2 March 2023

SRS Documenting process Report

ChemistryCalculator

Submitted To
Dipok Chandra Das
Assisstant Professor, IIT, NSTU

Team Members -
Emran Hossain [ASH1825002M]
Md Azad Hossain [ASH1825014M]
Al Shahriar Priyo [ASH1825020M]
Nadim Bhuiya [ASH1825034M]
Akash Chandra Debnath[ASH1825037M]

We followed the **Reverse Engineering (RE)** process to produce a **Software Requirement Specifications (SRS)** document from **ChemistryCalculator** application source code.

1. Reverse Engineering

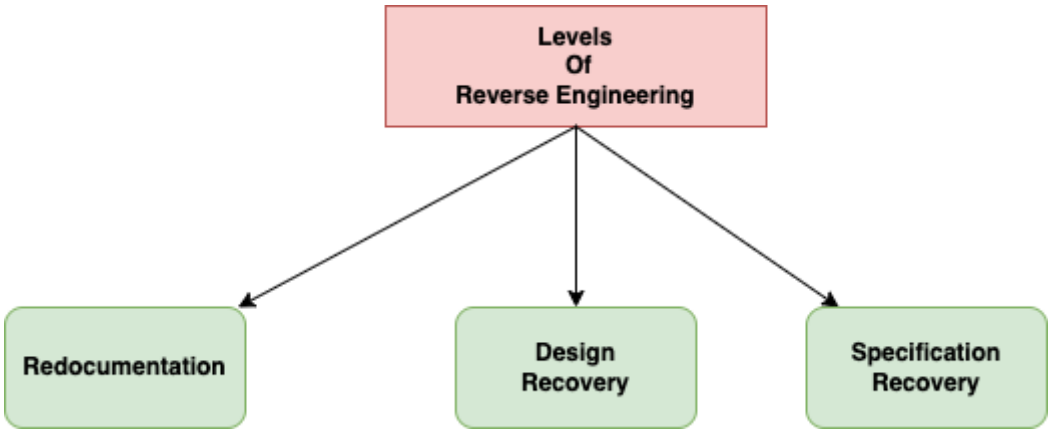
Reverse engineering is the process of dissecting a system to determine its constituent parts and how they relate to one another in order to represent the system in a different way or at a higher level of abstraction.

2. Objectives

- ➔ To recover lost information
- ➔ To generate documentation
- ➔ To reduce or analyse complexity
- ➔ To reduce maintenance effort

3. Levels of Reverse Engineering

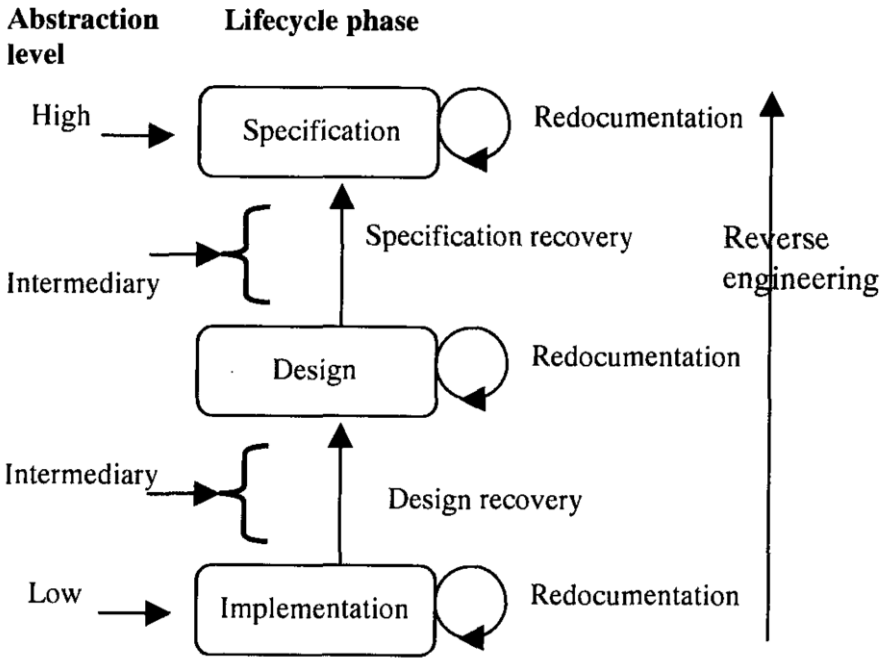
The levels of reverse engineering can vary depending on the scope and complexity of the object being analyzed, but generally, there are three main levels of reverse engineering.



