Masterclass for QA
Session #6

"Automation, Metrics,Analysis,Testability & BDD"

© 2020, STAG Software Pvt Ltd

www.stagsoftware.com

TOPICS

Thoughts on automation

Measures/metrics

Analysis & Improvement

Testability

Overview of BDD

Thoughts on Automation

What do we automate?
TEST types

Functional test

Non functional tests

What do we automate?
TEST types

Functional test

Non functional tests

Performance

Security

…

What do we automate?
TEST types

Functional test

Non functional tests

Performance

Security

…

Can only be done if automated

What do we automate?
TEST types

Functional test

- Behaviour
- UI Interface
- Data handling

Segregate scope

Non functional tests

- Performance
- Security
- …

Can only be done if automated

What part of TLC can be automated?

Setup

What part of TLC can be automated?

Setup

Execute

What part of TLC can be automated?

Setup

Execute

Oracle

What part of TLC can be automated?

Setup

Execute

Oracle

Report

What part of TLC can be automated?

Setup	Execute	Oracle	Report
Data creation *create data sets*			

What part of TLC can be automated?

Setup	Execute	Oracle	Report
Data creation *create data sets*			
Prerequisites *setup environment*			

What part of TLC can be automated?

Setup	Execute	Oracle	Report
Data creation *create data sets*	Front-end via CLI, GUI		
Prerequisites *setup environment*			

What part of TLC can be automated?

Setup	Execute	Oracle	Report
Data creation *create data sets*	Front-end via CLI, GUI		
Prerequisites *setup environment*	Back-end via API		

What part of TLC can be automated?

Setup	Execute	Oracle	Report
Data creation *create data sets*	Front-end via CLI, GUI	compare outcomes manually	
Prerequisites *setup environment*	Back-end via API		

What part of TLC can be automated?

Setup	Execute	Oracle	Report
Data creation *create data sets*	Front-end via CLI, GUI	compare outcomes manually	
Prerequisites *setup environment*	Back-end via API	compare outcomes real-time every case	

What part of TLC can be automated?

Setup	Execute	Oracle	Report
Data creation *create data sets*	Front-end via CLI, GUI	compare outcomes manually	
Prerequisites *setup environment*	Back-end via API	compare outcomes real-time every case	
		log outcomes and compare all at end	

What part of TLC can be automated?

Setup	Execute	Oracle	Report
Data creation *create data sets*	Front-end via CLI, GUI	compare outcomes manually	exploit framework reporting features
Prerequisites *setup environment*	Back-end via API	compare outcomes real-time every case	
		log outcomes and compare all at end	

To script rapidly & maintain them...

Levelise test scenario to keep it short and purposeful.

To script rapidly & maintain them…

Levelise test scenario to keep it short and purposeful.

Don’t attempt to do a lot in a script.

Good practices

Good practices

Parametrise environment info to shift environment easily

No magic data in script.

Good practices

Parametrise environment info to shift environment easily
No magic data in script.

Keep config/test data files for each environment
in a structured manner.

Good practices

Parametrise environment info to shift environment easily
No magic data in script.

Keep config/test data files for each environment
in a structured manner.

Data driven,
no data in script code.

Good practices

Parametrise environment info to shift environment easily
No magic data in script.

Keep config/test data files for each environment
in a structured manner.

Data driven,
no data in script code.

Individual logs per run, instead of giant log file.
Structured log info for easy search & analysis.

Good practices

Parametrise environment info to shift environment easily
No magic data in script.

Keep config/test data files for each environment
in a structured manner.

Data driven,
no data in script code.

Individual logs per run, instead of giant log file.
Structured log info for easy search & analysis.

Note human execution can meander, not automated script.
Test steps need to be automation friendly.

What to automate - Heuristics

Frequently executed scenarios

smoke, sanity

standard regression

Build test (dev)

What to automate - Heuristics

Frequently executed scenarios
smoke, sanity
standard regression
Build test (dev)

Non-functional tests
some of these like load, performance
cannot be done manually.

What to automate - Heuristics

Frequently executed scenarios
smoke, sanity
standard regression
Build test (dev)

Non-functional tests
some of these like load, performance
cannot be done manually.

Time consuming
those that are complex
those that are error prone
end-to-end flow test

In sprint vs. Post sprint automation

Do we script at a later sprint when an entity is complete and stable?

or

Do we automate in the current sprint and refactor subsequently?

