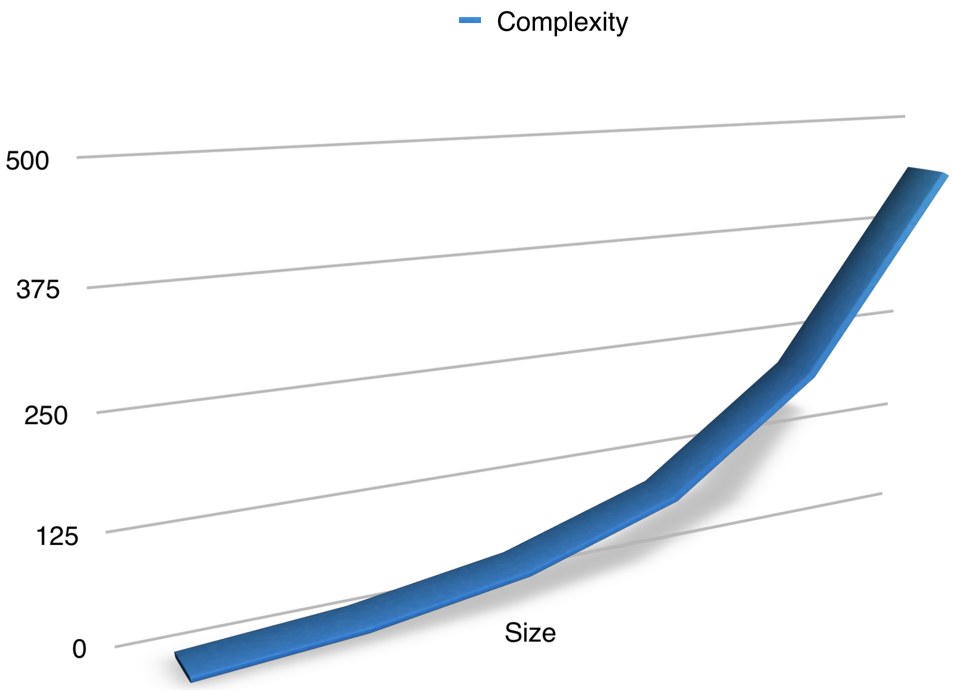
SE Assignment I

1.

A. **Trace the problem one would face, if he tries to develop a large software product without using software engineering principles. Define the term software engineering.**

Without using software engineering principles it would be difficult to develop large applications. The problem with developing large software is that the complexity and difficulty levels increase exponentially with the increase in software size.



For example a program of size 1,000 lines of code has some complexity. But a program with 10,000 LOC is not just 10 times more difficult to develop, but may as well turn out to be 100 times more difficult unless software engineering principles are used.

Software engineering is the application of engineering to the design, development, implementation, testing and maintenance of software in a systematic method.

B. **State four symptoms of the present software crisis. Show at least three important drawbacks of the exploratory programming style.**

Symptoms of the software crisis:

- Organizations are spending larger and larger portions of their budget on software.
- Software products turning out to be more expensive than hardware.
- Software products are difficult to alter, debug, and enhance; use resources non-optimally; often fail to meet the user requirements; are far from being reliable; frequently crash; and are often delivered late.
- The expenses that organizations all around the world are incurring on software purchases compared to those on hardware purchases.

Drawbacks of the exploratory style:

- Insufficient for developing large and complex programs.
- Difficult to write cost effective and correct programs using the exploratory style.
- Difficult to understand and maintain the programs which were written by others.

2.

A. **List at least four basic characteristics that differentiate a simple program from a software product. What is the aim of software engineering?**

Programs are **small** in size and have limited functionality but software products are extremely **large**.

In case of a program, the programmer himself is the **sole user** but on the other hand, in case of a software product, most users are **not involved** with the development.

In case of a program, a **single developer** is involved but in case of a software product, a **large number of developers** are involved.

