



SmartQA

Masterclass for QA

Session #4

"Test design"



© 2020, STAG Software Pvt Ltd

www.stagsoftware.com

TOPICS

Testing a feature

Test scenarios/cases - RECAP

Behaviour based design

Operational profiling

D1 - "Role of test documentation styles & organisation"


Test documentation

Test case architecture

Test case review

Testing a feature

BOOK INTERNATIONAL FLIGHTS

 **Flights Within India**

 **International Flights**

Round Trip One way Multi City

From:

To:

Include Nearby Airports

Include Nearby Airports

Departure:

Return:

My dates are flexible [+/- 3 days] **New!**

Travelers (up to 9 per booking)

Adults (12+ yrs):

Children(2-11 yrs):

Infants(0-2 yrs):

Economy Business First

Additional Search Options([Hide](#))

Non Stop Flights

Show Refundable tickets only

Preferred Airlines

Option 1.

Option 2.

Option 3.

Search for flights

[Check Flight Status](#)

NEW!

How do we go about testing this?
How many test cases?
Do we agree on a number?
or Is it based on experience?

L1 Input correctness

Boundaries, conditions, limits

BOOK INTERNATIONAL FLIGHTS

Flights Within India International Flights

Round Trip One way Multi City

From: To:

Include Nearby Airports Include Nearby Airports

Departure: DD/MM/YY Return: DD/MM/YY

My dates are flexible [+/- 3 days] **New!**

Travelers (up to 9 per booking)

Adults (12+ yrs): 1 Children(2-11 yrs): 0 Infants(0-2 yrs): 0

Economy Business First

Additional Search Options([Hide](#))

Non Stop Flights
 Show Refundable tickets only

Preferred Airlines

Option 1.

Option 2.

Option 3.

[Check Flight Status](#) **NEW!**

L2 Interface correctness

L1 Input correctness

defaults, tab order, layout..

Boundaries, conditions, limits

BOOK INTERNATIONAL FLIGHTS

Flights Within India International Flights

Round Trip One way Multi City

From: To:

Include Nearby Airports Include Nearby Airports

Departure:

My dates are flexible [+/- 3 days] **New!**

Travelers (up to 9 per booking)

Adults (12+ yrs): Children(2-11 yrs): Infants(0-2 yrs):

Economy Business First

Additional Search Options([Hide](#))

Non Stop Flights
 Show Refundable tickets only

Preferred Airlines

Option 1.

Option 2.

Option 3.

[Check Flight Status](#) **NEW!**

L3 Structural correctness

L2 Interface correctness

L1 Input correctness

error handling(connection)...

defaults, tab order, layout..

Boundaries, conditions, limits

BOOK INTERNATIONAL FLIGHTS

Flights Within India International Flights

Round Trip One way Multi City

From: To:

Include Nearby Airports Include Nearby Airports

Departure:

My dates are flexible [+/- 3 days] **New!**

Travelers (up to 9 per booking)

Adults (12+ yrs): Children(2-11 yrs): Infants(0-2 yrs):

Economy Business First

Additional Search Options([Hide](#))

Non Stop Flights
 Show Refundable tickets only

Preferred Airlines

Option 1.

Option 2.

Option 3.

[Check Flight Status](#) **NEW!**

L4 Behaviour correctness

L3 Structural correctness

L2 Interface correctness

L1 Input correctness

functional correctness

error handling(connection)...

defaults, tab order, layout..

Boundaries, conditions, limits

Trip Type 3

Unique From/To 2

Nearby airports? 2

Ret >= Dep >=Today 2

Flexible dates? 2

#Travelers valid? 2

Class? 3

Non stop? 2

Refundable? 2

Preferred airlines? 2

BOOK INTERNATIONAL FLIGHTS

Flights Within India | **International Flights**

Round Trip One way Multi City

From: To:

Include Nearby Airports Include Nearby Airports

Departure: Return:

My dates are flexible [+/- 3 days] **New!**

Travelers (up to 9 per booking)

Adults (12+ yrs): Children(2-11 yrs): Infants(0-2 yrs):

Economy Business First

Additional Search Options(Hide)

Non Stop Flights Show Refundable tickets only

Preferred Airlines

Option 1.

Option 2.

Option 3.

Search for flights

[Check Flight Status](#) **NEW!**

#TS (Functional) 13 - 1000+

L8 Deployment correctness

L7 Attribute correctness

L6 Environment correctness

L5 Flow correctness

L4 Behaviour correctness

L3 Structural correctness

L2 Interface correctness

L1 Input correctness

Shift Left?
performance, load,
responsiveness..

functional correctness

error handling(connection)...

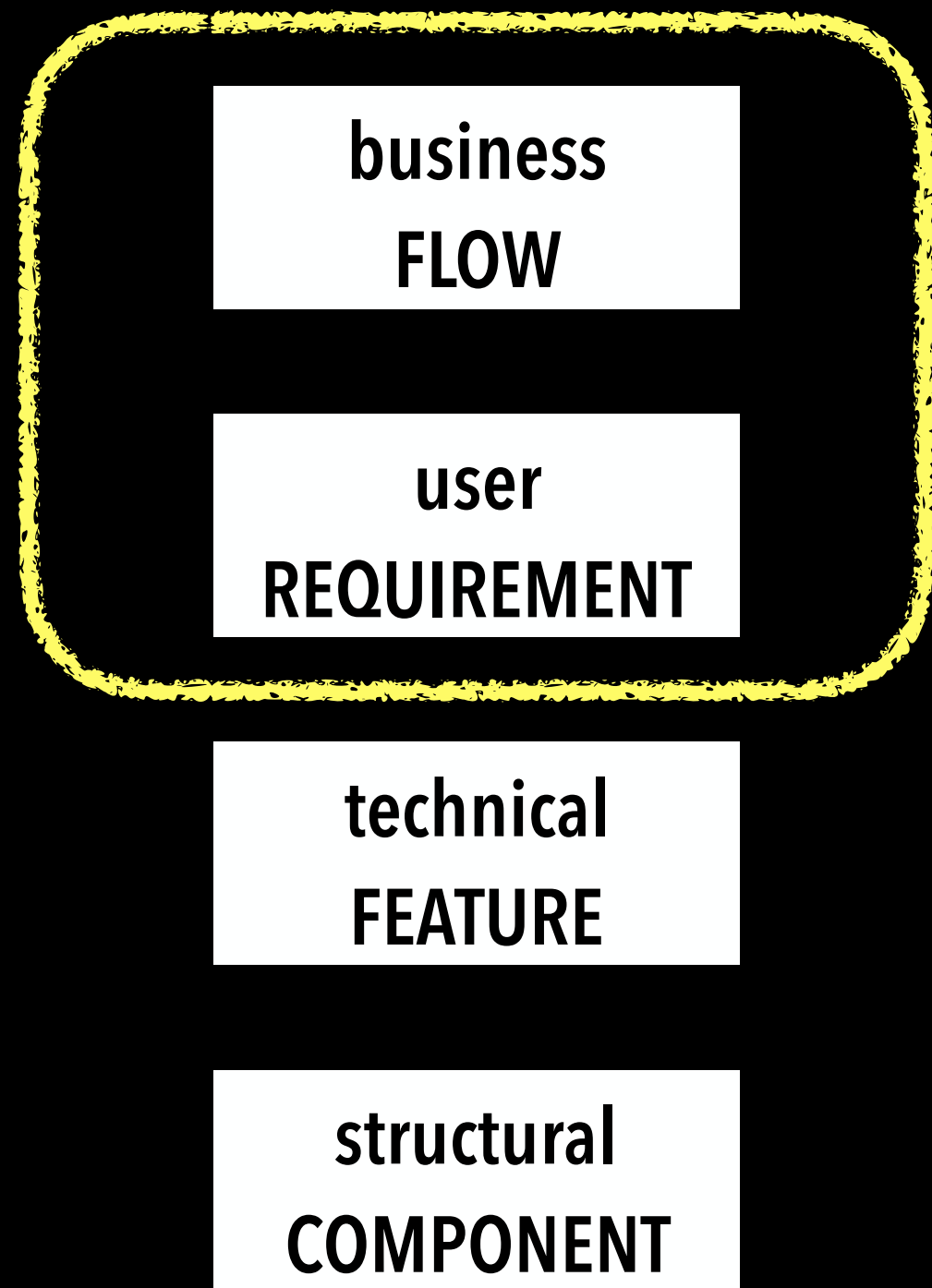
defaults, tab order, layout..

Boundaries, conditions, limits

The screenshot shows a flight booking interface with a yellow header and a blue background. It features two tabs: 'Flights Within India' and 'International Flights'. The 'International Flights' tab is active. Below the tabs, there are radio buttons for 'Round Trip' (selected), 'One way', and 'Multi City'. The 'From:' and 'To:' fields are empty. There are checkboxes for 'Include Nearby Airports'. The 'Departure:' and 'Return:' fields are empty, with a 'My dates are flexible [+/- 3 days] New!' option. The 'Travelers (up to 9 per booking)' section has dropdowns for 'Adults (12+ yrs): 1', 'Children(2-11 yrs): 0', and 'Infants(0-2 yrs): 0'. There are radio buttons for 'Economy' (selected), 'Business', and 'First'. The 'Additional Search Options(Hide)' section has checkboxes for 'Non Stop Flights' and 'Show Refundable tickets only'. The 'Preferred Airlines' section has three input fields labeled 'Option 1.', 'Option 2.', and 'Option 3.'. A green 'Search for flights' button is at the bottom right. A 'Check Flight Status' link with a 'NEW!' badge is at the bottom left.

Test scenarios, cases

Here is a EUT that could be:

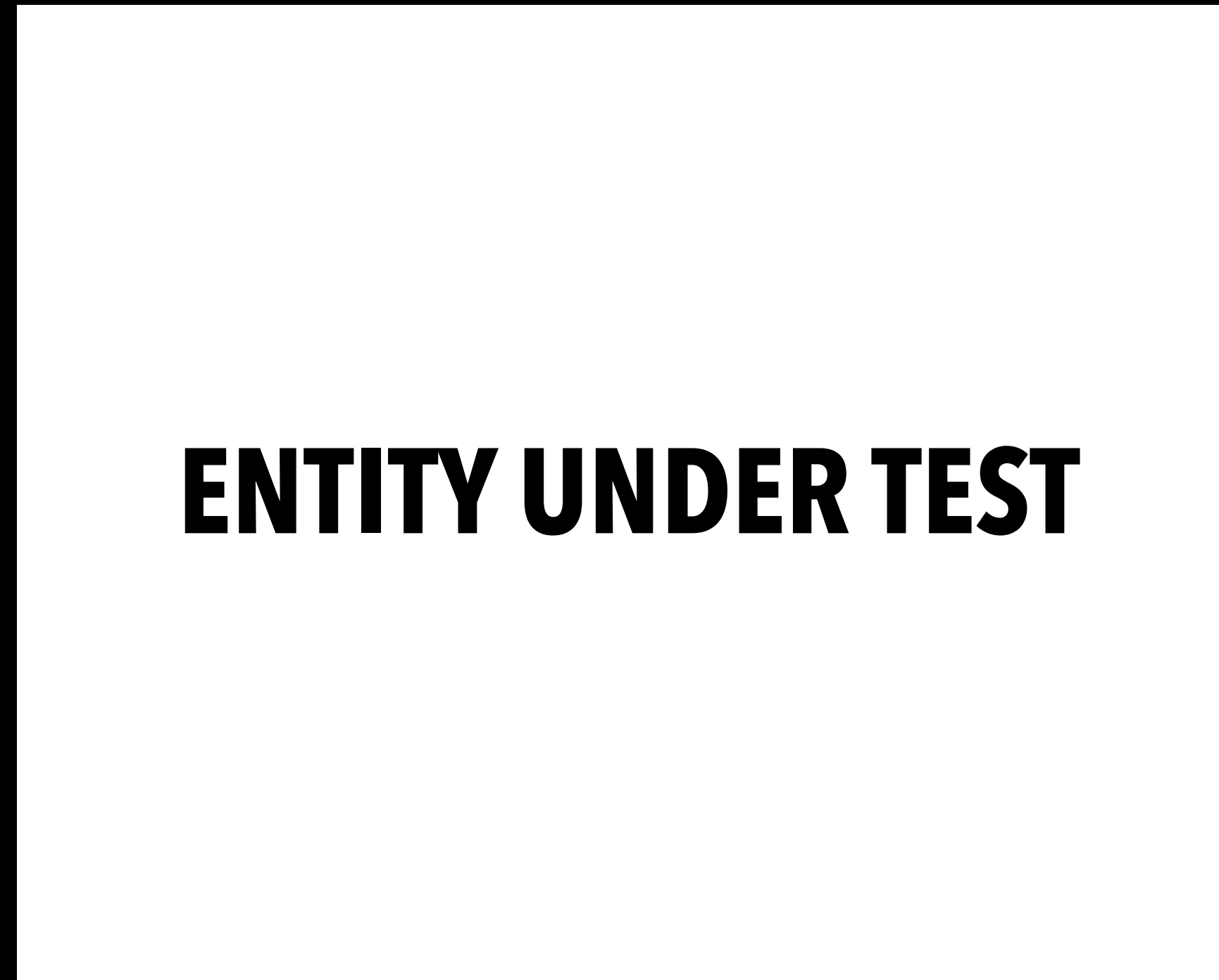


I1 →

I2 →

I3 →

I4 →



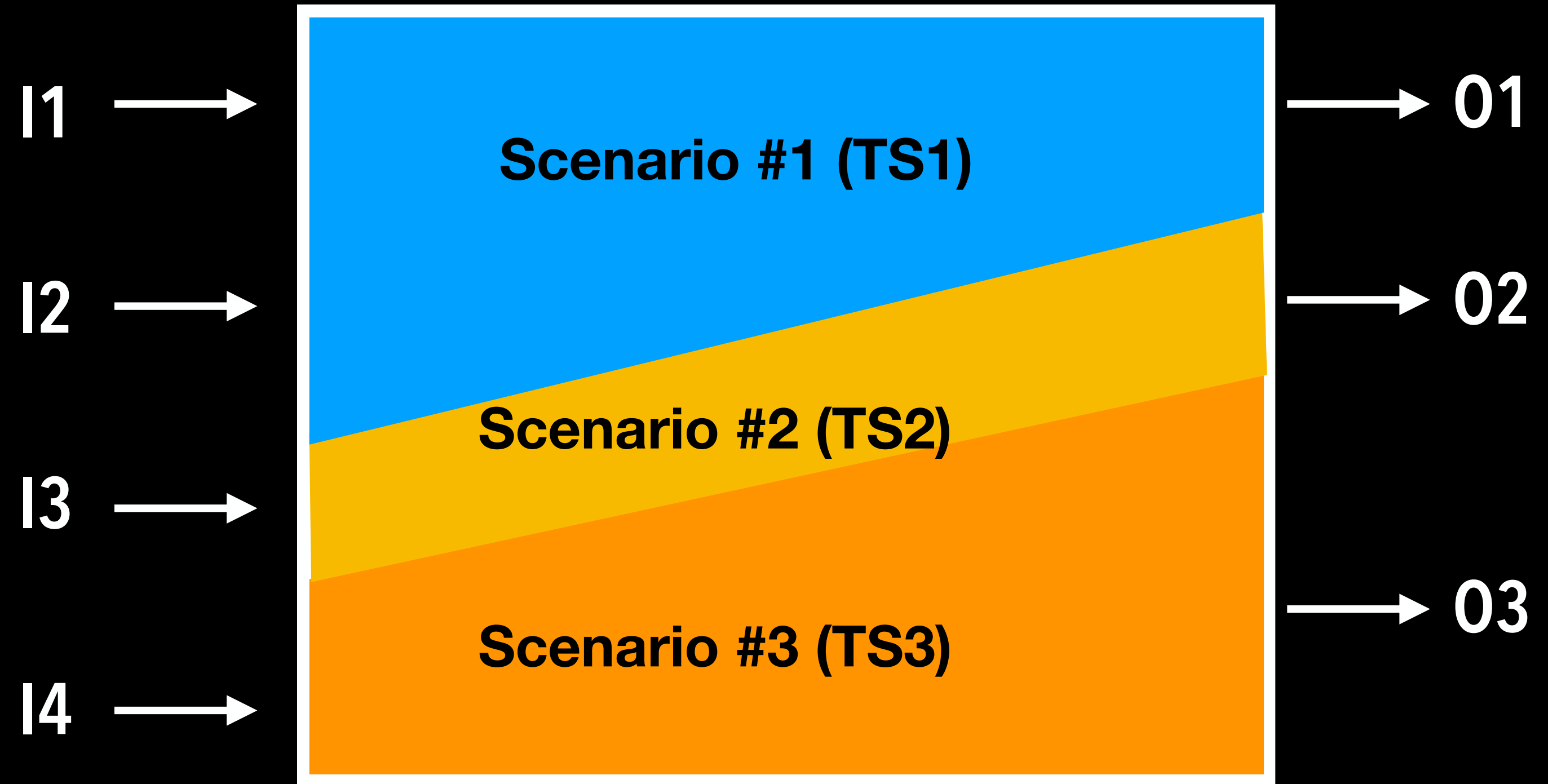
→ O1

→ O2

→ O3

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

ENTITY UNDER TEST



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

ENTITY UNDER TEST

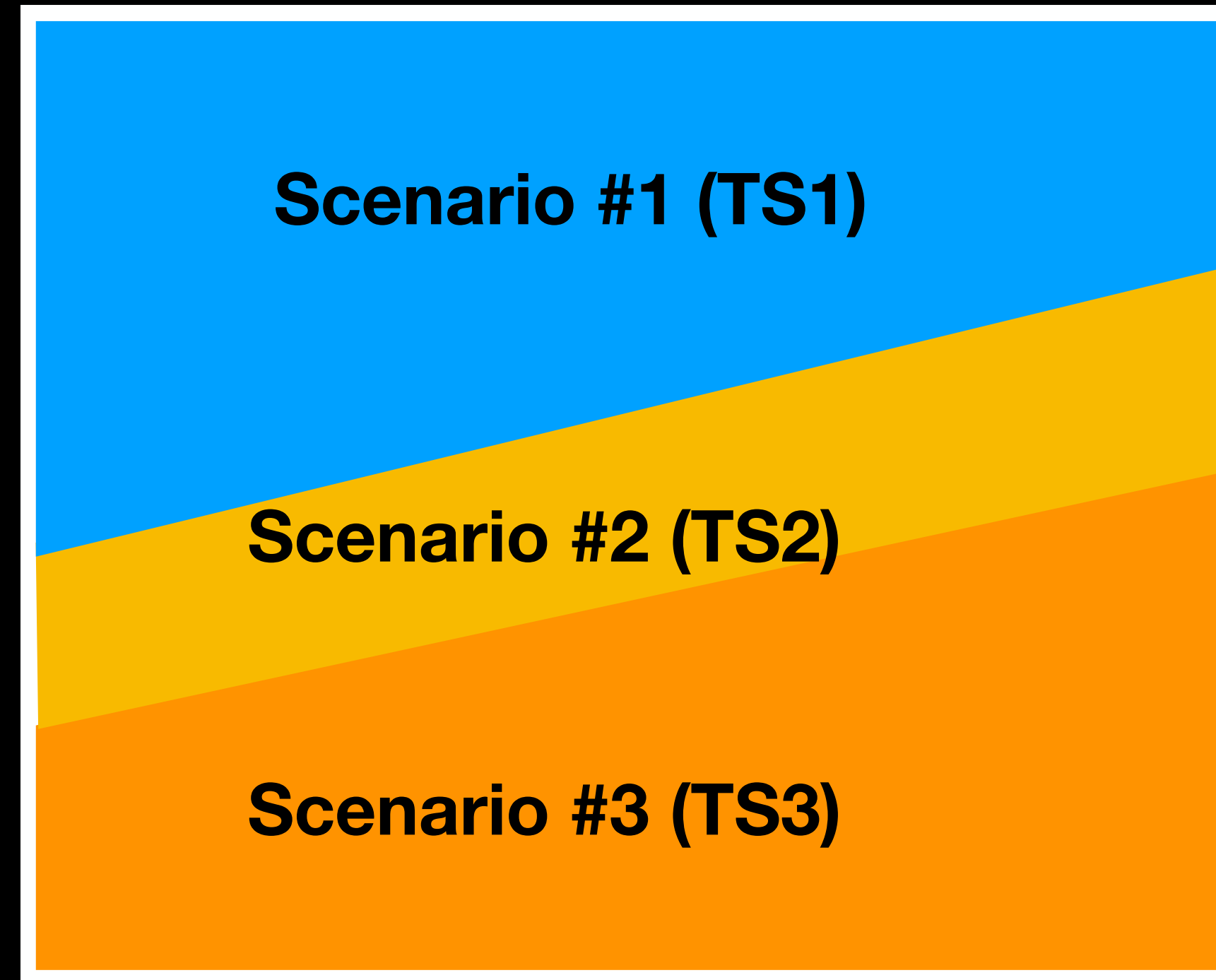
A1	A2	A3
B1	B2	B3
C1	C2	C3
D1	D2	D4

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

TS2

TS1

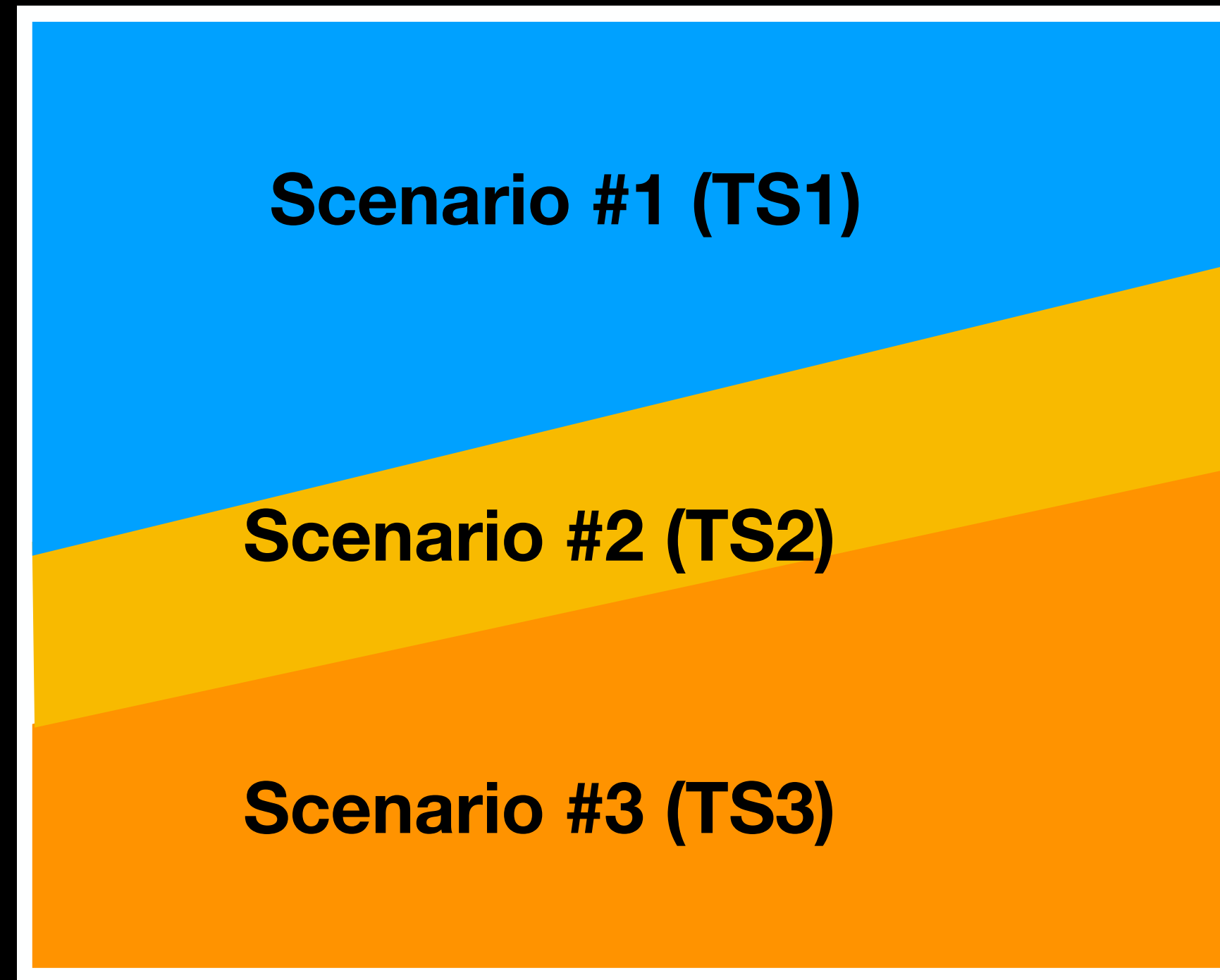
A6	A7
B6	B7
C6	C7
D6	D7

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 CASES for

TS3

TS2

TS1

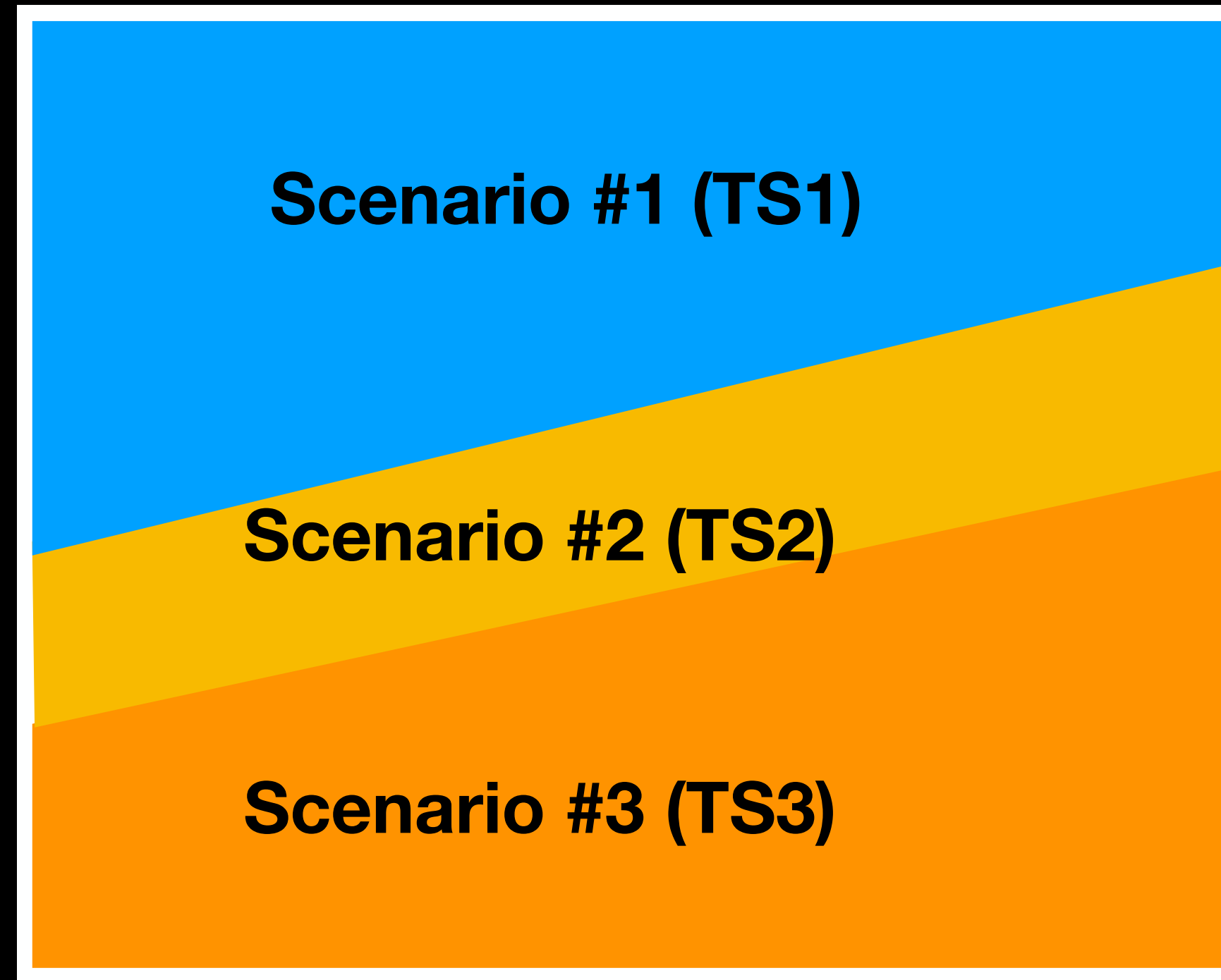
A6	A7
B6	B7
C6	C7
D6	D7

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

Behaviour based design

#1 Understand well

Let us use “Create wallet” feature as an example

CREATE MOBIKWIK WALLET

Enter your Email ID

Enter Password

Enter Mobile Number

Prepaid Postpaid

I agree to the [T&Cs](#) to create my wallet

[Create Wallet](#)

Already have a MobiKwik Wallet? [Login Now](#)

Remember we are designing test cases at Level - L4 for Function behaviour test.

So we need to model the behaviour i.e. state this as PRESCRIPTIVE BEHAVIOUR
To do this , we need to extract the behavioural conditions

So start with the expected outputs, then describe the behaviour, apply 5W1H to identify the conditions to state this as prescriptive behaviour.

CREATE MOBIKWIK WALLET

Enter your Email ID

Enter Password

Enter Mobile Number

Prepaid Postpaid

I agree to the [T&Cs](#) to create my wallet

Create Wallet

Already have a MobiKwik Wallet? [Login Now](#)

Let us first describe the behaviour...

DESCRIPTIVE behaviour

"This feature allows you to create a new wallet that can be pre or post paid. To create the wallet, enter your email ID and choose a password, enter your mobile no, choose pre/post paid and accept the terms & conditions. You will be sent an OTP on your mobile no which, when keyed will activate your wallet."

CREATE MOBIKWIK WALLET

Enter your Email ID

Enter Password

Enter Mobile Number

Prepaid Postpaid

I agree to the [T&Cs](#) to create my wallet

Create Wallet

Already have a MobiKwik Wallet? [Login Now](#)

DESCRIPTIVE behaviour

"This feature allows you to create a new wallet that can be pre or post paid. To create the wallet, enter your email ID and choose a password, enter your mobile no , choose pre/post paid and accept the terms & conditions. You will be sent an OTP on your mobile no which, when keyed will activate your wallet."

Now let us DESCRIBE this as series of actions/steps

1. Enter email ID
2. Enter mobile no
3. Choose wallet type (pre/post) :
4. Accept T&C
5. Create wallet
6. Enter OTP received to activate wallet

CREATE MOBIKWIK WALLET

Enter your Email ID

Enter Password

Enter Mobile Number

Prepaid Postpaid

I agree to the [T&Cs](#) to create my wallet

Create Wallet

Already have a MobiKwik Wallet? [Login Now](#)

DESCRIPTIVE behaviour

"This feature allows you to create a new wallet that can be pre or post paid. To create the wallet, enter your email ID and choose a password, enter your mobile no , choose pre/post paid and accept the terms & conditions. You will be sent an OTP on your mobile no which, when keyed will activate your wallet."

Now let us DESCRIBE this as series of actions/steps & ask 5W1H questions

1. Enter email ID : *What should it be? Already an existing one? Inactive?* **..and identify behaviour conditions**
2. Enter mobile no : *What should it be? Active? Non-existent?*
3. Choose wallet type (pre/post) : *Well, there are just two types that you have to choose from!*
4. Accept T&C : *What-if I do not accept?*
5. Create wallet : *Nothing to ask here!*
6. Enter OTP received to activate wallet : *When i.e. how long after I receive OTP? What-if I enter wrong one?*

Now. let us list the conditions that PRESCRIBE this behaviour.

CONDITIONS	VALUES	
	valid (Y)	invalid (N)
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)
Pwd valid?	Y (not null)	N (null)
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)
Type of account	Prepaid/Postpaid (PR/ PO)	
T&C accepted	Y	N
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)

Now let us DESCRIBE this as series of actions/steps & ask 5W1H questions

1. Enter email ID : What should it be? Already an existing one? Inactive?
2. Enter mobile no : What should it be? Active? Non-existent?
3. Choose wallet type (pre/post) : Well, there are just two types that you have to choose from!
4. Accept T&C : What-if I do not accept?
5. Create wallet : Nothing to ask here!
6. Enter OTP received to activate wallet : When i.e. how long after I receive OTP? What-if I enter wrong one?

Behaviour based design

#2 Model behaviour

Remember the conditions that PRESCRIBE this behaviour...

CREATE MOBIKWIK WALLET

Enter your Email ID

Enter Password

Enter Mobile Number

Prepaid Postpaid

I agree to the [T&Cs](#) to create my wallet

Create Wallet

Already have a MobiKwik Wallet? [Login Now](#)

CONDITIONS	VALUES	
	valid (Y)	invalid (N)
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)
Pwd valid?	Y (not null)	N (null)
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)
Type of account	Prepaid/Postpaid (PR/ PO)	
T&C accepted	Y	N
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)

Now let us DESCRIBE this as series of actions/steps & ask 5W1H questions

1. Enter email ID : What should it be? Already an existing one? Inactive?
2. Enter mobile no : What should it be? Active? Non-existent?
3. Choose wallet type (pre/post) : Well, there are just two types that you have to choose from!
4. Accept T&C : What-if I do not accept?
5. Create wallet : Nothing to ask here!
6. Enter OTP received to activate wallet : When i.e. how long after I receive OTP? What-if I enter wrong one?

