Documentation and Security Testing

Introduction

- Testing is an important step in any development process for both software and hardware products.
- Testing the functionalities of the end product is the main focus of all testing techniques. There are many other facets of testing which brings thoroughness to the product.
- Documentation testing is an example, which involves testing the accuracy of various kinds of documents which are part of the software application.

Introduction

- The process of software testing remains incomplete if the documentation related to the software is not tested.
- The tester has to make sure that the documentation elements accompanying the software are error free.

Introduction

Objective of any tester performing documentation testing

1. Whether the information mentioned in the documentation is available in the product.
2. Whether the required information of the product is provided in the documentation.

Introduction

- Deals with security testing that helps to bring the confidence in the product security.
- Software security is another major area of concern for any organization.
- Considered the be most important element that determine the quality of the software product.
- Any compromise or errors in software security will cause both financial and data losses to the users. The testing process must make sure that any vulnerability in the software is detected and resolved before the software is released to the market.

Documentation Testing

- Documentation refers to written information that defines, describes, specifies, reports or certifies the activities, requirements, procedures or results of the software application. It also includes pictorial information.
- Documentation provide information about the product such as design documents, code commands, white papers, and so on.
- It refers to the product's technical manuals which are made available both online and in the form of a printed book.

Documentation Testing - Example

- When you purchase a mobile phone there is a manual present along with the mobile phone.
- This manual will provide all the necessary information about the mobile product such as model number, list of features, information about the keys, safety tips, and so on.
- This ‘documentation’ helps the user to know and understand the features easily.

Documentation Testing- Note

- The end user can be a common man who might not understand the technology   and just uses the application. Alternatively, the end user can be a highly skilled technician, who will install or repair the system. Therefore, the type of
- documentation and information covered varies depending on the end users.

Documentation Testing

- Documentation meets its objective only if it provides necessary and complete information to the end users or customers.
- Very important to make sure that no error or incorrect information is included in the documentation.
- In order to remove such errors the process of documentation testing is carried out.

Documentation Testing

- Documentation is often considered a part of any software product.
  - Documentation deserves the same level of testing that is applied to software.
  - good documentation not only improves ease of use, but also improves customer satisfaction
  - Poor documentation can lead to extra work and costs for the vendor support desk, and it might put the software producer in legal troubles.

Documentation Testing

- Important characteristics of a documentation testing program

1. Frequent and early testing
2. Assessment of accuracy and ease of use
3. Evaluation by people who did not write the documentation

Types of Software Documentation

- Documentation contains information on software and its components. This information makes the software user friendly and makes it easier to use the software.
- **Testing text based documentation** is simpler compared to other forms of documentation.
  - The tester has to make sure that the information provided is correct and is meant for that particular group of audience for which it is developed.
  - Some software elements are also part of the software documentation then it becomes very important to carry out testing effectively.

Types of Software Documentation

- Elements of software that are part of software documentation testing

1. Packaging Text and Graphics
2. Marketing Material and Other Inserts
3. Warranty/Registration
4. End User License Agreement (EULA)
5. Labels and Stickers
6. Installation and Setup Instructions
7. User's Manual
8. Online Help
9. Tutorials, Wizards, and Computer Based Training
10. Samples, Examples, and Templates
11. Error Messages

Packaging Text and Graphics

- Packaging text and graphics are the text and graphics that are printed on the package of a product. The user receives a software or a hardware product inside a package. This package can be made using a box, carton, wrapped with paper or plastic.
- text and graphics used on the box or packaging conveys information such as company and product name, company logo, manufacturing data,
- text style of each organization varies from one organization to another and also from one product to another

Packaging Text and Graphics

- mobile phones are usually packed and sold in a box. These boxes not only contain the mobile phone but also the accessories such as USB connector, mobile charger, ear phones, and so on.
- User manual and mobile suite CD are the two other important things that come with the mobile phone package.
- Some information related to the mobile can also be seen on the package.

