Assignment:- 3

Student Name: Vivek Kumar
UID: 21BCS8129
Branch: BE-CSE (LEET)
Section/Group: 20BCS-809/A
Semester: 4th Sem
Date of Performance: 10/05/2022
Subject Name: Software Engineering
Subject Code: 20CSP-255

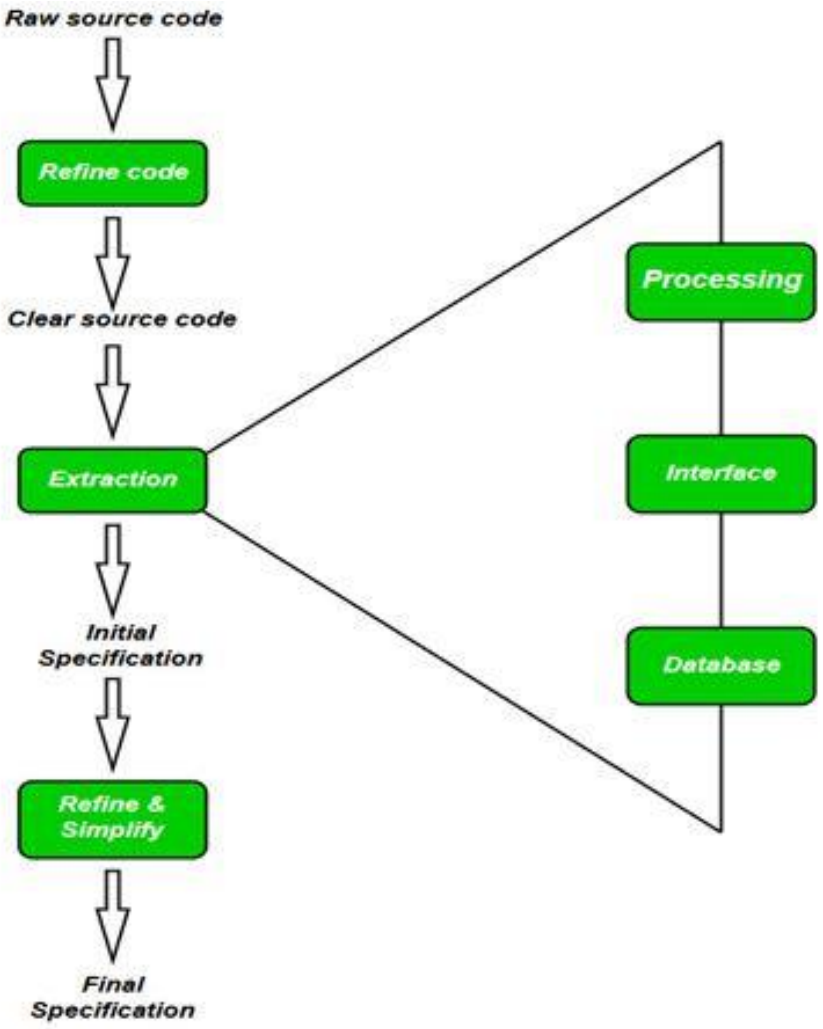
AIM: - Q1. Narrate the concept of software reuse briefly.

Reusing programs and other artifacts has been shown to be an effective strategy for significant reduction of development costs. This article reports on a survey of 128 developers to explore their experiences and perceptions about using other people's code: to what extent does the "not invented here" attitude exist? The survey was structured around a novel and simple "4A" model, which is introduced in this article: for an organization to obtain any benefits from reusing code, four conditions must obtain: availability, awareness, accessibility, and acceptability. The greatest impediments to reuse were shown to be awareness of reusable code and developers' perceptions of its acceptability for use on their new projects. For 72% of developers, the complexity of the old code was cited as a reason that the code was not reused. The survey also included developers' suggestions for ways to take greater advantage of existing code and related artifacts.

Software Reverse Engineering is a process of recovering the design, requirement specifications and functions of a product from an analysis of its code. It builds a program database and generates information from this. The purpose of reverse engineering is to facilitate the maintenance work by improving the understandability of a system and to produce the necessary documents for a legacy system.

Reverse Engineering Goals:

- Cope with Complexity.
- Recover lost information.
- Detect side effects.
- Synthesise higher abstraction.
- Facilitate Reuse.



Steps of Software Reverse Engineering:

1. **Collection Information:**

This step focuses on collecting all possible information (i.e., source design documents etc.) about the software.

