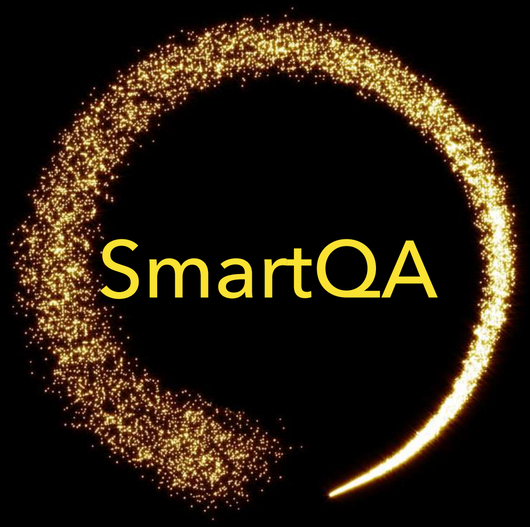


IST Masterclass
Session #6

Recoup phase in detail

© 2000-21, STAG Software Pvt Ltd

www.stagsoftware.com

TOPICS

Recoup - What is it?

Judging adequacy

Measuring quality

Analysing progress

Defect escapes

Check vs. Test effort

Analysis for clear actions

A quick summary of IST

IST Adoption Tips

Recoup - What is it?

RECONNAISANCE

EXPLORATION

RECOUP

Analyse , Reflect & Learn

RECONNAISANCE

EXPLORATION

RECOUP

Analyse , Reflect & Learn

Test adequacy
Execution progress
Product quality
Practice effectiveness (Check vs Test & Escapes)
Analysis for improvement

RECOUP Phase

“After a hard trek, recover”

RECOUP Phase

At the end of a session

Analyse actions & outcomes

RECOUP Phase

At the end of a session

Analyse actions & outcomes

Reflect how it went

RECOUP Phase

At the end of a session

Analyse actions & outcomes

Reflect how it went

Learn from experience

Judging adequacy

Adequacy

L9 Business 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

"various types of issues covered"

