Masterclass for DEV
Session #1

stag

www.stagsoftware.com

TOPICS

Testing vs. Checking

The BIG picture

Quality levels, Test types, Entities

Clarity of ‘Unit’ and therefore unit testing

Types of issues to focus in UT

Means to uncover issues

A look at defect escapes

Testing vs. Checking

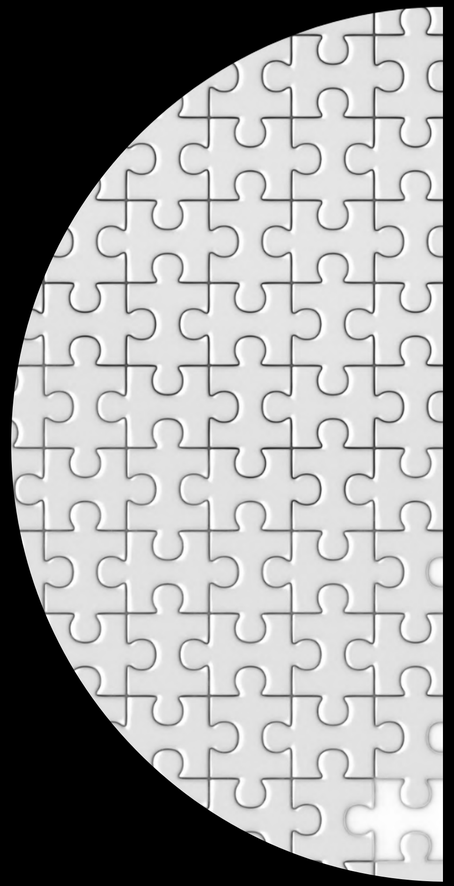
Great quality is about

ensuring compliance and discovering potential gaps

Great quality is about

ensuring compliance and discovering potential gaps

CHECKING

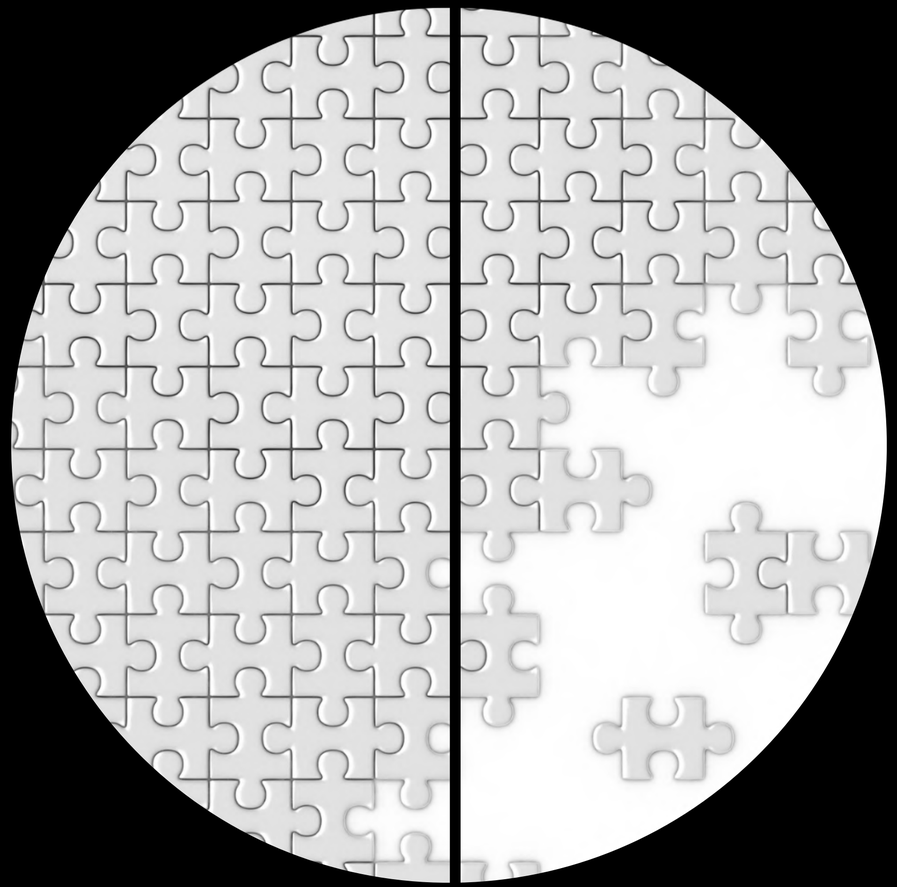


Great quality is about

ensuring compliance and discovering potential gaps

CHECKING

TESTING

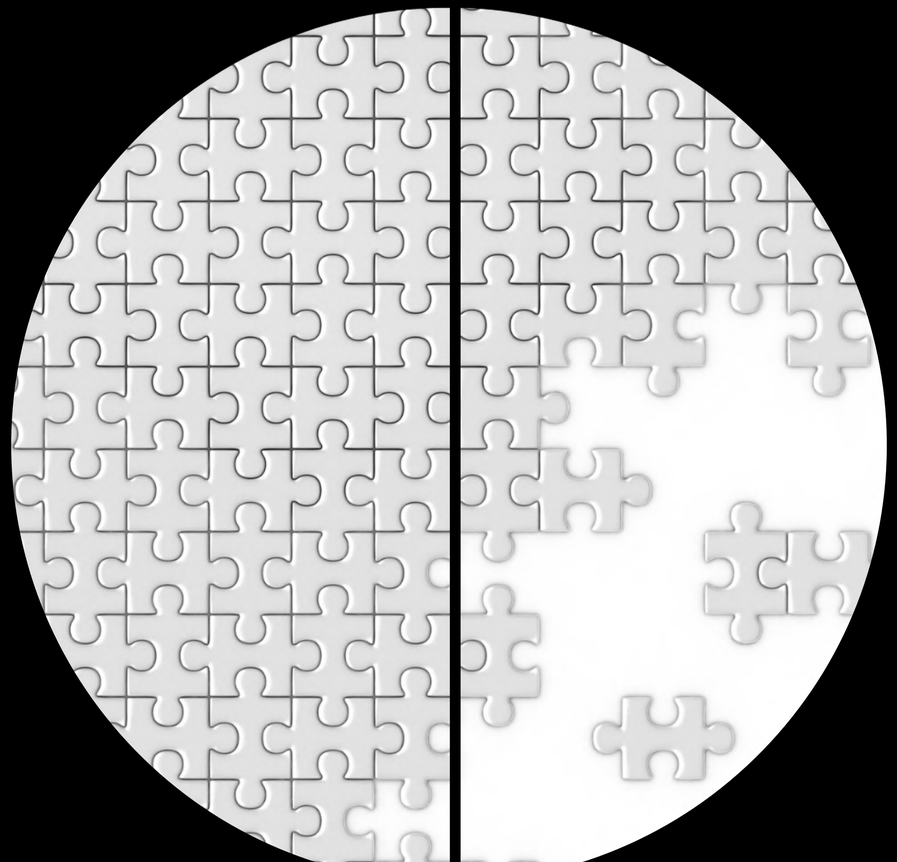


Great quality is about
ensuring compliance and discovering potential gaps

CHECKING

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

based on spec
wellness
AUTOMATED



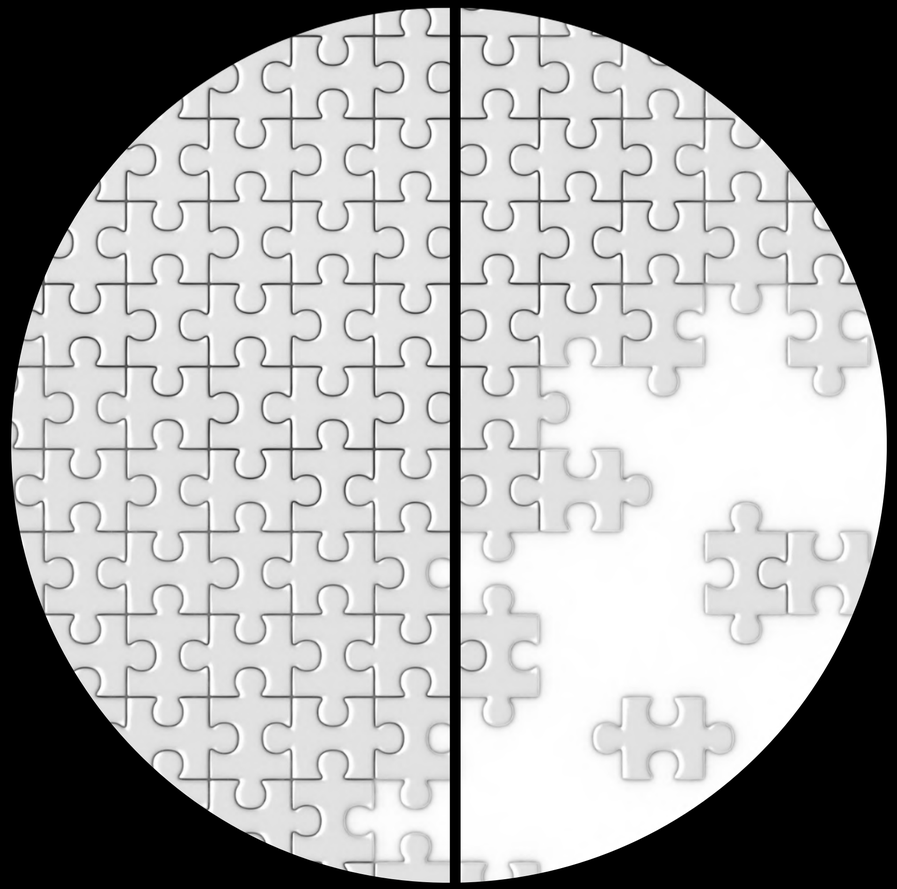
TESTING

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

The BIG picture

the BIG picture

Testing

Checking

Objective

the BIG picture

Testing	Checking
DevTest (UT,IT)	QA test (System Test)

Objective

Role

the BIG picture

Testing	Checking
DevTest (UT,IT)	QA test (System Test)
in the Small	in the Large

Objective

Role

Entity

the BIG picture

Testing	Checking
DevTest (UT,IT)	QA test (System Test)
in the Small	in the Large
Functional test	Non-functional test

Objective

Role

Entity

What to find

the BIG picture

Testing	Checking
DevTest (UT,IT)	QA test (System Test)
in the Small	in the Large
Functional test	Non-functional test
by Human	Automated

- Objective
- Role
- Entity
- What to find
- How to find

the BIG picture

Testing	Checking
DevTest (UT,IT)	QA test (System Test)
in the Small	in the Large
Functional test	Non-functional test
by Human	Automated
System POV	User POV

Objective

Role

Entity

What to find

How to find

POV

the BIG picture

Testing	Checking
DevTest (UT,IT)	QA test (System Test)
in the Small	in the Large
Functional test	Non-functional test
by Human	Automated
System POV	User POV
New	Modified