Marketing Material

- Marketing collaterals help a product to sell in the market
- Information that is presented on such collaterals should be
- informative and accurate.
- convey the most important and attractive feature of the product.
- makes the user aware of the product’s existence in the market and its special features.
- Must be correct
- presented in an attractive manner, to create an interest in the customer or end user to buy the product.

Marketing Material - Example

- The pop up online advertisements that appear on the Web page comes with the list of features of the product. These advertisements are targeted to generate interest in the end users to check for other features and purchase the
- product and hence the content has to be accurately tested.

Warranty/Registration

- This mainly deals with the terms and conditions for using the software. In many software products, the registration is done when the user tries to install the software. The information presented has to be easily understood by the user.

Warranty/Registration

- End User License Agreement (EULA)
- EULA is a legal license document. The customer has to agree and follow the terms and conditions defined in the license document. The details of the license will sometimes be printed on the envelope or package of software CD or it appears as pop up while installing the software

Labels and Stickers

- Labels and stickers are pictorial or graphical representations of the product and company icons or logos printed on the product cover, product, manual, or on the display screen of the monitor. Warranty information is also mentioned on the product cover using labels or stickers.

Windows® XP Professional
OEM Software
Product Key: XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
00000-001-966-247
X10-74762
Microsoft
www.microsoft.com/piracy/howtotell
Proof of License Certificate of Authenticity
Windows® XP Professional
OEM Software
YYY-12345
00000-004-505-542
Product Key: XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
X11-37156
Microsoft
Label not to be sold separately

Installation and Setup Instructions

- Installation and setup instruction information is provided to help the user to install and run the software. The instructions are usually printed on the product’s package material only if the installation and setup procedure is small. In case the procedure is lengthy, then it is provided in a separate manual. Such a manual is called an installation guide.

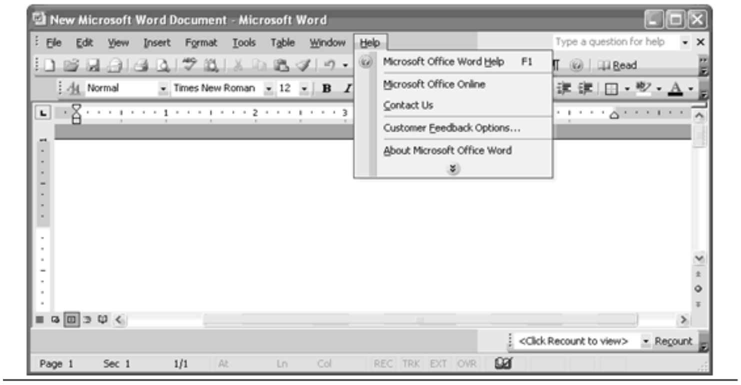
User Manual

- The most useful and mandatory document that all software products must accompany when the product is sold to a customer. This is available as both printed and online material. The manual provides all the information that the user would like to know about the software. This enables the user to acquire knowledge about the software functionalities and features.

- Online
- Ebooks

Online Help

- Online help is used along with the manual. It is easy to use and search for specific information easily and quickly, since it is indexed. Most online help allow users to ask the question in simple English language.


New Microsoft Word Document - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
Normal
Times New Roman
12
Read
Microsoft Office Word Help F1
Microsoft Office Online
Contact Us
Customer Feedback Options...
About Microsoft Office Word
<Click Recount to view>
Recount
Page 1 Sec 1 1/1 At Ln Col REC TRK EXT OVR

Tutorials, Wizards, and Computer Based Training (CBT)

- Consists of written documents and programming codes.
- They provide users with the useful information about the software that enables users to quickly learn steps or procedures to use the software.
- Documents are made available as online help, so that the users can access it easily. They also have high level macros which are programs that assist the users to perform a specific task. These macros work along with the online help system. This enables the user to quickly search and get the required information.

Samples, Examples, and Templates

- Software has certain inbuilt application samples that the user can refer or use to build his own application. Some software also provides templates which are pre-formatted examples based on which other applications can be developed.

Templates

- Microsoft Power point has a number of inbuilt templates for slide layout and
- design, which the user can apply to his presentation. These inbuilt templates come along with the software package.

