How scenarios are used to generate automation test script

1. Writing feature file(contains multiple scenarios)
2. Run the feature file. Get the step def skeleton
3. Copy those step def into your Java class
4. Implement methods till all scenarios are passed

In dev to implement step def then code needs to developed too. During dev there will be a cycle of TDD.

As a tester, you write step def and test those (test application using step def)

Page object?? 🡪 fits in step definition??

Challenges In maintaining feature files and scenarios

Page object design pattern

Symptoms - Problems

Some of our tests fail randomly-they fail without any reason and when they fail we cannot find out the reason easily as they are random – Flickering scenarios/flaky tests

We keep breaking tests unintentionally -something has been modified but something else is failing– Brittle features

Our features take too long to run – features become slow and slowness increases when adding more test cases – this is a challenge for maintenance– Slow features

Our stakeholders don’t read our features – stakeholders not interested-Bored stakeholders

Flickering scenarios

* Sometimes pass and sometimes fail
* Reproducing is difficult
* Leaky scenarios, shared environment

Brittle features

* Unrelated scenarios start breaking because of changes in code base
* They can be caused by fixture data, duplication, leaky scenarios,

Objective of Automation – reduces man power effort, identifying errors quicker,g ives faster feedback

Slow features

* Features take longer to run

Uninterested stake holders

* Lack of collaboration b/w business and technical teams
* Imperative steps, duplication, siloed features, lack of ubiquitous language (not readable by non-technical users)

How to avoid incidental scenarios – do scenario refactoring- you want your scenario to read better

Always let PA create a 1first draft which is in domain lang then testers and devs ask questions

Avoid Imperative Steps – sequence of commands giving what and how like in Java

Too many imperative steps lead to brittle tests

Make it Declarative – is what is to be done and not how