Do we catch up with automation?

or

Do we keep it current?

Automation velocity needs to match with dev velocity.

Closing thoughts.

Note automation is checking i.e. compliance.
Automation does not replace intelligent human testing.

Closing thoughts.

Note automation is checking i.e. compliance.
Automation does not replace intelligent human testing.

Script needs to kept in sync as system evolves.

Closing thoughts.

Note automation is checking i.e. compliance.
Automation does not replace intelligent human testing.

Script needs to kept in sync as system evolves.

Automation as being integral to testing than ‘add later’
fosters rapid dev with quality.

Discussion #1

“Measures/Metrics”

What do you measure?
What does it help in doing?
How does it help you improve?

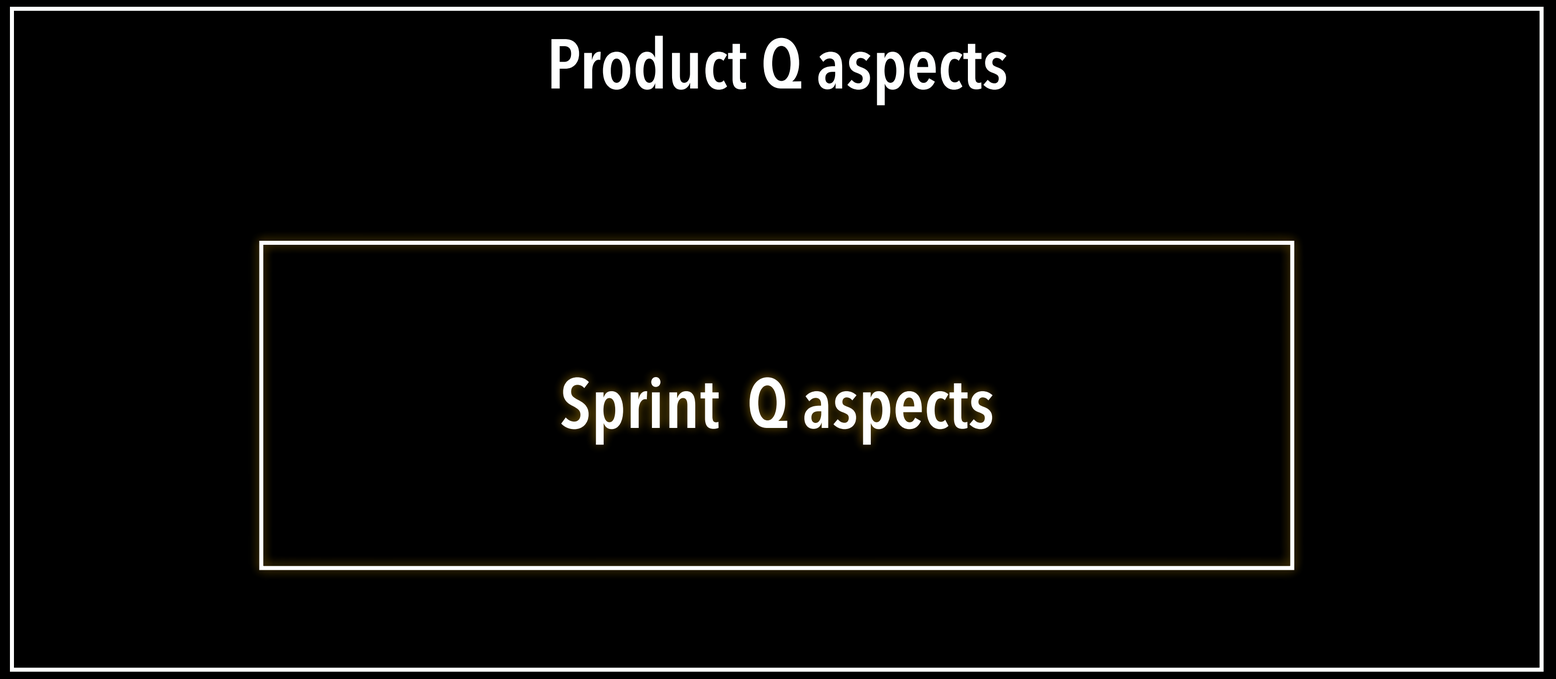
Your comments/thoughts please.