Now let us model the behaviour as a decision table



Now let us model the behaviour as a decision table :

CONDITIONS	VALUES	
	valid (Y)	invalid (N)
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)
Pwd valid?	Y(not null)	N (null)
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)
Type of account	Prepaid/Postpaid (PR/ PO)	
T&C accepted	Y	N
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)

CONDITIONS & their values

Now let us model the behaviour as a decision table

CONDITIONS	VALUES	
	valid (Y)	invalid (N)
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)
Pwd valid?	Y (not null)	N (null)
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)
Type of account	Prepaid/Postpaid (PR/ PO)	
T&C accepted	Y	N
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)
OUTPUTS		
Prepaid Wallet Created (PR-WC)		
Postpaid Wallet Created (PO-WC)		
Wallet not created, Email already registered (WNC1)		
Wallet not created, Email does not exist (WNC2)		
Wallet not created, Email not active (WNC3)		
Wallet not created, Password not provided (WNC4)		
Wallet not created, Phone number already registered (WNC5)		
Wallet not created, Phone number not in use(WNC6)		
Wallet not created, Terms & Conditions not accepted (WNC7)		
Wallet not created, OTP not received(WNC8)		
Wallet not created, incorrect OTP entered (WNC9)		
Wallet not created, OTP expired (WNC10)		

CONDITIONS & their values

OUTPUTS /ACTION

Now let us model the behaviour as a decision table

CONDITIONS	VALUES		R1
	valid (Y)	invalid (N)	
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y
Pwd valid?	Y(not null)	N (null)	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR
T&C accepted	Y	N	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C
OUTPUTS			
Prepaid Wallet Created (PR-WC)			*
Postpaid Wallet Created (PO-WC)			
Wallet not created, Email already registered (WNC1)			
Wallet not created, Email does not exist (WNC2)			
Wallet not created, Email not active (WNC3)			
Wallet not created, Password not provided (WNC4)			
Wallet not created, Phone number already registered (WNC5)			
Wallet not created, Phone number not in use(WNC6)			
Wallet not created, Terms & Conditions not accepted (WNC7)			
Wallet not created, OTP not received(WNC8)			
Wallet not created, incorrect OTP entered (WNC9)			
Wallet not created, OTP expired (WNC10)			

Now the RULES

Now let us model the behaviour as a decision table

CONDITIONS	VALUES		R1	R2
	valid (Y)	invalid (N)		
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y
Pwd valid?	Y (not null)	N (null)	Y	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO
T&C accepted	Y	N	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C
OUTPUTS				
Prepaid Wallet Created (PR-WC)			*	
Postpaid Wallet Created (PO-WC)				*
Wallet not created, Email already registered (WNC1)				
Wallet not created, Email does not exist (WNC2)				
Wallet not created, Email not active (WNC3)				
Wallet not created, Password not provided (WNC4)				
Wallet not created, Phone number already registered (WNC5)				
Wallet not created, Phone number not in use(WNC6)				
Wallet not created, Terms & Conditions not accepted (WNC7)				
Wallet not created, OTP not received(WNC8)				
Wallet not created, incorrect OTP entered (WNC9)				
Wallet not created, OTP expired (WNC10)				

...continue with combining the conditions

Now let us model the behaviour as a decision table

CONDITIONS	VALUES		R1	R2	R3
	valid (Y)	invalid (N)			
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	AR
Pwd valid?	Y(not null)	N (null)	Y	Y	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO	PR
T&C accepted	Y	N	Y	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C	C
OUTPUTS					
Prepaid Wallet Created (PR-WC)			*		
Postpaid Wallet Created (PO-WC)				*	
Wallet not created, Email already registered (WNC1)					*
Wallet not created, Email does not exist (WNC2)					
Wallet not created, Email not active (WNC3)					
Wallet not created, Password not provided (WNC4)					
Wallet not created, Phone number already registered (WNC5)					
Wallet not created, Phone number not in use(WNC6)					
Wallet not created, Terms & Conditions not accepted (WNC7)					
Wallet not created, OTP not received(WNC8)					
Wallet not created, incorrect OTP entered (WNC9)					
Wallet not created, OTP expired (WNC10)					

... negative test scenario

Now let us model the behaviour as a decision table

CONDITIONS	VALUES		R1	R2	R3	R4
	valid (Y)	invalid (N)				
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	AR	DNE
Pwd valid?	Y(not null)	N (null)	Y	Y	Y	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y	Y	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO	PR	PO
T&C accepted	Y	N	Y	Y	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C	C	C
OUTPUTS						
Prepaid Wallet Created (PR-WC)			*			
Postpaid Wallet Created (PO-WC)				*		
Wallet not created, Email already registered (WNC1)					*	
Wallet not created, Email does not exist (WNC2)						*
Wallet not created, Email not active (WNC3)						
Wallet not created, Password not provided (WNC4)						
Wallet not created, Phone number already registered (WNC5)						
Wallet not created, Phone number not in use(WNC6)						
Wallet not created, Terms & Conditions not accepted (WNC7)						
Wallet not created, OTP not received(WNC8)						
Wallet not created, incorrect OTP entered (WNC9)						
Wallet not created, OTP expired (WNC10)						

Now let us model the behaviour as a decision table

CONDITIONS	VALUES		R1	R2	R3	R4	R5	R6
	valid (Y)	invalid (N)						
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	AR	DNE	NA	Y
Pwd valid?	Y(not null)	N (null)	Y	Y	Y	Y	Y	N
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y	Y	Y	Y	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO	PR	PO	PR	PO
T&C accepted	Y	N	Y	Y	Y	Y	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C	C	C	C	C
OUTPUTS								
Prepaid Wallet Created (PR-WC)			*					
Postpaid Wallet Created (PO-WC)				*				
Wallet not created, Email already registered (WNC1)					*			
Wallet not created, Email does not exist (WNC2)						*		
Wallet not created, Email not active (WNC3)							*	
Wallet not created, Password not provided (WNC4)								*
Wallet not created, Phone number already registered (WNC5)								
Wallet not created, Phone number not in use(WNC6)								
Wallet not created, Terms & Conditions not accepted (WNC7)								
Wallet not created, OTP not received(WNC8)								
Wallet not created, incorrect OTP entered (WNC9)								
Wallet not created, OTP expired (WNC10)								

Now let us model the behaviour as a decision table

CONDITIONS	VALUES		R1	R2	R3	R4	R5	R6	R7
	valid (Y)	invalid (N)							
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	AR	DNE	NA	Y	Y
Pwd valid?	Y(not null)	N (null)	Y	Y	Y	Y	Y	N	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y	Y	Y	Y	Y	AR
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO	PR	PO	PR	PO	PR
T&C accepted	Y	N	Y	Y	Y	Y	Y	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C	C	C	C	C	C
OUTPUTS									
Prepaid Wallet Created (PR-WC)			*						
Postpaid Wallet Created (PO-WC)				*					
Wallet not created, Email already registered (WNC1)					*				
Wallet not created, Email does not exist (WNC2)						*			
Wallet not created, Email not active (WNC3)							*		
Wallet not created, Password not provided (WNC4)								*	
Wallet not created, Phone number already registered (WNC5)									*
Wallet not created, Phone number not in use(WNC6)									
Wallet not created, Terms & Conditions not accepted (WNC7)									
Wallet not created, OTP not received(WNC8)									
Wallet not created, incorrect OTP entered (WNC9)									
Wallet not created, OTP expired (WNC10)									

Now let us model the behaviour as a decision table

CONDITIONS	VALUES		R1	R2	R3	R4	R5	R6	R7	R8
	valid (Y)	invalid (N)								
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	AR	DNE	NA	Y	Y	Y
Pwd valid?	Y(not null)	N (null)	Y	Y	Y	Y	Y	N	Y	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y	Y	Y	Y	Y	AR	NIU
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO	PR	PO	PR	PO	PR	PO
T&C accepted	Y	N	Y	Y	Y	Y	Y	Y	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C	C	C	C	C	C	C
OUTPUTS										
Prepaid Wallet Created (PR-WC)			*							
Postpaid Wallet Created (PO-WC)				*						
Wallet not created, Email already registered (WNC1)					*					
Wallet not created, Email does not exist (WNC2)						*				
Wallet not created, Email not active (WNC3)							*			
Wallet not created, Password not provided (WNC4)								*		
Wallet not created, Phone number already registered (WNC5)									*	
Wallet not created, Phone number not in use(WNC6)										*
Wallet not created, Terms & Conditions not accepted (WNC7)										
Wallet not created, OTP not received(WNC8)										
Wallet not created, incorrect OTP entered (WNC9)										
Wallet not created, OTP expired (WNC10)										

Now let us model the behaviour as a decision table

CONDITIONS	VALUES		R1	R2	R3	R4	R5	R6	R7	R8	R9
	valid (Y)	invalid (N)									
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	AR	DNE	NA	Y	Y	Y	Y
Pwd valid?	Y(not null)	N (null)	Y	Y	Y	Y	Y	N	Y	Y	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y	Y	Y	Y	Y	AR	NIU	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO	PR	PO	PR	PO	PR	PO	PR
T&C accepted	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C	C	C	C	C	C	C	C
OUTPUTS											
Prepaid Wallet Created (PR-WC)			*								
Postpaid Wallet Created (PO-WC)				*							
Wallet not created, Email already registered (WNC1)					*						
Wallet not created, Email does not exist (WNC2)						*					
Wallet not created, Email not active (WNC3)							*				
Wallet not created, Password not provided (WNC4)								*			
Wallet not created, Phone number already registered (WNC5)									*		
Wallet not created, Phone number not in use(WNC6)										*	
Wallet not created, Terms & Conditions not accepted (WNC7)											*
Wallet not created, OTP not received(WNC8)											
Wallet not created, incorrect OTP entered (WNC9)											
Wallet not created, OTP expired (WNC10)											

Now let us model the behaviour as a decision table

CONDITIONS	VALUES		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
	valid (Y)	invalid (N)										
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	AR	DNE	NA	Y	Y	Y	Y	Y
Pwd valid?	Y (not null)	N (null)	Y	Y	Y	Y	Y	N	Y	Y	Y	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y	Y	Y	Y	Y	AR	NIU	Y	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO	PR	PO	PR	PO	PR	PO	PR	PO
T&C accepted	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C	C	C	C	C	C	C	C	NR
OUTPUTS												
Prepaid Wallet Created (PR-WC)			*									
Postpaid Wallet Created (PO-WC)				*								
Wallet not created, Email already registered (WNC1)					*							
Wallet not created, Email does not exist (WNC2)						*						
Wallet not created, Email not active (WNC3)							*					
Wallet not created, Password not provided (WNC4)								*				
Wallet not created, Phone number already registered (WNC5)									*			
Wallet not created, Phone number not in use (WNC6)										*		
Wallet not created, Terms & Conditions not accepted (WNC7)											*	
Wallet not created, OTP not received (WNC8)												*
Wallet not created, incorrect OTP entered (WNC9)												
Wallet not created, OTP expired (WNC10)												

Now let us model the behaviour as a decision table

CONDITIONS	VALUES		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
	valid (Y)	invalid (N)												
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	AR	DNE	NA	Y	Y	Y	Y	Y	Y	Y
Pwd valid?	Y(not null)	N (null)	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y	Y	Y	Y	Y	AR	NIU	Y	Y	Y	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO	PR	PO	PR	PO	PR	PO	PR	PO	PR	PO
T&C accepted	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C	C	C	C	C	C	C	C	NR	I	E
OUTPUTS														
Prepaid Wallet Created (PR-WC)			*											
Postpaid Wallet Created (PO-WC)				*										
Wallet not created, Email already registered (WNC1)					*									
Wallet not created, Email does not exist (WNC2)						*								
Wallet not created, Email not active (WNC3)							*							
Wallet not created, Password not provided (WNC4)								*						
Wallet not created, Phone number already registered (WNC5)									*					
Wallet not created, Phone number not in use(WNC6)										*				
Wallet not created, Terms & Conditions not accepted (WNC7)											*			
Wallet not created, OTP not received(WNC8)												*		
Wallet not created, incorrect OTP entered (WNC9)													*	
Wallet not created, OTP expired (WNC10)														*

POSITIVE scenarios

Decision Table

CONDITIONS	VALUES		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
	valid (Y)	invalid (N)												
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	AR	DNE	NA	Y	Y	Y	Y	Y	Y	Y
Pwd valid?	Y(not null)	N (null)	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y	Y	Y	Y	Y	AR	NIU	Y	Y	Y	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO	PR	PO	PR	PO	PR	PO	PR	PO	PR	PO
T&C accepted	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C	C	C	C	C	C	C	C	NR	I	E
OUTPUTS														
Prepaid Wallet Created (PR-WC)			*											
Postpaid Wallet Created (PO-WC)				*										
Wallet not created, Email already registered (WNC1)					*									
Wallet not created, Email does not exist (WNC2)						*								
Wallet not created, Email not active (WNC3)							*							
Wallet not created, Password not provided (WNC4)								*						
Wallet not created, Phone number already registered (WNC5)									*					
Wallet not created, Phone number not in use(WNC6)										*				
Wallet not created, Terms & Conditions not accepted (WNC7)											*			
Wallet not created, OTP not received(WNC8)												*		
Wallet not created, incorrect OTP entered (WNC9)													*	
Wallet not created, OTP expired (WNC10)														*

Decision Table

POSITIVE scenarios

NEGATIVE scenarios

CONDITIONS	VALUES		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
	valid (Y)	invalid (N)												
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	AR	DNE	NA	Y	Y	Y	Y	Y	Y	Y
Pwd valid?	Y(not null)	N (null)	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y	Y	Y	Y	Y	AR	NIU	Y	Y	Y	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO	PR	PO	PR	PO	PR	PO	PR	PO	PR	PO
T&C accepted	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C	C	C	C	C	C	C	C	NR	I	E
OUTPUTS														
Prepaid Wallet Created (PR-WC)			*											
Postpaid Wallet Created (PO-WC)				*										
Wallet not created, Email already registered (WNC1)					*									
Wallet not created, Email does not exist (WNC2)						*								
Wallet not created, Email not active (WNC3)							*							
Wallet not created, Password not provided (WNC4)								*						
Wallet not created, Phone number already registered (WNC5)									*					
Wallet not created, Phone number not in use(WNC6)										*				
Wallet not created, Terms & Conditions not accepted (WNC7)											*			
Wallet not created, OTP not received(WNC8)												*		
Wallet not created, incorrect OTP entered (WNC9)													*	
Wallet not created, OTP expired (WNC10)														*

Behaviour based design

#3 Generate scenarios/cases

POSITIVE scenarios

Decision Table

CONDITIONS	VALUES		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
	valid (Y)	invalid (N)												
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	AR	DNE	NA	Y	Y	Y	Y	Y	Y	Y
Pwd valid?	Y(not null)	N (null)	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y	Y	Y	Y	Y	AR	NIU	Y	Y	Y	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO	PR	PO	PR	PO	PR	PO	PR	PO	PR	PO
T&C accepted	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C	C	C	C	C	C	C	C	NR	I	E
OUTPUTS														
Prepaid Wallet Created (PR-WC)			*											
Postpaid Wallet Created (PO-WC)				*										
Wallet not created, Email already registered (WNC1)					*									
Wallet not created, Email does not exist (WNC2)						*								
Wallet not created, Email not active (WNC3)							*							
Wallet not created, Password not provided (WNC4)								*						
Wallet not created, Phone number already registered (WNC5)									*					
Wallet not created, Phone number not in use(WNC6)										*				
Wallet not created, Terms & Conditions not accepted (WNC7)											*			
Wallet not created, OTP not received(WNC8)												*		
Wallet not created, incorrect OTP entered (WNC9)													*	
Wallet not created, OTP expired (WNC10)														*

Decision Table

POSITIVE scenarios

NEGATIVE scenarios

CONDITIONS	VALUES		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
	valid (Y)	invalid (N)												
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	AR	DNE	NA	Y	Y	Y	Y	Y	Y	Y
Pwd valid?	Y(not null)	N (null)	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y	Y	Y	Y	Y	AR	NIU	Y	Y	Y	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO	PR	PO	PR	PO	PR	PO	PR	PO	PR	PO
T&C accepted	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C	C	C	C	C	C	C	C	NR	I	E
OUTPUTS														
Prepaid Wallet Created (PR-WC)			*											
Postpaid Wallet Created (PO-WC)				*										
Wallet not created, Email already registered (WNC1)					*									
Wallet not created, Email does not exist (WNC2)						*								
Wallet not created, Email not active (WNC3)							*							
Wallet not created, Password not provided (WNC4)								*						
Wallet not created, Phone number already registered (WNC5)									*					
Wallet not created, Phone number not in use(WNC6)										*				
Wallet not created, Terms & Conditions not accepted (WNC7)											*			
Wallet not created, OTP not received(WNC8)												*		
Wallet not created, incorrect OTP entered (WNC9)													*	
Wallet not created, OTP expired (WNC10)														*

Decision Table

CONDITIONS	VALUES		R1	R2
	valid (Y)	invalid (N)		
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y
Pwd valid?	Y (not null)	N (null)	Y	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO
T&C accepted	Y	N	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C
OUTPUTS				
Prepaid Wallet Created (PR-WC)			*	
Postpaid Wallet Created (PO-WC)				*
Wallet not created, Email already registered (WNC1)				
Wallet not created, Email does not exist (WNC2)				
Wallet not created, Email not active (WNC3)				
Wallet not created, Password not provided (WNC4)				
Wallet not created, Phone number already registered (WNC5)				
Wallet not created, Phone number not in use(WNC6)				
Wallet not created, Terms & Conditions not accepted (WNC7)				
Wallet not created, OTP not received(WNC8)				
Wallet not created, incorrect OTP entered (WNC9)				
Wallet not created, OTP expired (WNC10)				

Generate test scenarios (POSITIVE)

TS1 (R1): Ensure that prepaid wallet is created if prepaid wallet chosen and all conditions are satisfied.

Decision Table

CONDITIONS	VALUES		R1	R2
	valid (Y)	invalid (N)		
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y
Pwd valid?	Y(not null)	N (null)	Y	Y
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)	Y	Y
Type of account	Prepaid/Postpaid (PR/ PO)		PR	PO
T&C accepted	Y	N	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)	C	C
OUTPUTS				
Prepaid Wallet Created (PR-WC)			*	
Postpaid Wallet Created (PO-WC)				*
Wallet not created, Email already registered (WNC1)				
Wallet not created, Email does not exist (WNC2)				
Wallet not created, Email not active (WNC3)				
Wallet not created, Password not provided (WNC4)				
Wallet not created, Phone number already registered (WNC5)				
Wallet not created, Phone number not in use(WNC6)				
Wallet not created, Terms & Conditions not accepted (WNC7)				
Wallet not created, OTP not received(WNC8)				
Wallet not created, incorrect OTP entered (WNC9)				
Wallet not created, OTP expired (WNC10)				

Generate test scenarios (POSITIVE)

TS1 (R1): Ensure that prepaid wallet is created if prepaid wallet chosen and all conditions are satisfied.

TS2(R2): Ensure that postpaid wallet is created if postpaid wallet chosen and all conditions are satisfied.