2. **Examining the information:**

The information collected in step-1 as studied so as to get familiar with the system.

3. **Extracting the structure:**

This step concerns with identification of program structure in the form of structure chart where each node corresponds to some routine.

4. **Recording the functionality:**

During this step processing details of each module of the structure, charts are recorded using structured language like decision table, etc.

5. **Recording data flow:**

From the information extracted in step-3 and step-4, set of data flow diagrams are derived to show the flow of data among the processes.

6. **Recording control flow:**

High level control structure of the software is recorded.

7. **Review extracted design:**

Design document extracted is reviewed several times to ensure consistency and correctness. It also ensures that the design represents the program.

8. **Generate documentation:**

Finally, in this step, the complete documentation including SRS, design document, history, overview, etc. are recorded for future use.

AIM: - Q2. How software reverse engineering is helpful in development of software. Discuss in detail?

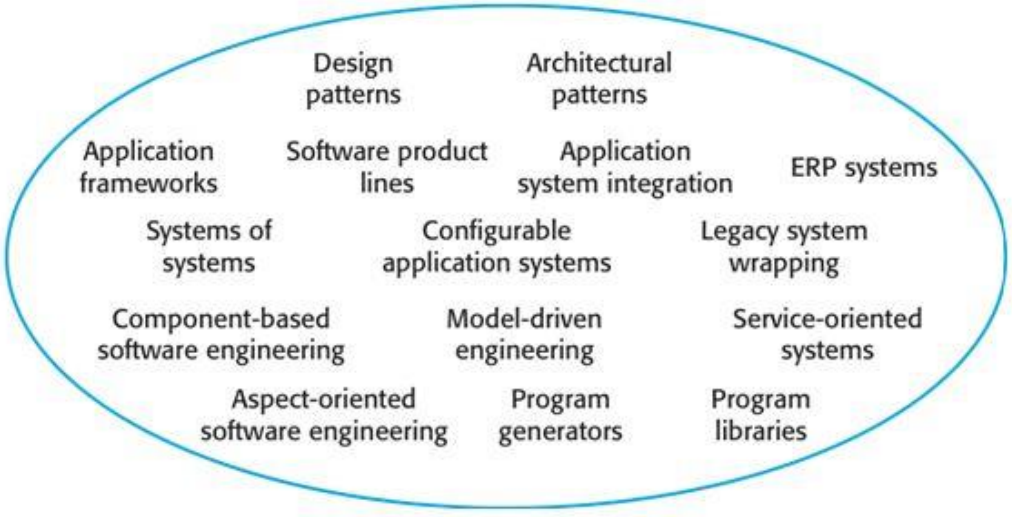
The purpose of reverse-engineering is to find out how an object or system works. There are a variety of reasons to do this. Reverse-engineering can be used to learn how something works and to recreate the object or to create a similar object with added enhancements.

Often the goal of reverse-engineering software or hardware is to find a way to create a similar product more inexpensively or because the original product is no longer available. Reverse-engineering in information technology is also used to address compatibility issues and make the hardware or software work with other hardware, software or operating systems that it wasn't originally compatible with.

How does the reverse-engineering process work?

The reverse-engineering process is specific to the object on which its being performed. However, no matter the context, there are three general steps common to all reverse-engineering efforts. They include:

- **Information extraction.** The object being reverse-engineered is studied, information about its design is extracted and that information is examined to determine how the pieces fit together. In software reverse-engineering, this might require gathering source code and related design documents for study. It may also involve the use of tools, such as a disassembler to break apart the program into its constituent parts.
- **Modeling.** The collected information is abstracted into a conceptual model, with each piece of the model explaining its function in the overall structure. The purpose of this step is to take information specific to the original and abstract it into a general model that can be used to guide the design of new objects or systems. In software reverse-engineering this might take the form of a data flow diagram or a structure chart.
- **Review.** This involves reviewing the model and testing it in various scenarios to ensure it is a realistic abstraction of the original object or system. In software engineering this might take the form of software testing. Once it is tested, the model can be implemented to reengineer the original object.



Although reuse is often simply thought of as the reuse of system components, there are many different approaches to reuse that may be used. Reuse is possible at a range of levels from simple functions to complete application systems. The reuse landscape covers the range of possible reuse techniques.

Key factors for reuse planning:

- The development schedule for the software.
- The expected software lifetime.
- The background, skills and experience of the development team.
- The criticality of the software and its non-functional requirements.
- The application domain.
- The execution platform for the software.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			
4.			