Measures/metrics

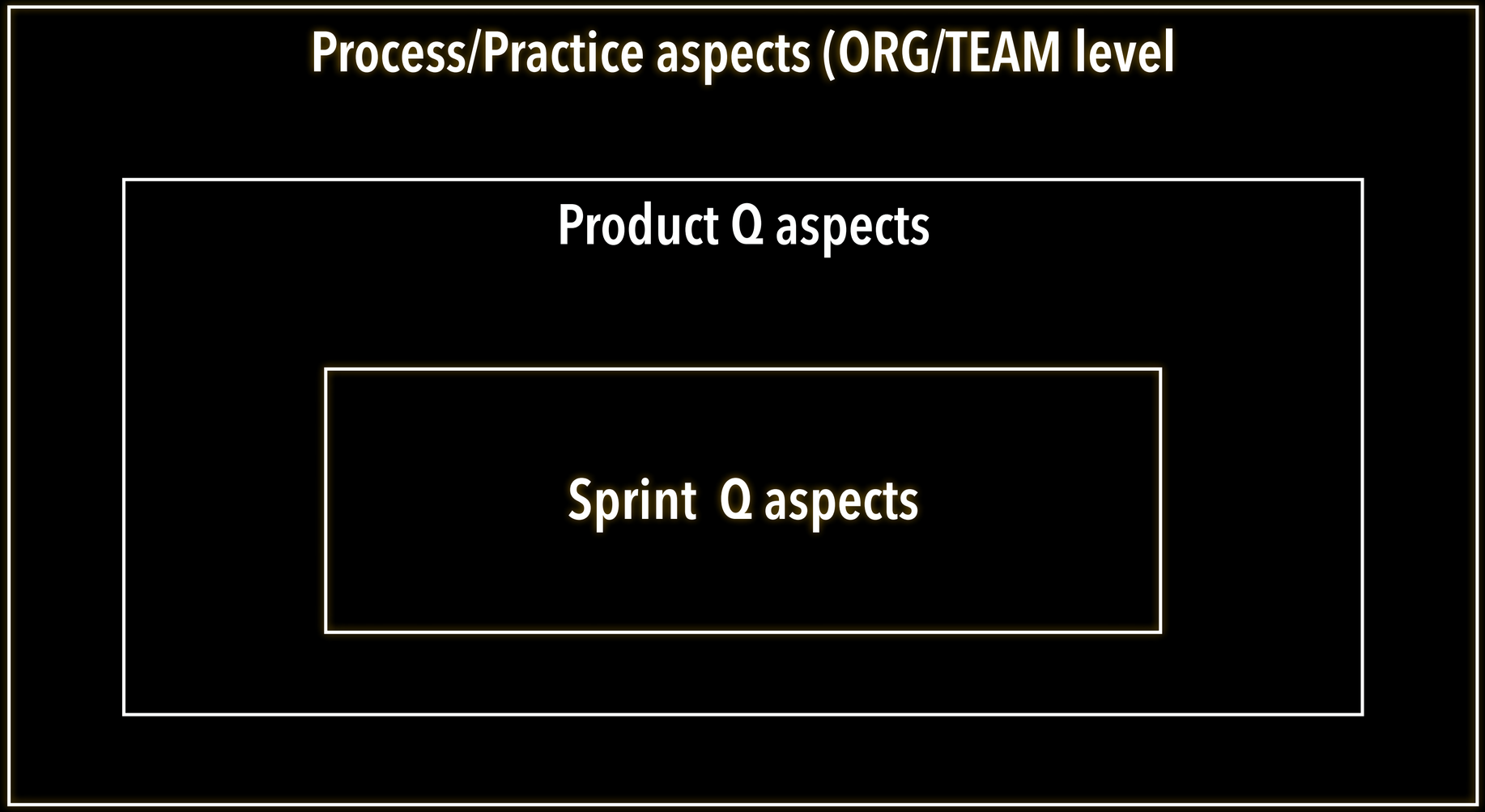
Measures of?

Sprint Q aspects

Measures of?



Measures of?


Process/Practice aspects (ORG/TEAM level
Product Q aspects
Sprint Q aspects

Measure what?

Measure what?

Design effectiveness

Measure what?