Adequacy

L9 Business value

L8 Deployment correctness

L7 Attribute correctness

L6 Environment correctness

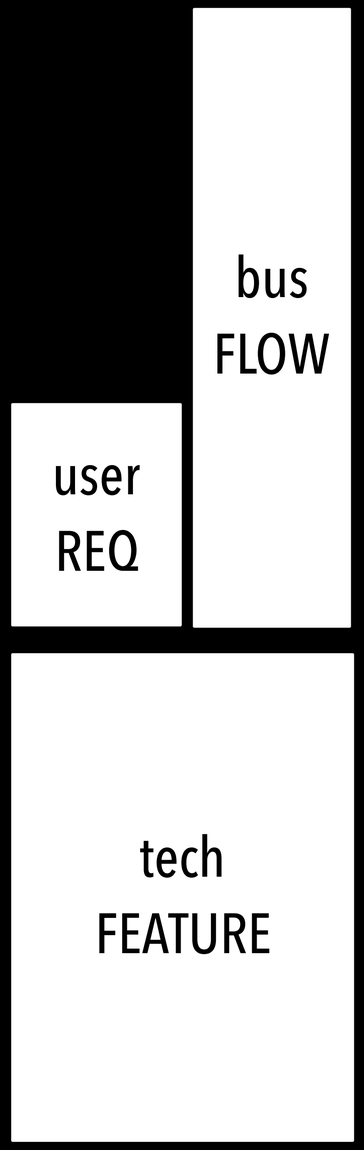
L5 Flow correctness

L4 Behaviour correctness

L3 Structural correctness

L2 Interface correctness

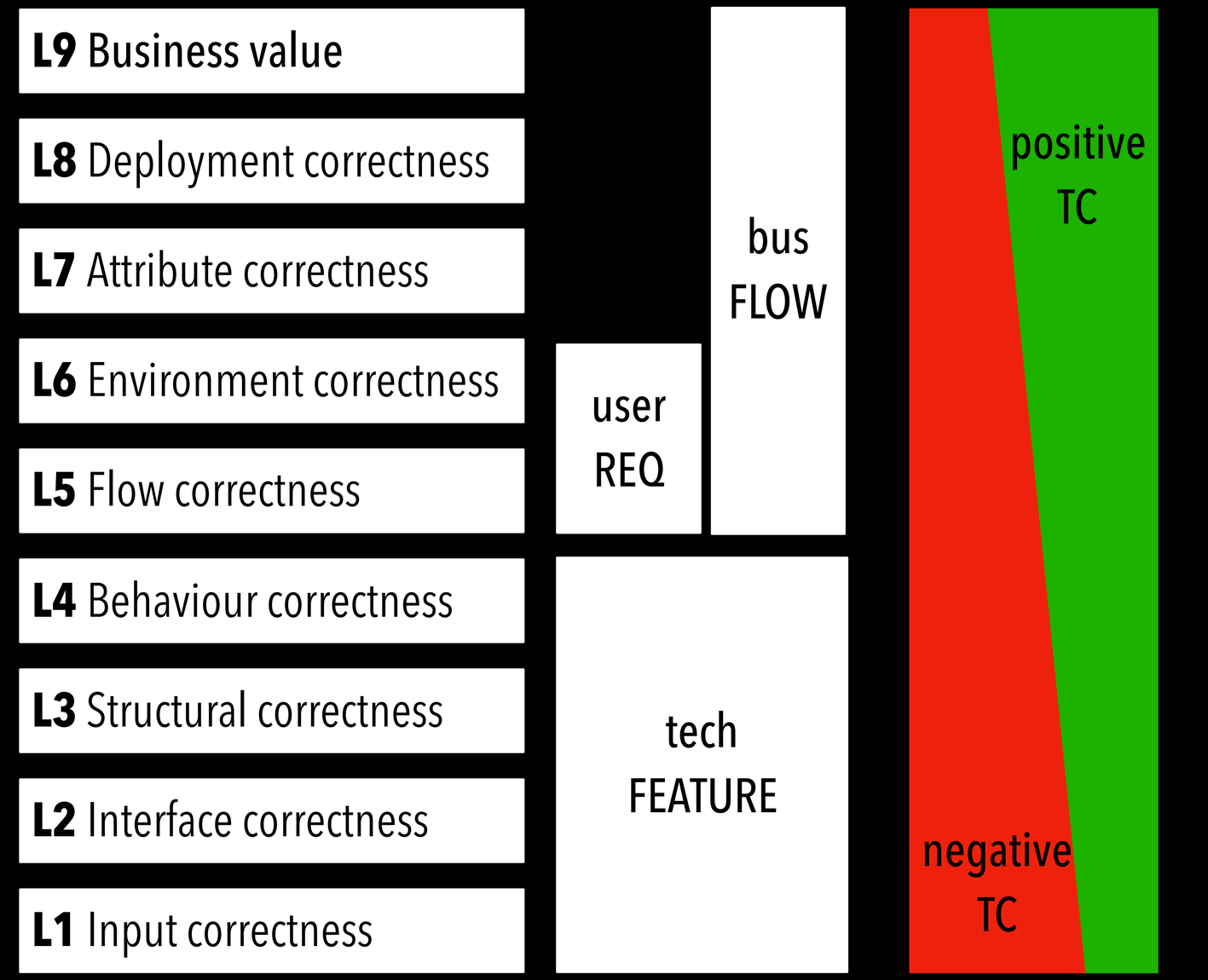
L1 Input correctness



"various types of issues covered"

"tested small to big"

Adequacy


L9 Business 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
user REQ
bus FLOW
tech FEATURE
positive TC
negative TC
"various types of issues covered"
"tested small to big"
"conforms & robust"

Adequacy

L9 Business 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

bus FLOW

user REQ

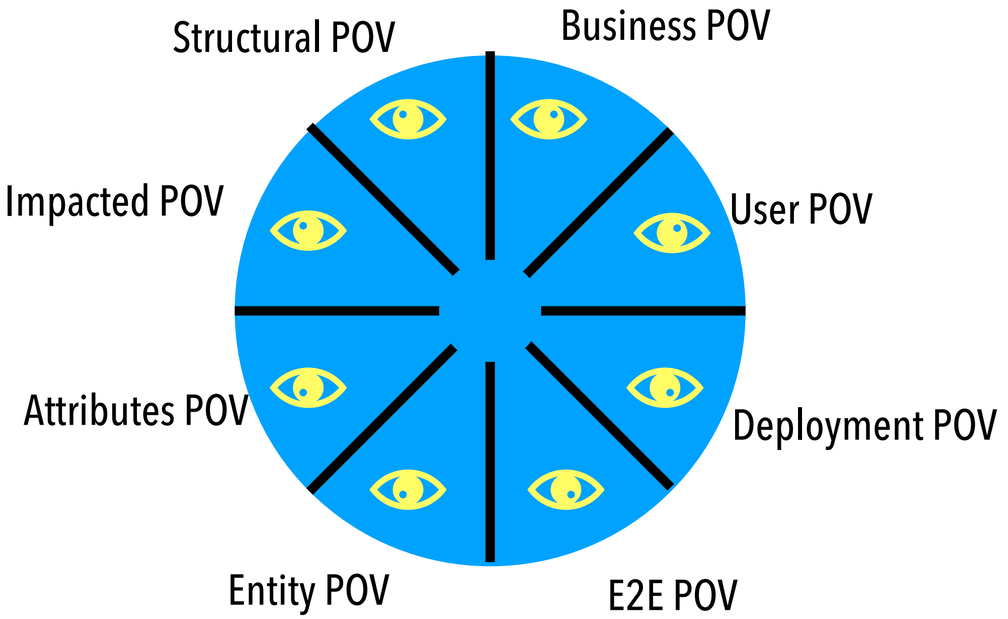
tech FEATURE

positive TC

negative TC

"various types of issues covered" **"tested small to big"** **"conforms & robust"**

