**Beet Seed**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Functional** | **Non-functional** | **Change-related** |
| What is being checked? | Tests if the system is working as expected and accordingly to the requirements. | Tests the performance of the system, like its speed, security,, load. | Tests if changes on the system during the development phase didn’t impacted negatively the functionalities that already existed. |
| When applicable? | During all the software development (when there’s already actual code and functionalities to test). | Often after the functional testing but also during all the development process as well. | After changes are made, like bugs fixed, new functionalities and/or updates. It must be done before new releases, too. |
| Restrictions | It only tests functional features and it does not include features like security, speed, performance, load (non-functional).  And it’s also very common to make some mistakes on this testing because exhaustive testing is impossible, the testers might miss some scenarios/outputs. | Only focuses on general features, rather than specific features, like on functional testing.  It may require more complex tools than the functional testing (*when, for instances, we want to simulate a large numbers of users visiting the website?*) | Only tests functionalities that already exist. If new features are added to the system, we can’t use this type of testing because it only covers already existing functionalities. |
| Pecularities | It focuses on testing functionalities that are documented on the requirements.  It tests features that are mainly related to the user experience, like if it’s easy to use and if it meets the users needs. | It focuses on testing non-functional features, like the performance, the security. | It’s more commonly used with automated testing.  The test cases must be updated when it comes to this testing stage. |

1. Although it may look similar, regression testing and re-test are two different concepts.

Regression testing makes sure that the code changes (updates on the system, bugs fixed) doesn’t impact negatively on a certain existing functionality of a software.

Re-testing, on the other hand, makes sure that a specific defect has been successfully fixed.

So, the regression testing focuses on a more general approach of the whole functionality, and the re-testing focuses on a specific defect found on a certain functionality.

**Beet Sprout**

1. ✅
2. I believe that it might depend on the projects. I think that it may be possible to discard the non-functional testing (or at least minimize it). Maybe on smaller and not so complex projects. However, it may constitute risks for the project, like possible defects on the security, performance and the overall user experience. So, it’s always preferable to include both functional and non-functional tests.
3. I believe that smoke testing really matters a lot on the overall testing of a project.

Smoke testing is performed at the early stages of the SDLC and its main focus is to ensure that the main functionalities are working correctly. Usually it’s done when there are new addictions to the code, to certify that the most important functionalities of the system that is being developed are working as expected.

I think it’s always appropriate to avoid later risks and issues on the project, that may be more expensive and more difficult to fix on the late stages.

**Mighty Beet**

1. ✅
2. Functional:

<https://beetrootqa2024.testrail.io/index.php?/cases/view/284&group_by=cases:section_id&group_order=asc&display_deleted_cases=0&group_id=308>

<https://beetrootqa2024.testrail.io/index.php?/cases/view/285&group_by=cases:section_id&group_order=asc&display_deleted_cases=0&group_id=315>

1. Non-functional:

<https://beetrootqa2024.testrail.io/index.php?/cases/view/288&group_by=cases:section_id&group_order=asc&display_deleted_cases=0&group_id=321>

<https://beetrootqa2024.testrail.io/index.php?/cases/view/286&group_by=cases:section_id&group_order=asc&display_deleted_cases=0&group_id=322>