Error Messages

- Information about error messages are important part of any software documentation. The software displays the error messages when it encounters unusual or exceptional events. These events can be triggered while using the software or hardware incorrectly or due to software related problems. The users must be able to understand the error message to know the actual problem that the system has encountered. This enables them to take necessary action to resolve the error quickly.

Importance of Documentation Testing

- enhances the quality of the various elements of software application.
- documentation is very essential at all stages of any software development process.
- provides information to end users about the product and also acts as a marketing tool for product promotion.
- Any error in this documentation will not only impact the product sales but also the reputation of the organization.

Importance of Documentation Testing - Example

- Let us assume that a typical software product released to the market by an organization can run with a minimum of 256 Mega Bytes (MB) of Random Access Memory (RAM).
- However, in the installation manual of the software product, if it is mentioned that the minimum RAM required for the software to run is 512 MB, then it is a serious mistake in the documentation.
- *This mistake is considered as an error, which the software tester must rectify and make necessary correction.*

Importance of Documentation Testing

- If the documentation of the software is good, then it contributes to the product in the following ways:

a) It Improves the Software's Usability: The documentation helps the user to know how the software should be used. If it communicates the required information effectively, then the user will be able to use the software easily without much difficulty.

Importance of Documentation Testing

b) It Improves the Software Reliability: When the user reads the documentation he/she will come to know about the various features and applications the software has. The user will read the documentation and judge the software functionalities based on the information provided. Any errors in the documentation will certainly result in poor reliability. Therefore, it is very important to test the documentation against the software to find errors in it.

Importance of Documentation Testing

c) It Decreases the Product Support Cost:

The cost incurred on the error found by the customer is 10 to 100 times more than the cost incurred to find the same error before it is released to the market. If the documentation fails to communicate the information clearly to the users, then the user will be confused and will face problems while using the software. Moreover, the organization which has developed the software has to provide customer support to resolve the problems that the customer is facing – which proves to be expensive. Therefore, in order to overcome such problems, good documentation must be provided with the software product. This documentation helps the customers to easily understand and use even the most difficult application or feature of the software.

Importance of Documentation Testing

- Documentation testing
- checks for the correctness of facts and figures mentioned in the documentation.
- make sure that all the instructional steps are explained clearly and effectively.
- check whether or not the documentation meets all the requirements of the end user.
- helps in finding errors in the documentation and helps in providing correct, accurate, and effective information to the users. It improves customer satisfaction

Security Testing

- Security testing must not be confused with safety testing. The aspect of security deals with how well the software is protected from external elements such as hackers who make use of virus to affect the normal operation of the software.

Security Testing

- A type of computer viruses named resident virus such as Randex, CMJ, Meve, and MrKlunky attack the RAM memory of the system. From RAM, it affects the normal operation of the entire operating system. It will corrupt the files currently used by the operating system.

Security Testing

- A tester who is performing a software security test must apply risk based approach to find the bugs in the software architecture and design.
- Testers must have the mindset of an attacker (hacker) to find bugs related to software security.
- This requires identifying the risks that the software is prone to during an attack and creating test cases based on the risks identified. Such test cases will enable the tester to focus on a particular area of code where the possible attack can be successful.

Security Testing

- Banks use smart card technology to overcome fraud and misuse of account holder’s freedom. These smart cards use the Crypto System to carryout transactions and verify the identities of the cardholder and the bank. The card holder information and account have to be very well protected, as the transactions happen online and there is always a possible risk of the system being hacked or a virus attack. If any such incident happens, the bank will lose valuable information and also its account holder’s money. Testing for possible risk in the software will help to overcome and stop any attack on the software and prevent theft of information.

Security Testing

- Software security testing tests the software behavior when the software is attacked by some external element. Sometimes, software failure occurs without any external interference. This may occur due to weak design or coding. Therefore, software security aims to protect the software from any such failures. Any software designed with poor logic is prone to external attack from hackers.

