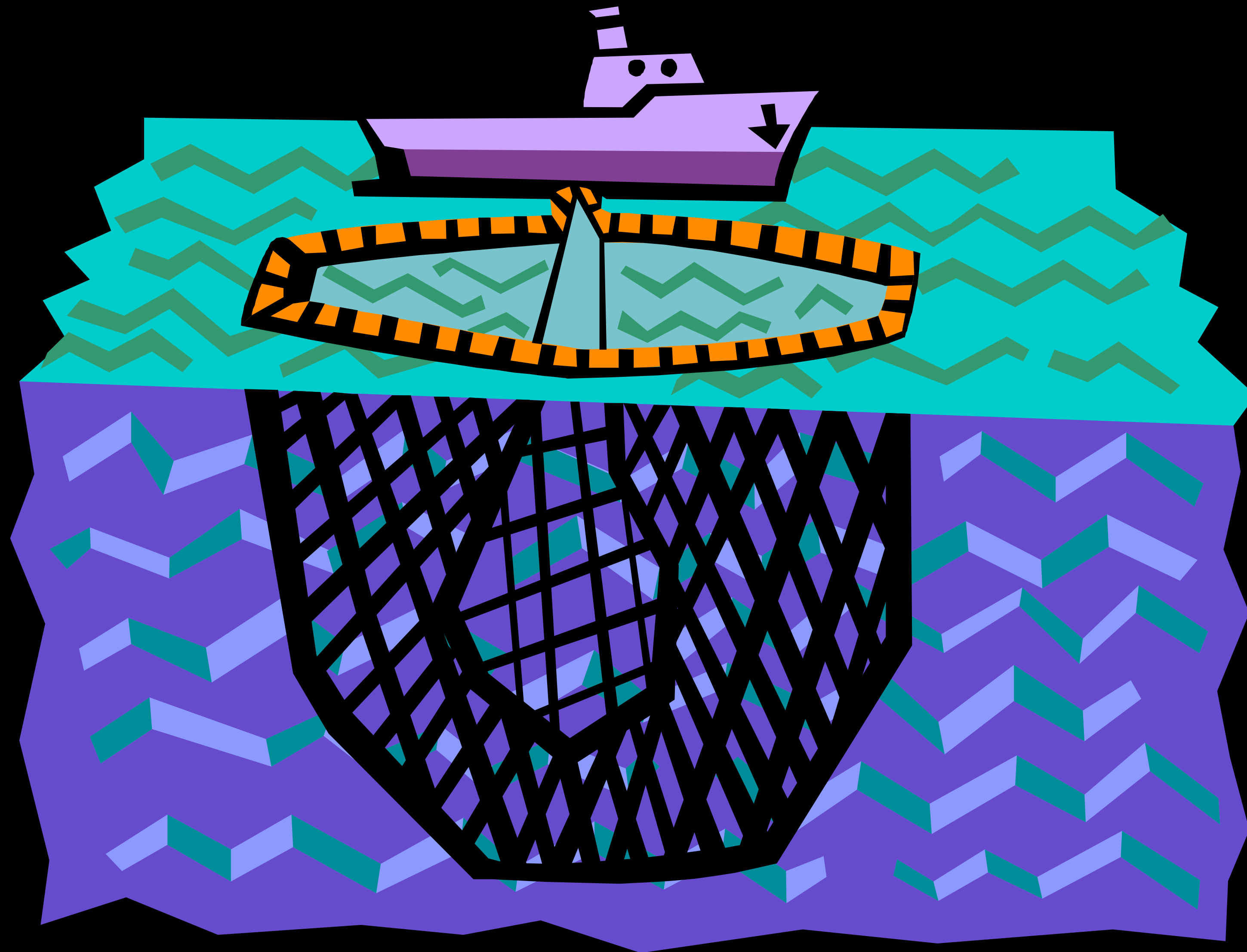
Evaluation

- Setup+Execution +Oracle
- Setup | Execution | Oracle