Design effectiveness

Validation progress

Measure what?

Design effectiveness

Validation progress

Product quality

Measure what?

Design effectiveness

Validation progress

Product quality

Process/Practice

Measure what?

Measure what?

Design effectiveness
at Sprint/Product Distribution of TS/TC by EUT, Levels, Types, +:-

Measure what?

Design effectiveness
at Sprint/Product Distribution of TS/TC by EUT, Levels, Types, +:-

Validation progress
by EUT by Test type by Level by Test/Re-test

Measure what?

Design effectiveness	Validation progress	Product quality
at Sprint/Product Distribution of TS/TC by EUT, Levels, Types, +:-	by EUT by Test type by Level by Test/Re-test	Sprint & Product Quality by Levels, EUT, conformance/robustness

Measure what?

Design effectiveness	Validation progress	Product quality
at Sprint/Product Distribution of TS/TC by EUT, Levels, Types, +:-	by EUT by Test type by Level by Test/Re-test	Sprint & Product Quality by Levels, EUT, conformance/robustness

Process/Practice
Efficiency/Productivity - automated/manual execution, yield Effectiveness - escapes, defects via scripted/unscripted, prevent:detect Reusability - Checklists, Scripts, Frameworks, Data sets

Analysis & Improvement

First understand issues before figuring out how to address them.

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.

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

on Testability

Software testability is:

Software testability is:

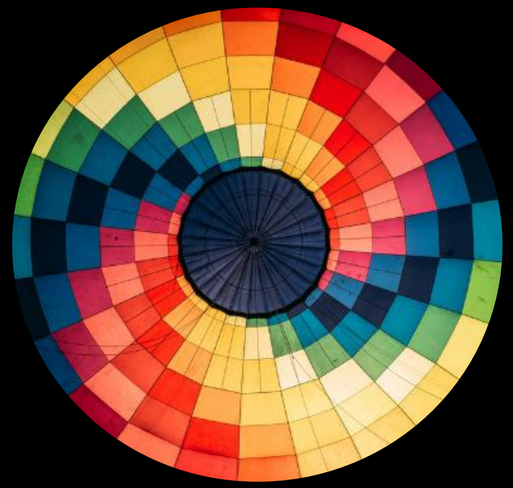
the degree to which a software artefact
(i.e. SW system/module, requirements /design document)
supports testing in a given test context.

Software testability is:

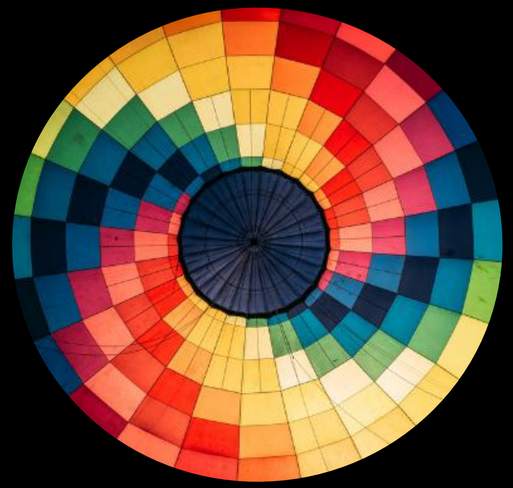
the degree to which a software artefact
(i.e. SW system/module, requirements /design document)
supports testing in a given test context.

If testability of a software artefact is high,
then finding faults in the system by testing is easier.

