DOCUMENTATION OF API TESTING IN SOFTWARE TESTING

TESTING BETWEEN TWO PROGRAMS

MRS. JYOTHI ESWARI N
ASSISTANT PROFESSOR, BACHELOR OF COMPUTER APPLICATIONS
COMMUNITY INSTITUTE OF COMMERCE AND MANAGEMENT STUDIES
BENGULURU, KARNATAKA-560011
jyothikumar.mca@gmail.com

***Abstract*—Engineering disciplines pair design and development stages which check early, intermediate and final product so that bugs and defects can be identified and removed. The product which meets the requirements is validated and verified to build high quality software. Testing is classified into many types if testing, where testing is done repetitively till it meets the software requirements. In API testing, both the application programs should be tested individually and later tested their communication. Testing is a process to know the developed Product meets the expected requirements. In that case, communication between two applications is tested using API testing. It is done both manual and automated testing.**

Keywords—Testing, interface, application, request, response, security, integration

I. INTRODUCTION (*Heading 1*)

API stands for APPLICATION PROGRAMMING INTERFACE. API acts as an interface between two software programs which creates bridge to communicate each other. It works as a intermediate, taking the request from one software program and respond to the other program, vice versa.

The three different layers are PRESENTATION LAYER, BUSINESS LAYER AND DATABASE LAYER. API is the middle layer in between Business Layer and Database Layer, where communication and data exchange between the two different applications. Example: If I am browsing Social Media, and if I want to share any information with my friends network, I can send the link to my contacts in Whatsapp.

A. Aims of API Testing

- Issues faced by Multithreaded programs
- Maintains Security and performance of the applications.
- Checks with Reliability issues.
- Checks for the duplication or if any missing functionality.
- Maintains the good response time.

B. Benefits of API testing

- Don't repeat: Avoid repeating code. When test require you to address the same components or code for similar actions, create a common library or procedure to wrap your requests.
- Example: If I am booking for grocery in BIGBASKET and making online payment through another application using UPI., where Integration is done between two different applications.
- Platform Independent -/The test should run on any of the environments under different configuration.
- Avoid repetition - Avoid repetiting code. When tests require you to address the same components or same functionality.
- Earlier validation of correctness in response and reply by the software programs.
- Reduced testing cost - The cost is reduced as repeating code is minimised.
- Works faster - Code development is faster as it can service more requests without any major changes in the application code.
- Faster time to resoultion- When failure in API testing, can easily find out where system got struck and find out the reason for the defect .

II. TYPES OF API TESTING

1. UNIT TESTING: Individual components are tested and the low-level bugs are found and fixed.Each component is tested independently, without checking with other components in the system.
2. INTEGRATION TESTING: It consists of collection of subsystems which have been integrated and testing is done all together as one system.
3. CONTRACT TESTING: In this testing phase, it validates and checks whether two separate systems are able to communicate and respond each other or not.
4. PERFORMANCE TESTING: It is a technique used to test the Speed, Request time, Response time, Stress, Stability, Reliability, Scalability of the application and

also evaluates the network of the device under which the application is being executed.
5. FUZZ TESTING: In this testing, random data and massive amount of data is given and checks for the vulnerabilities found, if any.
6. FUNCTIONAL TESTING: It derives from software specifications. Usually functional testing techniques produce test case derived from specifications that identify and produces individual test cases.

III. PROGRAM ANALYSIS TOOLS

A program analysis tool is an automated tool which takes the source code of a program as an input, generates and produces several output called as output, which is very important to a program such as
1. Time complexity
2. Space complexity
3. Efficiency
4. Performance
5. Quality Assurance
6. Adaptability
7. Security standards
8. Reliability and so on.

Program analysis tools are of two types.

A. Static analysis tools: Static analysis tool represents properties of a developed software product without generating the product. It analyzes the working structural representation of an application in all analytical conditions. It checks the procedural call, stack parameters, local and global variables, code reviews, code inspection, veriafibility and validating the specifications with the product.

B. Dynamic analysis tool: This tool is used for the actual execution of the program and its behaviour of the system is been recorded during execution. Its output is recorded and produce a report using statement coverage, structural analysis, branching loop, boundary value analysis and condition coverage. Dynamic analysis reports are generated in the form of histograms, which is graphic tool to summarize discrete and continuous data in the given intervals. And it is similar to barcharts. API testing adopts both static and dynamic tools to perform its operation.

IV. CONCLUSION

API testing is handled by Quality assurance team and is concerned with ensuring that the required level of quality is achieved in a software product, and defining the appropriate quality standards to meet its requirements. API testing is Black box testing, which is used to derive test cases based on analysis of the specification, either functional or nonfunctional without reference to its internal structure of both the applications.

REFERENCES

[1] Mauro Pezze, Michal Young:Software Testing and Analysis - Process, Principles and Techniques, 1st edition, John Wiley & sons, 2011.
[2] Paul C. Jorgensen: Software Testing, , A Crafstman's Approach, 3rd Edition, Auerbach Publications, 2012.
[3] API Testing and Development, A practical guide to crating, testing, and managing APIs for automated software testing,by DAVE WESTERVELD, Packt Publishing Limited, 2021.