SYNOPSIS

During the eighties, the computer has changed the fabric of our society because of incredible computing power, now available on the 'chip'. The software technology is also imparting revolutionary impact on our lives and causing 'software exigency'.

The dilemma is that the software problems we face today are similar to those which were encountered in the sixties and were responsible for 'software crisis'. Although we are improving the computer hardware, the problems of software project management - cost overrun, schedule slippage, user unfriendly and unreliable software are still growing. This is because of complex nature of software and misunderstood behaviour of software projects resulting in wrong and unpragmatic estimates of resources.

Life cycle behaviour is the characteristic of all software projects i.e. a software has to go through seven phases namely, requirements analysis, specification, design, coding, testing, documentation, and operation & maintenance. In this thesis our endeavour has been to apply a quantitative macroscopic methodology - a pragmatic approach to software projects entire life cycle for resource estimation. Conceptually, given initial parameters - certain facts about a software project that are
available in the beginning - this approach generates an expected curve of entire life cycle effort against time. This dynamic and multi-variable approach provides answers to management questions in terms of management parameters and can be very well used for real-time dynamic control of software projects.

Putnam has applied Rayleigh-Norden model to selected phases of software life cycle. Our approach has been an integrated one and we have applied R-N model to entire life cycle. The analogous behaviour of software projects has also been investigated which is found to be very useful for project simulation.

An estimating system (SILES) has been evolved using effort and development time parameters which are related to system size - code and documents - through a constant representing development and software engineering environment. This methodology is quite pragmatic and useful for managing software projects.

This thesis entitled "Software Project Resource Estimation: An Integrated Life Cycle Methodology" is divided into nine chapters. Chapter 1 is an introduction to the thesis. Chapter 2 gives an overview of a project in general and software project in particular. Chapter 3 is an investigation into various phases of software life cycle. Chapter 4 presents our software progress model. Chapter 5 gives our estimation methodology. Chapter 6 deals with our investigation into project dynamics. Chapter 7 discusses our observation of analogous behaviour of software projects. Chapter 8 briefly describes the software
package SILES developed for software resource estimation. Chapter 9 concludes the thesis. A summary of each chapter is given below:

Chapter 1 - Introduction

In this chapter, a brief account of the fundamental ideas and features characterising the software problems and historical perspective of software crisis and software exigency is given. This forms the background of our work and explains why this particular area for research has been identified.

Chapter 2 - Software Project and its Management

This chapter conceptualizes a project. It distinguishes between programs, system and product. The effort-time involvement, their complexity, and the problems of project management and their solutions are investigated.

Chapter 3 - Life Cycle of a Software Product

It gives the historical background of the life cycle approach and includes an example of Airline Software Product. The various phases of software life cycle and the software tools which are helpful in these phases are investigated.

Chapter 4 - Quantitative Project Progress Model

This chapter is a basis for our mathematical formulation of Rayleigh curve for behaviour of software projects. It gives analysis of Progress data, concepts of success function, progress rate and objectivity, generalized success models and the particular case of linearly increasing progress.

viii
Chapter 5 - Integrated Life Cycle Methodology for Resource Estimation

This chapter contains reference to conventional estimation technique used in engineering disciplines and the approach applied for software projects. It contains our investigation into resource models, micro and macro approaches, quantization, Integrated Life Cycle equation, empirical evidence, linear ILC equation; Software complexity, manageability, and dexterity; Iso-tech lines, ILC throughput equation and effort-time trade-off relation.

Chapter 6 - Dynamic Control of Software Projects and Analogous Systems

This chapter includes our investigation into Project dynamics, dynamic ILC equation, dynamic physical model of software process - mechanical and electrical systems; experimental analogs, project data, requirements changes and systems growth.

Chapter 7 - Simulation of Software Projects

This chapter describes software projects as continuous systems and our techniques of using CSMP for simulation and includes feedback control, auto-controller, interactive and real-time simulation.

Chapter 8 - Software Project Integrated Life Cycle Resource Estimation System

This chapter briefly describes the better known estimation systems and gives functional overview of a comprehensive Software Project Integrated Life Cycle Estimation System
(SILES) developed by the author. It is interactive and gives information about size, time, effort, manpower, cost, risk and trade-off of a software project. An output of the interactive session with SILES is included as appendix.

Chapter 9 - Conclusion

This summarizes main features of the work carried out - our findings; applicability of ILC Methodology and our recommendations.

Summarily, the main contributions embodied in this thesis are as follows:

1. Exposure of current state of affairs in Software Technology.
2. Identification of problems of Software Project Management (SPM) and qualitative guidelines.
3. Analysis of phases of Software Project Life Cycle and useful software tools.
4. Formulation of quantitative project progress model.
5. Formulation of Integrated Life Cycle (ILC) Equation in terms of effort and development time, linear ILC equation, Iso-tech lines, ILC throughput equation in terms of development effort, time and Software Engineering Environment Factor.
6. Dynamic ILC equation, analogous models and experimental analogs for software engineering laboratory.
7. Simulation of software projects using CSMP.
Our research during the last couple of years shows conclusively that the phenomenon of software development is quite complex - dynamic and stochastic. The developed ILC Methodology is significant from software users as well as developers point of view for estimating software project resource and its monitoring.