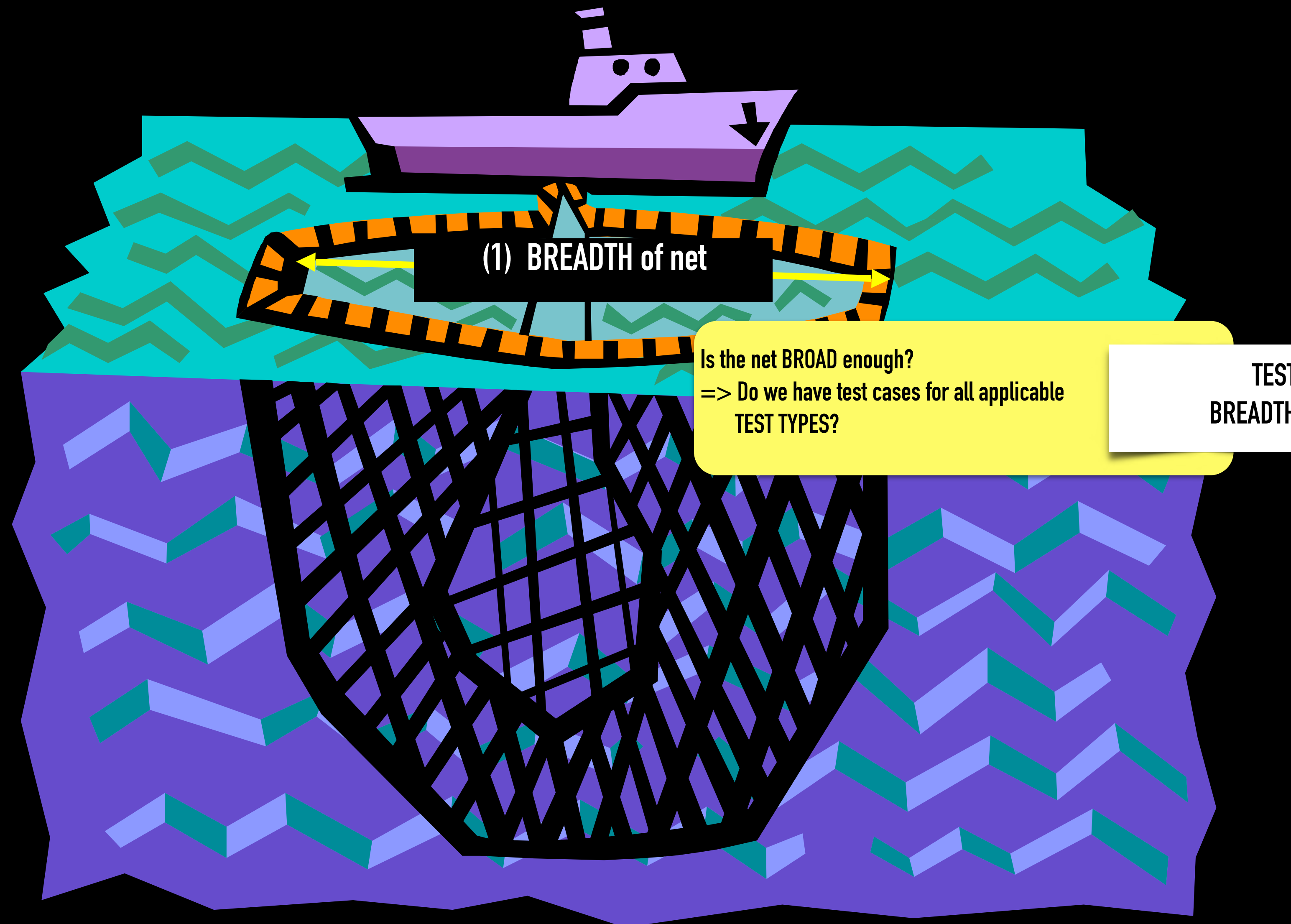
**Metrics of interest
related to effectiveness, efficiency**