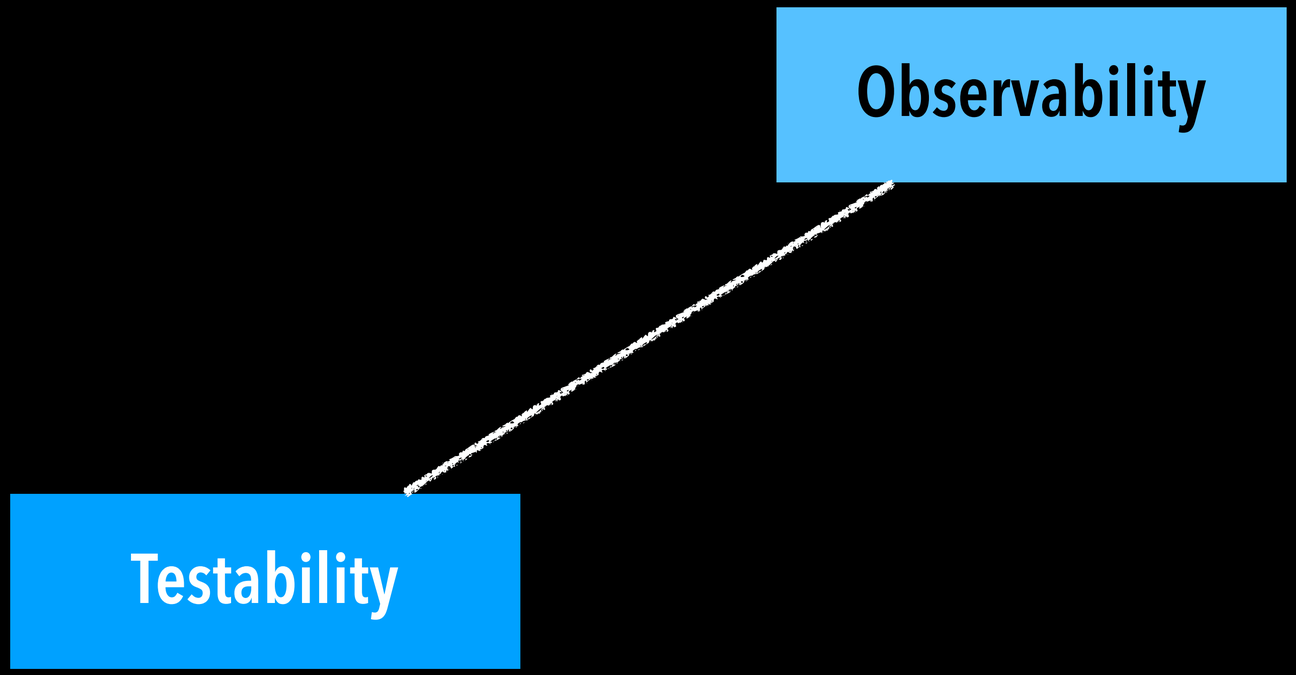
At the most basic level,
testability is about how easy software is to test.
Another way to look at it is, how likely testing will expose application faults.



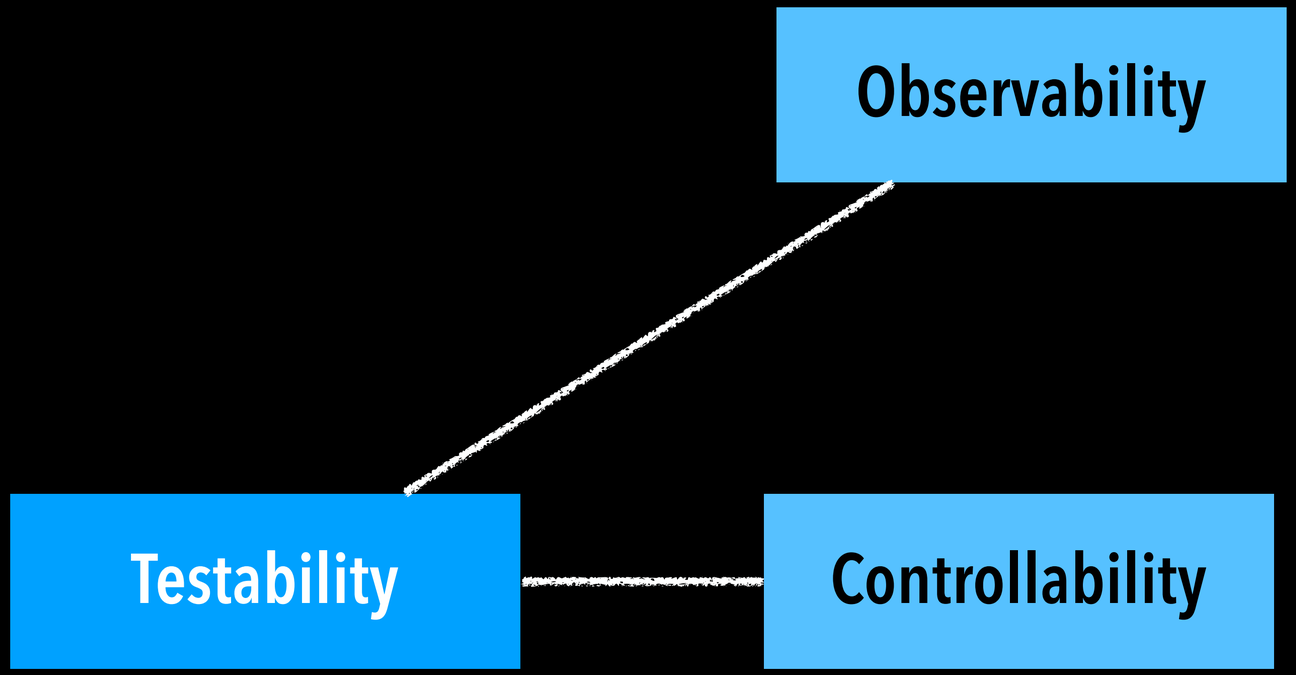
At the most basic level,
testability is about how easy software is to test.
Another way to look at it is, how likely testing will expose application faults.



