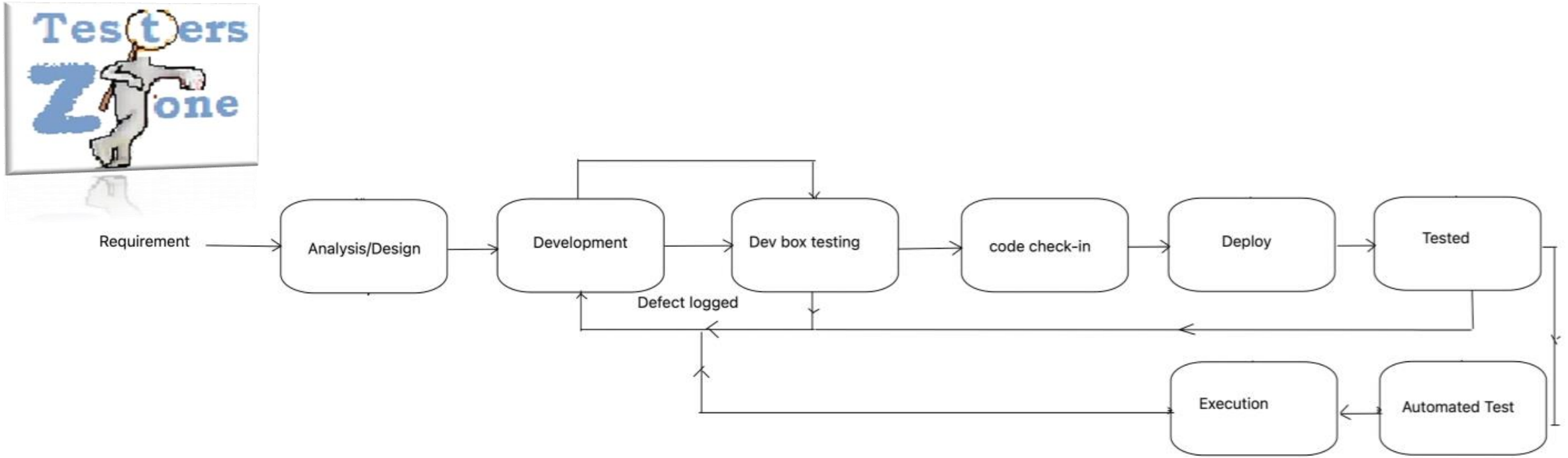
Dev box testing

Testers Zone

By Mithilesh Singh



What?

- Dev Box is an abridged version of ‘Developer Machine’. In Dev Box testing, a quality engineer validates, tests, and verifies a feature in scope on the developer’s machine. It’s as simple as that. A basic approach through which one ensures the quality of a feature.

How?

- The process of successful Dev Box testing starts early in the development cycle, when the feature is being discussed between the Product, Dev, and QA teams. While a story is being groomed, a quality engineer ensures the following are in place:
- Acceptance criteria.
- In scope and out of scope features.
- Pre-conditions, if any
- Once the above parameters are concluded, the dev team can start working on it. After the development is finished, the SDE will call out for dev box testing.

Steps:

- The quality engineer reviews the story and checks the preconditions.
- Ask the developer to show the workflow to ensure the required functionality is working
- Review the unit, functional, integration tests and execute them.
- Ask questions to the developer

  - When would this fail in production? What would be possible scenarios?
  - Would this perform as expected under performance-stress-load tests?
  - Are there any known issues with the code, module, or integrations?
  - What edge cases have they handled in the implementation?
  - Is there a better way to implement the same code?
  - Are there any platform or hardware constraints?
  - Questions around scenarios of localization, etc.

Note:

The questions would depend on what type of story you're testing, for example: front-end or back-end stories.

While going through the above points, note the details such as discussion points, issues found, suggested changes, action items, and share them as minutes of meeting.

This would ensure the required updates are made. If no issues are found, the developer can then merge the code to the staging branch where the quality engineer would execute functional tests.

Benefits?

- Technical understanding of the product by the quality engineer
- Upskilling quality engineer
- provides faster feedback to the team
- Improve vertical testing
- Improve cross-team communication and build trust

Drawbacks?

It may take time to get each team member in sync

If the team is not following the TDD approach, it may prove difficult or futile

The team should consist of at least two quality engineers (**Dev Box is quality-centric, hence, would require enough QEs to accommodate all tasks**)

End to End Testing Steps

- The team will be there in the sprint backlog grooming meeting and decide the Stories that are all can pick in the upcoming sprint.
- Dev and Tester will be in sync during Scope analysis and they will divide the feature into testable components.
- The developer will be creating the Jira task based on those testable components[backend and front end development tasks].
- Testers will be adding all those testable components in the Google sheet with sprint and JIRA task details. he/shill will also mention ETA for every task/subtask for their reference.
- The designer will share the Figma for that feature and that will be the road map for the testers and developer for further activity.
- The developer will be developing the feature as per the timeline mentioned in the google sheet and the tester meanwhile writing the test cases for that feature.

End to End Testing Steps

Note: Tester can draft all the cases at once no matter which component is going to come first for the testing but make sure test case writing is done before receiving any build.

Once the developer developed the component he/she can call tester for dev box testing, testers can follow slide 4 and 5 to perform the dev box testing.

After dev box Build will be pushed to the respective env--> netlify or staging.

The tester should verify Test environment details [Testing portal link, test data, .apk, release notes, dependent areas list, bug fixes list, JIRA ticket etc.]before starting the test execution.

The tester will start testing the component based on the written test case and update the test case status in the test case sheet.

If Testers encounter any bug, he/she will report it to the respective dev and add the JIRA in front of the test case.

End to End Testing Steps

This development and testing process with bug life cycle will be continued and after executing all the cases we will have execution results for all the cases.

There are possibilities to get some Adhoc requests in between for the same developing components or some already developed feature also.

if there is some enhancement needed in between in current feature, the developer has to convey their QA regarding this so that he/she can modify test cases accordingly and till the time those changes will reflect tester will have a case ready for testing

- **Older feature Adhoc request:** if this is on priority and we cannot put on hold, Analyse and decide whether it is non-functional changes like UI/UX or functional changes. for UI/UX changes we can skip test case writing and direct update the JIRA ticket based on activity performed but if there are functional changes try to add the test case first, that kind of changes will not contain more cases so will hardly take 30-40 mins time to draft/modify the cases.

End to End Testing Steps

Note: dev box testing will help to avoid major bugs in staging or pre prod and dev will have enough time to fix it if any

01

Once Development and testing are done, the tester will add the deliverables in test plan with documents like test case sheet, execution status etc.

02

In the test plan, the only tester can mark the sign off date and status and share with the dev and team lead.

03

Note: Tester who does not follow or not aware the Dev box testing can directly perform deep testing on staging or preprod but dev might get less time to fix logged bugs and sprint will get delayed.


Thank You