Decision Table

CONDITIONS	VALUES	
	valid (Y)	invalid (N)
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)
Pwd valid?	Y (not null)	N (null)
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)
Type of account	Prepaid/Postpaid (PR/ PO)	
T&C accepted	Y	N
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)

OUTPUTS
Prepaid Wallet Created (PR-WC)
Postpaid Wallet Created (PO-WC)
Wallet not created, Email already registered (WNC1)
Wallet not created, Email does not exist (WNC2)
Wallet not created, Email not active (WNC3)
Wallet not created, Password not provided (WNC4)
Wallet not created, Phone number already registered (WNC5)
Wallet not created, Phone number not in use (WNC6)
Wallet not created, Terms & Conditions not accepted (WNC7)
Wallet not created, OTP not received (WNC8)
Wallet not created, incorrect OTP entered (WNC9)
Wallet not created, OTP expired (WNC10)

NEGATIVE scenarios

R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
AR	DNE	NA	Y	Y	Y	Y	Y	Y	Y
Y	Y	Y	N	Y	Y	Y	Y	Y	Y
Y	Y	Y	Y	AR	NIU	Y	Y	Y	Y
PR	PO	PR	PO	PR	PO	PR	PO	PR	PO
Y	Y	Y	Y	Y	Y	N	Y	Y	Y
C	C	C	C	C	C	C	NR	I	E

Generate test scenarios (NEGATIVE)

TS12 (R12): Ensure that wallet is NOT created if OTP is entered after a long time

Decision Table

CONDITIONS	VALUES	
	valid (Y)	invalid (N)
Email ID valid?	Unique & Active (Y)	Already Registered (AR)/ Does not exist(DNE)/not active (NA)
Pwd valid?	Y (not null)	N (null)
Ph# valid?	Unique and in-use (Y)	Already Registered (AR)/ Not in-use (NIU)
Type of account	Prepaid/Postpaid (PR/ PO)	
T&C accepted	Y	N
OTP valid?	Y: Correct value & Not expired (C)	Not received (NR)/Incorrect (I)/ Expired (E)

OUTPUTS
Prepaid Wallet Created (PR-WC)
Postpaid Wallet Created (PO-WC)
Wallet not created, Email already registered (WNC1)
Wallet not created, Email does not exist (WNC2)
Wallet not created, Email not active (WNC3)
Wallet not created, Password not provided (WNC4)
Wallet not created, Phone number already registered (WNC5)
Wallet not created, Phone number not in use(WNC6)
Wallet not created, Terms & Conditions not accepted (WNC7)
Wallet not created, OTP not received(WNC8)
Wallet not created, incorrect OTP entered (WNC9)
Wallet not created, OTP expired (WNC10)

NEGATIVE scenarios

R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
AR	DNE	NA	Y	Y	Y	Y	Y	Y	Y
Y	Y	Y	N	Y	Y	Y	Y	Y	Y
Y	Y	Y	Y	AR	NIU	Y	Y	Y	Y
PR	PO	PR	PO	PR	PO	PR	PO	PR	PO
Y	Y	Y	Y	Y	Y	N	Y	Y	Y
C	C	C	C	C	C	C	NR	I	E

Generate test scenarios (NEGATIVE)

TS12 (R12): Ensure that wallet is NOT created if OTP is entered after a long time

TS3 (R3): Ensure that wallet is NOT created if the entered email ID already exists

Let us generate test cases for POSITIVE scenario TS1

TS1 (R1): Ensure that prepaid wallet is created if prepaid wallet chosen and all conditions satisfied.

(1) Use each rule to generate a scenario

CONDITIONS	VALUES	R1
	valid (Y)	
Email ID valid?	Unique & Active (Y)	Y
Pwd valid?	Y (not null)	Y
Ph# valid?	Unique and in-use (Y)	Y
Type of account	Prepaid/Postpaid (PR/ PO)	PR
T&C accepted	Y	Y
OTP valid?	Y: Correct value & Not expired (C)	C

Let us generate test cases for POSITIVE scenario TS1

TS1 (R1): Ensure that prepaid wallet is created if prepaid wallet chosen and all conditions are satisfied.

CONDITIONS	VALUES	R1	Values	#Values
	valid (Y)			
Email ID valid?	Unique & Active (Y)	Y	Y	1
Pwd valid?	Y (not null)	Y	Y	1
Ph# valid?	Unique and in-use (Y)	Y	Y	1
Type of account	Prepaid/Postpaid (PR/ PO)	PR	PR	1
T&C accepted	Y	Y	Y	1
OTP valid?	Y: Correct value & Not expired (C)	C	correct one entered before time => Immediately (IMM) Just before expiry time (JBE) (boundary)	2

**(2) Generate values
for each condition of the rule**

Let us generate test cases for POSITIVE scenario TS1

TS1 (R1): Ensure that prepaid wallet is created if prepaid wallet chosen and all conditions are satisfied.

CONDITIONS	VALUES	R1	Values	#Values
	valid (Y)			
Email ID valid?	Unique & Active (Y)	Y	Y	1
Pwd valid?	Y (not null)	Y	Y	1
Ph# valid?	Unique and in-use (Y)	Y	Y	1
Type of account	Prepaid/Postpaid (PR/ PO)	PR	PR	1
T&C accepted	Y	Y	Y	1
OTP valid?	Y: Correct value & Not expired (C)	C	correct one entered before time => Immediately (IMM) Just before expiry time (JBE) (boundary)	2

(2) Generate values for each condition of the rule

(3) Combine these values optimally to generate test cases

TEST CASE(S)

TC ID	Other Inputs	OTP	Exp Result
TS1.TC1	Enter unique & active email ID, with valid pwd and valid valid Ph# accepting T&C	IMM	Prepaid Wallet Created (PR-WC)
TS1.TC2	Enter unique & active email ID, with valid pwd and valid valid Ph# accepting T&C	JBE	

Let us generate test cases for **NEGATIVE** scenario TS12

TS12 (R12): Ensure that wallet is NOT created if OTP entered has expired.

CONDITIONS	VALUES	R12
	invalid(N)	
Email ID valid?	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y
Pwd valid?	N (null)	Y
Ph# valid?	Already Registered (AR)/ Not in-use (NIU)	Y
Type of account		PO
T&C accepted	N	Y
OTP valid?	Not received (NR) Incorrect (I) Expired (E)	E

(1) Use each rule to generate a scenario

Let us generate test cases for **NEGATIVE** scenario TS12

TS12 (R12): Ensure that wallet is NOT created if OTP entered has expired.

CONDITIONS	VALUES	R12	Values	#Values
	invalid(N)			
Email ID valid?	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	1
Pwd valid?	N (null)	Y	Y	1
Ph# valid?	Already Registered (AR)/ Not in-use (NIU)	Y	Y	1
Type of account		PO	PR	1
T&C accepted	N	Y	Y	1
OTP valid?	Not received (NR) Incorrect (I) Expired (E)	E	correct one entered after a long time just after expiry time (JEX) much after expiry (MAF)	2

(1) Use each rule to generate a scenario

**(2) Generate values
for each condition of the rule**

Let us generate test cases for **NEGATIVE** scenario TS12

TS12 (R12): Ensure that wallet is NOT created if OTP entered has expired.

CONDITIONS	VALUES	R12	Values	#Values
	invalid(N)			
Email ID valid?	Already Registered (AR)/ Does not exist(DNE)/not active (NA)	Y	Y	1
Pwd valid?	N (null)	Y	Y	1
Ph# valid?	Already Registered (AR)/ Not in-use (NIU)	Y	Y	1
Type of account		PO	PR	1
T&C accepted	N	Y	Y	1
OTP valid?	Not received (NR) Incorrect (I) Expired (E)	E	correct one entered after a long time just after expiry time (JEX) much after expiry (MAF)	2

(1) Use each rule to generate a scenario

(2) Generate values for each condition of the rule

(3) Combine these values optimally to generate test cases

TEST CASE(S)

TC ID	Other Inputs	OTP	Exp Result
TS12.TC1	Enter unique &active email ID, with valid pwd and valid valid Ph# accepting T&C	JEX	Wallet not created, OTP expired (WNC10)
TS12.TC2	Enter unique &active email ID, with valid pwd and valid valid Ph# accepting T&C	MAF	

Operational Profiling

Load, Stress, Performance, Scalability test design

Operational profiling is about understanding:

Who the various end **users** are

and the various **operations** done by them

Operational profiling is about understanding:

Who the various end **users** are

and the various **operations** done by them

The **concurrency** of operations

and therefore the number of concurrent users

Operational profiling is about understanding:

Who the various end **users** are

and the various **operations** done by them

The **concurrency** of operations

and therefore the number of concurrent users

Rate of arrival of operations

and seasonal variations

Objective : To mimic real life operational load scenarios

Creating operational profile

WHAT
(operations)

01

02

03

04

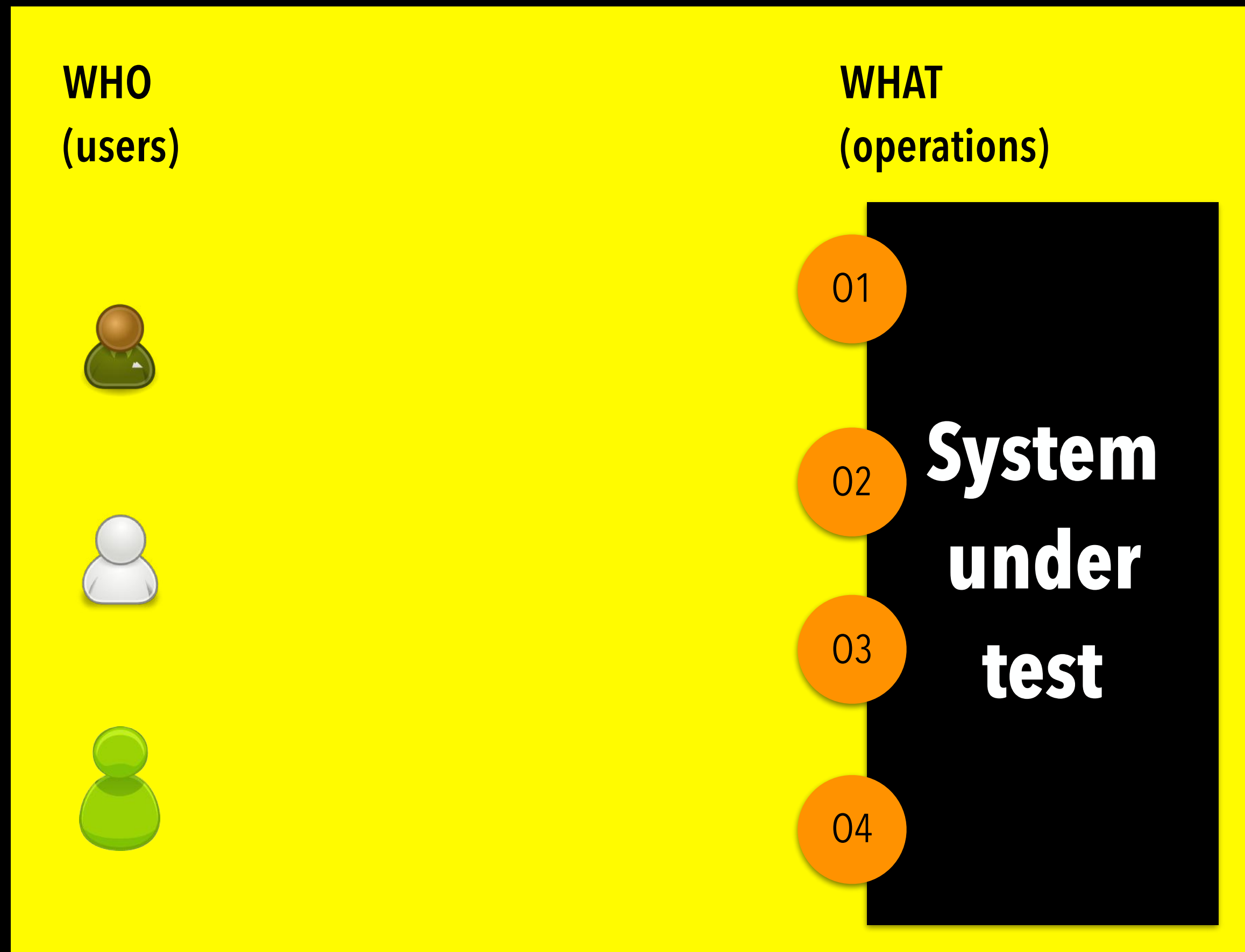
**System
under
test**

**1. List key operations -
business flows/ user requirements
(what)**

2. List the various types of users
(who)

3. Connect the various types of
users to the operations
(who-does-what)

Creating operational profile



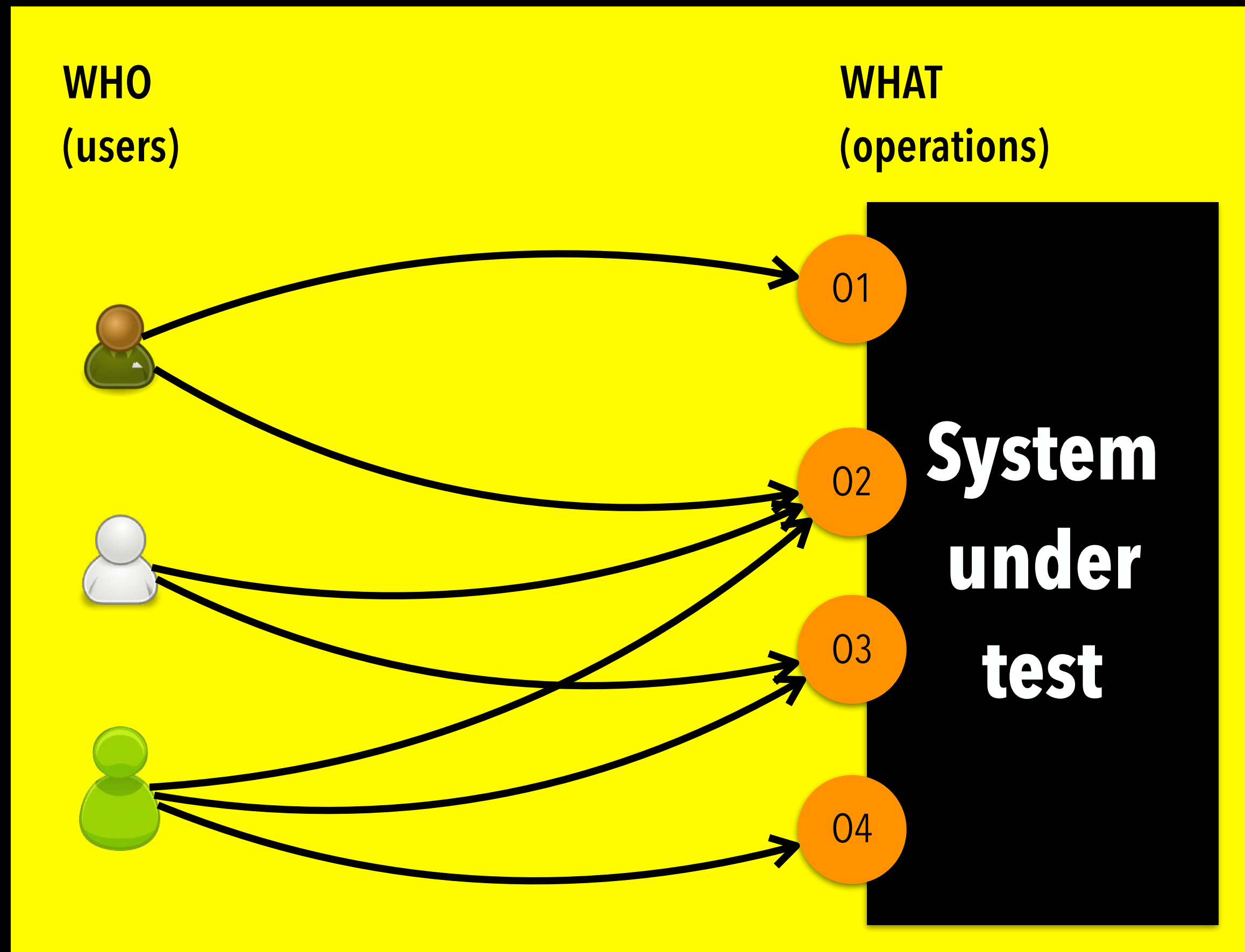
1. List key operations -
business flows/ user requirements
(what)

**2. List the various types of users
(who)**

3. Connect the various types of
users to the operations
(who-does-what)

NOTE that users does not imply only humans,
machines/other-systems could be users too.
e.g. automatic file upload to a remote machine