Objective

Role

Entity

What to find

How to find

POV

State

the BIG picture

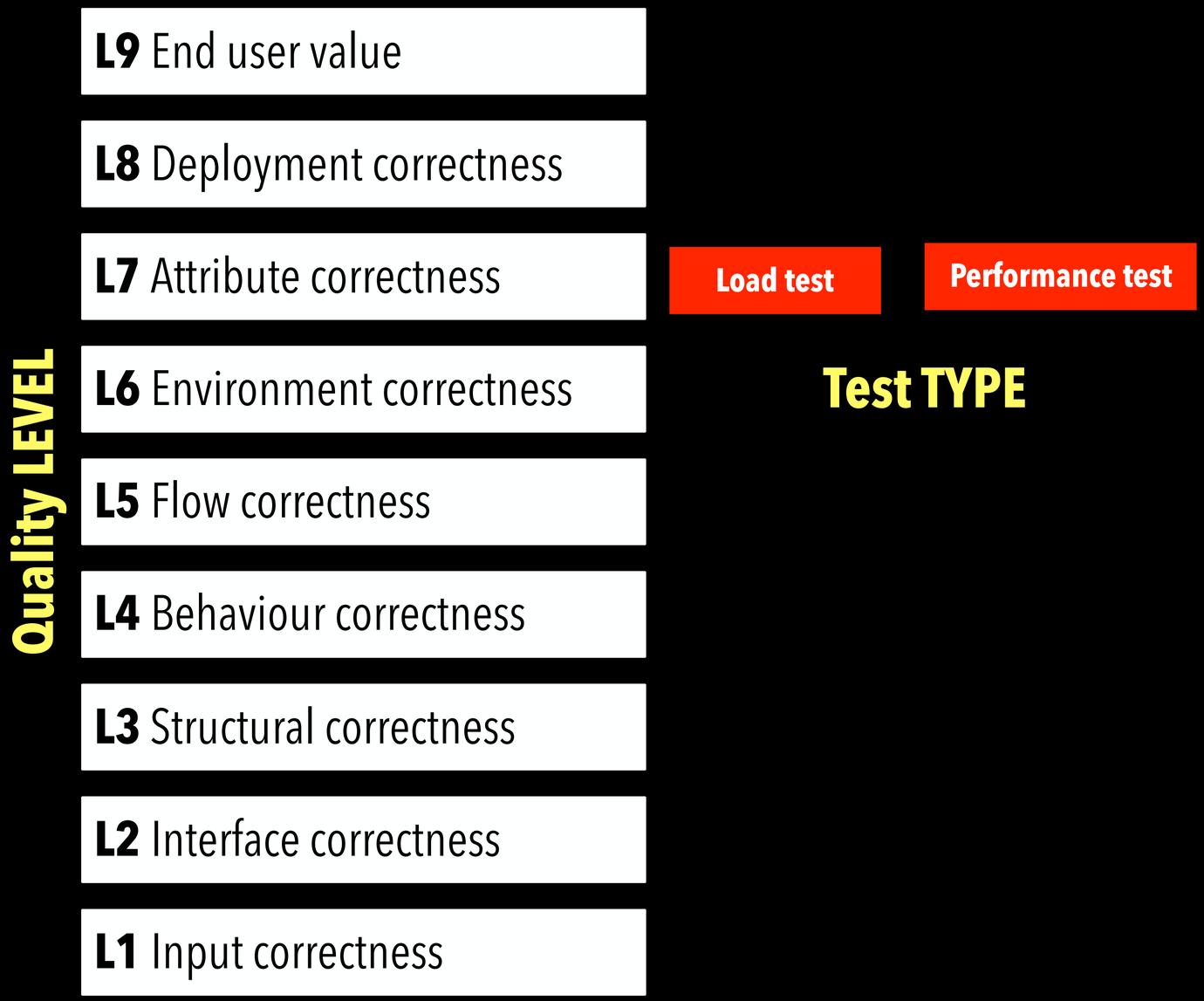
Testing	Checking	Objective
DevTest (UT,IT)	QA test (System Test)	Role
in the Small	in the Large	Entity
Functional test	Non-functional test	What to find
by Human	Automated	How to find
System POV	User POV	POV
New	Modified	State
Detect	Prevent (Shift-left)	Value

Quality Levels,
Test Types &
Test TECHNIQUES

Quality LEVEL

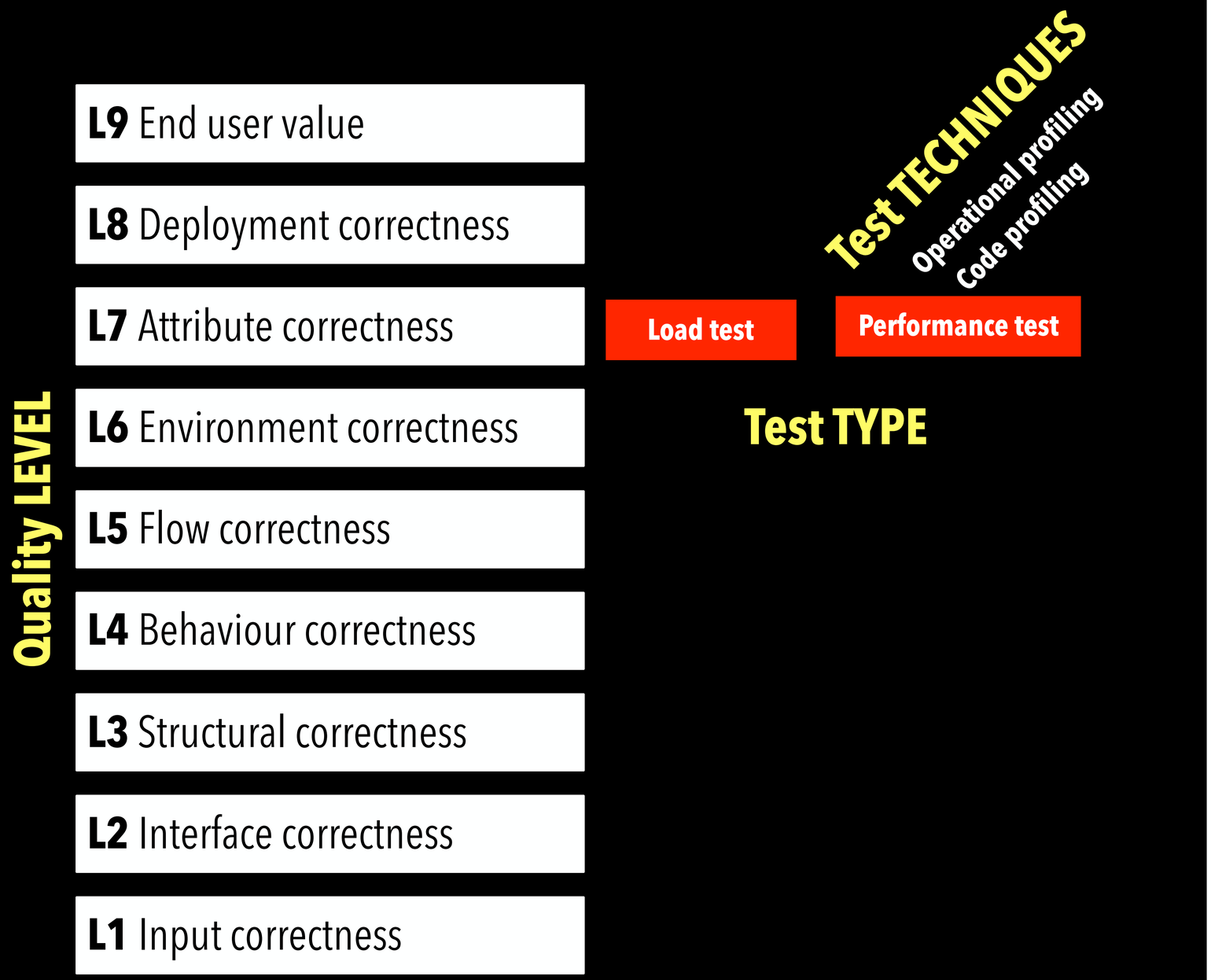
- **L9** End user value
- **L8** Deployment correctness
- **L7** Attribute correctness
- **L6** Environment correctness
- **L5** Flow correctness
- **L4** Behaviour correctness
- **L3** Structural correctness
- **L2** Interface correctness
- **L1** Input correctness