POV



"examined from various points of view"

Dashboard #1 : Test quality

	#TC	#Positive	#Negative
L9			
L8			
L7			
L6			
L5			
L4			
L3			
L2			
L1			

Dashboard #1 : Test quality

	#TC	#Positive	#Negative
L9			
L8			
L7			
L6			
L5			
L4			
L3			
L2			
L1			

	#Entities
FLOWS	
REQS	
FEATURES	

Measuring quality

Can we quantify quality ?

Hmmm, seems very amorphous.

Can we quantify quality ?

Hmmm, seems very amorphous.

Guess we can come with indicator as a measure.

Can we quantify quality ?

Hmmm, seems very amorphous.

Guess we can come with indicator as a measure.

If we have a reasonable reference
of what may be adequate
then we can setup a simple indicator.

Dashboard #2- System quality

	#TC	%Pass
L9		
L8		
L7		
L6		
L5		
L4		
L3		
L2		
L1		

The reasonable reference
of what may be adequate is **TC**
The simple indicator of quality is **%Pass**

Analysing progress

Remember the **session plan**, it was really:

	EUT	TEST	ENV
1	E1	T1	V1

Remember the **session plan**, it was really:

	EUT	TEST	ENV
1	E1	T1	V1
2	E2	T1	V1

Remember the **session plan**, it was really:

	EUT	TEST	ENV
1	E1	T1	V1
2	E2	T1	V1
3	E2	T2	V2
..			
N			

Remember the **session plan**, it was really:

	EUT	TEST	ENV
1	E1	T1	V1
2	E2	T1	V1
3	E2	T2	V2
..			
N			

Progress

is how much we have done wrt plan

Remember the **session plan**, it was really:

	EUT	TEST	ENV
1	E1	T1	V1
2	E2	T1	V1
3	E2	T2	V2
..			
N			

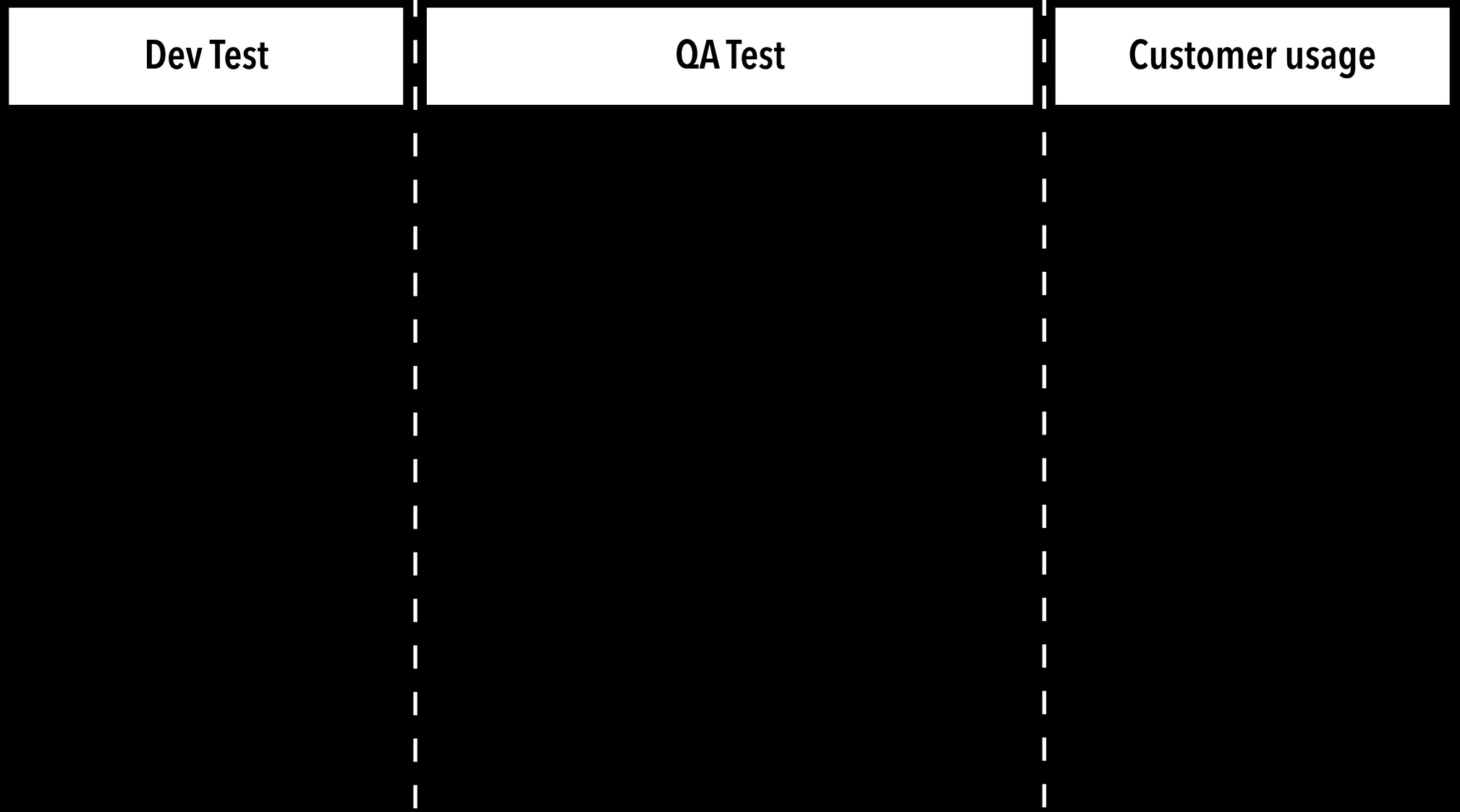
Progress

is how much we have done wrt plan

is %age of ‘TC’ executed in a test

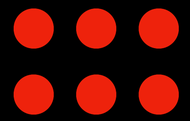
Defect escapes

Escapes



Escapes

Dev Test	QA Test	Customer usage



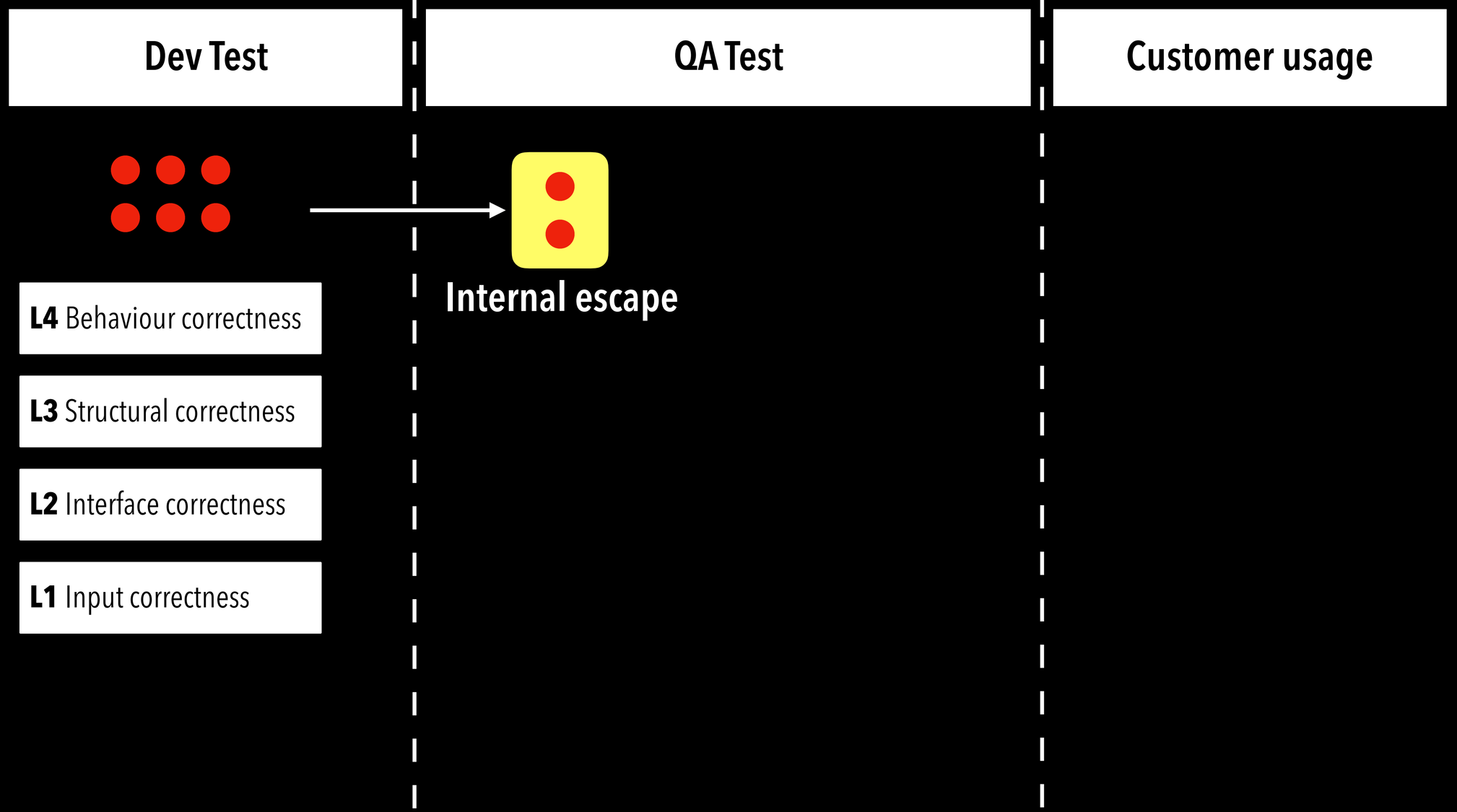
L4 Behaviour correctness

L3 Structural correctness

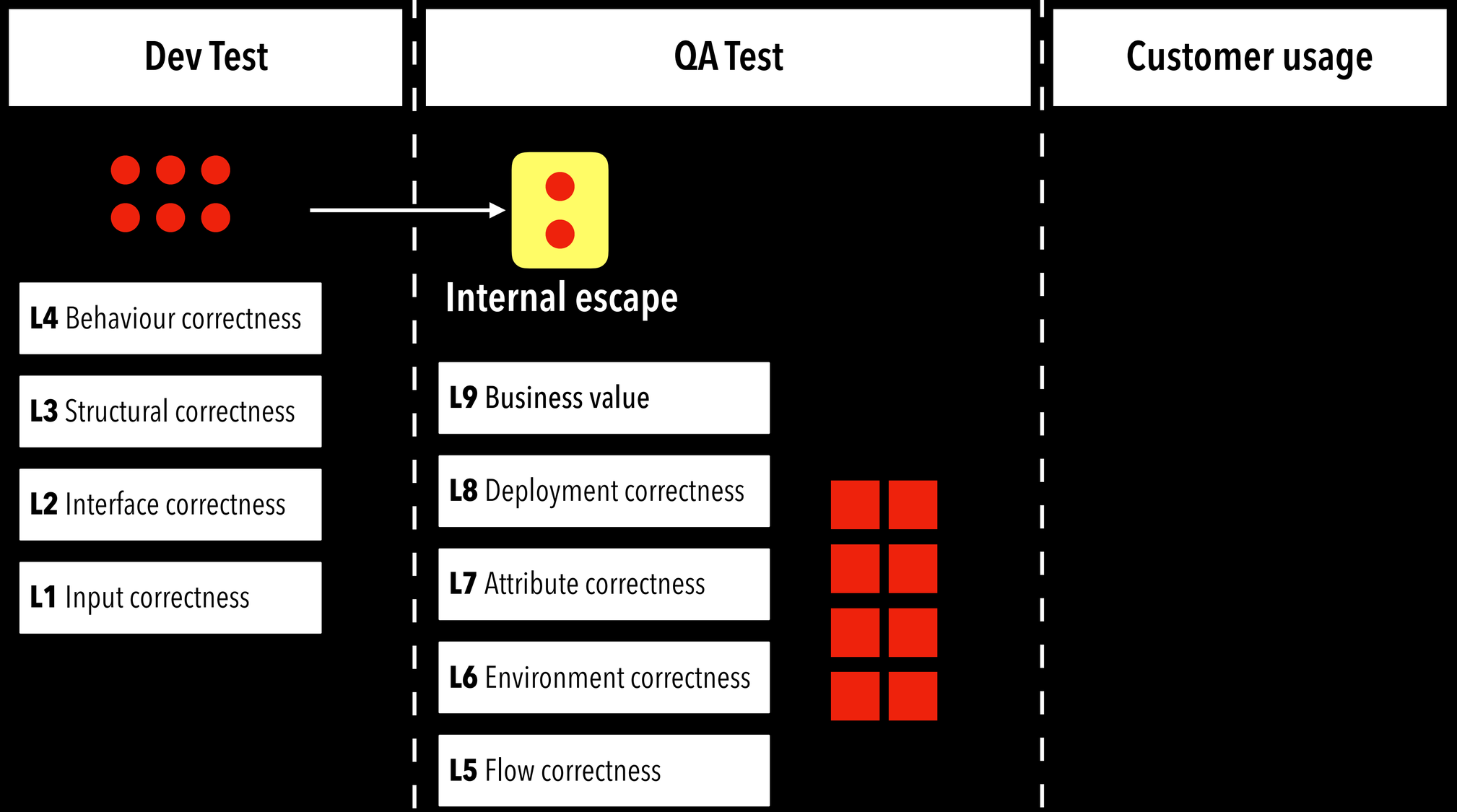
L2 Interface correctness