Testability is a product of effective communication
between product, dev, product & testing teams.
More the ability to test is considered when creating a feature,
more the team members ask for test inputs making testing effective.

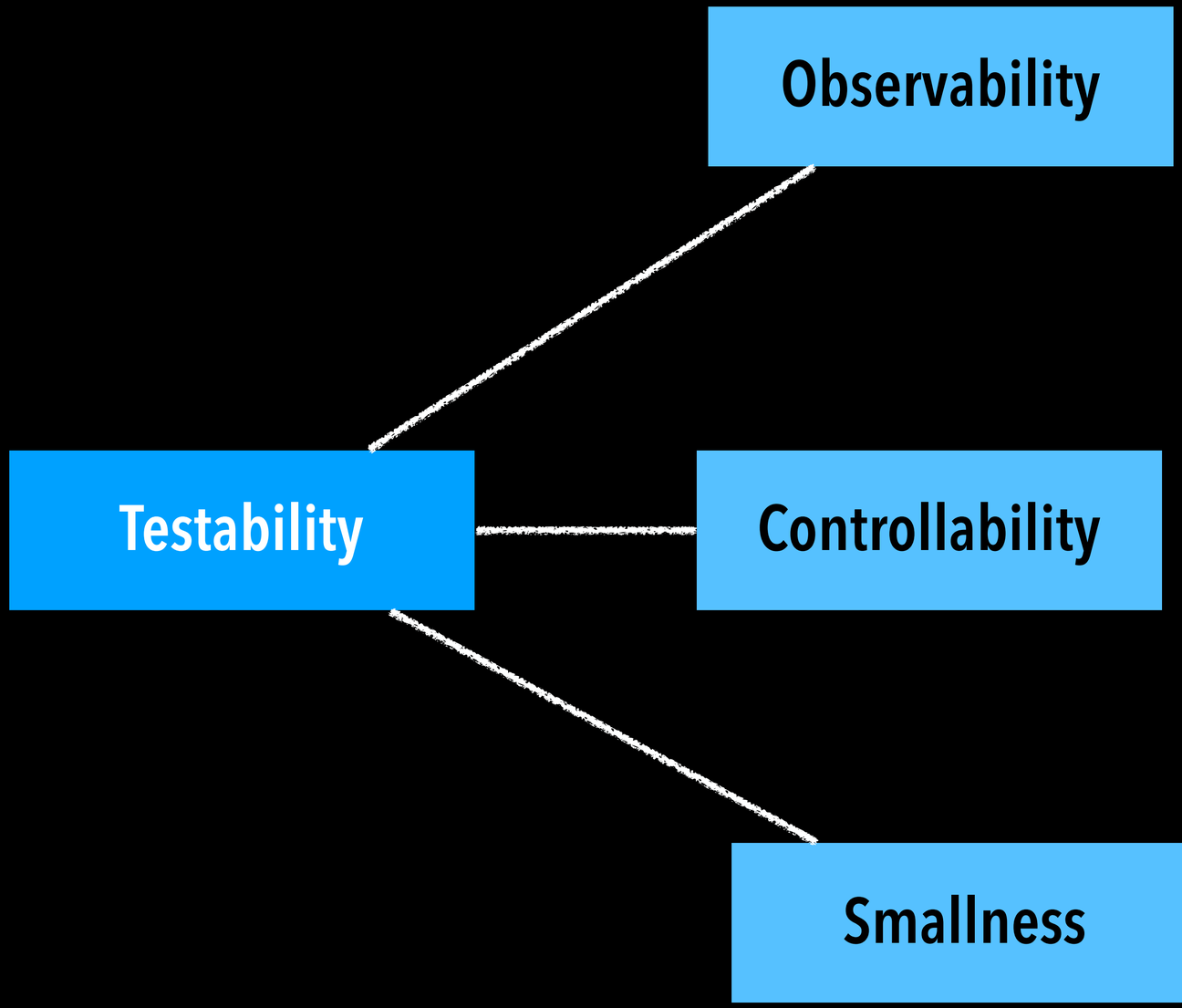


the intrusiveness to observe better



the intrusiveness to observe better

ability to set system into state
inject inputs to stimulate system
facilitate easy deployment to test in the large



the intrusiveness to observe better

ability to set system into state
inject inputs to stimulate system
facilitate easy deployment to test in the large

smaller the SW, lesser to test
singularity of code
abstraction & reuse

An overview of
Behaviour Driven Development (BDD)

An user story is seen as a modern way of communicating the end user's needs and expectations in a sweet and simple format that can be easily modified.

This brevity/simplicity hides information leading to understanding in the small and potentially missing the big picture.

What is BDD?

It's using examples to talk through how an application behaves...
And having conversations about those examples.

Source: Wikipedia

BDD specifies that business analysts and developers should collaborate in this area and should specify behavior in terms of user stories, which are each explicitly written down in a dedicated document.
Each User Story should, in some way, follow the following structure:

Source: Wikipedia

BDD specifies that business analysts and developers should collaborate in this area and should specify behavior in terms of user stories, which are each explicitly written down in a dedicated document.
Each User Story should, in some way, follow the following structure:

Title
An explicit title.

Narrative
A short introductory section with the following structure:
- **As a**: the person or role who will benefit from the feature;
- **I want**: the feature;
- **so that**: the benefit or value of the feature.

Acceptance criteria
A description of each specific scenario of the narrative with the following structure:
- **Given**: the initial context at the beginning of the scenario, in one or more clauses;
- **When**: the event that triggers the scenario;