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



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

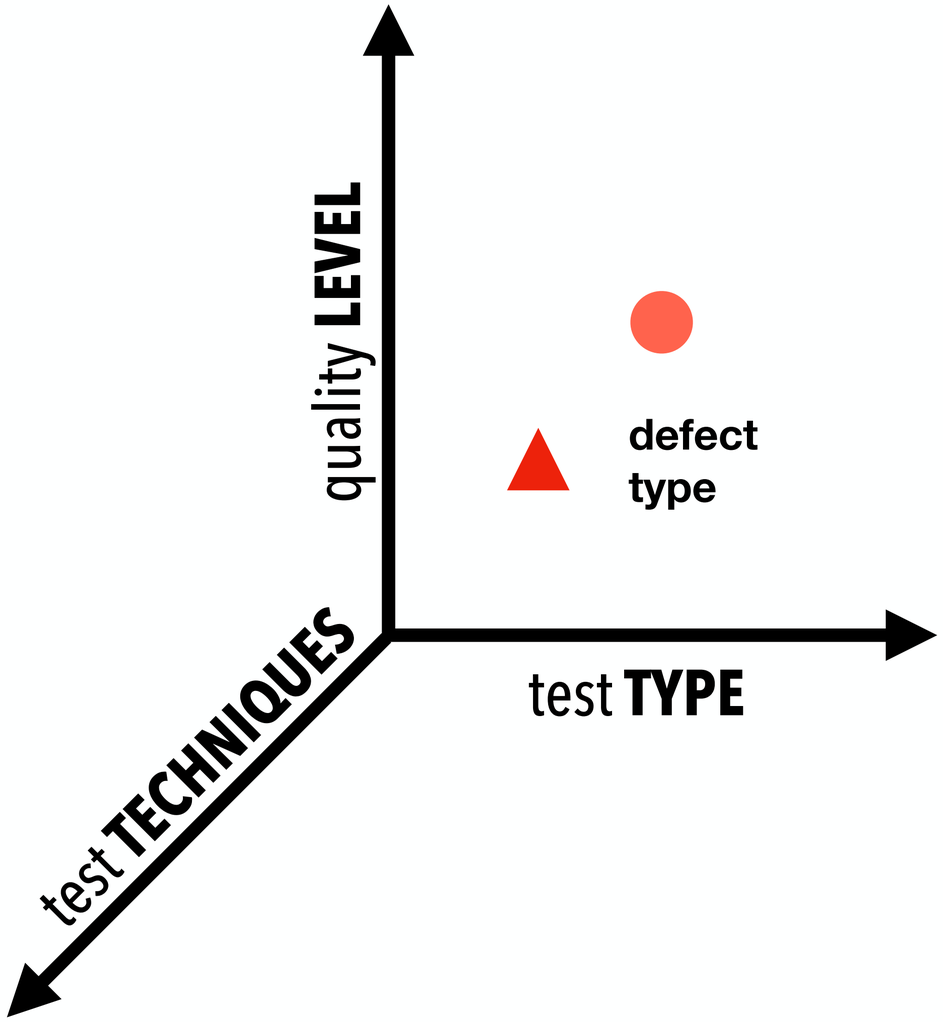
To attain a level, defects that affect this level must not be present
=> we must conduct **specific** tests

