

9.13 REVERSE ENGINEERING

Reverse engineering (RE) is the process of discovering the technological principles of a device, object or system through analysis of its structure, function and operation. It often involves taking something (e.g., a mechanical device, electronic component, or software program) apart and analyzing its workings in detail to be used in maintenance or to try to make a new device or program that does the same thing without copying anything from the original. Reverse engineering has its origins in the analysis of hardware for commercial or military advantage.

The purpose is to deduce design decisions from end products with little or no additional knowledge about the procedures involved in the original production. The same techniques are currently being researched for application to legacy software systems, not for industrial or defence ends, but rather to replace incorrect, incomplete, or otherwise unavailable documentation http://en.wikipedia.org/wiki/Reverse_engineering - cite_note-1#cite_note-1.

Today the market of software is covered by an incredible number of protected applications, which don't allow you to use all features of programs if you aren't a registered user of these. Reverse engineering is simply the art of removing protection from programs also known as "cracking". In some other words cracking is described as follows: "When you create a program you engineer it, in fact you build the executable from the source-code. The reverse engineering is simply the art of generate a source-code from an executable. Reverse engineering is used to understand how a program does an action, to bypass protection etc. Usually it's not necessary to disassemble all code of the application not only the part of the application that we are interested must be reversed. Reverse engineering used by a cracker to understand the protection scheme and to break it, so it's a very important thing in the whole world of the crack." In short: "Reverse Engineering referred to a way to modify a program such that it behaves as the way a reverse engineer wish." "Cracking is a method of making a software program function other than it was Originally intended by means of investigating the code, and, if necessary, patching it."

Software reverse engineering is done to retrieve the source code of a program because the source code was lost, to study how the program performs certain operations, to improve the performance of a program, to fix a bug (correct an error in the program when the source code is not available), to identify malicious content in a program such as a virus or to adapt a program written for use with one microprocessor for use with another. Reverse engineering for the purpose of copying or duplicating programs may constitute a copyright violation. In some cases, the licensed use of software specifically prohibits reverse engineering.

Hardware reverse engineering involves taking apart a device to see how it works. For example, if a processor manufacturer wants to see how a competitor's processor works, they can purchase a competitor's processor, disassemble it, and then make a processor similar to it. However, this process is illegal in many countries. In general, hardware reverse engineering requires a great deal of expertise and is quite expensive.

SHORT ANSWER TYPE QUESTIONS

1. What is Software Reliability? How we can achieve the Reliable Software?
2. What is ISO Standards?
3. What are the different CASE Tools? Explain in Details.
4. What is the importance of CMM Model? Explain.
5. What is the Reverse Engineering? Why we use it?
6. Explain the following:
 - (a) CASE Environment
 - (b) Difference between CMM and ISO

LONG ANSWER TYPE QUESTIONS

1. What is the importance of CASE in Development of Software? Give the suitable example to support your answer.
2. Explain the concept of Case Environments in details.
3. What is the use of Case tools in Software Development? Explain in details.
4. Mention the Comparison between SEI and CMM in details.
5. What is the role of Software Configuration Management (SCM) in software development? Explain in details.

OBJECTIVE TYPE QUESTIONS

1. Which of these are valid software configuration items?
 - (a) Software tools
 - (b) Documentation
 - (c) Executable programs
 - (d) Test data
 - (e) All of these
2. Which of the following is not considered one of the four important elements that should exist when a : configuration management system is developed?
 - (a) Component elements
 - (b) Human elements
 - (c) Process elements
 - (d) Validation elements
3. Once a software engineering work product becomes a baseline it cannot be changed again.
 - (a) True
 - (b) False
 - (c) Can not say