FISHNET ANALOGY

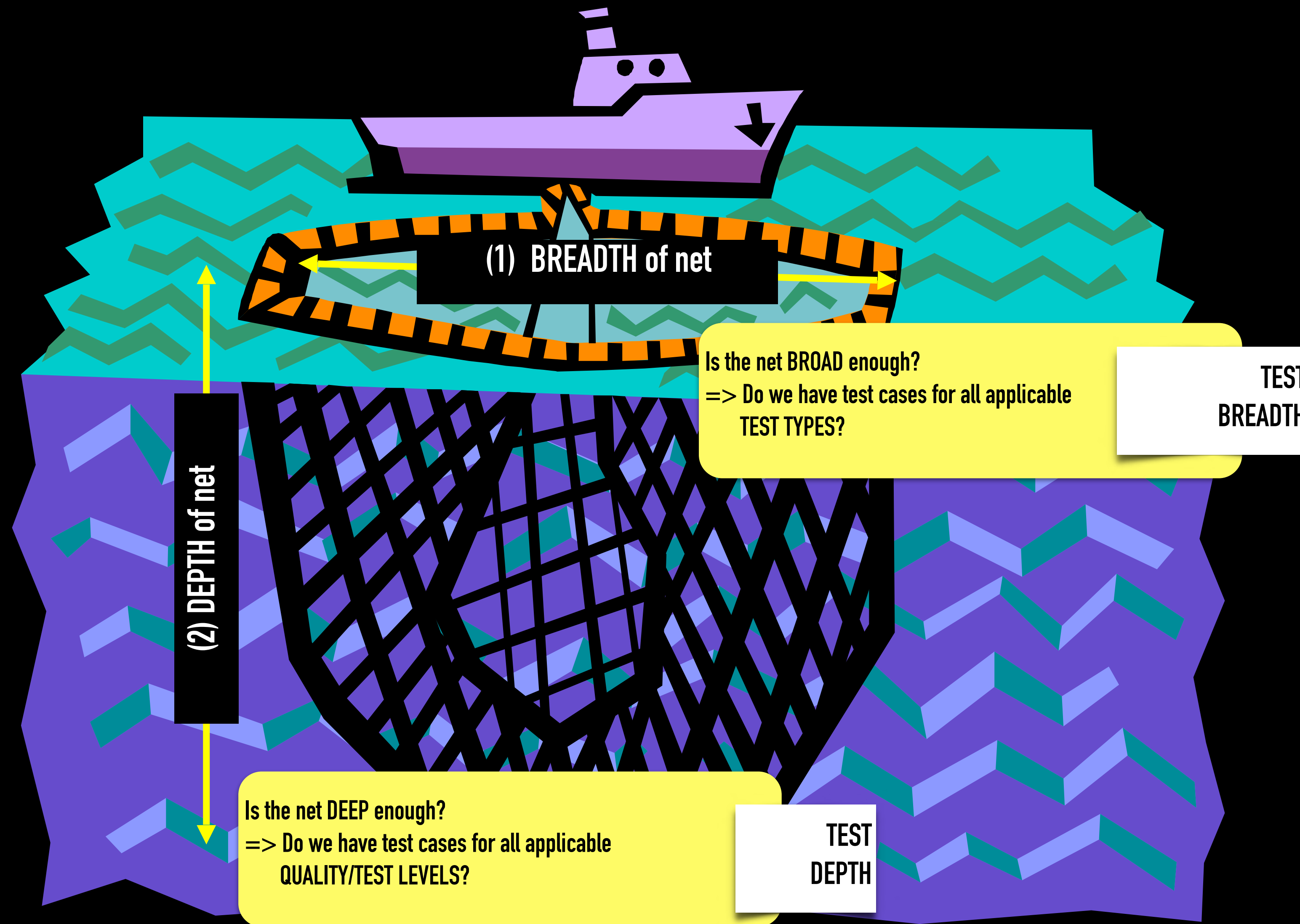
what decides how EFFECTIVE a net will be to catch fishes?