I. Redocumentation: Redocumentation is the term used to describe the operation if it is performed at the same level as the original system.

II. Design Recovery: The operation is referred to as "design recovery" if the final product is at a higher level of abstraction and the implementation phase is at a lower level.

III. Specification Recovery: The operation is referred to as "specification recovery" if the final product is at a higher level of abstraction and the design phase is at a lower level.



We followed "**design recovery**" and mainly "**specification recovery**" to produce SRS of the **'ChemistryCalculator'** desktop application.

3.1 Details of the Process

As source code was provided so we were finding and extracting useful higher-level abstractions in addition to those found through a direct examination of the source code is what design recovery entails. The specification can be obtained during this procedure either directly from the source code or by applying **backward transformations** to current design representations. The recovery process can be greatly aided by information discovered from other sources, such as system and design documentation, prior knowledge, and problem and application domain expertise.

3.2 Problems Faced during the Process

There are many issues that occurred during the process of reverse engineering, which is difficult and complex. Several typical issues encountered during reverse engineering include -

1. Missing design documents
2. Unnecessary source code complexity
3. Poorly structured source code
4. Worked on others source code without any system documentation
5. Time constraints

Since we are provided only the source code of the ChemistryCalculator application, we must **understand source code** to produce artifacts (SRS) and get interconnections among them.

4. Program Comprehension

Program comprehension is the process of understanding how a software program works by analyzing its source code. It involves a combination of techniques, such as reading, navigating, and visualizing the code, as well as understanding the underlying algorithms and data structures used in the program.

5. Program Comprehension Process Model

There are three actions in this model.

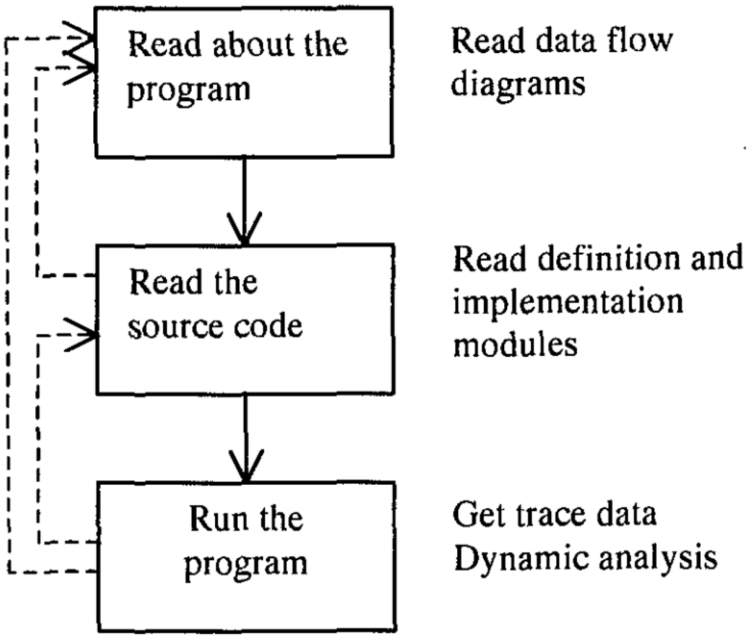


Figure: Comprehension Process Model

Step-1: Read About the Program - In order to develop an overview or general understanding of the system at this stage of the process, the "understander" browses and peruses various information sources, such as the specification and design documents for the system. Structure charts, data flow diagrams, and control flow diagrams are examples of documentation aids.

Step-2: Read the Source Code - The "global" and "local" views of the program can be obtained at this stage. The "global" view is used to get a high-level understanding of the system and to estimate the size of any potential ripple effects on other system components. Programmers can concentrate their attention on a particular area of the system using the "local" view. This viewpoint provides details on the architecture, data types, and algorithmic patterns of the system. During this stage, tools like static analyzers, which look at source code, are used.

Step-3: Run the Program - This step aims to investigate the dynamic behavior of the program while it is being used, such as by running the program and gathering trace information. Running the program has the advantage of revealing system characteristics that are difficult to discern from the source code alone.

We followed all the three steps for making the **SRS** of ChemistryCalculator application.

5.1 Problem Faced During Program Comprehension

We encountered a number of issues while attempting to comprehend a software program from its source code because program comprehension is a difficult process.

- Limited resources [No advanced tool]
- Code complexity
- Lack of proper code structure and architecture
- No design documents or schema