L1 Input correctness

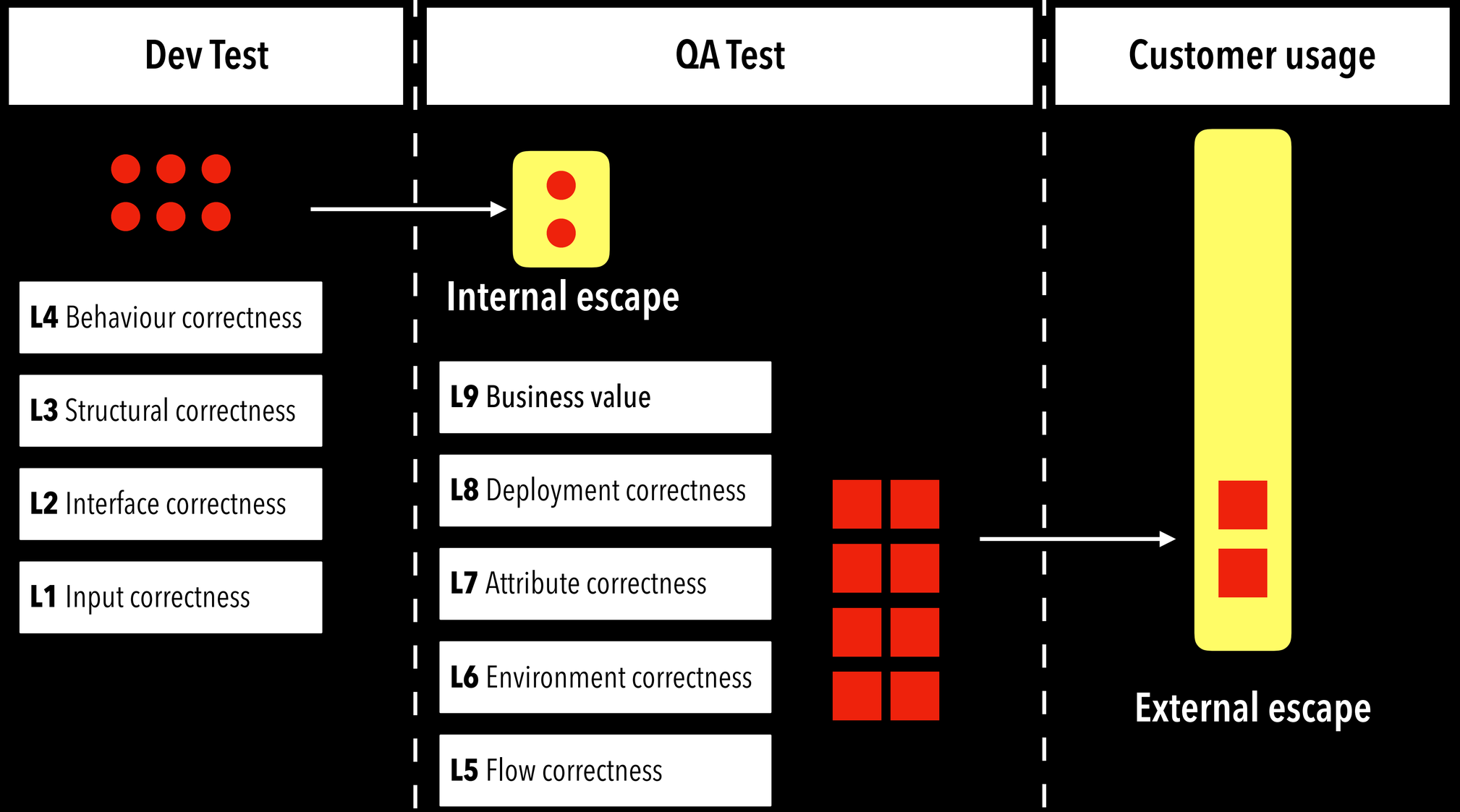
Escapes


Dev Test
QA Test
Customer usage
Internal escape
L4 Behaviour correctness
L3 Structural correctness
L2 Interface correctness
L1 Input correctness

