Giới thiệu

As you know. I have been working for FPT software in the position of full stack qa since 2017. So I have more than 5 years of experience in both manual test and automation test. My projects run on agile scrum model and use the azure devops system for test management. In latest project, I have managed 4 members. I do plan and estimation, then assign task to team member, review TC, test script and mornitoring member progress. About automation technical, I am familiar with using Katalon, Selenium tool along with BDD cucumber framework, POM. I have experience in setup CI/CD for automation in my project. I used to work with api, database as well.

I think that’s a little bit about myself, thanks for listening and looking forward to answering your question

SCRUM

1.  Agile/Scrum is a development model that

+ break many sprints, each sprint repeats actions as: planing, Analysis and design, execution, testing and reporting

+ Time of sprint: about 2-3 weeks , Usually it is 2-3 weeks

+ Role: development team(BA, dev, test, stakeholder), scrum master, Product Owner

**+ There are some types of meeting:**

1. Refinement Meeting: product backlog items need refinement for the next Sprint, to make the team understand them better for successful execution.

2. Sprint Planning: decide the amount of work

3. Daily Scrum: the team gathers together to report any issues and  progress on their tasks

4. Sprint Review:  a live demonstration of the work completed.  Product Owner, Scrum Master and stakeholders are present to review the product and suggest changes or improvements.

5. Sprint Retrospective:  The team speaks openly about their organizational concerns and teamwork

TEST PROCESS

**process of testing:**  5 levels:

•     Planning and control

•     Analysis and design

•     Implementation and execution

•     Evaluating exit criteria and reporting

•     Test closure activities

Test Activities

First I study requirement. If something is not clear I will ask a question for BA

After everything in requirement is clear, I start writing test design and peer review test design with my team member.

Next I will create test plan on ADO, and create test suite base on user story to write detailed test case on it.

After the developer changes the status of the user story to ready to test, I start executing the test and mark the test result to the test plan. During test execution, if I find a bug. I will confirm and log the bug on system.

Actually, I will inform to developer list of bugs and monitoring status of bugs

LEADING/Manage team

I used to lead some project with around 4 members. My main task are setup my team, make testing plan and estimation, assign task to team member, mornitor working progress and review output of member such as Test Case, test script, test result, report and support in problem solving for member. Also, I need to define risk and mitigation. Besides, I need to give technical advice to members and training for new member

Test Type

In previous projects, I usually do the main test types like system test, regression test, smoke test, performance test

- Smoke test: we do the smoke test whenever we have the new build and run the smoke test to check the main function for whole app to make sure that the app is working fine after the new build

We also do smoke test every time we deploy to production to check if the main function is working fine after deployment and we can check further if necessary

- RT: normally, we do RT when we have request to deploy to production and test whole function to make sure the system work normally when the code change

- Performance:

+ do when having request from customer

+ usually, do when we do when finish all functional test and we want to deploy to production

+ we need to run PT to measure that the app can load how many concurrent user in the period time, st like this

- System test/functional test: Do ST after front end & backend have been integrated and ready to work as a whole system, Usually do ST in each sprint

TEST LEVEL

**Unit testing**: is a type of software testing where individual units or components of a software are tested   
-> purpose: to validate that each unit of the software code perform as expected (method, module, object,..)

- **Integration testing**: tests interface between components, interactions to diffident parts of a system perform after component testing   
-> purpose: to find defects in the interaction between modules when they are integrated

- **System testing**: is a level of testing that validates the complete and fully integrated software product

-> purpose: to evaluate the end-to-end system requirements

- **Acceptance testing**: to evaluate that the system fulfill all the contractual requirement or not

ISSUE

I had a few issues in the project.

For example, the bug is reopened too many times. In this case I will review the information in the bug first to make sure the bug is clear and easy to understand. Then I can call directly the developer to clear the actual and expected result of that bug

The second issue I often encounter is too many user stories in a sprint making. At this point, I will discuss with the whole team to consider keeping the urgent us, and the unnecessary us can be moved to the next sprint. Because if we don't have enough time to test, we can't guarantee the quality

RISK & MITIGATION

Risk: Many User Stories might be moved to the current sprint and it takes times for developer to deploy for testing all of User Stories in just one time

mitigation: Dev team separate in to 2 builds and select which User Story should be built in " Build 1" or "Build 2"

ESTIMATE

1. Estimate time of testing based on  productivity in create/ execute test case of each member in the team

2. Estimate story point base on the complexity of User Story, follow to Fibonacci number(1, 3, 5, 7 and so on) and 1 story point equals to 4 hours

3. Estimation base on historical data: can base on the similar project with similar function or st like this

Plan Test

+ Timeline

+ US

+ Test Level

+ Resources

+ Environment

+ Risks and Mitigation

+ Issue

+ Test result

TEST PLAN

**Master Test Plan**: Scope of Testing, Testing Type, Test Criteria (Test Entry Criteria, Test Exit Criteria), Risk and mitigation, Resources, Test Environment, Feature and Schedule.