Creating operational profile



1. List key operations -
business flows/ user requirements
(what)

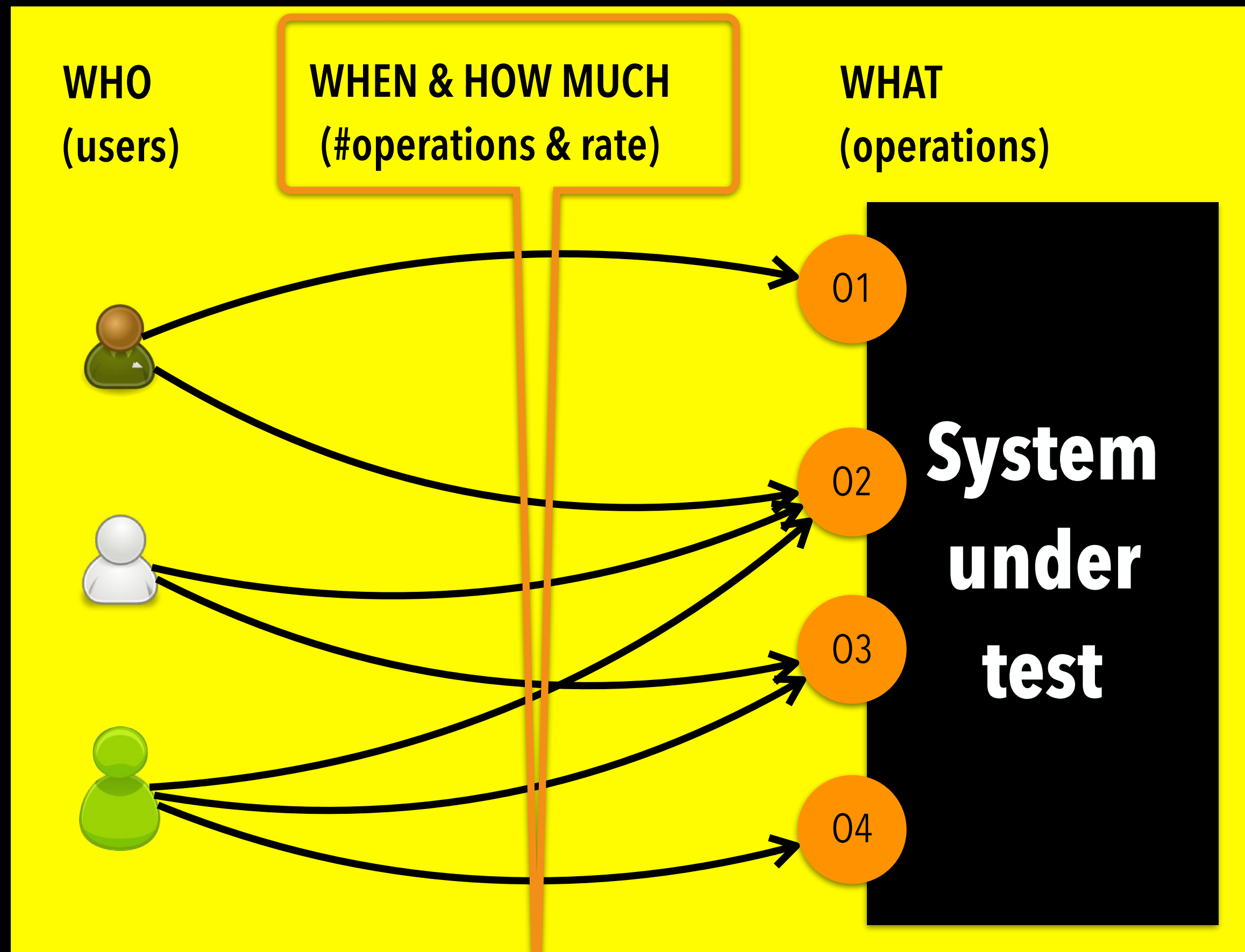
2. List the various types of users
(who)

**3. Connect the various types of
users to the operations
(who-does-what)**

Who the various end **users** are

and the various **operations** done by them

Creating operational profile



1. List key operations -
business flows/ user requirements
(what)

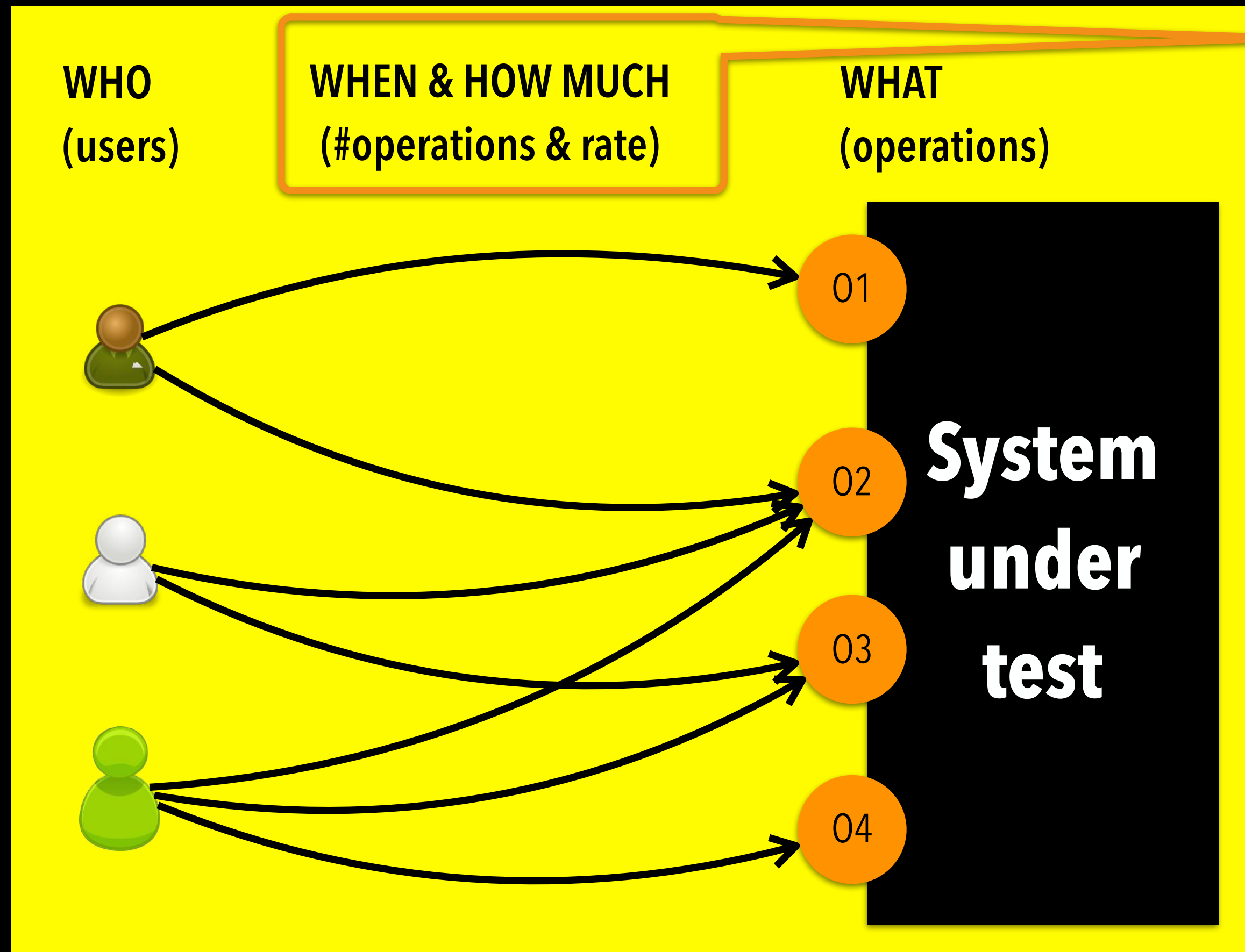
2. List the various types of users
(who)

3. Connect the various types of
users to the operations
(who-does-what)

Now we need to figure out:

- How many operations per unit time by each user?
- How many concurrent users may be there?

Creating operational profile



List the various modes

For each mode

1. List #concurrent-users

2. Now calculate total #operations

3. Now figure out the rate of arrival of operations. The simplest distribution is one where we state equally spaced arrivals.

Now we have the operational profile.

The **concurrency** of operations & therefore #concurrent users

Rate of arrival of operations and seasonal variations

Now we know : **WHAT, WHO, HOW-MUCH, WHEN**

Who the various end **users** are

and the various **operations** done by them

The **concurrency** of operations

and therefore the number of concurrent users

Rate of arrival of operations

and seasonal variations

Objective : To mimic real life operational load scenarios

Road traffic analogy

WHO & WHAT

① Trailer



② Truck



③ Car

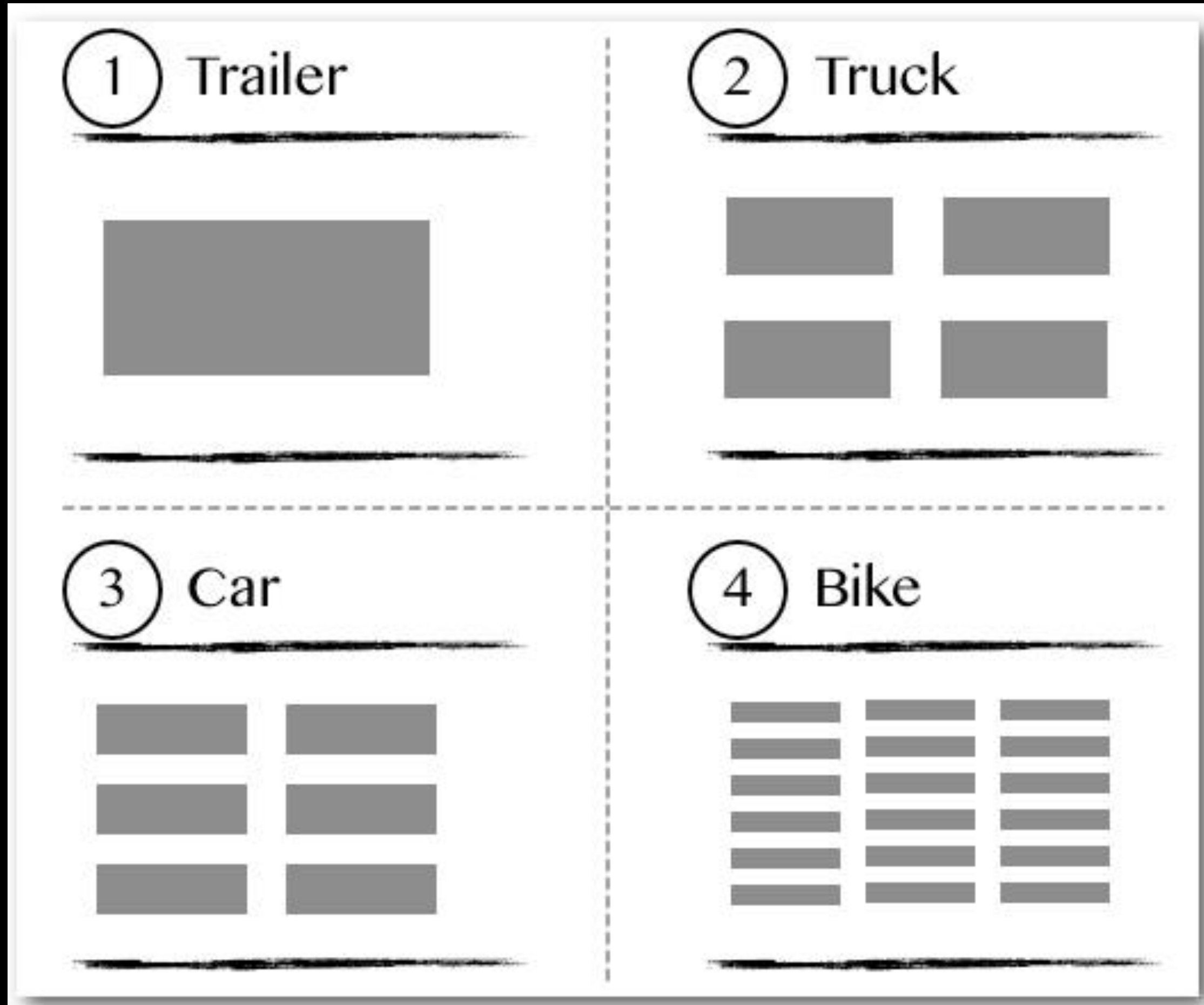


④ Bike



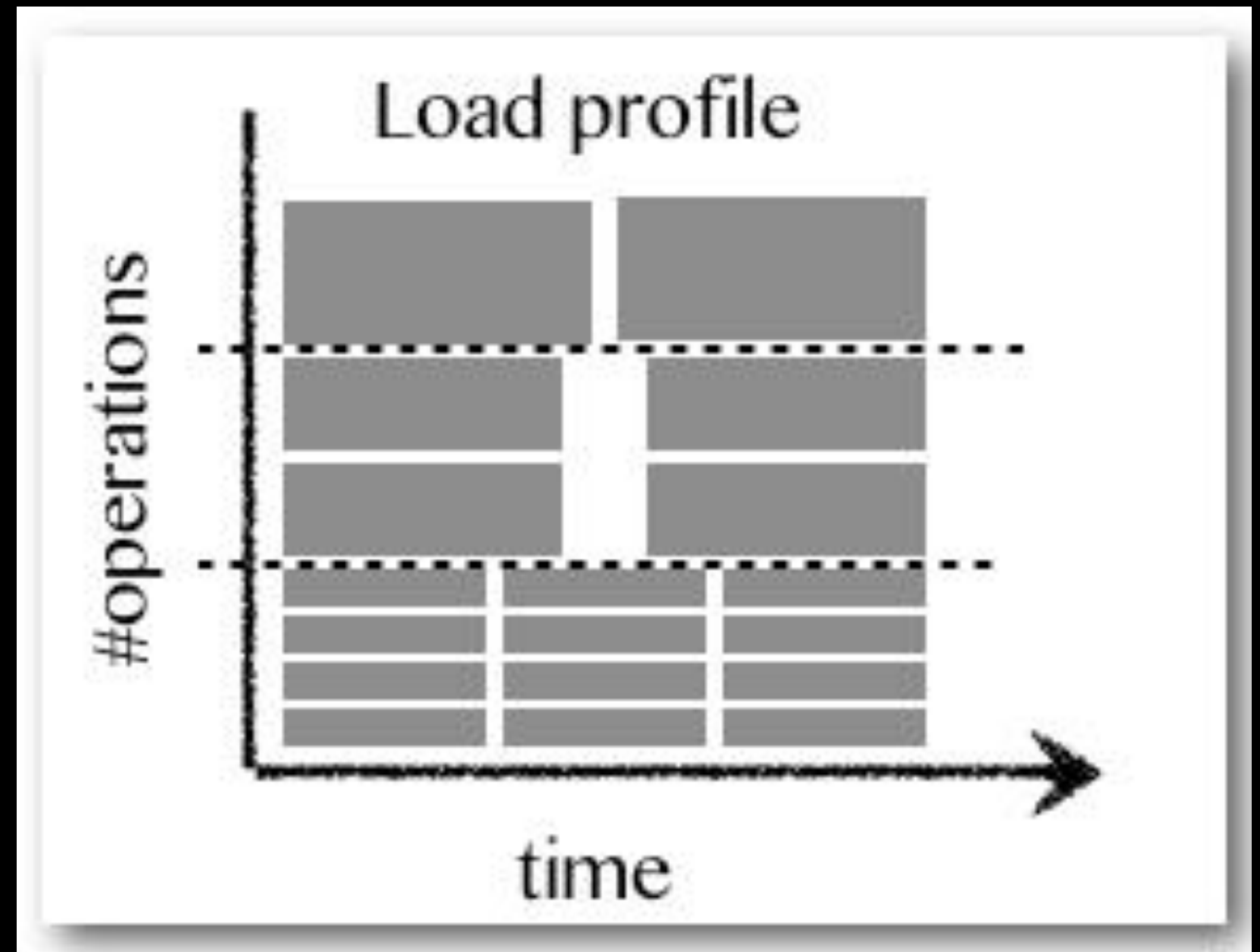
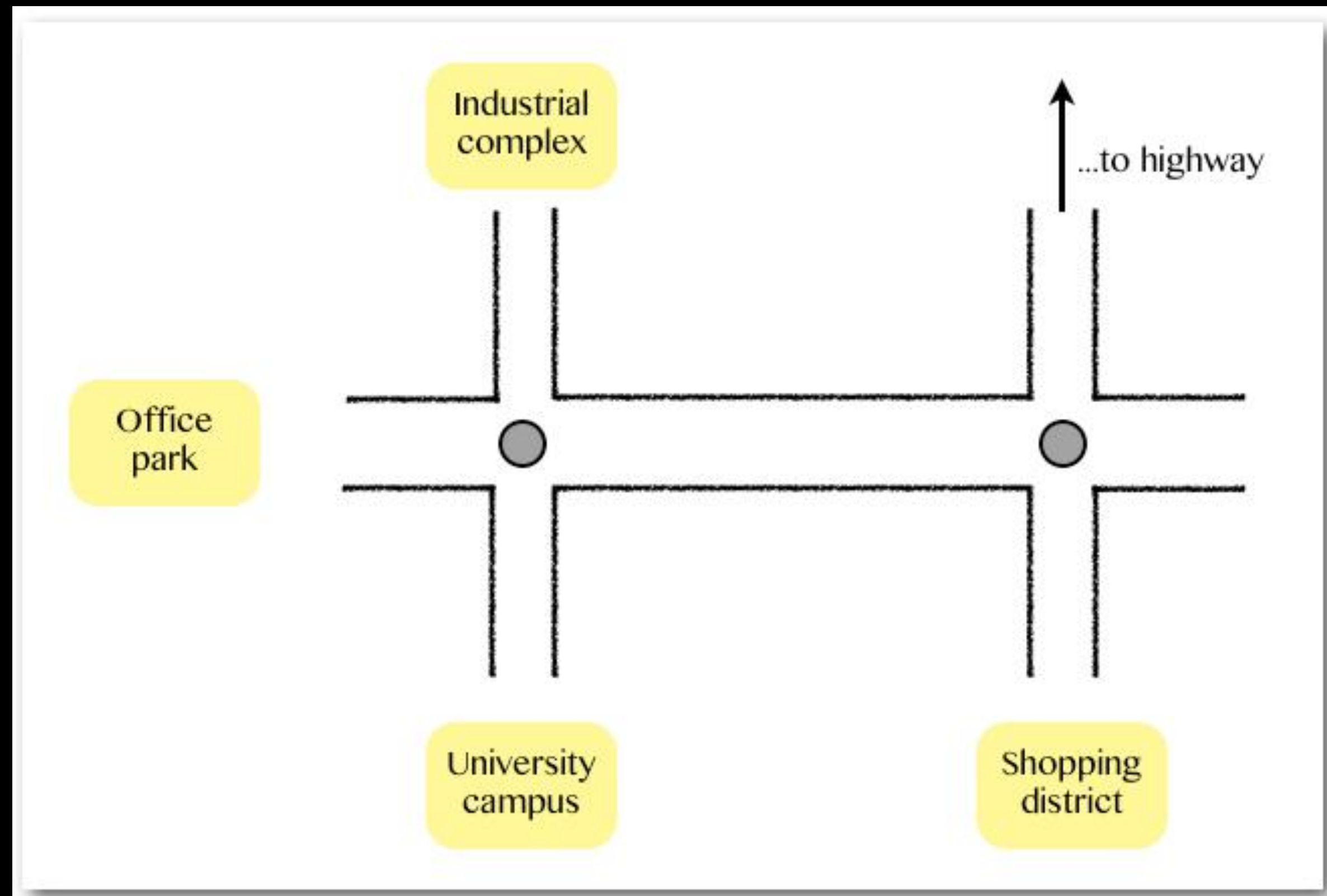
Road traffic analogy