- **Then**: the expected outcome, in one or more clauses.

Source: Wikipedia

Title: Returns and exchanges go to inventory.

As a store owner,
I want to add items back to inventory when they are returned or exchanged,
so that I can track inventory.

Scenario 1: Items returned for refund should be added to inventory.
Given that a customer previously bought a black sweater from me
and I have three black sweaters in inventory,
when they return the black sweater for a refund,
then I should have four black sweaters in inventory.

Scenario 2: Exchanged items should be returned to inventory.
Given that a customer previously bought a blue garment from me
and I have two blue garments in inventory
and three black garments in inventory,
when they exchange the blue garment for a black garment,
then I should have three blue garments in inventory
and two black garments in inventory.

Source: Wikipedia

Gherkin, Cucumber

Cucumber is a software tool that supports behavior-driven development .
Central to the Cucumber BDD approach is its ordinary language parser called Gherkin.
It allows expected software behaviors to be specified in a logical language that customers can understand.

Gherkin, Cucumber

Cucumber is a software tool that supports behavior-driven development .
Central to the Cucumber BDD approach is its ordinary language parser called Gherkin.
It allows expected software behaviors to be specified in a logical language that customers can understand.

Gherkin is the language that Cucumber uses to define test cases.
Designed to be non-technical & human readable, it describes use cases relating to a software system

Cucumber tests are divided into individual FEATURES.
FEATURES are subdivided into SCENARIOS, which are sequences of STEPS.
Each FEATURE is made of a collection of scenarios.

Scenario: Eric wants to withdraw money from his bank account at an ATM
Given Eric has a valid Credit or Debit card
And his account balance is $100
When he inserts his card
And withdraws $45
Then the ATM should return $45
And his account balance is $55

Cucumber tests are divided into individual FEATURES.
FEATURES are subdivided into SCENARIOS, which are sequences of STEPS.
Each FEATURE is made of a collection of scenarios.