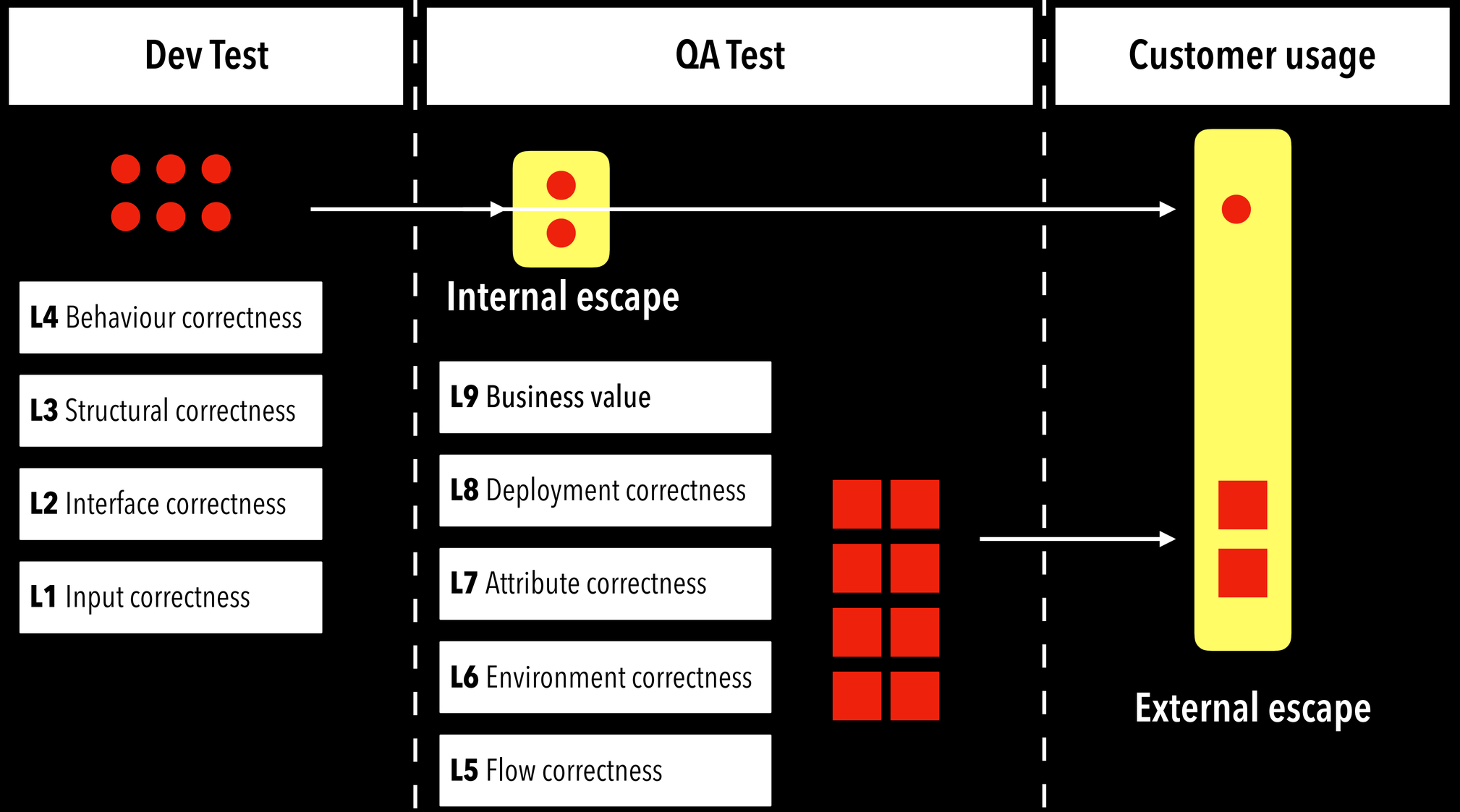
Escapes


Dev Test
QA Test
Customer usage
Internal escape
L4 Behaviour correctness
L3 Structural correctness
L2 Interface correctness
L1 Input correctness
L9 Business value
L8 Deployment correctness
L7 Attribute correctness
L6 Environment correctness
L5 Flow correctness

Escapes


Dev Test
QA Test
Customer usage
L4 Behaviour correctness
L3 Structural correctness
L2 Interface correctness
L1 Input correctness
Internal escape
L9 Business value
L8 Deployment correctness
L7 Attribute correctness
L6 Environment correctness
L5 Flow correctness
External escape

Escapes


Dev Test
QA Test
Customer usage
L4 Behaviour correctness
L3 Structural correctness
L2 Interface correctness
L1 Input correctness
Internal escape
L9 Business value
L8 Deployment correctness
L7 Attribute correctness
L6 Environment correctness
L5 Flow correctness
External escape

Sprint X

	#Issues DevTest	#Issues QATest	Internal Escape (%)
L9			
L8			
L7			
L6			
L5			
L4	a	b	b/(a+b)
L3			
L2			
L1			