WHO & WHAT



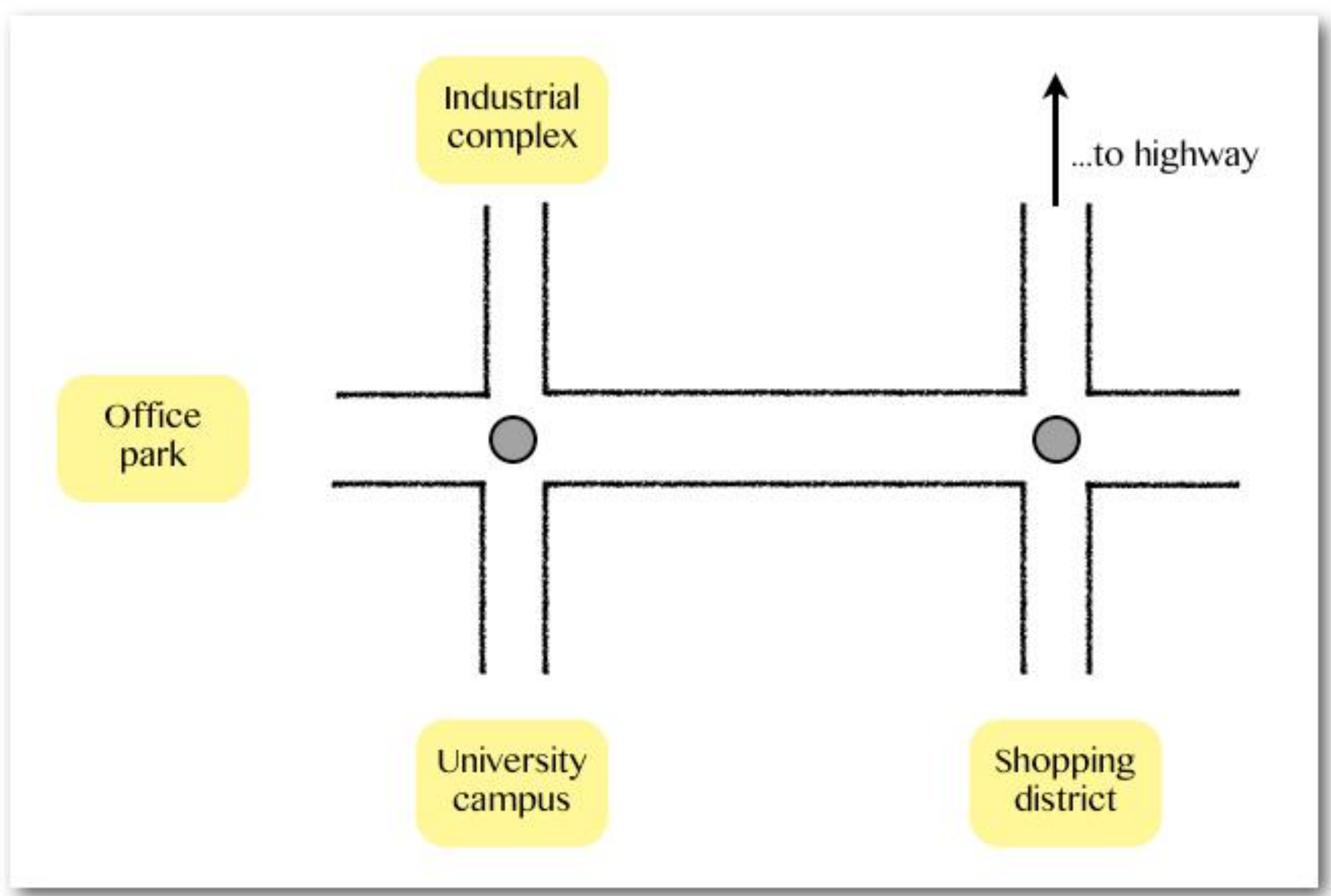
Road traffic analogy

USAGE CONTEXT

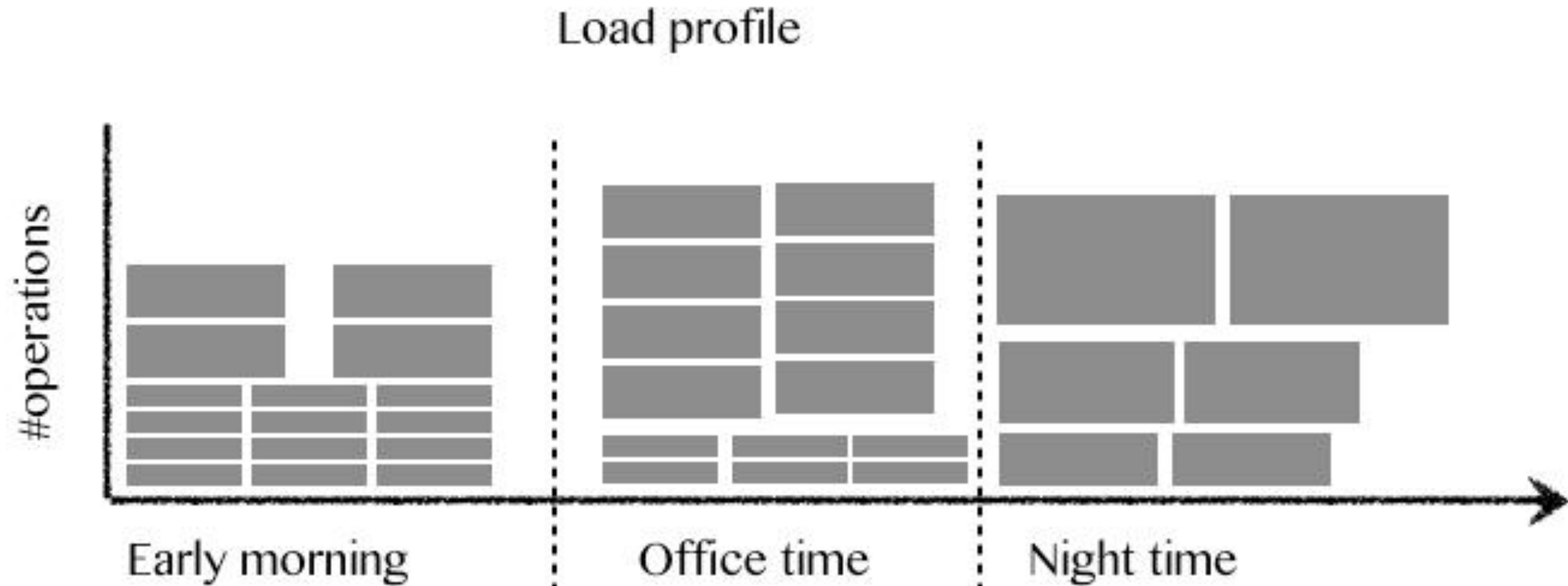


Road traffic analogy

USAGE CONTEXT



Road traffic analogy the LOAD PROFILE



Discussion #1

“Role of test documentation styles and organisation”

Test documentation is often weary, dreary and lengthy.

Does writing style and organisation impact test effectiveness?

Does it have a bearing on the ability to be agile?

Should it be structured? or free format?

Can it be structured, yet be creative?

Your comments/thoughts please.

Test documentation

Contents of test case document

Test objective

This is really a TEST SCENARIO

Pre-requisites

What is to be done before we stimulate this behaviour

Test data

INPUT-1	INPUT-2	INPUT-3	EXP. RESULT
A1	B1	C1	ER1
A2	B2	C2	ER2

Test steps

Actions to be done to execute this behaviour

Expected result

List 'what to check for' to assess correctness

Contents of test case document - How much do we need?

Test objective

This is really a TEST SCENARIO

Pre-requisites

What is to be done before we stimulate this behaviour

Test data

INPUT-1	INPUT-2	INPUT-3	EXP. RESULT
A1	B1	C1	ER1
A2	B2	C2	ER2

Test steps

Actions to be done to execute this behaviour

Expected result

List 'what to check for' to assess correctness

Typical contents of test documentation

Entity #1 (say Feature 1)

Entity #2 (say Feature 2)

Entity #3 (say Feature 3)

1. Do test wise
2. For each test, entity-wise
3. For each entity, Scenario-wise
4. For each scenario, list pre-requisites
5. For each scenario, list test cases using tabular notation
6. For each scenario, list detailed steps as needed
7. Tag scenario tag :
 - a. to type Conformance (+ve)/ Robustness (-ve)
 - b. with PRIORITY (say 1/2/3)
 - c. to PDT as appropriate

Entity #1

Scenario#1

Pre-requisites

...

Test steps

1.

2.

Test Cases

Scenario#2

...

Writing a test scenario - One liner

A test scenario reflects a behavior and is therefore "the path from beginning to end"

Write this in ONE sentence using IMPERATIVE style

Ensure that
the system does ..."

+ve

Ensure that
the system does NOT ..."

-ve

"Ensure that email is delivered when sender and receiver information is complete,
PDF file attached, payment method selected and terms/conditions agreed"

...add attributes to test scenario like TYPE, PRIORITY

Degree of importance, maybe rank as 1/2/3

What type of scenario? Positive or Negative

Objective

TS1	Ensure that email is delivered when sender and receiver information is complete, PDF file attached, payment method selected and terms/conditions agreed.	+ve	1
-----	--	-----	---

TC ID	SEND By	Pgs2Print	PaperSize	Two Sided?	Colour?	EXP. RESULT
TS1.TC1	Normal Post	1	A4	Yes	Yes	Successful
TS1.TC2	Speed Post	2	A5	No	No (B/W)	Successful

Writing test cases - Tabular

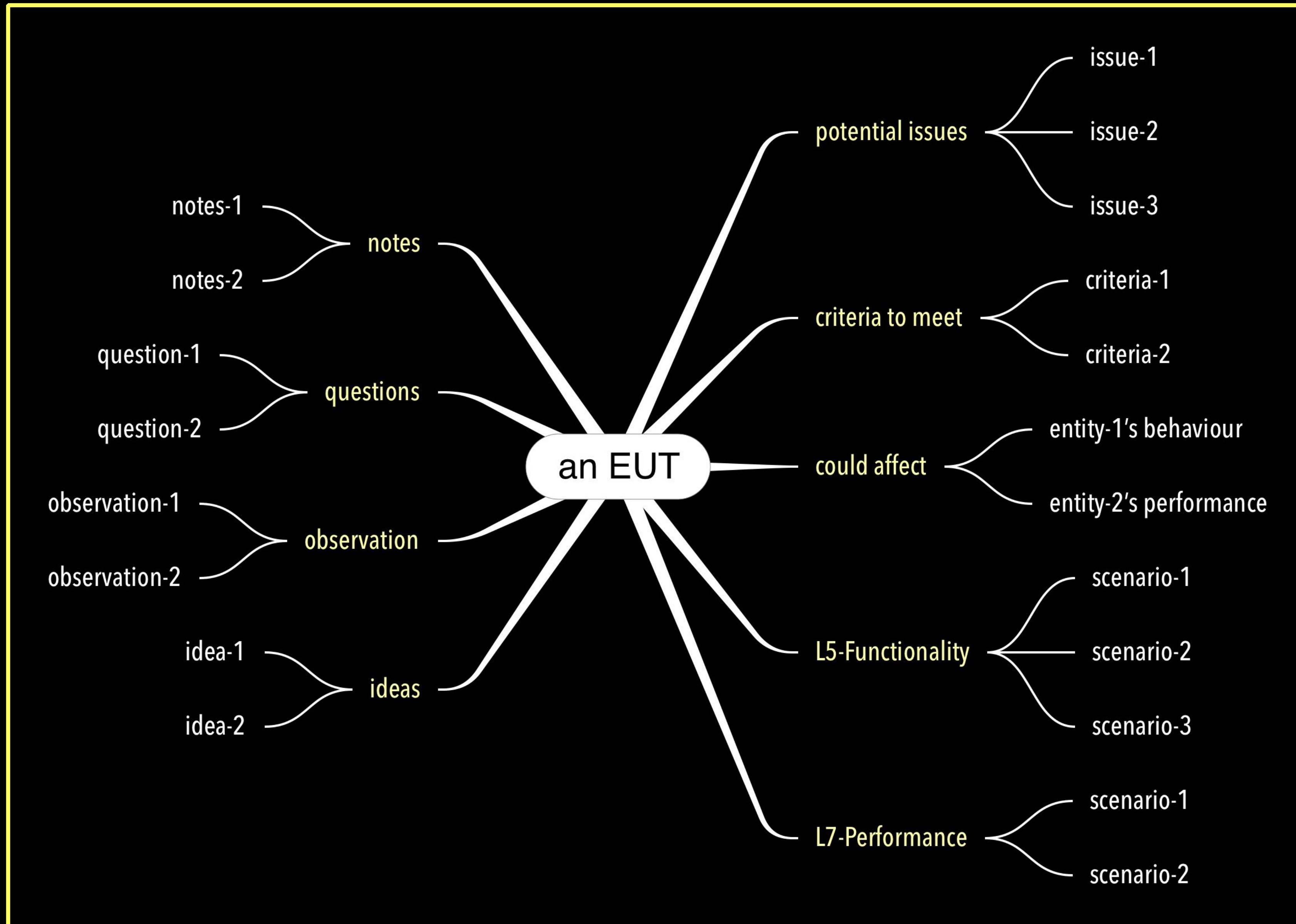
A test case is an unique combination of a set of inputs to stimulate a behavior

Consider the example where the following inputs vary for this scenario

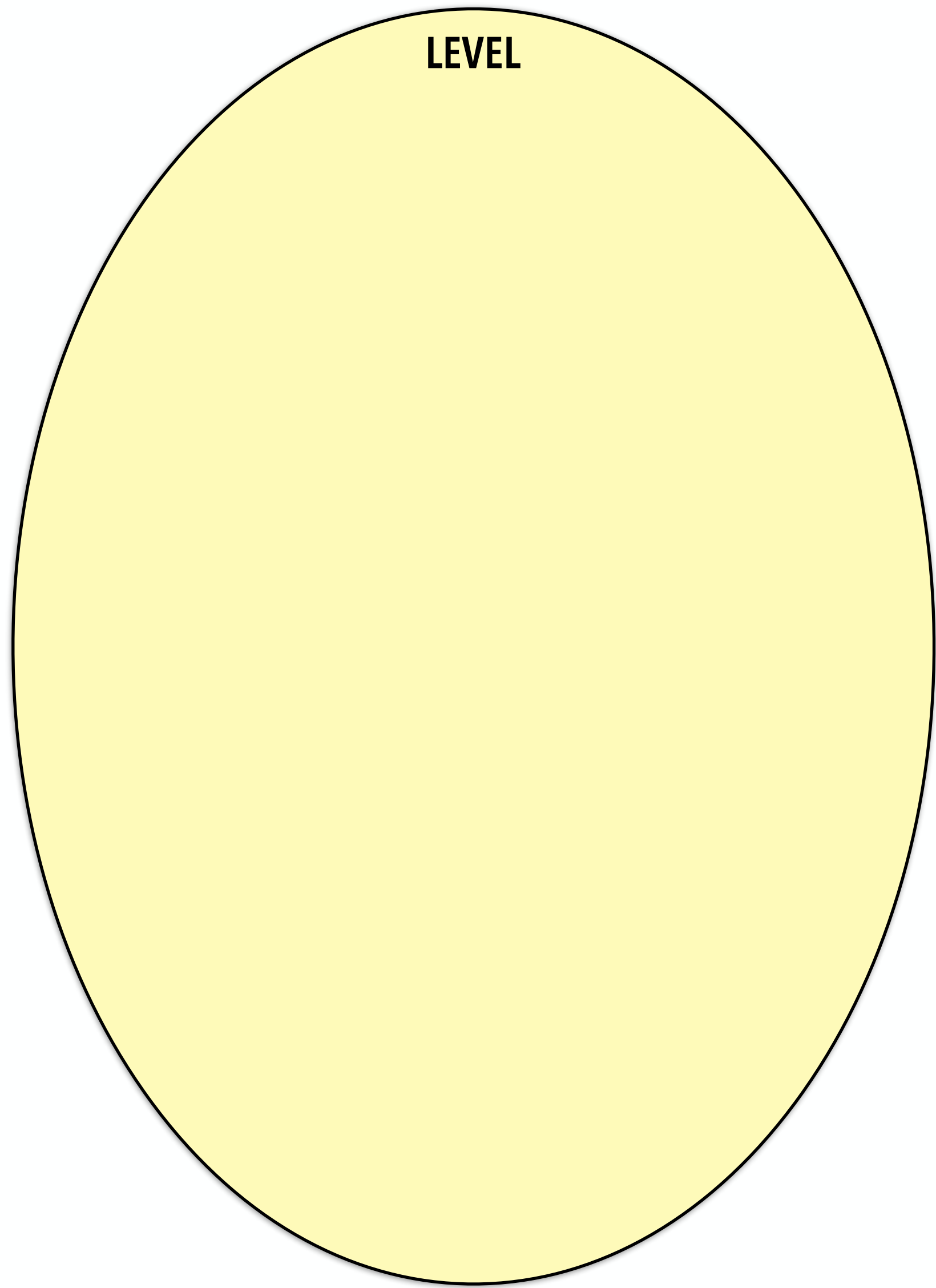
INPUT	Values
SendBy	SpeedPost, NormalPost
Pgs2Print	1, 2
PaperSize	A4, A5
Two-sided?	Yes/No
Colour?	C/BW

TC ID	SEND By	Pgs2Print	PaperSize	Two Sided?	Colour?	EXP. RESULT
TS1.TC1	Normal Post	1	A4	Yes	Yes	Successful
TS1.TC2	Speed Post	2	A5	No	No (B/W)	Successful

Could we ideate & document non-linearly?