Scenario: Eric wants to withdraw money from his bank account at an ATM
Given Eric has a valid Credit or Debit card
And his account balance is $100
When he inserts his card
And withdraws $45
Then the ATM should return $45
And his account balance is $55

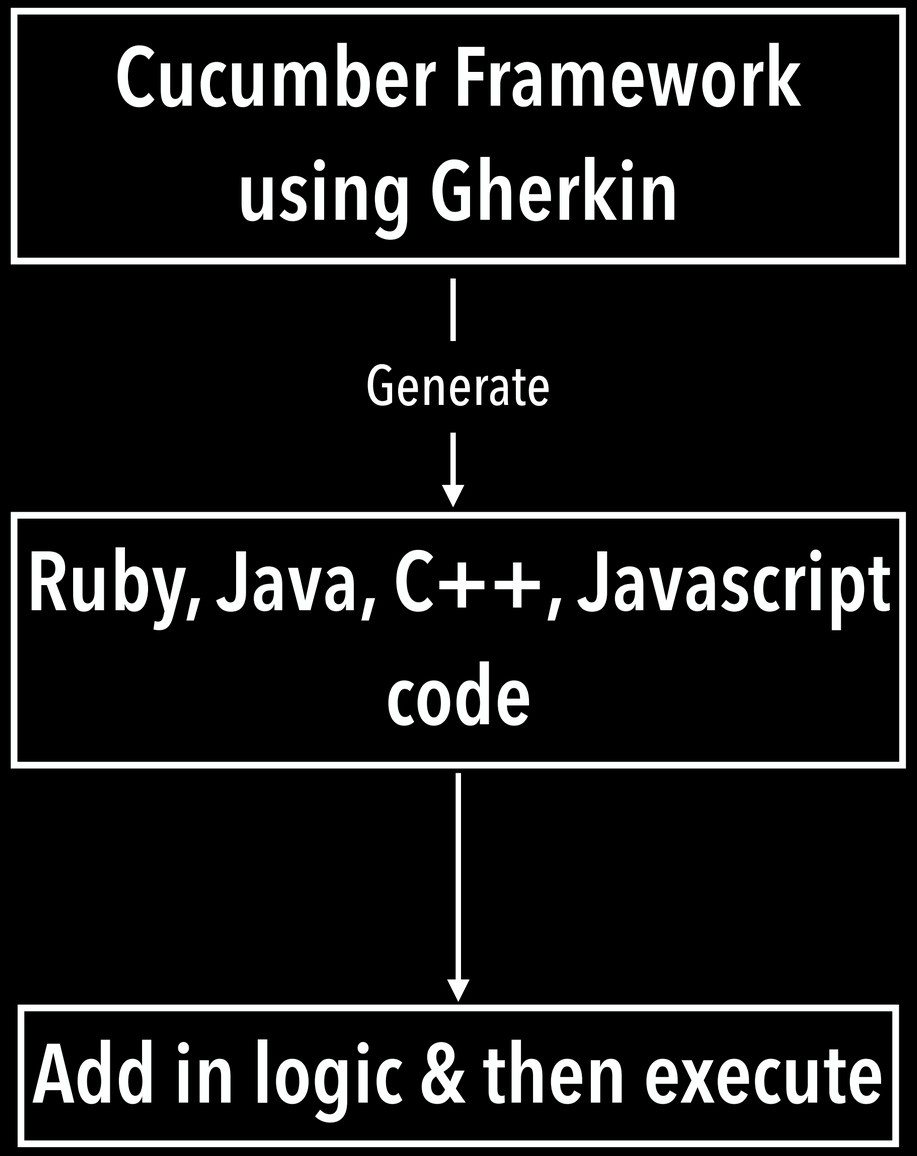
A **Scenario Outline** provides a technique to specify multiple examples to test against a template scenario by using placeholders.

Scenario Outline: A user withdraws money from an ATM
Given <Name> has a valid Credit or Debit card
And their account balance is <OriginalBalance>
When they insert their card
And withdraw <WithdrawalAmount>
Then the ATM should return <WithdrawalAmount>
And their account balance is <NewBalance>

Examples:

Name	OriginalBalance	WithdrawalAmount	NewBalance
Eric	100	45	55
Gaurav	100	40	60
Ed	1000	200	800

Automation in BDD



Thank you.

© 2020, STAG Software Pvt Ltd

www.stagsoftware.com