Smoke Testing

Vinod Kumar Chauhan

Quality Assurance (QA), Impetus InfoTech Pvt. Ltd. (India)

Abstract- Smoke testing is an end-to-end testing which determine the stability of new build by checking the crucial functionality of the application under test and used as criteria of accepting the new build for detailed testing.

Index Terms- Software testing, acceptance testing, agile, build, regression testing, sanity testing

I. Introduction

The purpose of this research paper is to give details of smoke testing from the industry practices.

II. SMOKE TESTING

The purpose of smoke testing is to determine whether the new software build is stable or not so that the build could be used for detailed testing by the QA team and further work by the development team. If the build is stable i.e. the smoke test passes then build could be used by the QA and development team. On the stable build QA team performs functional testing for the newly added features/functionality and then performs regression testing depending upon the situation. But if the build is not stable i.e. the smoke fails then the build is rejected to the development team to fix the build issues and create a new build.

Smoke Testing is generally done by the QA team but in certain situations, can be done by the development team. In that case the development team checks the stability of the build and deploy to QA only if the build is stable. In other case, whenever there is new build deployment to the QA environment then the team first performs the smoke test on the build and depending upon the results of smoke the decision to accept or reject the build is taken. Generally if smoke fails then the build is rejected otherwise build is accepted.

Smoke testing is preliminary testing to reveal simple failures severe enough to reject a prospective software release. [1] This is the first testing performed on the build and all other kinds of testing follow it.

While checking the stability of the build, we do end-to-end testing to check if the code breaks. If there is any code break in any of the end-to-end flows then smoke fails. Smoke testing is conducted to ensure whether the most crucial functions of a program are working, but not bothering with finer details. [2] Because if crucial functionality is not working it will not be possible to perform the detailed testing so the build has to be rejected to fix the issues. For example login functionality in Facebook, launching the Facebook URL, navigate to account home page etc. are examples of smoke test cases.

Smoke testing performed on a particular build is also known as a build verification test (BVT). [1] The focus of the testing is the verification of the crucial functionality and not on the finer details. We touch all areas of application without going into deep. Smoke test cases are subset of the total test cases since we don’t have to go in details rather check the crucial functionality only.

Smoke Testing has a great importance in Agile. We have frequent build deployments which need to perform the smoke tests before performing the detailed testing. In certain cases the frequency of the deployments could be very high; more than one build per day. One feature/functionality is implemented or some issues are fixed and then new build is deployed to the QA environment to know if the new build is stable and functionality is implemented correctly. During this time another feature/functionality could be implemented or more issues can be fixed by the development team and then again new build could be deployed to QA environment. This way deploying more than one build per day, first ensures the quality of product and secondly ensure that the QA team is not sitting idle. Such situation occurs only when we have critical client releases but generally we have one build per day in agile.

The plumbing industry started using the smoke test in 1875 [3] but some believe this term originated in the electronic hardware testing. "The phrase smoke test comes from electronic hardware testing. You plug in a new board and turn on the power. If you see smoke coming from the board, turn off the power. You don't have to do any more testing." [4] When a new hardware component is added to a device then if smoke comes out then it fails. Similarly in plumbing if there are any breaks in the pipe fitting then smoke comes out. To serve the same purpose in software field, smoke testing is used to find the code breaks in the end-to-end testing of the product.

Sanity testing is the term which is related to smoke testing but they are different. One similarity between two is that both are used as criteria for accepting/rejecting the new build. If sanity test cases fail then build is rejected because if the build is not containing the required changes then there is no point of doing regression testing on the build. Smoke is being part of regression testing, checks the crucial functionality while sanity testing is part of acceptance testing, checks whether the newly added functionality is working or not. Generally smoke is performed on relatively unstable product while sanity testing is done on the relatively stable product.

Moreover generally only one of them is performed but if we need to do both then smoke testing is performed first and then sanity testing is done.