Security Testing

- Hackers make use of weak codes in the software to carry out an attack on the software and cause software failure. Therefore, any such vulnerability is considered as a bug.
- It is the responsibility of the tester to make sure that any such bug in the software is detected and reported to the development team.
- Information and services are the two important aspects of software security. Protecting both the information and services from possible external attack is vital.
- risk analysis should be carried out at the design level so as to identify any potential security problems and resolve them at the earliest.

Security Testing - Example

- In 2008, hackers from Russia robbed an Automatic Teller Machine (ATM) in New York. The hackers were able to obtain the PIN numbers of customers after they hacked the main server of the ATM company. They managed to steal $180,000 from ATM machine.

Security Testing

- Software security testers perform many different tasks to manage risks related to software security such as:

1. Creating security abuse/misuse cases.
2. Listing normative security requirements.
3. Performing architectural risk analysis.
4. Building risk-based security test plans.
5. Wielding static analysis tools.
6. Performing security tests.
7. Performing penetration testing in the final environment.
8. Cleaning up after security breaches.

Security Testing

- security testing involves two approaches:

1. Testing software security mechanisms to check whether the functionalities of this mechanism are properly implemented during the software product design and coding.
2. Performing the risk based security testing with the perspective of an attacker and developing test cases to check for possible risks.

- The tester should make sure that they identify all possible bugs in the software to minimize the risk of software being prone to any external attack from hackers.

Threat Modelling

- The process of assessing and documenting the system's vulnerability to security risks is known as security threat modeling.
- Allows organization or developers to understand the various threats a system can face.
- the tester will analyze the system with an attacker’s perspective.
- identify the possible threats and rate them based on order of threat severity i.e., greater the risk, higher the order of severity.

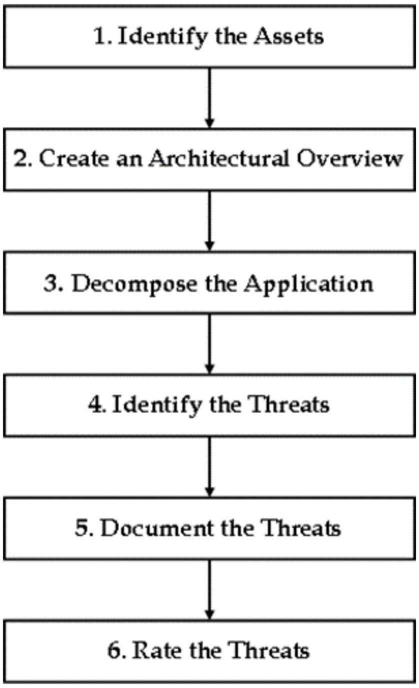
Threat Modelling

- Threat modeling is a highly structured and organized approach of threat identification. It is also a highly cost effective approach, which helps to identify the possible threats efficiently and effectively. The principle behind the model is that "one cannot make the system secure until they know the threats the system can face".

Threat Modelling

- Threat modeling helps to identify and tackle the risk during software evolution. It should be carried out at every level of software development life cycle, as identifying and resolving threats of a fully developed product require both time and cost.

Threat Modelling



1. Identify the Assets
2. Create an Architectural Overview
3. Decompose the Application
4. Identify the Threats
5. Document the Threats
6. Rate the Threats

Identify Assets

- the tester will identify all the assets that are associated with the system that has to be protected.
- Confidential customer data, orders, and employee database are some areas that are identified.

Create an Architecture Overview

- The tester will use representations like simple diagrams, tables, and graphical representation to describe and document the architecture of the system. This helps the tester to understand the actual working of the system.