**Sprint Test Plan**: Sprint number, Sprint timeline, User Story, Human resource, Browser, Schedule, risk and mitigation, issue and solution, test result

The Master Plan is common for all sprints, while the sprint plan will only cover each specified sprint

DOCUMENT

We have a lot of testing documents such as: test plan, test design, test case, UAT script, User guideline, testing daily report, weekly report, deployment test report, test close report

UAT – End User

I usually join UAT session to support end user

Sometimes I need to connect with the end user to understand and reproduce the production issue if any

REPORTING

Regarding report testing activites, I usually do: daily report, weekly report, closure report, report on demand any time to team

In each report, I also need test results including the number of test cases passed, the number of failed tests, the number of test cases not tested yet and a list of the bugs along with status, severiy, priprity of bugs.

I need to note more reasons why those bugs have not been fixed

CLOSURE REPORT

Yes, I have experience making test closure reports

Purpose of test closure report is Describe the testing activities performed by QA team

I create test closure report when the testing phase is completed successfully (sprint/release completed)

I think all team members who are interested in how testing happened, future QAs who take over the testing

about Main content in the test closure are Features to test, Test types conducted, Test results, Bug list with status, Known Issues and Notes

RELEASE NOTE

Time (what sprint), list of released functions, links of test plan, related link of dev team

TC - Manage

I manage test cases on azure devops. Every sprint I create a test plan on ado, then add test cases that link directly to the user story. When executing test case, I mark test result in the test case and if I find a bug, I will link the bug to the test case

TC – Template

ID, title or summery, pre-condition (if have), steps and expected results, attachment file(if have)

TC – Design Tech

**- Black-box:**

+ Equivalence partitioning

+ Boundary Value Analysis

+ Decision table

+ State transaction

**- Experience- based**

+ Error Guessing

+ Exploratory Testing

TC - GOOD

**In my opinion, to have a good TC, we should consider some points, such as:**

1.    Keep things simple and transparent.

2.    Make test cases reusable.

3.    Keep test case IDs unique.

4.    Peer review is important.

5.    Test cases should have the end user or defined requirements in mind.

6.    Specify expected results and assumptions.

REQUIREMENT - Clarify

To clarify requirement, I Study US and make Q&A first

Then I Work with BA to clarify. If BA can't give the answer, they will reach the user, SM

After that, All the team have a grooming meeting to make sure all US are ready for coding and testing

BUG - Manage

In the azure devops, I query all the bugs in current sprint. I need to check the status of bug every day. The bugs that are ready to test I will assign to team member or myself to verify the bug. If there a bug has todo status for a long time, I need to notify the developer to remind them about the bug

If by the end of the sprint there are still unfixed bugs, I need to inform the whole team about the situation to decide which bugs need to be fixed immediately and which can be moved to the next sprint

BUG - Life cycle

In my experience, a bug usually has 4 main status: New, in-progress, ready for test, closed.

A new bug is found and logged on the system it will have a status of New. When the developer starts to fix the bug, it will have a status of in-progress. After the developer fixes the bug, it will have a status of ready for test, And when the tester verify and confirm the bug has been fixed, the status can be changed to closed.

In some cases, if I verify but the bug is still not fixed as expected required, I can re - open the bug and notify the developer

Sometime, the bug is rejected by developer. In this case, I will check the requirement related to this bug again. If this bug is really NOT a bug, I will close the bug. But if it is real bug. I will discuss with developer to show expected requirement for them one more time. If they still NOT accept this bug, I have to involve BA and Scrum Master to resolve.

BUG – Priority & Severity

- Priority is the order that a bug need to be fixed (1,2,3,4)

- Severity is how serious a bug can impact to the system (critical, high, medium, low)

to be more detail about each level:

\* **Severity:**

- critical: Prevents testing or usage of certain areas of the application such as Crashes, loss of data

- high: when it causes termination of one or more system components or the complete system. However, an acceptable alternative method exists to achieve required results.

- medium: when it causes the system to produce incomplete or inconsistent results.

- low: it has acceptable workarounds to achieve required results and does not affect much on user experience.

\* **Priority:** 1,2 are the highest levels; 3, 4 are lower levels

- 1: when it happens, Product cannot release without the successful resolution of the work item, and it should be fixed as soon as possible.