Sprint X

	#Issues DevTest	#Issues QATest	Internal Escape (%)
L9			
L8			
L7			
L6			
L5			
L4	a	b	b/(a+b)
L3			
L2			
L1			
		Cumulative IE	Sx-IE

Sprint X

	#Issues DevTest	#Issues QATest	Internal Escape (%)
L9			
L8			
L7			
L6			
L5			
L4	a	b	b/(a+b)
L3			
L2			
L1			
		Cumulative IE	Sx-IE

Trend

Sprint #	Internal Escape (%)
S1	S1-IE
S2	S2-IE
S3	S3-IE
..	
Sn	Sn-IE

Check vs. Test effort

How much checking? How much testing?

Given that in a session we would be doing both

how do they look like?

How much checking? How much testing?

Given that in a session we would be doing both

how do they look like?

How much regression effort?

How much of scripted test effort?

How much unscripted test effort?

How much checking? How much testing?

Given that in a session we would be doing both

how do they look like?

How much regression effort?

How much of scripted test effort?

How much unscripted test effort?

How does this help?

Keeps us aware, so that we don’t get into doing ‘rote’.

Analysis for clear actions

First understand issues before figuring out how to address them.

Understand what issue types and what EUT .

Then go onto why and what-to-do.

Do not jump into RCA immediately.

Understand WHAT of escapes, then WHY & then WHAT-TO-DO

Understand WHAT of escapes, then WHY & then WHAT-TO-DO

Analyse issues missed for 'what'

1 which level these belong to

2 which test do these fall into

3 what EUT do these belong to (TF | UR | BF)

Understand WHAT of escapes, then WHY & then WHAT-TO-DO

Analyse issues missed for 'what'

1 which level these belong to

2 which test do these fall into

3 what EUT do these belong to (TF | UR | BF)

Then analyse for 'why'

1 did not have TS/TC

2 did not do (time issue?)

3 did not do correctly (incorrect understanding?)

4 did not regress well

Understand WHAT of escapes, then WHY & then WHAT-TO-DO

Analyse issues missed for 'what'

1 which level these belong to

2 which test do these fall into

3 what EUT do these belong to (TF | UR | BF)

Then analyse for 'why'

1 did not have TS/TC

2 did not do (time issue?)

3 did not do correctly (incorrect understanding?)

4 did not regress well

Now figure out 'what-to-do'

1 tighten quality gates

2 improve test quality

3 enhance efficiency

A quick summary of IST

RECONNAISANCE | **EXPLORATION** | **RECOUP**

Get a big picture of system and create maps to explore

who are the end users (persona)
what are the needs i.e.system elements (entities)
what are the expectations (attributes)
where will it be used (environment)

Landscaping

who uses what	Persona Map
what is expected of	Scope Map
what affects what	Interaction Map
where is it used	Environment Map

Mapping

RECONNAISANCE

EXPLORATION

RECOUP

Dive deep to understand entities and then evaluate them

understand what entity does/should do
what are conditions governing behaviour?
what are the acceptance criteria?
what may be potential issues probable?

Deep dive

come up with scenarios to try out
come up with smart checklist to check/test
create suites, review, revise
note down issues, suggestions, observations

Design & validate

RECONNAISANCE

EXPLORATION

RECOUP

Analyse , Reflect & Learn

Test adequacy

Execution progress

Product quality

Practice effectiveness (Check vs Test & Escapes)

Analysis for improvement

IST Adoption Tips

IST Adoption Tips

Be sensitive to Check vs Test

IST Adoption Tips

Be sensitive to Check vs Test

View testing as questioning

IST Adoption Tips

Be sensitive to Check vs Test

View testing as questioning

Be lightweight in writing, be non-linear

IST Adoption Tips

Be sensitive to Check vs Test

View testing as questioning

Be lightweight in writing, be non-linear

Think of issues level-wise

IST Adoption Tips

Be sensitive to Check vs Test

View testing as questioning

Be lightweight in writing, be non-linear

Think of issues level-wise

Behaviour based design, augment with experience

IST Adoption Tips

Be sensitive to Check vs Test

View testing as questioning

Be lightweight in writing, be non-linear

Think of issues level-wise

Behaviour based design, augment with experience

Scripted & Unscripted in tandem

IST Adoption Tips

Be sensitive to Check vs Test

View testing as questioning

Be lightweight in writing, be non-linear

Think of issues level-wise

Behaviour based design, augment with experience

Scripted & Unscripted in tandem

See different POV

Discussion

Your views, questions, comments …

SmartQA

Thank you.

© 2000-21, STAG Software Pvt Ltd

www.stagsoftware.com