For a program, the **user interface may not be very important**, because the programmer is the sole user. On the other hand, for a software product, **user interface must be carefully designed** and implemented because developers of that product and users of that product are totally different.

The main aim of software engineering is to apply engineering to the design, development, implementation, testing and maintenance of software in a systematic method.

B. **State at least two basic differences between control flow-oriented and data flow-oriented design techniques and State at least two advantages of object-oriented design techniques.**

Control flow oriented design deals with carefully designing the program's control structure.

A program's control structure refers to the sequence in which the program's instructions are exegguted – the control flow of the program.

But data flow oriented design technique identifies:

Different processing stations (functions) in a system.

The data items that flow between processing stations.

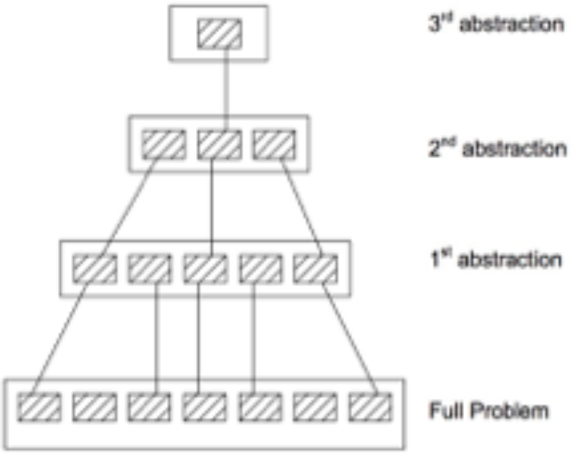
3.

A. **State at least three differences between the exploratory style and modern styles of software development. Illustrate the use of abstraction principle in software engineering.**

An important difference is that the exploratory software development style is based on error correction while the software engineering principles are primarily based on error prevention.

In the exploratory style, coding was considered synonymous with software development. Developing a software product with exploratory programming style believed in developing a working system as quickly as possible and then successively modifying it until it performed satisfactorily. In the modern software development style, coding is regarded as only a small part of the overall software development activities. There are several development activities such as design and testing which typically require much more effort than coding.

Abstraction in simplifying a problem by omitting ~~useless shit~~ unnecessary details; focussing attention on only one aspect of the problem and ignoring other aspects and irrelevant details. It is used in the same way in software engineering, by hiding the unnecessary details of the project and thus helping in completion of the project by making it simpler.



B. **Mention two important features that a program must satisfy to be called as a structured program. Illustrate the use of decomposition principle in software engineering?**

A program is called a structured when it uses only the following types of constituents:

Sequence, **Selection** and **Iteration**.

Structured programs avoid unstructured control flow statements like JUMP or GOTO. They consist of a well partitioned set of modules.

In decomposition, a complex problem is divided into several smaller problems and then the smaller problems are solved one by one. The problem has to be decomposed such that each component of the decomposed problem can be solved independently and then the solution of the different components can be combined to get the full solution. A good decomposition of a problem should minimize interactions among various components.

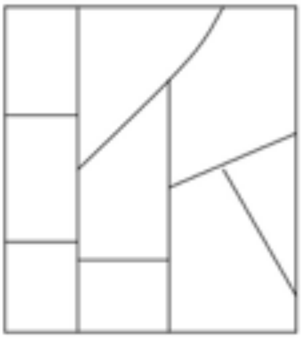
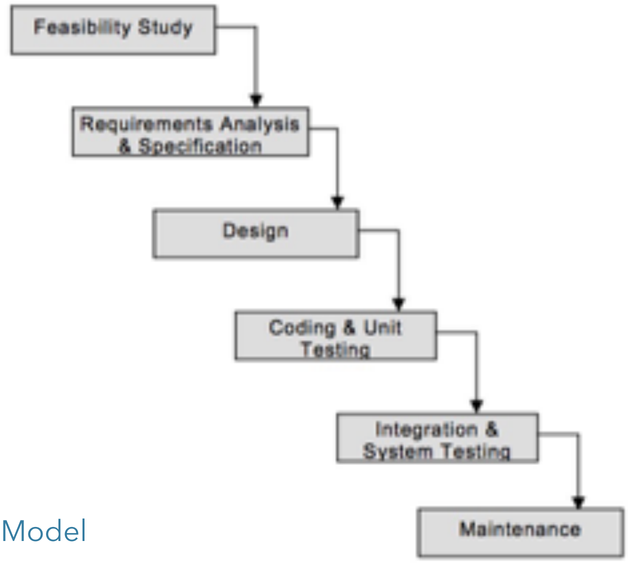