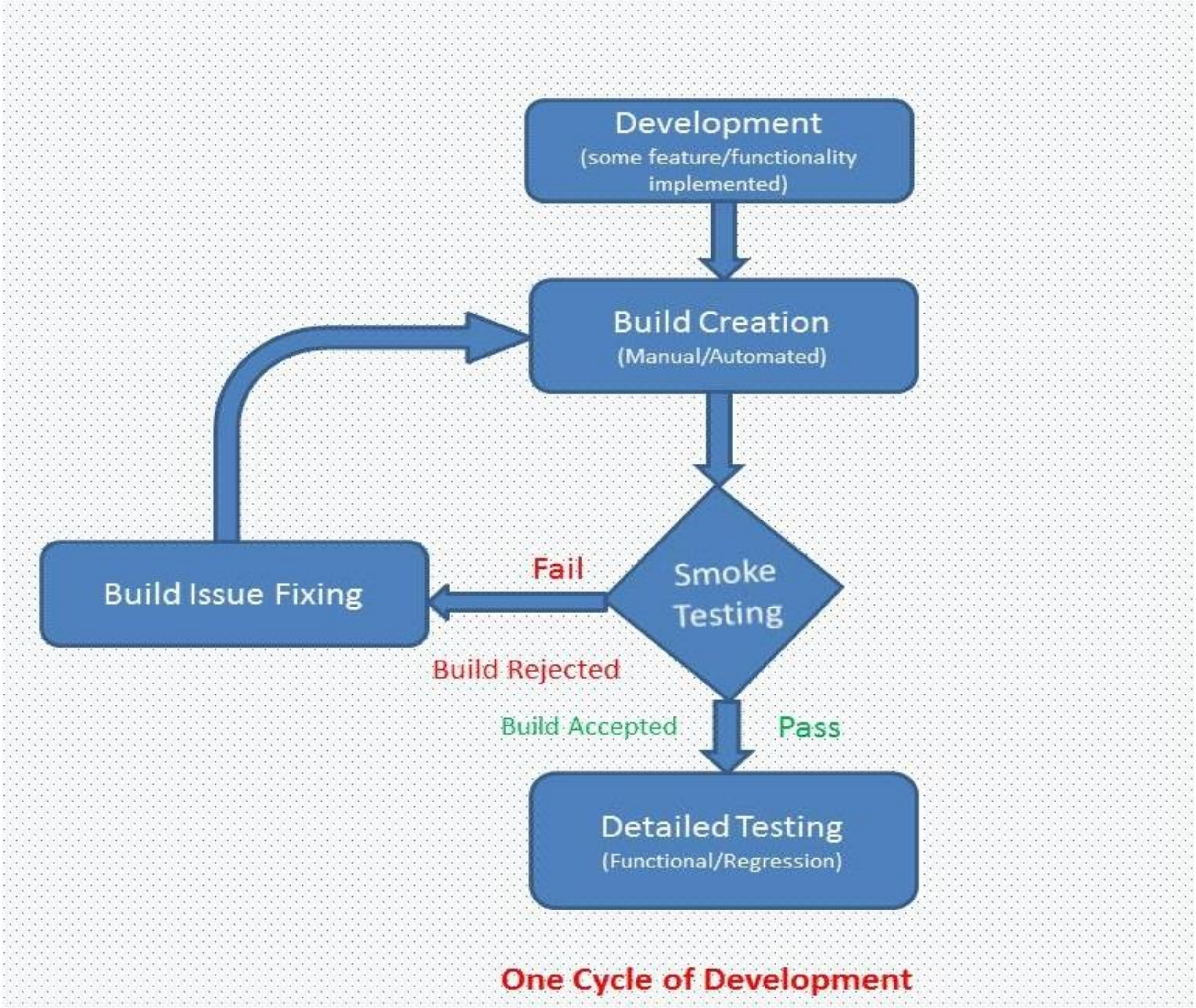

Figure 1: One cycle of Development

III. SCOPE OF SMOKE TESTING

We can define the scope of smoke testing w.r.t. when, where and how smoke testing is performed as follows:

1) When

Smoke testing is performed immediately after the build deployment. This is the first testing done on the build. First smoke testing is performed followed by other testing like functional testing (to test the newly added features), regression testing and user acceptance testing etc. It is a preliminary testing. [1]

2) Where

Smoke testing is generally done by the QA team in QA environment as criteria of accepting the build but in certain cases, it can be performed by development team.