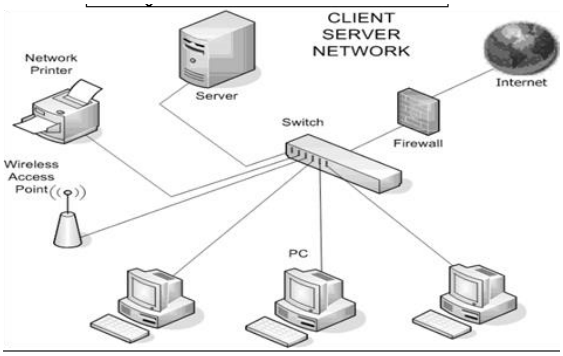
Create an Architecture Overview

- Tasks
- (a) Identify what the system does and how it accesses the various subsystems.
- Example
- In an employee database management system, the employee views various subsystems such as financial data, personal data, and project data. The managers view employee details, employee log in times, and so on.

Create an Architecture Overview

(b) Create an architecture diagram of the system. The architecture diagram includes subsystems, boundaries of operation, data flow channels, and so on.

- Example
- The network architecture overview comprises various computers that are connected to the local server and access the data from the protected system.



Create an Architecture Overview

(c) Identify the technologies associated with the system as these are the technologies that are used to implement the system.

Create an Architecture Overview

- Decompose the Application
- Decomposing the application refers to understanding the platform on which the system operates and designing appropriate standards.
- In a network system, the tester identifies the network and host infrastructure design.

- This helps them to create a security profile for the application. This profile is used to detect the vulnerabilities in the various area of the system such as design, implementation, and configuration.

Create an Architecture Overview

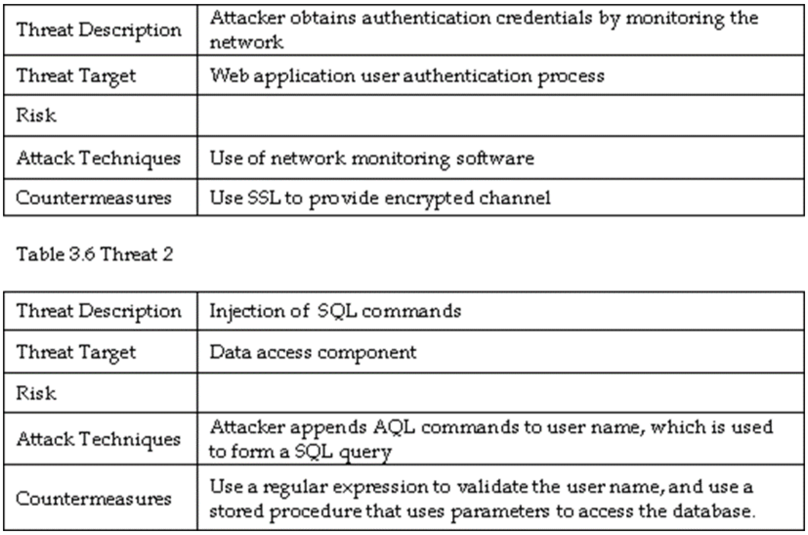
- Identify the Threats
- The tester should think and act like a hacker and find the vulnerabilities in the system. The tester should have the knowledge of the entire system architecture and potential vulnerabilities of the system. This enables him/her to identify the threats that could affect the system.
- Example
- In a network system the tester identifies the threats such as network threats, host threats, application threats, and so on.

Create an Architecture Overview

- Document the Threats

1. Threat Description: It defines the threat that has been detected.
2. Threat Target: It specifies the actual target of the attacker.
3. Risk: This is used to mention the priority based on the criticality of the risk.
4. Attack Techniques: These are the techniques that are used by the attackers to carry out the attack.
5. Attack Techniques: These are the techniques that are used by the attackers to carry out the attack.

Template to Record Threat



Threat Description	Attacker obtains authentication credentials by monitoring the network
Threat Target	Web application user authentication process
Risk	
Attack Techniques	Use of network monitoring software
Countermeasures	Use SSL to provide encrypted channel

Table 3.6 Threat 2

Threat Description	Injection of SQL commands
Threat Target	Data access component
Risk	
Attack Techniques	Attacker appends AQL commands to user name, which is used to form a SQL query
Countermeasures	Use a regular expression to validate the user name, and use a stored procedure that uses parameters to access the database.