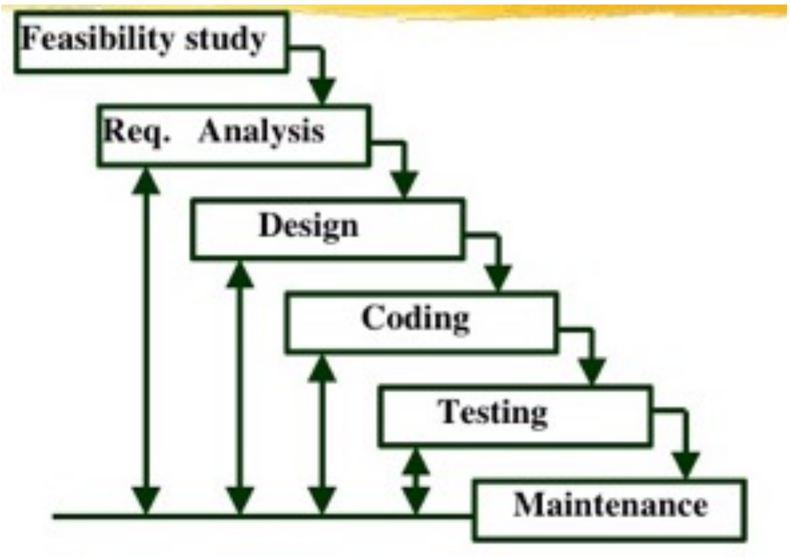
Fig. 1.5: Decomposition of a large problem into a set of smaller problems.

4.

A. **Illustrate waterfall model and Iterative waterfall model and list the differences.**


Feasibility Study
Requirements Analysis & Specification
Design
Coding & Unit Testing
Integration & System Testing
Maintenance

Waterfall Model



Iterative Waterfall Model

We use the classical waterfall model when we're well aware of the requirements beforehand, and iterative waterfall when we're unsure of the project requirements.

The iterative waterfall model is cyclic, it's possible to go from one stage to the previous one, whenever a project is not satisfying the demands of the customer.

B. What do you mean by Phase Containment of Errors? Which particular model will eliminate this error?

Phase containment in a nutshell is finding and removing bugs/defects early in the process of SDLC. It is the act of containing faults in one phase of software development before they escape and are found in subsequent phases. An error is a fault that is introduced in the current phase of software development.

To achieve phase containment of errors we have to take periodic reviews. Iterative waterfall model will eliminate this error.

5.

A. **What are the objectives of feasibility study phase of software development? Explain the important activities that are carried out during feasibility study.**

The main aim of feasibility study is to determine whether it would be financially an technically be achievable to develop the product.

Activities carried out:

- Managers and the team leaders visit the client to understand the project.
- They study the shit the clients wants them to do, whilst being high.
- With a good knowledge of the project, they think of various solutions of the project.
- They then analyze these solutions and come up on a model to solve the given problem, well under the client's budget, giving the client the best value for his money.

B. **Explain the problems that might be faced by an organization if it does not follow any software life cycle model.**

- The development of the software would not be in systematic manner.
- The design of the software may not be proper.
- The software might not be properly tested.
- The software might not be well maintained.
- There might be chaos among the developers leading to excess drug consumption, which may result in improperly made software, which may perhaps lead to destruction of this universe.