3) How

The focus of smoke testing is to perform end-to-end testing of the build without going in finer details of the modules. This is high level testing. We simply navigate the different application flows from start to end without testing the details of the individual modules. Details of the modules are tested in the functional and regression testing. Here we just try to touch all the parts of the application.

IV. IMPORTANCE OF SMOKE TESTING

1) Saves Time

Smoke testing uses subset of total test cases to determine whether the build can be accepted for further testing and work, or not. Suppose we don’t perform smoke testing and start with the detailed testing. Now if there is issue on the last page of the application then we won’t be able to complete the regression testing of the application. The build has to be rejected to fix the issue. Now when new build will be deployed then that will need to perform the regression again. In this way a lot of effort and time, spent on the regression testing of earlier build is wasted. But if we perform smoke testing on the earlier build then we could detect the issue in lesser time and with lesser effort because smoke test cases are subset of total test cases.

2) Saves Effort

Smoke testing leads to saving of effort also as discussed above.

3) Saves Cost

Saving time and effort lead to saving cost of testing the application. In the software life cycle the earlier the errors are discovered and removed, the lower is the cost of their removal. [5]

4) Integration Risk

We perform end-to-end testing on each build so the blocker type issues can be uncovered early. So the risk of having integration issues is reduced.

5) Quality Improvement

Here the major issues are detected and corrected much earlier in the software test cycle thereby increasing the quality of software.

6) Progress Assessment

It is easier for project managers to assess the development progress. Since with each build we certify that end-to-end product is working and new features are added correctly.

V. TECHNIQUES OF SMOKE TESTING

1) Manual Approach

Here the smoke test cases are run manually. Manual approach is mainly used where the product is developed from the scratch and is unstable. Since during each development cycle new features/functionality is added to the product so using automation for this scenario will be very costly affair. This is because a lot of effort will be required on each build to maintain the scripts. For each newly added feature, we may have to update the scripts or may need to create new scripts. So for a new and unstable product, it is better to use the manual approach.

2) Automation Approach

Here the smoke test cases are automated and run with the help of some automation tools. In some cases the smoke scripts can be integrated with the automated build creation tools like Jenkins so that whenever a new build is deployed, the smoke suite automatically start execution without manual intervention and without wasting any time.

Automated smoke test cases are used in those environments where the product has become stable or product is the customization of some base product. No doubt even in this scenario, smoke scripts require maintenance but the maintenance effort is less.

3) Hybrid Approach

This approach is the combination of manual and automation approaches.

VI. PROCESS OF SMOKE TESTING

1) Identify Smoke Test Cases

Smoke test cases are subset of total test cases. To create a smoke, we need to identify the minimum number of test cases, to cover the crucial functionality of the product so that they could be executed in least amount of time. This is a very important step of the smoke testing. If we create large set of smoke test cases then it may take more time in execution and if we create small set of smoke test cases then it may not be effective in covering the crucial functionality.