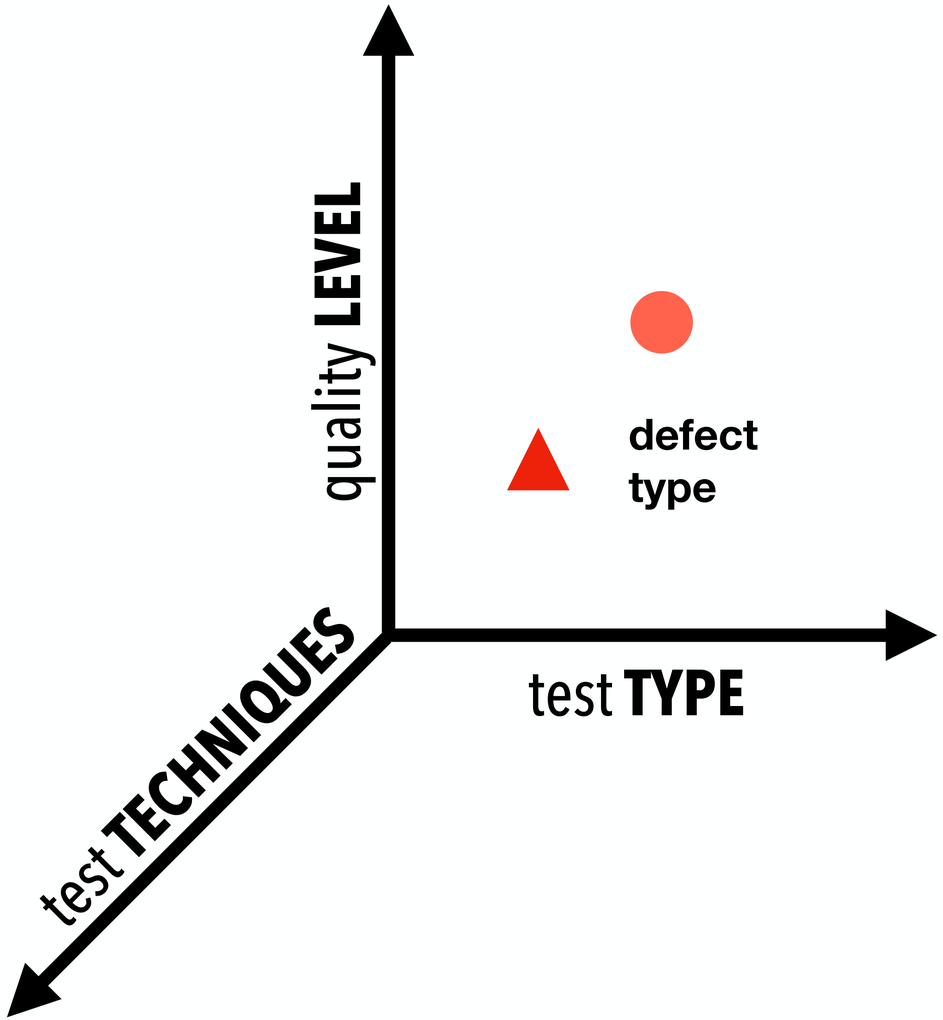


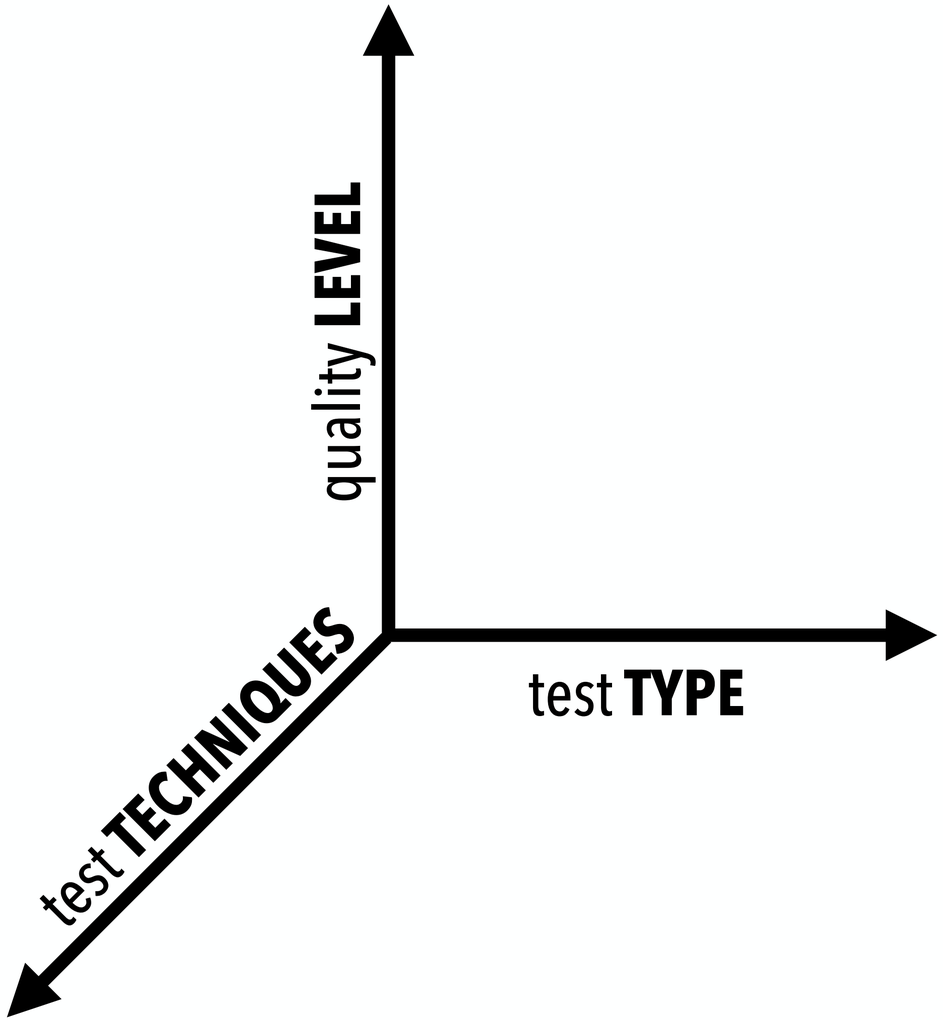
View user's expectation of quality as a series of **levels** to attain.

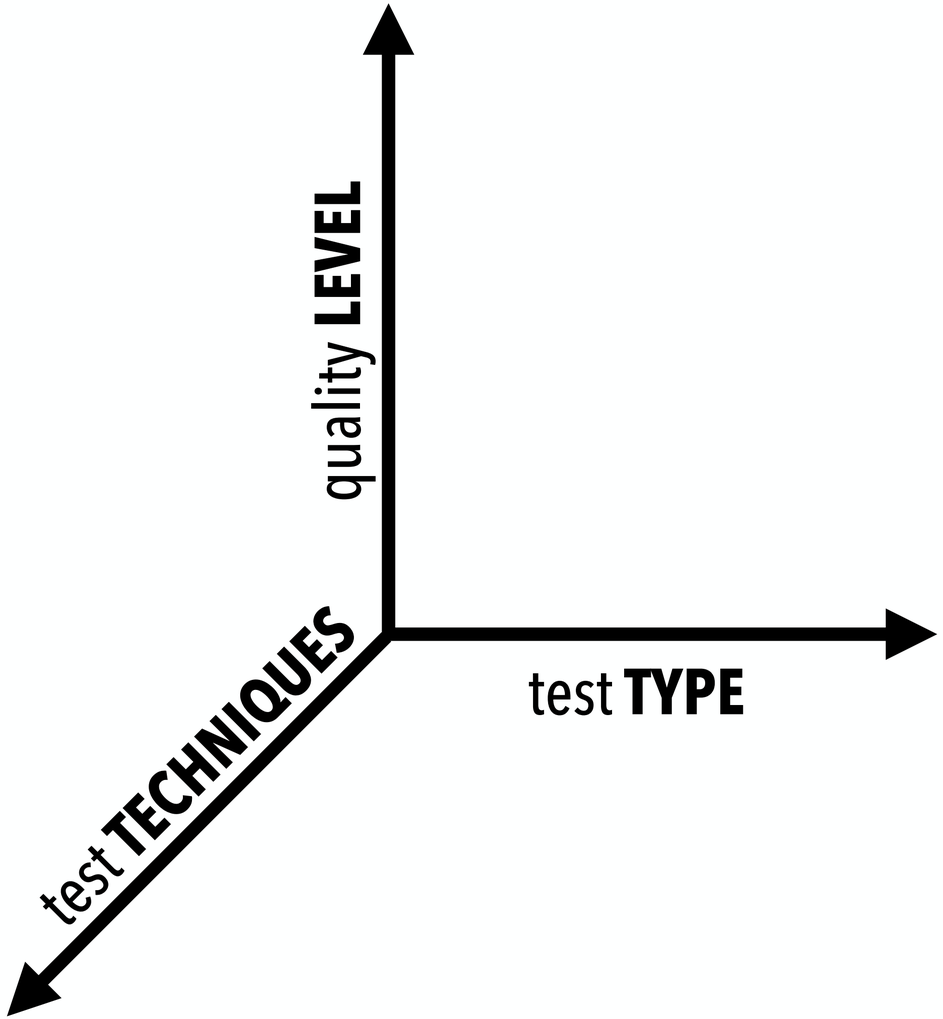
To attain a level, defects that affect this level must not be present
=> we must conduct **specific** tests