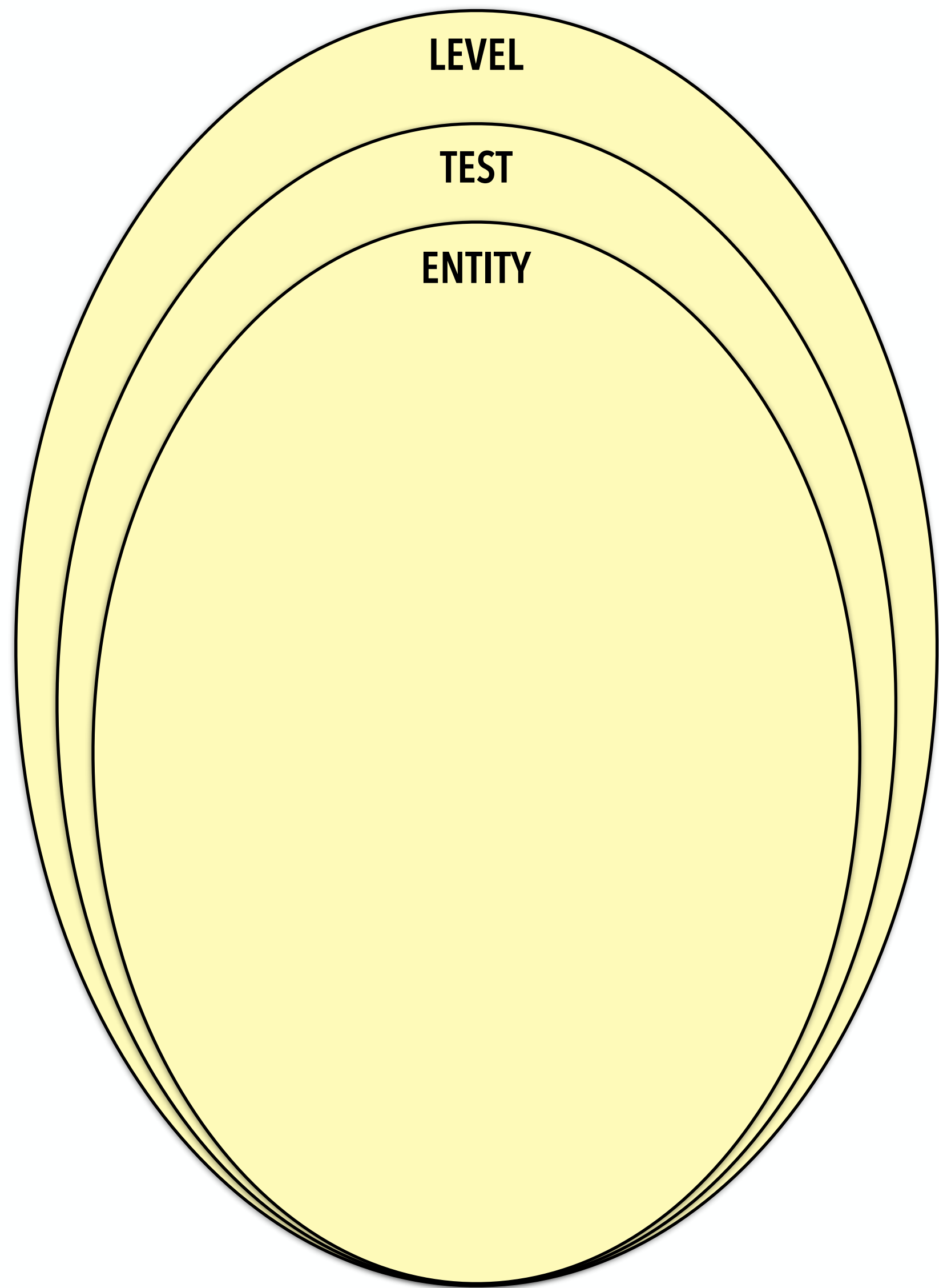


Test case architecture



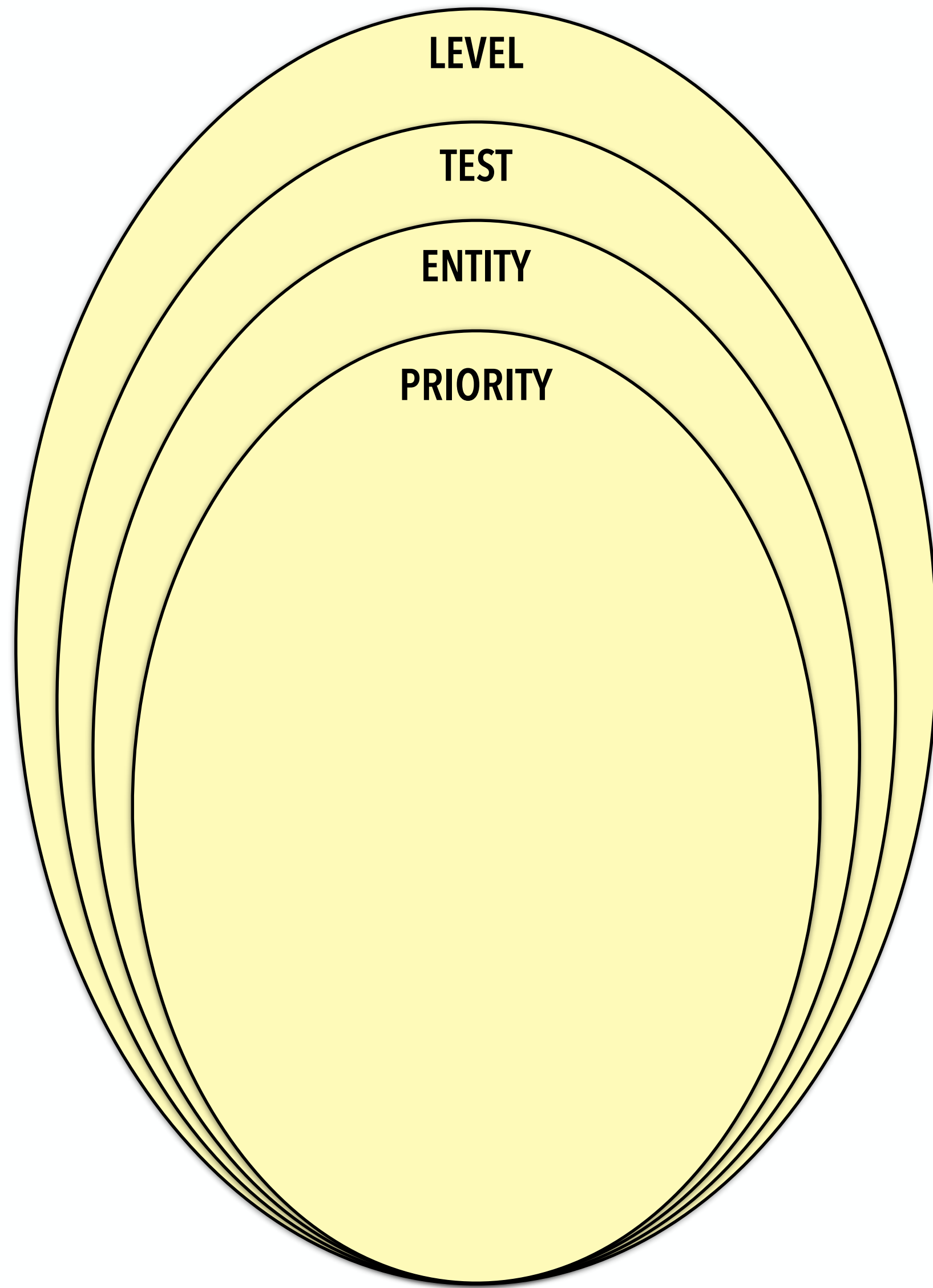
TS

Arranged
by LEVELS,



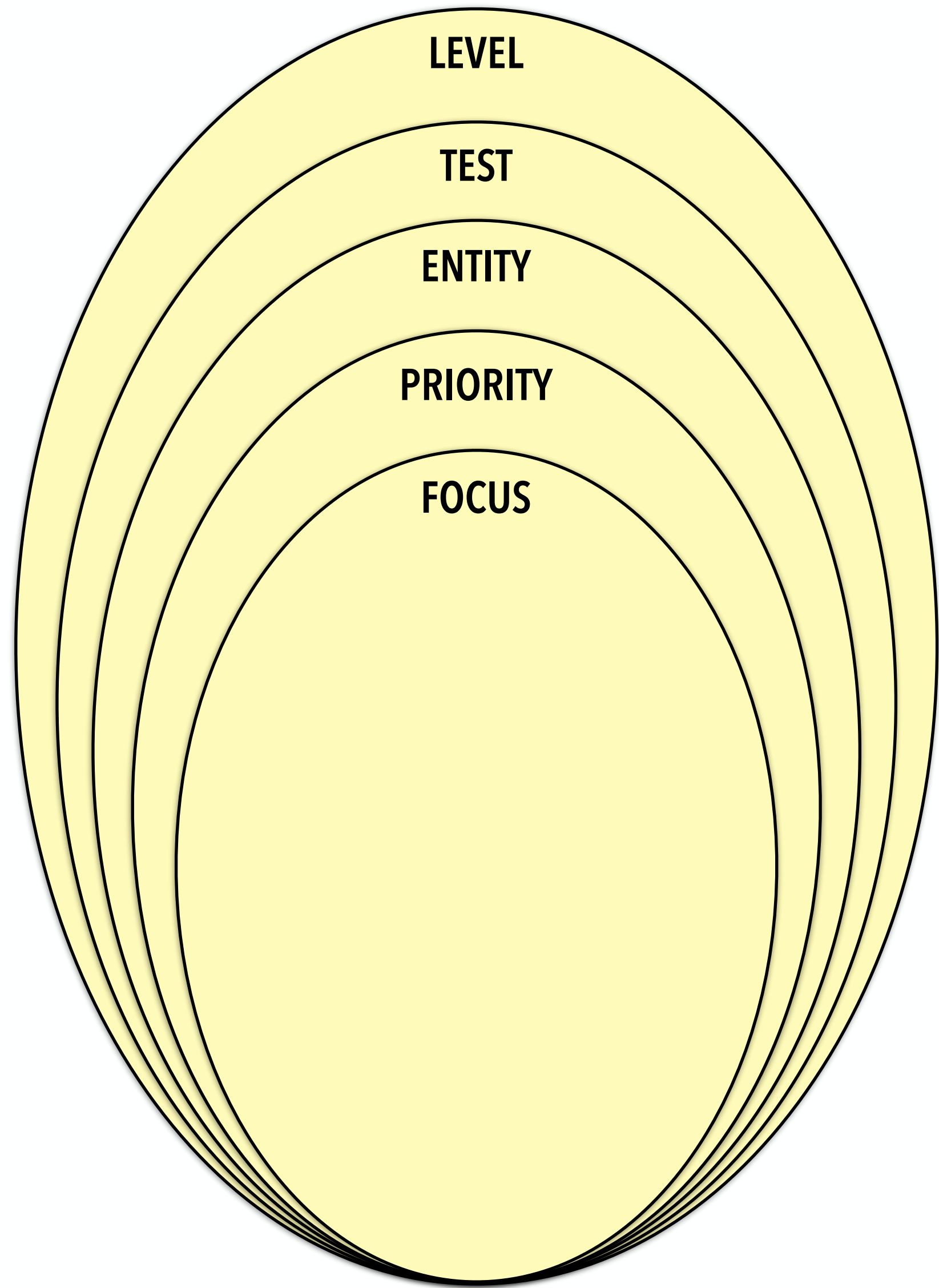
TS

Arranged
by LEVELS,
with ENTITIES
grouped by TEST,



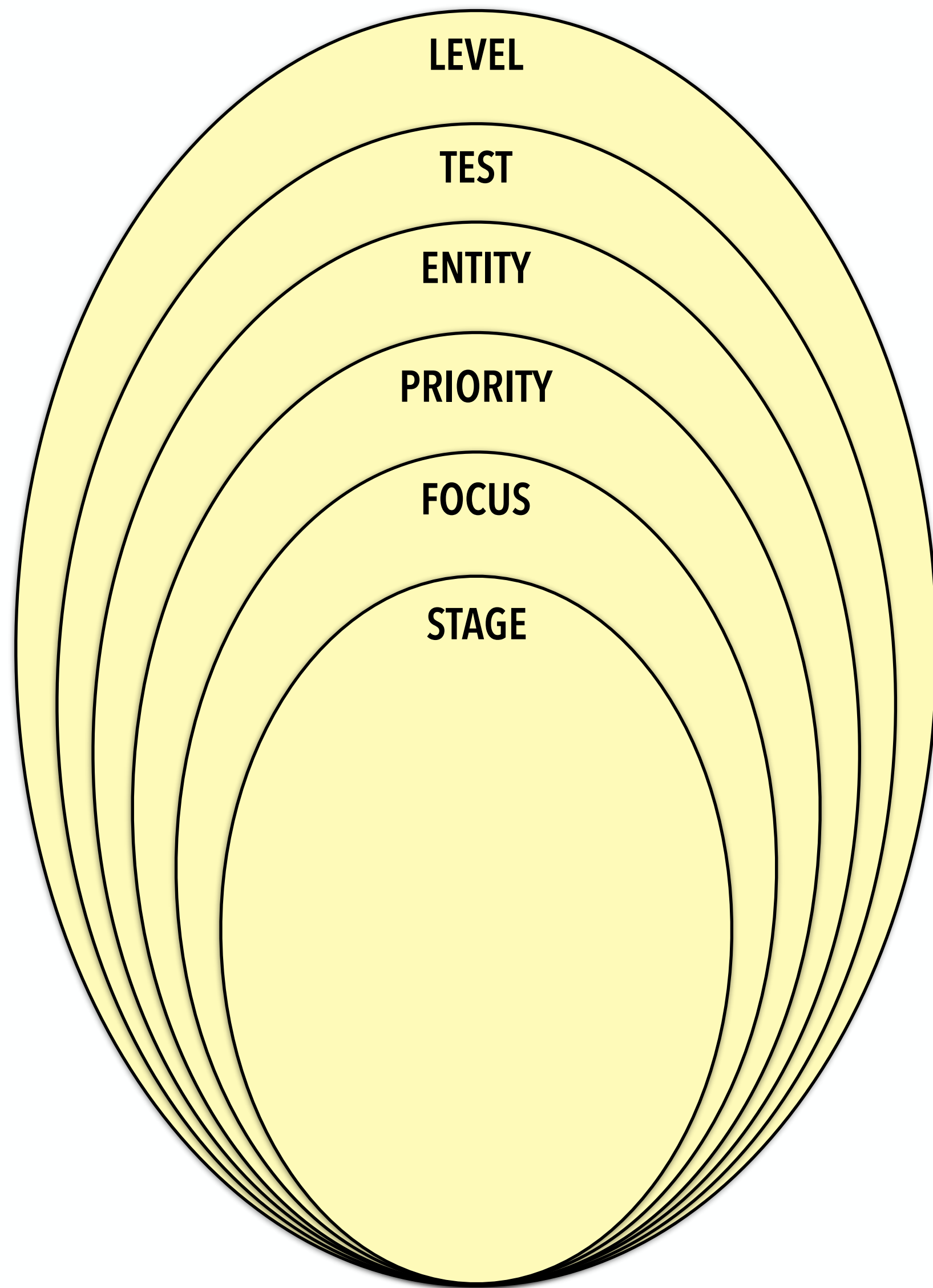
TS

Arranged
by LEVELS,
with ENTITIES
grouped by TEST,
ranked by PRIORITY,



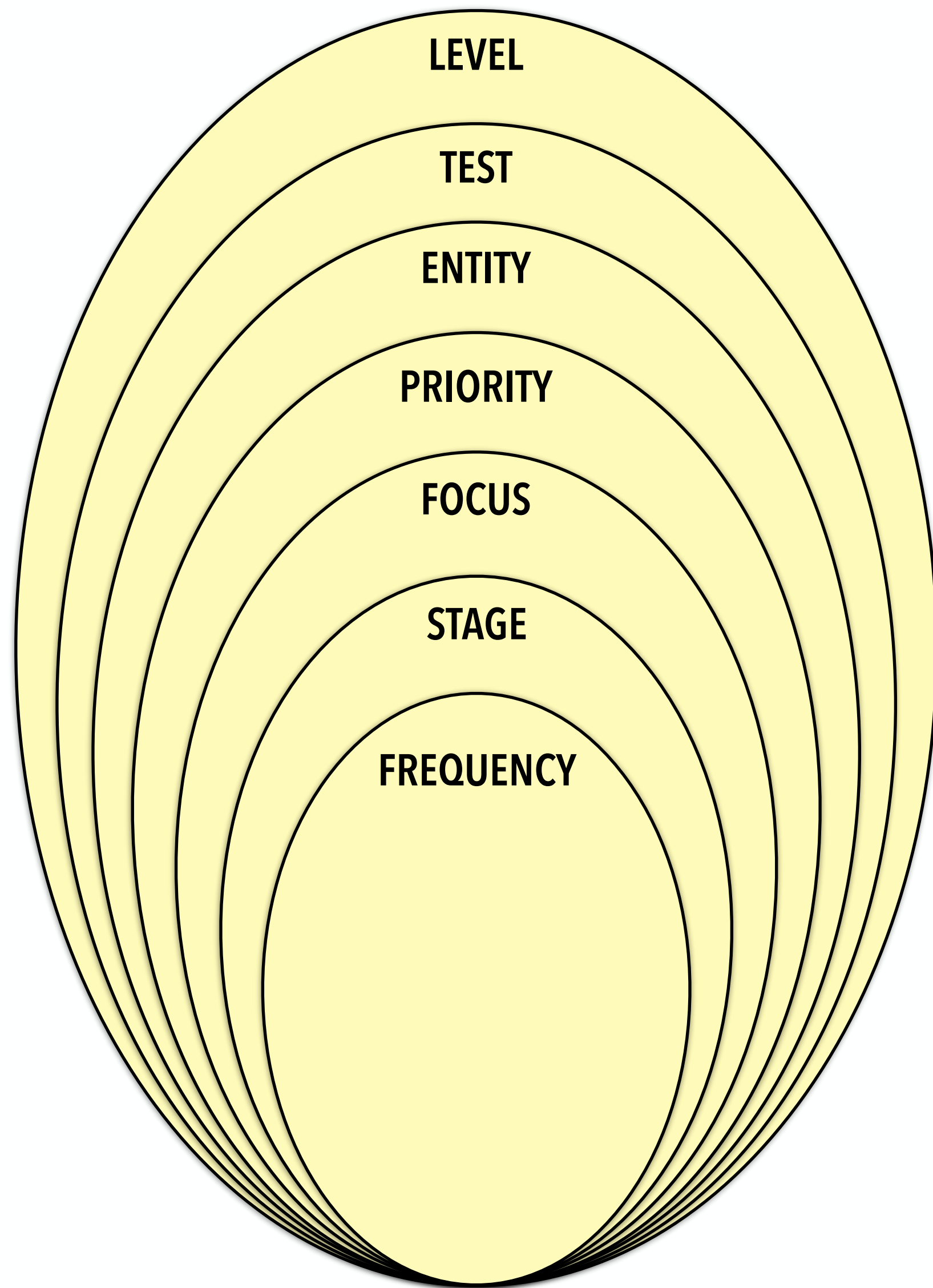
TS

Arranged
by LEVELS,
with ENTITIES
grouped by TEST,
ranked by PRIORITY,
categorised by FOCUS,