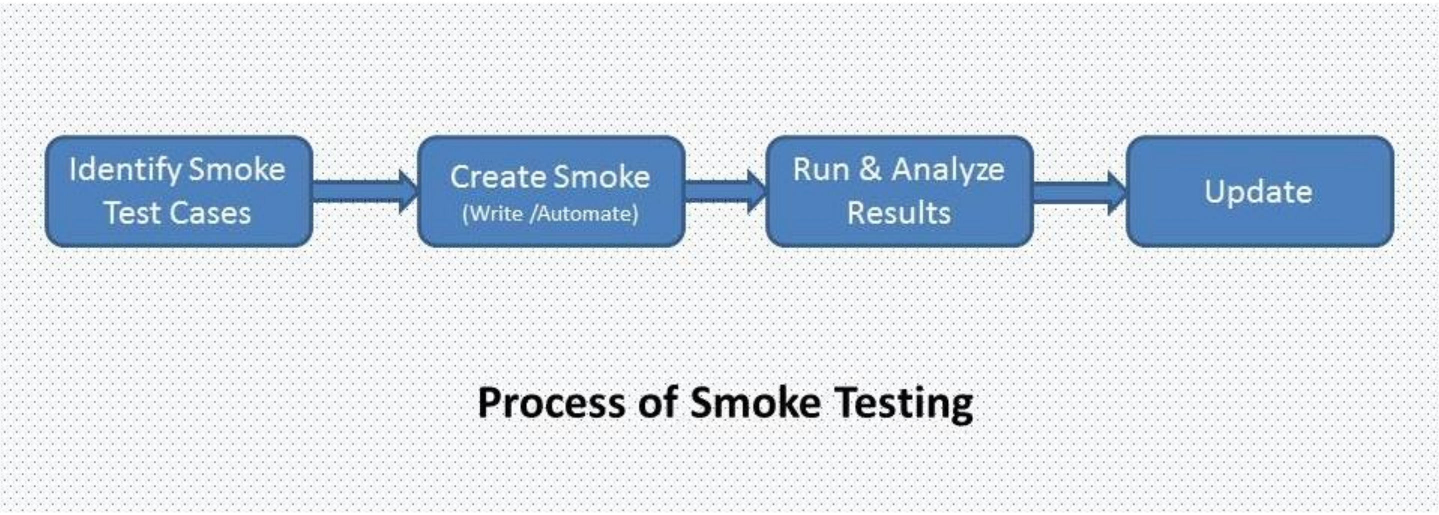

Figure 2: Process of smoke testing

2) Create Smoke Test Cases

Once the smoke test cases have been identified, the next step is to create test cases. We write smoke test cases and automate them if required. But as discussed in 'V. Techniques of Smoke Testing', it's not always possible to automate the smoke test cases.

3) Run and Analyze the Results

Once the smoke test cases are ready then whenever there is a new build, smoke is run on the build and results are analyzed to take the decision of accepting or rejecting the build.

4) Maintenance

Maintenance is used to maintain the value of smoke test cases over the period of time. Whenever new crucial functionality is added, we need to create new smoke test cases. Similarly whenever there are changes which affect the smoke scripts, we have to fix them.

VII. CHARACTERISTICS OF SMOKE TESTING

1) Smoke testing is preliminary testing. [1]
2) It is one type of integration testing because it involves end-to-end testing.
3) It checks only the crucial functionality without going into the finer details.
4) Generally done by the QA team in QA environment.
5) It checks the stability of build.
6) It checks if the build could be used for detailed testing and further work or not.
7) It is performed on the new build only.
8) Smoke test cases are a subset of total test cases.
9) It is also known as Build Verification Test (BVT). [1]
10) Smoke testing is like General Health Check Up. [6]
11) Smoke testing is used as criteria for accepting or rejecting the new build.
12) It is a high level testing.
13) It is widely used in agile.

REFERENCES

[1] WIKIPEDIA: http://en.wikipedia.org/wiki/Smoke_testing_(software)

[2] "Smoke testing and sanity testing – Quick and simple differences", http://www.softwaretestinghelp.com/smoke-testing-and-sanity-testing-difference/

[3] Buchan, W.P., " Institution of the Smoke-Test for Drains".

[4] Kaner, Bach, and Pettichord, "Lessons Learned in Software Testing". Wiley Computer Publishing, 2002.

[5] K.K. Aggarwal and Yogesh Singh, "Software Engineering", New Age International Publishers.

[6] Guru99, "Smoke and Sanity Testing - Introduction and Differences", http://www.guru99.com/smoke-sanity-testing.html

[7] Roger S. Pressman, "Software Engineering - A Practitioner's Approach".

AUTHOR

Vinod Kumar Chauhan, B.Sc. (Maths), MCA, HPU Shimla, Impetus InfoTech Pvt. Ltd. (India) jmdvinodjmd@gmail.com /vinod.kumar@impetus.co.in +919816383222.