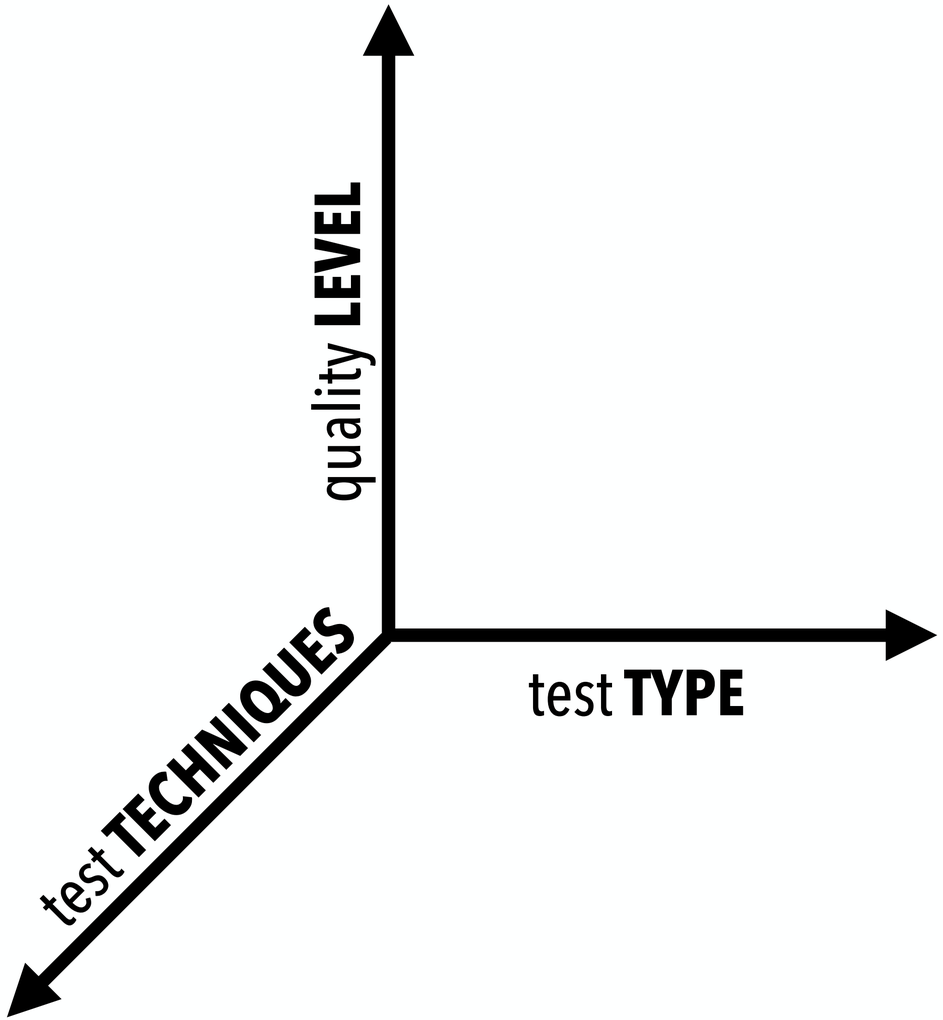
To do a test, we need to come up with test scenarios/cases
=> we need test **TECHNIQUES**