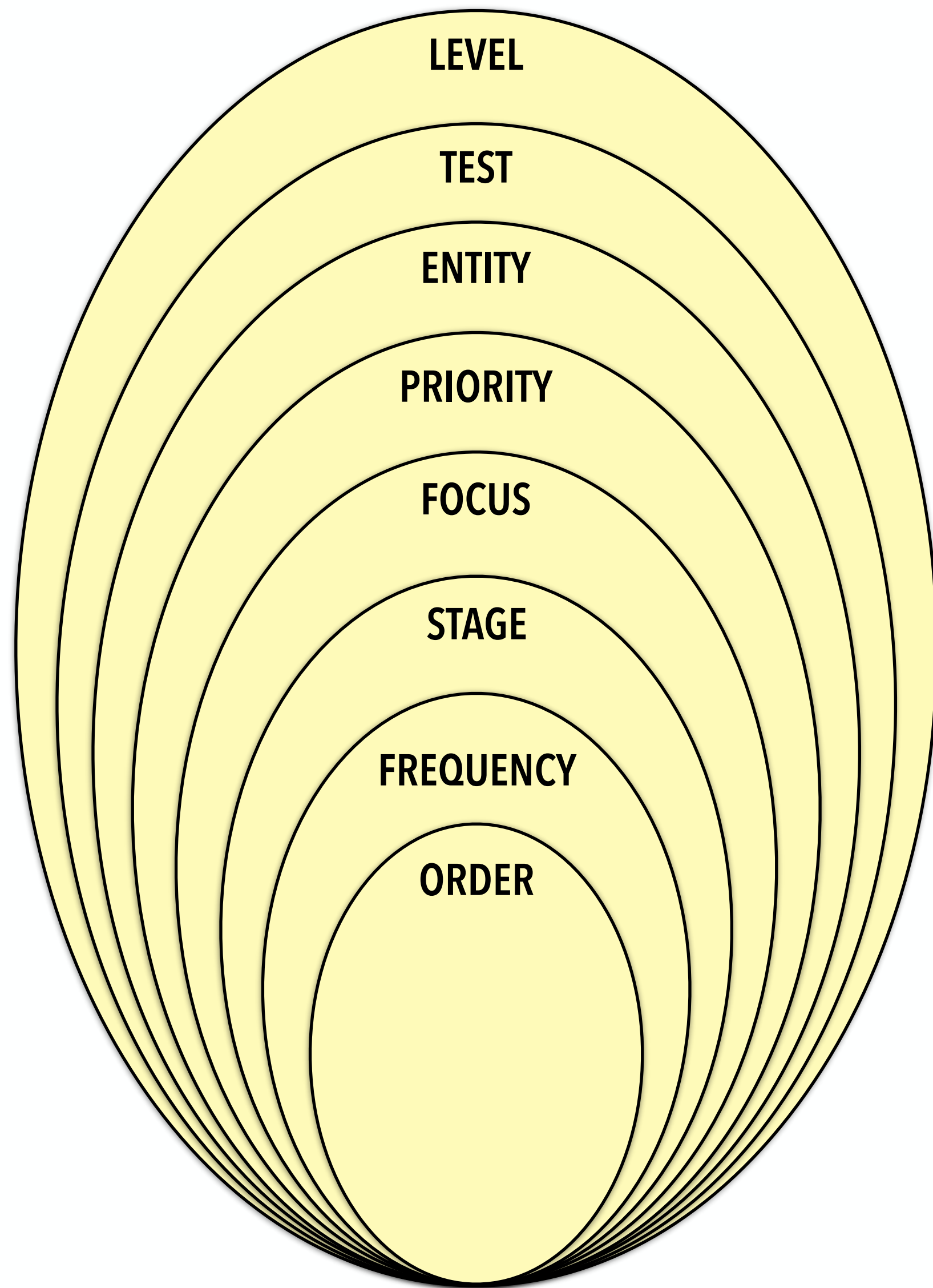
TS

Arranged
by LEVELS,
with ENTITIES
grouped by TEST,
ranked by PRIORITY,
categorised by FOCUS,
classified by evaluation STAGE,



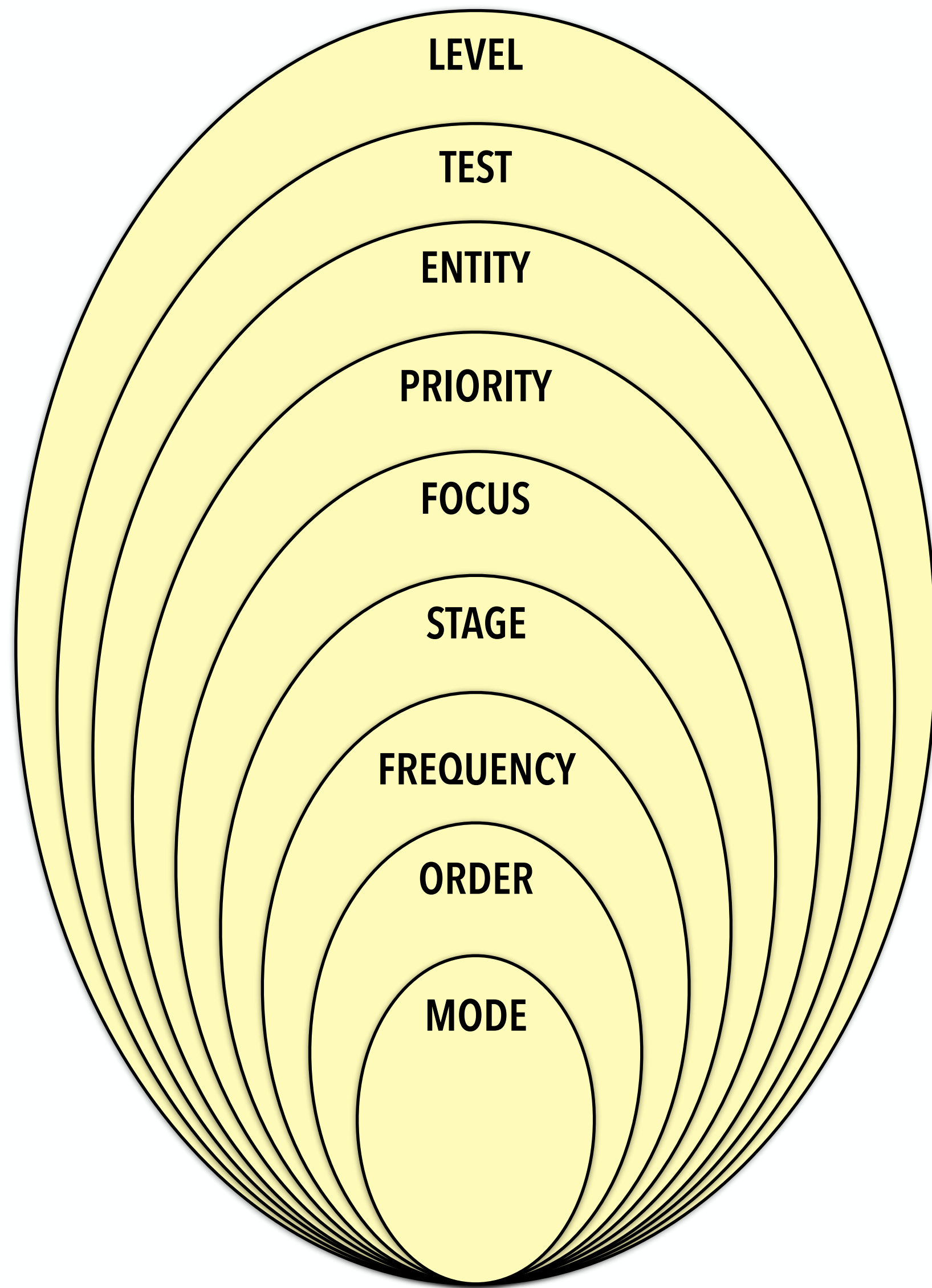
TS

Arranged
by LEVELS,
with ENTITIES
grouped by TEST,
ranked by PRIORITY,
categorised by FOCUS,
classified by evaluation STAGE,
tagged with execution FREQUENCY,



TS

Arranged
by LEVELS,
with ENTITIES
grouped by TEST,
ranked by PRIORITY,
categorised by FOCUS,
classified by evaluation STAGE,
tagged with execution FREQUENCY,
linked by optimal run ORDER,

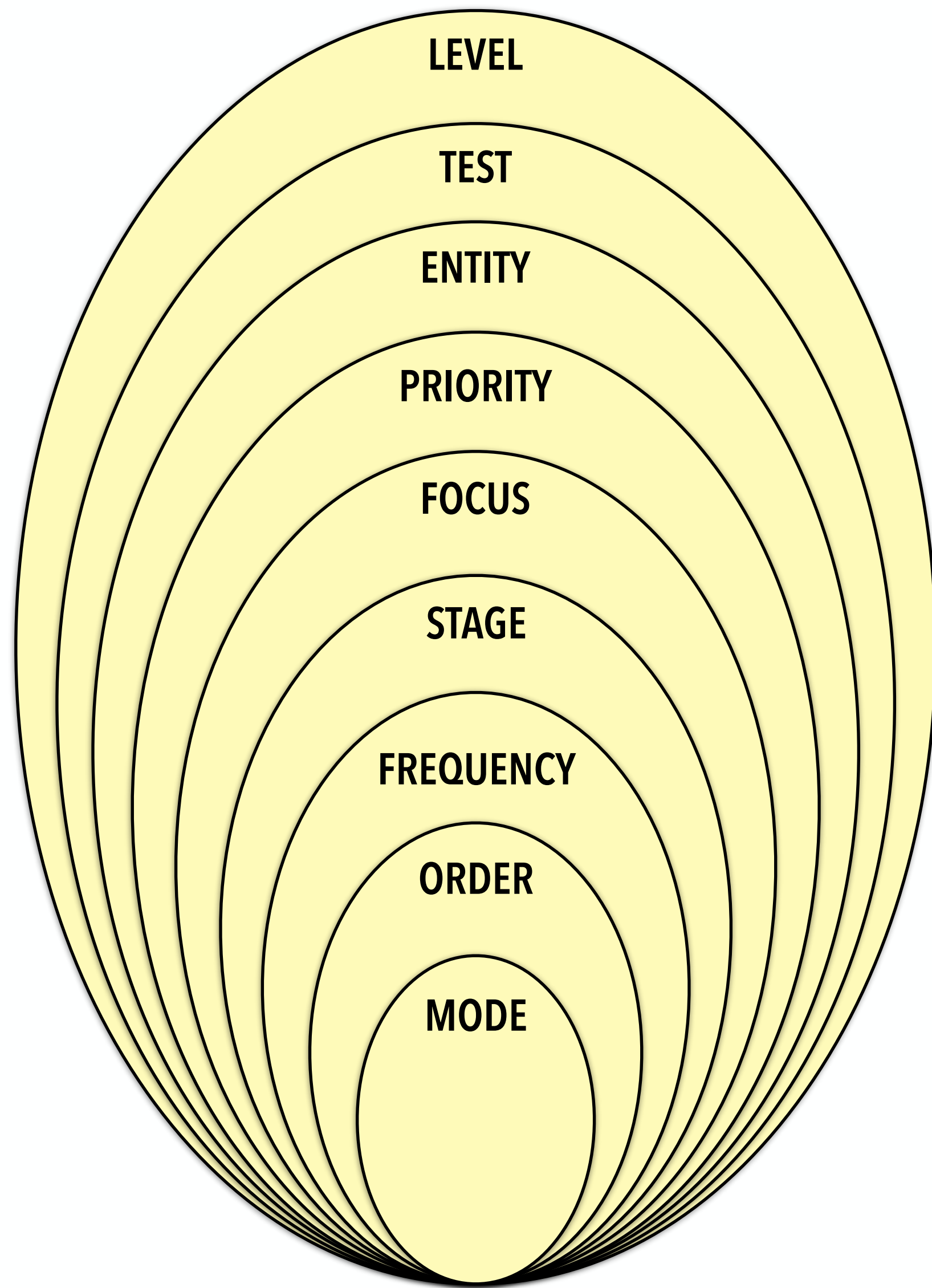


HBT test architecture (stagsoftware.com/hbtcentral)

TS

Arranged
by LEVELS,
with ENTITIES
grouped by TEST,
ranked by PRIORITY,
categorised by FOCUS,
classified by evaluation STAGE,
tagged with execution FREQUENCY,
linked by optimal run ORDER,
collated by execution MODE.

A well architected set of test cases is like a effective
bait that can 'attract' defects in the system.



HBT test architecture (stagsoftware.com/hbtcentral)

See aesthetics in this arrangement?

TS

Arranged
by LEVELS,
with ENTITIES
grouped by TEST,
ranked by PRIORITY,
categorised by FOCUS,
classified by evaluation STAGE,
tagged with execution FREQUENCY,
linked by optimal run ORDER,
collated by execution MODE.

A well architected set of test cases is like a effective
bait that can 'attract' defects in the system.

Test case review

What do you review? What is the objective?

What do you review? What is the objective?

STRUCTURE

IS IT WELL ORGANISED?

A good structure aids clarity making it effective.

Well organised test cases foster easier automation and better test selection.

What do you review? What is the objective?

STRUCTURE

IS IT WELL ORGANISED?

A good structure aids clarity making it effective.

Well organised test cases foster easier automation and better test selection.

CONTENT

IS IT COMPLETE/ADEQUATE?

It is very necessary to ensure that we do NOT miss out test cases.

All applicable tests considered?
Are test cases complete?
Covered all requirements?

What do you review? What is the objective?

STRUCTURE

IS IT WELL ORGANISED?

A good structure aids clarity making it effective.

Well organised test cases foster easier automation and better test selection.

CONTENT

IS IT COMPLETE/ADEQUATE?

It is very necessary to ensure that we do NOT miss out test cases.

All applicable tests considered?
Are test cases complete?
Covered all requirements?

DOCUMENTATION

IS IT CLEAR/WELL-WRITTEN?

Good documentation enables consistency and serves as a good repository for future use.

Clear objective, instructions?
Test data sets clear?

How do you review?

How do you review?

STRUCTURE

IS IT WELL ORGANISED?

Are test cases testing
"A SPECIFIC WHAT-TO-TEST"?

Are test cases testing
"A SPECIFIC TEST-FOR-WHAT"?

Are test cases tagged well?

How do you review?

STRUCTURE

IS IT WELL ORGANISED?

Are test cases testing
"A SPECIFIC WHAT-to-TEST"?

Are test cases testing
"A SPECIFIC TEST-For-WHAT"?

Are test cases tagged well?

CONTENT

IS IT COMPLETE/ADEQUATE?

Are there test cases for each
"WHAT-to-TEST"?

Are there test cases for each
"TEST-for-WHAT"?

Are CONDITION/DATA
combinations good enough?

How do you review?

STRUCTURE

IS IT WELL ORGANISED?

Are test cases testing
"A SPECIFIC WHAT-to-TEST"?

Are test cases testing
"A SPECIFIC TEST-For-WHAT"?

Are test cases tagged well?

CONTENT

IS IT COMPLETE/ADEQUATE?

Are there test cases for each
"WHAT-to-TEST"?

Are there test cases for each
"TEST-for-WHAT"?

Are CONDITION/DATA
combinations good enough?

DOCUMENTATION

IS IT CLEAR/WELL-WRITTEN?

Are prerequisites, test steps, test
data, how-to check outcome clear?

Will it be easy to modify/add
existing/new test cases?

Thank you.



© 2020, STAG Software Pvt Ltd
www.stagsoftware.com

SmartQA