

Gamanza Interview Questionnaire

Interviewee: Gjore Janevski

1. What bug reporting tools have you used in the past and what do you recommend the most and why?

- Me personally have not used any bug reporting tools yet, but by doing a huge amount of research, I came to the conclusion that JIRA is a pretty good bug testing tool.

It is easily integrable, allows for screenshots and video feedback, you can run scans weekly and it supports a large scope of programming languages.

2. Are you familiar with test scenarios? Have you ever used any tool for that? How would you introduce writing test scenarios for our apps?

- After conducting research, I became familiar of test scenarios and the difference between test scenarios and test cases. I have not used any tool for that yet.

The first thing I am going to do is put myself in the shoes of the user, so it can give me a clear idea of the actual scenarios that the software will come across to after the release.

So first, for tasks that are done repeatedly, I will use test scripts. For testing new ideas I will use test cases, and for imitating the user I will use test scenarios.

So to create a test scenario I will divide it into 4 steps:

- * Collect the data regarding the documentation related to the specifications of the system
- * Define all the positive and negative scenarios that cover all the features of the software
- * Ensure that each requirement has a test scenario
- * Get an approval from the team leader, supervisor or the manager that the test scenario is good to go.

3. How important is writing test scenarios for you? Why?

- Test scenarios are very important to me because of a simple reason. Ensuring uninterrupted software testing, while also verifying the complete functionality of the application per the requirements.

4. Assume you find a problem while testing the application, what would be your recommended

approach to get that problem resolved?

- If I found a bug while testing, my first approach would be to report the problem as early as possible. Add it to the bug reporting tool, report the problem, and then write the bug details afterwards.

5. You find and report a problem within the application, but the developer doesn't believe this to be a problem and resolve the task as won't fix? How would you deal with this scenario?

- So in this case, I will simply document the bug into jira, which will contain all the screenshots and/or videos of the bug and the event log containing the errors so the developer will have no excuse.

After that, I will add the whole team into jira, so they can discuss it, and the conclusion will be between the developer and the whole team constructed of immediate supervisors and managers.

6. It is late in the evening. You are responsible for the GO/NO GO decision. The workflow is very strict and there are numerous tasks to be done:

- **Verifying bugs found in the very last stage of testing**
 - **One round of regression testing, etc.**
 - **Conclude the testing**
 - **Filling out the report,**
 - **Check everything is ready for release**

You just discovered that some part of the app doesn't feel right - UI/UX it's not catchy enough.

What do you do? Why?

You know if following the procedure the app won't be released on time, but timing is everything on this release. What do you do? Why?

- I will calculate the Risk and the Impact. It depends a lot on how likely it is that the part that does not feel right will occur in production and how much damage it will cause if it occurs.

See if there is a workaround, and find it. For example, if the "bad" part is extremely likely to occur, and there is no work around then this is a show-stopper. I will immediately escalate to

the stakeholders and block the release. If there is a work around, document the release and notify the stakeholders.

7. Assume you are the QA lead for a new major product with a set project deadline. Briefly describe

how you would go about the following:

- **Implementing QA process into the project**
- **Ensuring product successfully passes QA**
- **Making sure the project deadline is met.**

- First I will calculate how much time do I have to do all the testing and arrange my time accordingly. I will analyze the requirements, plan and design the tests and execute tests and report the defects.

After the defects are fixed, I will rerun the tests and do regression tests.

At the end I will run a release test. If the release test passed, and the deadline is met, then we are good to go. I will make sure

that the release test is done way before the deadline, so if it does not pass, the developers have time to fix the problems.

8. Assume that only one area of the application has been changed, how much time do you think should be spent testing each segment of the app (new functionality, core functionality, all functionality, etc.)? Give an estimate in % for each segment and explain your decision.

- Assuming the application was tested before the area was changed, I will spread the segments into: 60% testing the new functionality, 20% core functionality and 10% the all functionalities.

Simply because, if everything was working before the area has changed, put more time into that new area, and do regression testing to ensure that, that area did not affect the rest of the application.

9. Are you familiar with any test automation tool? Have you ever used automated testing in apps? How would you introduce automated testing to our apps?

I am not yet familiar with test automation tools yet, and never used them. After conducting a research, I read about Selenium and its implementation in Eclipse for web application testing.

10. For the Demo Testing, I am not yet familiar to the tools I need to use, so I just used Selenium Add-on for Chrome to test the front-end of the web application.