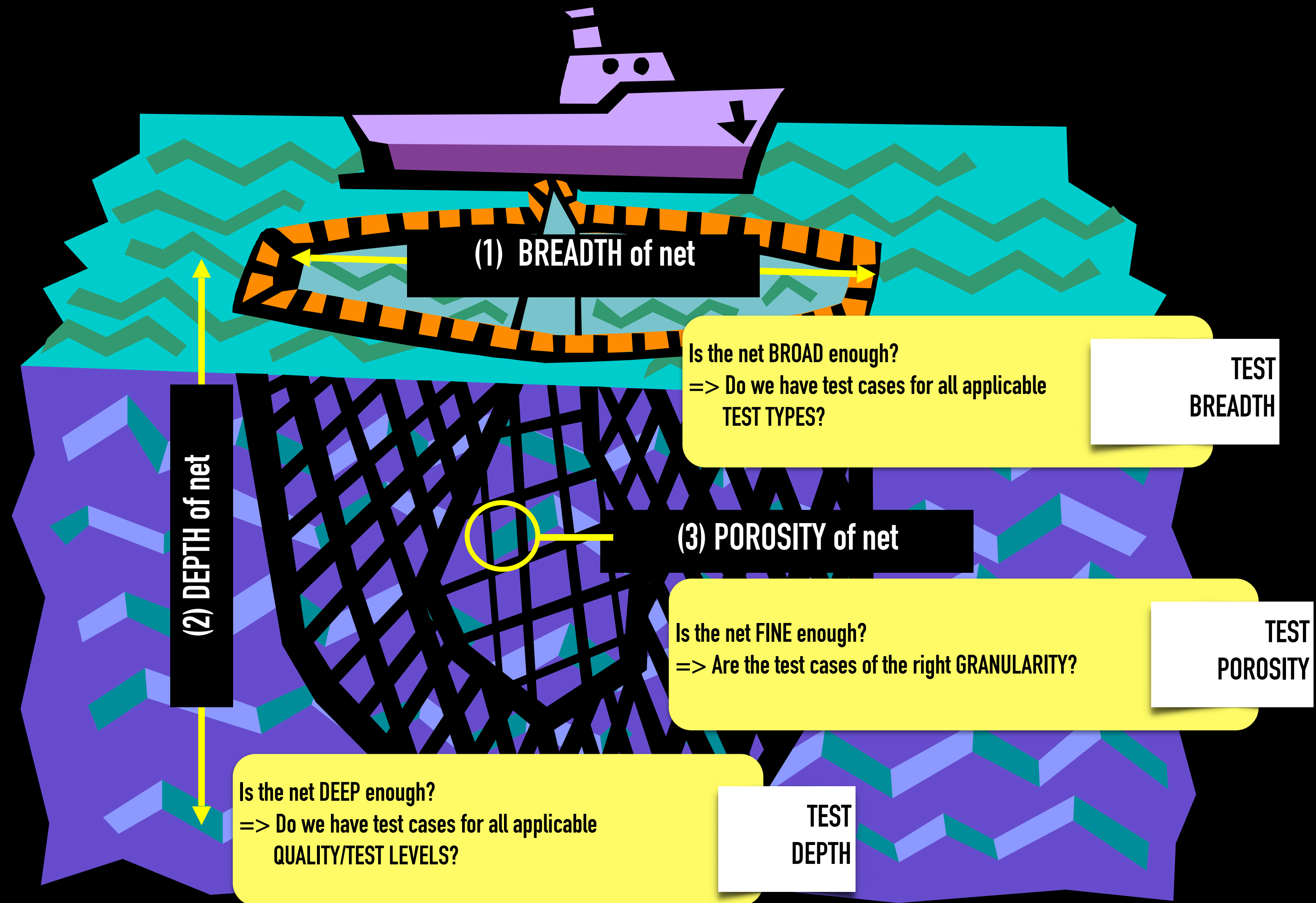
Test asset measures - EFFECTIVENESS



Test asset measures - EFFECTIVENESS



Test asset measures - EFFECTIVENESS



Test asset analysis for **EFFECTIVENESS**

**Measures
related to**

How can we use the measures?

Counts		Can we justify why we have this number of scenario/cases?
Distribution	by ENTITIES	Do we have appropriate number of test cases for each entity?
	by LEVELS	Are there more test cases at lower levels? as #TC is proportional to number of inputs.
	by TEST TYPES	Do we have test cases for different types of tests and are they justifiable
	by OBJECTIVE (positive/negative)	is there any skew with respect to the distribution of positive/negative scenarios/cases?
	by PRIORITY	Is there a meaningful correlation between the number of test cases and the importance of the entity?

The objective is to enable us to spot skewness and take suitable actions.

Outcome measure for EFFECTIVENESS

L9 End user value

L8 Deployment correctness

L7 Attribute correctness

L6 Environment correctness

L5 Flow correctness

L4 Behaviour correctness

L3 Structural correctness

L2 Interface correctness

L1 Input correctness

(1) How are defects distributed by QUALITY LEVELS?

**(2) Given that QA is L5 and upwards,
how much is leaking from earlier levels?**

(3) How do customer reported escapes stratify?