Rate the Threats

- The threats are rated using simple high, medium, or low ratings. This rating is based on the risk value which is calculated using the following simple formula:
- Risk= Probability * Damage Potential
- where

(a) Probability represents the probability of occurrence of the threat. This is rated using the scale of 1 to 10, where 1 indicates a threat that is very rare to occur and 10 indicates a common and definite threat.

(b) Damage Potential represents the damage caused by the threat. This is also rated using the scale of 1 to 10 where 1 represents minimal damage and 10 represents a catastrophe to the system.

Create an Architecture Overview

- If Probability = 5 and Damage Potential = 3 then Risk = 15.
- Similarly, if Probability = 7 and Damage Potential = 3 then Risk = 21.

Buffer Overrun

- Buffer overrun is one of the most common security problems today. Buffer is a memory area shared by microprocessors and other hardware devices. Buffer overrun is caused by buffer overflow. Buffer overflow occurs when the value saved in the buffer exceeds the size of the buffer memory. The excess memory values either overwrite the existing buffer memory or other buffer memory. Such buffer overruns might crash the system.

Buffer Overrun

- Software developed using programming languages such as C and C++ have no built-in protection against accessing or overwriting data during buffer overruns. This is because there are no inbuilt functions to check whether or not the data written to an array is within the defined boundary limits of the array.

Buffer Overrun

- Bugs are caused when excess data is saved in a limited buffer memory. This will overwrite the adjacent stack or memory of the buffer and will often cause the program to behave incorrectly or crash. This not only affects the execution of the program, but also throws serious security vulnerability.
- Example
- A hacker can overrun the buffer of the running program by supplying un- trusted data and can corrupt the stack. Later the hacker could overwrite the buffer with an un-trusted executable code and thereby can take control of the system.

Safe String Functions

- Safe string functions are functions that are used to overcome buffer overruns in software programs. Poor handling of buffers causes a buffer overrun and leads to many system security problems. Such poor handling is related to string manipulation operations. The safe string functions perform extra processing of the input data for proper handling of buffers in the software.   Since, C or C++ language   do not have proper control over the data being stored in the buffer, these strings replace the standard string functions that are available such as strcat, strcpy, sprintf, and so on.

Buffer Overrun

- The strcpy is a function used to copy the string value from one variable to another variable. If the destination variable array length is small compared to the source variable, then the problem of overflow occurs in the destination variable.

Advantages of using safe string functions

1. Along with the input data, the functions also receive the destination buffer's size as input. This makes sure that the destination buffer does not overrun if the input data exceeds the normal size of the destination buffer.
2. The string functions terminate all output strings with a Null character, which indicates the end of the string. Other functions using these strings can assume that they will encounter null character. Therefore, the data before the null character is a valid data and null character terminates the string without allowing it to run indefinitely.
3. NTSTATUS value is returned by all safe string functions. This value indicates the calling function that the safe string function has performed the operation successfully.
4. The safe string functions are available in two versions. One version supports double-byte Unicode characters and the other supports single-byte American Standard Code for Information Interchange characters.

Advantages of using safe string functions

- When the tester performs the white box test of the software, then the tester has to check for unsafe strings in the program code and how they are used in the program logic. This enables to develop test cases to check whether or not these unsafe string functions cause overruns.

Computer Forensics

This is a technique of computer investigation and analysis that is used to gather substantial evidence against a cyber crime for presenting it in a court of law. The main aim of computer forensics is to conduct a structured investigation of a cyber crime to find out what happened and who was responsible for it. This is done to protect the security of software.
Computer forensics deals with identifying and solving crimes that are carried out by using computer technology. The governments across the globe have imposed many laws to check cyber crimes. However, lack of evidence has made it difficult to prosecute the people responsible for the crimes. Computer forensics helps to overcome such difficulties. It helps to gather evidence to take legal actions against those who carry out such crimes.

Computer Forensics

- The tester should check for security vulnerability issues related to test software from a computer forensic perspective. Sometimes, hackers do not really need to break into your system to steal the data, since some data can be easily accessed. If the hacker knows where exactly to look for a particular data then he/she can easily get the data from the software.