- 2: Product cannot release without the successful resolution of the work item, but it does not need to be fixed immediately.

- 3: Resolution of the work item is optional based on resources, time, and risk.

- 4: Resolution of the work item may be not necessary, based on resources, time, and risk.

**Example**:

Low Severity, High Priority: Error in the logo of a delivery website, low Severity because it will not affect the functionality of the website, but High Priority because even if the logo is wrong, customers will not have trust and do not want to use anymore.

High Severity, Low Priority: For a website, the error function has a high Severity because it affects one of the main functions but the Priority is low because this function needs to be released in the next phase, not the current phase.

BUG - Template

A bug must include: Title, steps to reproduce, actual result, expected result, environment, evidence like image or video record, priority, severity and assignee

BUG – Clarify

* Discuss with team member -> To make sure it is a real bug
* Log to system
* Inform developer
* Monitoring status of bug

RT >< System Test

The difference between system test and regression test is that

Doing ST in each sprint after front end & backend has been integrated and ready to work as a whole system, I test only function in this sprint not other

RT is usually do before PROD deployment, purpose is to make sure that new code changes do not impact old functions.

RT >< Re-Test

Re-test is test the function that used to have defect to prove it is no longer buggy.   
In contrast, regression test is test the function that used to work normally to find defects in it

RT – bao nhiêu lần

As you know, when we have to run RT it take a lot of time and effort because we need to run all test case from first sprint to current sprint. But normally we don’t have enough time. So we only do RT before releasing function to the production. But I would run RT every 2 sprint to check the stability of the code. So that, if I apply the automation to reduce time and effort for running RT. Normally if we don’t have enough time to run all test case we can pick the test case priority 1 and 2 to run RT. But we need to inform the whole scrum team about this decision. Because the risk can happen when I run RT with only test case priority 1 and 2

RT – Pick TC

Purpose of RT is to make sure that new code changes do not impact old functions.

Normally, regression test should be run all test cases from first sprint to current sprint

But it takes a lot of time and effort

- So If I need to run in a short time, i have to pick test case with priority 1 and 2 to Run RT. And I have to inform the whole team about this decision to be aware of the risk of this one

Why testing important

I think testing is so important in the project. Because Human errors can cause a defect or failure at any stage of the software development life cycle. So, let’s list the important reasons as to why software testing must be considered mandatory

To gain customer confidence

To check software adaptability

To identify errors

To avoid extra costs

To accelerate software development

To avoid risks

To optimize business

Cận kề deadline -> phải làm gì

When I have very tight deadline, to ensure my project meet the dealine,

First solution, I need to discuss with team to prioritize, get focus of everyone to release-related task. Tasks not related or not in high priority at the moment can be handled later/after release deadline. Prioritize tasks, bugs, testing carefully to make the most value for the release.

Second Solution, I Negotiate to reduce scope of testing OR shift the deadline to have more time for testing

3rd solution, I can get help by asking for more testers who can support (e.g, from neighbor team around)

And the final solution is working overtime. But this is always the last option if the above solutions are not met

so I need to Try to avoid the same situation again by raising In retrospective meeting to analyze the cause, try to improve in planning & estimation

Positive >< Negative test

Positive test is happy case or the test case with valid data. In constract, negative test is unhappy case or test case with invalid data

START DỰ ÁN

I need to outline for project in three areas: People, Process and Product.

People: we need to shape our team, know the stakeholder, meeting, communications, report

Process: we need to know the methodology to work on (agile or traditional), tools will use (jira, azure devops)

Product: we need to know about requirement, scope and deliverables, budget and timing, definition of done

Dự án 2 tester

i think teamwork is very importantevery day, we can have the daily meeting to devive the work btw us and also break tasks, such as study, test, log bug and so on. besides that, we also peer review TD, TC each other or double check bugs to make more confidencely. **can pick a**representative gathers task of all members to report or can change-over

EXPLORATORY

Exploratory: is the test that we test the product without any knowledge about it, we have to explore, study and analyze and test it all at one

Functional – non functional

Functional testing: is the testing of a function to be performed by the components or system and the goal is to verify functional requirements

e.g: unit, smoke, sanity, integration, regression

\* Non-functional Testing: is the testing of a non-functional quality characteristic like: reliability or usability and the goal is to verify how well does the software perform its function

e.g: load test, volume test, stress test

Smoke – Sanity

- Smoke testing is performed to ensure that the main functions of the system is working fine. (stability)

- Sanity testing is performed to check new function or bugs have been fixed. (feasibility)