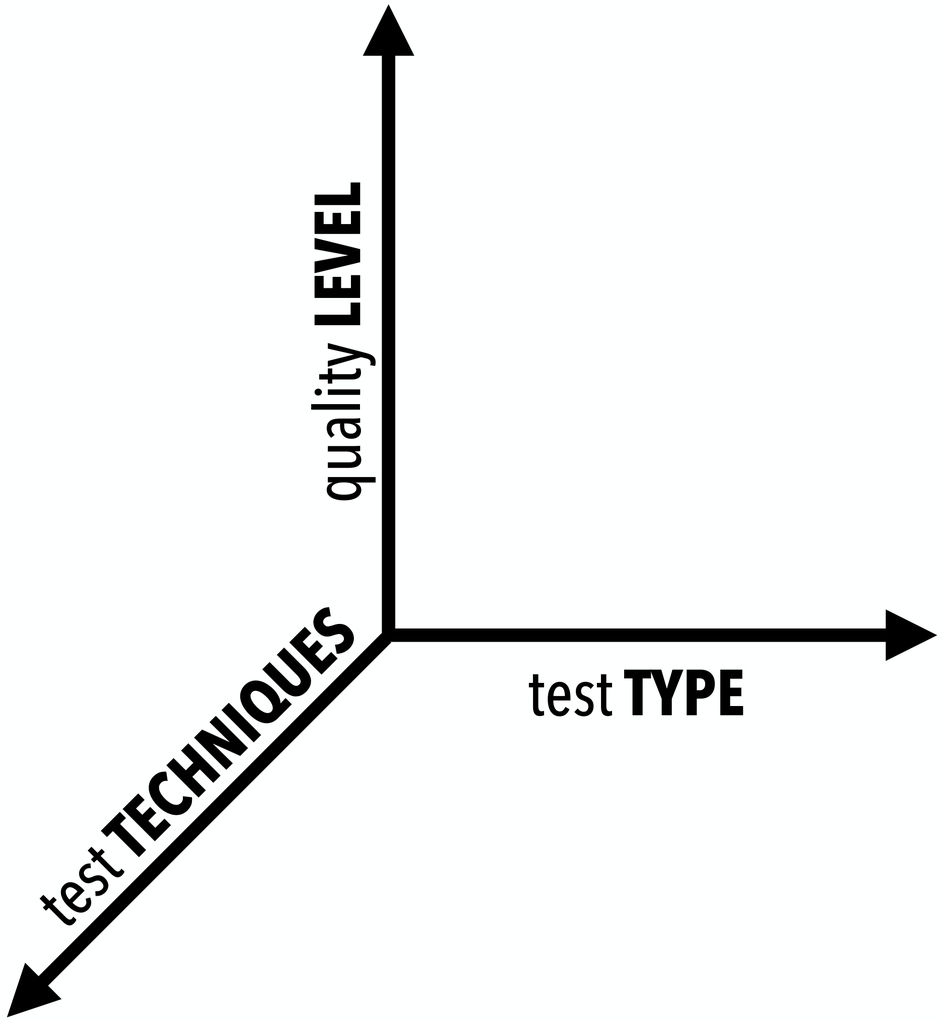

quality LEVEL
defect
type
test TYPE
test TECHNIQUES


quality LEVEL
defect
type
test TYPE
test TECHNIQUES


quality **LEVEL**
test **TYPE**
test **TECHNIQUES**


quality **LEVEL**
test **TYPE**
test **TECHNIQUES**


quality **LEVEL**
test **TYPE**
test **TECHNIQUES**


quality **LEVEL**
test **TYPE**
test **TECHNIQUES**

Clarity of ‘Unit’ and therefore unit testing

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

WHAT TO TEST? Entity Under Test (EUT)

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

Dev Test

- **L4** Behaviour correctness
- **L3** Structural correctness
- **L2** Interface correctness
- **L1** Input correctness

WHAT TO TEST? Entity Under Test (EUT)

technical FEATURE	Basic offering from system
structural COMPONENT	Basic building block

TEST FOR WHAT?

+ Shift Left	**L9** End user value
	L8 Deployment correctness
	L7 Attribute correctness
	L6 Environment correctness
	L5 Flow correctness
Dev Test	**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?

QA Test (System)	**L9** End user value
	L8 Deployment correctness
	L7 Attribute correctness
	L6 Environment correctness
	L5 Flow correctness
Dev Test	**L4** Behaviour correctness
	L3 Structural correctness
	L2 Interface correctness
	L1 Input correctness

Types of issues to focus in UT

satisfy QUALITY level

by TEST for ISSUES

L1 Input correctness

Input validation test

Off limits, duplicates, data type, non-unique

dependency on other data

dependancy on past transaction, boundaries

satisfy QUALITY level

by TEST for ISSUES

L2 Interface correctness

L1 Input correctness

Interface correctness test (Data/UI)
UI interface, Data, Message, File format

Input validation test
Off limits, duplicates, data type, non-unique
dependency on other data
dependancy on past transaction, boundaries

satisfy QUALITY level

by TEST for ISSUES

L3 Structural correctness

L2 Interface correctness

L1 Input correctness

Structural correctness test
System resources (availability, consumption)
Exceptions/Errors (Timeouts, error handling..)
Synchronisation (Timing, Concurrency)
Side effects, Code coverage,

Interface correctness test (Data/UI)
UI interface, Data, Message, File format

Input validation test
Off limits, duplicates, data type, non-unique
dependency on other data
dependancy on past transaction, boundaries

satisfy QUALITY level

by TEST for ISSUES

L4 Behaviour correctness

L3 Structural correctness

L2 Interface correctness

L1 Input correctness

Functional behaviour test
Behaviour correctness

Structural correctness test
System resources (availability, consumption)
Exceptions/Errors (Timeouts, error handling..)
Synchronisation (Timing, Concurrency)
Side effects, Code coverage,

Interface correctness test (Data/UI)
UI interface, Data, Message, File format

Input validation test
Off limits, duplicates, data type, non-unique
dependency on other data
dependancy on past transaction, boundaries

satisfy QUALITY level

L6-L8 : Attributes

L4 Behaviour correctness

L3 Structural correctness

L2 Interface correctness

L1 Input correctness

by TEST for ISSUES

Shift Left-ing Attributes

Functional behaviour test
Behaviour correctness

Structural correctness test
System resources (availability, consumption)
Exceptions/Errors (Timeouts, error handling..)
Synchronisation (Timing, Concurrency)
Side effects, Code coverage,

Interface correctness test (Data/UI)
UI interface, Data, Message, File format

Input validation test
Off limits, duplicates, data type, non-unique
dependency on other data
dependancy on past transaction, boundaries

Means to uncover issues

satisfy QUALITY level	by TEST for ISSUES	using METHOD
L6-L8 : Attributes	**Shift Left-ing Attributes**	**Code review**
L4 Behaviour correctness	**Functional behaviour test** Behaviour correctness	
L3 Structural correctness	**Structural correctness test** System resources (availability, consumption) Exceptions/Errors (Timeouts, error handling..) Synchronisation (Timing, Concurrency) Side effects, Code coverage,	
L2 Interface correctness	**Interface correctness test (Data/UI)** UI interface, Data, Message, File format	
L1 Input correctness	**Input validation test** Off limits, duplicates, data type, non-unique dependency on other data dependancy on past transaction, boundaries	

