CHAPTER 1:

INTRODUCTION

1.1 LARGER SOFTWARE PRODUCTS SCENARIO

Software has become a critical issue in modern society. Everyone seems to need more and better software faster and cheaper. Many development projects are now so large and complex that a few brilliant specialists can no longer handle them. Unfortunately, there is no sign of a magic new technology to solve these problems. The alternatives are to improve the performance of practicing software engineers while encouraging more people to enter the field. This thesis provides a framework based on the experimentations and observations for understanding why we make errors and how best to find them. We can determine the quality of our reviews, the error types we miss, and the methods that are most effective for us [49, 26]. Some organizations have addressed the problem of developing large-scale software systems by adopting the concept, taken from the manufacturing community of the defined and managed process. By properly managing the software process at the project and laboratory level, these organizations have successfully improved the capabilities of their development groups [48].

As our products become larger and are used in increasingly critical applications, the potential of damaging errors will almost certainly increase; just about any software engineer could commit one. The software industry has responded to this threat by resorting to increasingly rigorous and time-consuming tests. However, this testing strategy has not been totally effective. The only responsible choice left is to improve the working disciplines of software professionals.
Software Development Strategy with High Quality Design for Large Scale Projects

The software development strategy may be defined as a simplified description of a software process, presented from a particular perspective. Software development strategies play a very important role in the development of large scale software engineering projects [21, 52]. Choosing the adequate development strategy is an important managerial decision for the success of the project. The development strategy determines the sequence of tasks required to accomplish the project.

Historically development strategies were originally proposed to streamline the entire phase of software development. At present there are many software development strategies which are used depending on particular needs of the software being developed.

In spite of 50 years of software development methodology and process improvement, we need a new paradigm to develop large scale software systems. [14, 18] Productivity gains in software development have not kept up with the Performance increases in hardware. New hardware technology enables and encourages new applications, which require much larger and more complex programs. Perhaps a dozen models of software development aim to improve development productivity and/or enhance quality. [12] They all work reasonably well when faithfully and diligently applied. [9, 10]

The objective of the software developers is to use the software development strategy to deliver the quality product within deadlines and budget. “Software development strategy is an abstract representation of a software process”.

1.2 MOTIVATION

- **In spite of 50 years of software development** methodology and process improvement, we need a new paradigm to develop **large scale software systems**.

- **Productivity falls** off sharply when we develop larger programs.

- Due to the added tasks that come with larger projects, a significant part of the problem is caused by **defects**.

- Write larger programs, the **difficulty of finding and fixing problems** increases exponentially.

- If allowed to develop **consistently very-high quality small programs**, produce better products also substantially improve individual and organization’s productivity.

- Unfortunately because of **various constraints** and the work environment such kind of freedom is very rare and developer needs to work on large scale projects directly.
1.3 PROBLEM DOMAIN

- Smallest project programs without highest quality, hard to test, take time to integrate into larger systems, and be cumbersome to use.
  
  – Then in case of large scale projects?

- As we work to improve the quality of the software we develop, we should focus the ability of our process to produce quality products.

- In case of large scale projects seek the most effective development strategies with proper process model to find defects as well as the most effective ways to prevent them.
  
  – It will definitely improve the quality of the design.

- Also recognize that the costs of finding and fixing defects escalates rapidly the longer we leave them in the product.
1.4 PROBLEM STATEMENT

- One potential problem with all the strategies when apply to the large scale projects is that some critical function may not be developed until the later development cycles.

- Double problem of delaying exposure to the principal project risks and of testing the most complex element for the first time.

- If the rest of the system has many defects, this could greatly complicate testing of the later slices.

- Here we are suggesting the improvement in the strategy that tests the most difficult components with the smallest amount of untested code.

- The defects are then more easily identified and fixed which enhance the software engineering performance.

- The suggested development strategy is thus to ensure our program is of the highest quality design with cyclic reviews and verification approach when we first produce it.