Atlassian QA Questions :- Mindset Exercises
(Tushar Sappal)

Q1:- When you are testing a new feature, under what circumstances would you deviate from a script while performing manual testing?

According to me there can be the following scenarios under which I can deviate from the script (test plan) when testing a new feature.

If there is a sudden change in the specification of the new feature from product management and if there are chances that the new changes may affect the old workflows , I may deviate from the original testing temporarily and would rigorously test the old workflows .

If there is an immediate testing requirement for a hotfix release , and the testing resources are not present / scarce , I would work to test the hotfix code.

If the testing can be more optimized with automation scripts and there is too redundant work to be done , I would drop testing the feature manually and would write automation scripts for the same.

Q2:- If you joined a team of 10 developers as the only QA engineer and could implement and change any process(es) you'd like, how would you ensure that the team delivers high quality software?

Ans:-- Since I am the only QA for the 10 developers , the first task for me would be to quickly ramp up on the overall functionality of the product along with the features on which the developers are working on . This task may involve reading the documentation , viewing the reference videos and talking to the respective fellow

developers (if they are available) . If the reference material is not documented I will be going ahead and document those .The next task would be to gain insight on the feature set that the 10 developers are working on . 10 developers would be pushing a lot of code on the daily basis so I would be going forward and would be automating the common QA tasks like creating build out of the code on a scheduled basis (By Setting up Jenkins / Bamboo Machines). That would help me in getting a ready made build every time a change is committed in the repo.

Since I am the QA for the project , I will go ahead and document the test-cases and the use cases of the features on which the developers are working on , Since the new feature set should be compatible with the old feature set in terms of performance and usability parameters , I will be closely investigating if the new (under –development use cases) are not breaking any of the old working use-cases

As I mentioned earlier since there would be a lot of code that developers would be pushing daily, after getting an understanding of the existing features of the product I will be automating the old workflows (if they are not automated)and would incorporate those in the CI setup as a part of Build Generation Process , so that the new ones are more rigorously tested across various levels.If the old workflows are automated already I will incorporate them in the CI as well and will start writing the Automation Code for the new feature set that such that as soon as the new code is ready we should be having our automation ready to test it .

This would be my approach to the above situation to deliver the high quality software in a timely manner.

Q3:- A developer closes a defect you've reported as "Won't fix". How would you respond?

Ans:-- I will be approaching the developer for seeking explanation around the issue and the reason for marking the issue as won't fix. If the bug is reported by a customer and it is getting reproduced at customer's end then it becomes our priority to fix the issue.

There can be a few explanations for marking the issue as won't fix from the developer's perspective , the QA team should have a clear understanding why the issue was marked won't fix and whether it would be affecting any old feature set or any new features in the pipeline.

First can be , that the issue is reproduced on an environment that is too old to support and no customer is actively using the environment on which the issue is reproducible.
The other reason could be the fix is to be needed from the external party (like Operating System , Browser etc .) , in this case we would be opening the bug on their end and if the bug exists we would be following the issue and till the time the issue is not fixed , in this scenario I will be suggesting the developer to not mark the bug as won't fix and change the fix version to future-fix only if bug is important else won't fix can work fine. For e.g there are some features on Google Chrome that are not present in Internet Explorer and these issues can be ignored if they do not hamper the customer .

The third reason could be that the bug which was marked won't fix requires a lot of architectural change in the product at various levels and the developer is not confident of the fix he has and the effect the fix would cause on the overall product so he marks that as won't fix . In this scenario I will discuss within team to figure out should we mark it as won't fix or should we mark it as future if ultimately we are going to make architectural changes.

With adequate evidence why the issue is marked as won't fix , I can have discussions with product management and other managers to pull out the feature out so that half baked feature set around this issue is not live and till we do not have a solution for the bug we do not release a half baked functionalities.

Q4:- If you can, provide examples of where you have personally:

a. **Attempted to convince someone to change the way they were working, even though they didn't want to.**

When I was the part of my first team , the QA Team followed a waterfall model for testing the features , i.e. the developers would build the features in the designated time frame (sprint) and only when the feature was ready the QA team would start writing the automated test cases around the feature , this sometimes caused a bottleneck for testing as they started testing late so a thorough testing was not performed and a round of hardening was required very often before even minor releases . I requested the team members to follow the test driven development process in which the QA writes the automation code as soon as we have a single deliverable from developers such that the QA team has a ready framework for testing with essential pillars built in. I was questioned that this pattern would not be a feasible one for the product but after my request the team followed the pattern for a couple of release cycles and they realised that this pattern is helping the team to test , automate the workflows and deliver the product on time . Since then the team is actively following the test driven development pattern .

b. **Identified a problem in a process and implemented a change to improve it.**

I worked on a product that was highly UI Centric. When I joined the QA Team , they only tested the functional workflows and aspects of the product . But there were many bugs from the support team mentioning that the UI crashes a lot at the customer end very often and the UI remains in Hung State thereafter. The issue was passed to the QA team to investigate. On having discussions within the team it was discovered that the UI Performance testing was not done on a regular basis and the response time of the UI and the benchmarking of the UI code was not captured .

We re-wrote the Performance scripts from scratch and used the UI code benchmarking tools like Y-slow and Page Speed to get the feedback on whether the UI code was written in an efficient manner. With the new scripts and the correct benchmarking values we were able to optimize the response times and the frequent crashes of the UI code as we started following standard practices . We had a hotfix for the above changes and the team has followed the practise of running the benchmarking tool and the performance scripts on a weekly basis.

As the part of finding optimizing the UI code the developers moved the css to the top of the page and the js files to the bottom of the page as per the recommendations from the QA team to make sure the users have a better load time and page performance.

c. Implemented new testing tools or frameworks.

When I was part of Adobe Connect QA team I developed and implemented the Nightly Performance suite that performed Performance Testing of Adobe Connect on a Nightly basis , the tests had covered 10,000 users simultaneously using Adobe Connect and the resources to cater that demand were allocated in a dynamic and efficient manner .

I also developed the mobile automation framework for Adobe Connect .
For Adobe Target , I implemented the bullet tracer framework to run the end to end workflows for testing during and after the deployment windows

d. Made Changes to improve the quality of code before it reaches the testing stage.

With the test driven development approach , Continuous Integration / Delivery and more focus on automating the workflows I with my teams were able to test the features in more fast , efficient manner , thus giving us time to

think out of the box and work together with the development teams to further improve the features .

With the above benefit we worked closely with the dev team to work on improving the architecture / design thus making code less prone to bugs and covering all the features set requested in the minimum round of iterations
So each time the code reaches the QA team for testing it is better in quality as both QA and Dev teams have worked collaboratively to make sure the code and feature set is complete and robust .

e. Recognised a pattern of bugs occurring (e.g. cross-browser rendering problems) and implemented a change to stop the pattern from recurring.

Sorry I don't have any example for this .. :)

f. Any other achievement in your career that illustrates your suitability for our Quality Assistance role.

I was awarded the Special Contribution and Spot Awards for my efforts to automate the crucial workflows and developing the tools to assist and enhance the testing purposes .
I have a good grasp of testing workflows and an ability to quickly learn the product .