satisfy QUALITY level	by TEST for ISSUES	using METHOD
L6-L8 : Attributes	**Shift Left-ing Attributes**	**Code review**
	Functional behaviour test Behaviour correctness	
L4 Behaviour correctness	**Structural correctness test** System resources (availability, consumption) Exceptions/Errors (Timeouts, error handling..) Synchronisation (Timing, Concurrency) Side effects, Code coverage,	**Static/Dynamic analysis**
L3 Structural correctness	**Interface correctness test (Data/UI)** UI interface, Data, Message, File format	
L2 Interface correctness	**Input validation test** Off limits, duplicates, data type, non-unique dependency on other data dependancy on past transaction, boundaries	
L1 Input correctness		

satisfy QUALITY level	by TEST for ISSUES	using METHOD
L6-L8 : Attributes	**Shift Left-ing Attributes**	**Code review**
L4 Behaviour correctness	**Functional behaviour test** Behaviour correctness	
L3 Structural correctness	**Structural correctness test** System resources (availability, consumption) Exceptions/Errors (Timeouts, error handling..) Synchronisation (Timing, Concurrency) Side effects, Code coverage,	**Static/Dynamic analysis**
L2 Interface correctness	**Interface correctness test (Data/UI)** UI interface, Data, Message, File format	**Unit Test** (Automated/Human)
L1 Input correctness	**Input validation test** Off limits, duplicates, data type, non-unique dependency on other data dependancy on past transaction, boundaries	

satisfy QUALITY level	by TEST for ISSUES	using METHOD
L6-L8 : Attributes	**Shift Left-ing Attributes**	**Code review**
	Functional behaviour test Behaviour correctness	
L4 Behaviour correctness	**Structural correctness test** System resources (availability, consumption) Exceptions/Errors (Timeouts, error handling..) Synchronisation (Timing, Concurrency) Side effects, Code coverage,	**Static/Dynamic analysis**
L3 Structural correctness		**Unit Test** (Automated/Human)
L2 Interface correctness	**Interface correctness test (Data/UI)** UI interface, Data, Message, File format	
L1 Input correctness	**Input validation test** Off limits, duplicates, data type, non-unique dependency on other data dependancy on past transaction, boundaries	**Checklists**

satisfy QUALITY level	by TEST for ISSUES	using METHOD
L6-L8 : Attributes	**Shift Left-ing Attributes**	**Code review**
	Functional behaviour test Behaviour correctness	
L4 Behaviour correctness	**Structural correctness test** System resources (availability, consumption) Exceptions/Errors (Timeouts, error handling..) Synchronisation (Timing, Concurrency) Side effects, Code coverage,	**Static/Dynamic analysis** **Unit Test** (Automated/Human)
L3 Structural correctness	**Interface correctness test (Data/UI)** UI interface, Data, Message, File format	**Checklists**
L2 Interface correctness	**Input validation test** Off limits, duplicates, data type, non-unique dependency on other data dependancy on past transaction, boundaries	**Spec review**
L1 Input correctness		

satisfy QUALITY level	by TEST for ISSUES	using METHOD	via PROCESS
L6-L8 : Attributes	**Shift Left-ing Attributes**	**Code review**	
	Functional behaviour test Behaviour correctness		
L4 Behaviour correctness	**Structural correctness test** System resources (availability, consumption) Exceptions/Errors (Timeouts, error handling..) Synchronisation (Timing, Concurrency) Side effects, Code coverage,	**Static/Dynamic analysis** **Unit Test** (Automated/Human)	**Scripted**
L3 Structural correctness	**Interface correctness test (Data/UI)** UI interface, Data, Message, File format	**Checklists**	
L2 Interface correctness	**Input validation test** Off limits, duplicates, data type, non-unique dependency on other data dependancy on past transaction, boundaries	**Spec review**	
L1 Input correctness			

satisfy QUALITY level	by TEST for ISSUES	using METHOD	via PROCESS
L6-L8 : Attributes	**Shift Left-ing Attributes**	**Code review**	
	Functional behaviour test Behaviour correctness		
L4 Behaviour correctness	**Structural correctness test** System resources (availability, consumption) Exceptions/Errors (Timeouts, error handling..) Synchronisation (Timing, Concurrency) Side effects, Code coverage,	**Static/Dynamic analysis**	**Scripted**
L3 Structural correctness		**Unit Test** (Automated/Human)	**Unscripted**
L2 Interface correctness	**Interface correctness test (Data/UI)** UI interface, Data, Message, File format	**Checklists**	
L1 Input correctness	**Input validation test** Off limits, duplicates, data type, non-unique dependency on other data dependancy on past transaction, boundaries	**Spec review**	

satisfy QUALITY level	by TEST for ISSUES	using METHOD	via PROCESS
L6-L8 : Attributes	**Shift Left-ing Attributes**	**Code review**	
	Functional behaviour test Behaviour correctness		
L4 Behaviour correctness	**Structural correctness test** System resources (availability, consumption) Exceptions/Errors (Timeouts, error handling..) Synchronisation (Timing, Concurrency) Side effects, Code coverage,	**Static/Dynamic analysis**	**Scripted**
L3 Structural correctness		**Unit Test** (Automated/Human)	**Unscripted**
L2 Interface correctness	**Interface correctness test (Data/UI)** UI interface, Data, Message, File format	**Checklists**	**Testability**
L1 Input correctness	**Input validation test** Off limits, duplicates, data type, non-unique dependency on other data dependancy on past transaction, boundaries	**Spec review**	

A look at defect escapes

We analysed defect data by stratifying INTERNAL & EXTERNAL defects into QL and classified them as escapes from DEV & QA stage.

	PBN (Winter 19)		PBF (Fall 20)		RCHT (Summer 19)	
	INT (276)		INT (189)		INT (130)	
DEV	32% (87)		34% (64)		34% (45)	
QA	68%		66%		66%	

We analysed defect data by stratifying INTERNAL & EXTERNAL defects into QL and classified them as escapes from DEV & QA stage.

	PBN (Winter 19)		PBF (Fall 20)		RCHT (Summer 19)	
	INT (276)	EXT (81)	INT (189)	EXT (78)	INT (130)	EXT (151)
DEV	32% (87)	22% (18)	34% (64)	24% (19)	34% (45)	26% (39)
QA	68%	78%	66%	76%	66%	74%

Request you to answer Dev Questionnaire #1 today.

This short questionnaire is
to enable you to REFLECT on your unit/dev testing practice.

https://bit.ly/mcdev1

Thank you.

www.stagsoftware.com