Outcome measure for EFFECTIVENESS

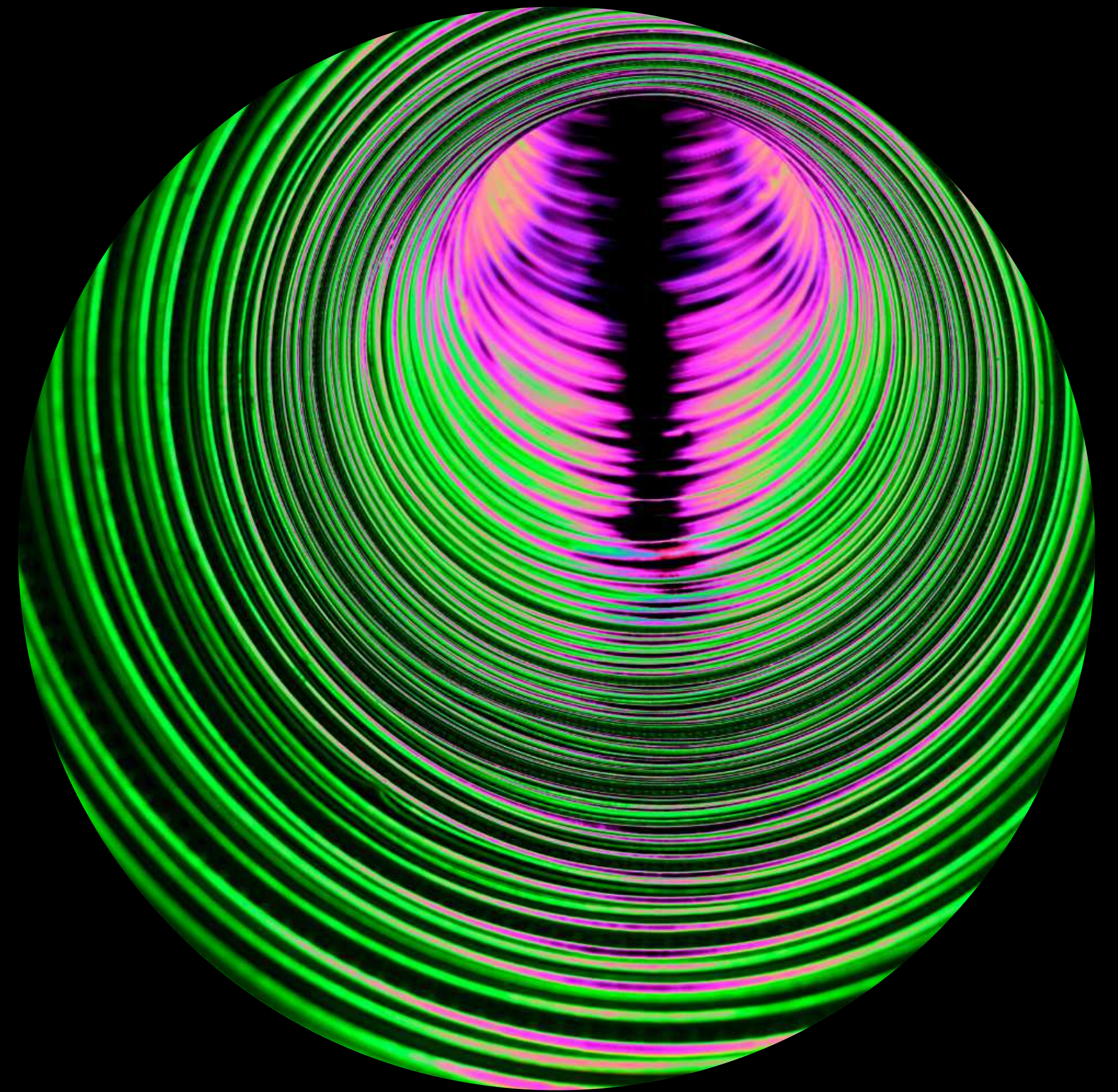
Escape Analysis (INT & EXT) is a great mirror



Stratify escapes by quality levels to act effectively

Smart QA - Do less, Accomplish More

Treat testing as good work to
produce systems of value



Thank you.



© 2020, STAG Software Pvt Ltd

www.stagsoftware.com

smartqa.org

A large, stylized graphic on the right side of the slide. It consists of a thick, glowing golden ring made of many small particles, forming a circle. The text "SmartQA" is written in a yellow, sans-serif font across the center of the ring.

SmartQA