Testing strategies – Bean and Brew

Contents

[Types of Testing. 1](#_Toc148618369)

[Software Testing 1](#_Toc148618370)

[Security Testing 1](#_Toc148618371)

[Unit Testing 1](#_Toc148618372)

[White Box Testing 1](#_Toc148618373)

# Types of Testing.

## Software Testing

This is the process of verifying and evaluating the functions of the bean and brew application. This type of testing will be used on the bean and brew application to assure the quality of the software and make sure that the requirements are met correctly. This type of testing is used to identify and fix defects in the application.

## Security Testing

This type of testing it used to uncover vulnerabilities in the software being developed. This is used to prevent any malicious attacks that can happen. Security testing must identify all loopholes to ensure that the program cannot be attacked. An attack we should use on the Bean and brew application is penetration testing and having a risk assessment. The first test that should be done will be penetration testing. This will simulate a malicious hacker externally trying to harm the system. Then after we find the vulnerabilities and develop mitigations, run a risk assessment to make sure the application has no vulnerabilities.

## Unit Testing

This process is where the individual would focus on a certain unit or component of the bean and brew application. This is used to meet the requirements and needs of the program which allows to determine if it is suitable for user or not. This will be efficient and effective as there are multiple components used in the application which allows the user to easily log down and find out if the application can be used to improve the quality of the application.

## White Box Testing

This testing technique is where the software’s internal structure, design and functionality and security are tested to verify input and output flow to improve the design and to determine if the program is suitable for use. This allows the individual to see how the program works from the inside allowing them to see what happens inside the program when running the tests.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Test Description | Test Data/Input | Type of test | Expected outcome. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |