CONCLUSION

"The tar pit of software engineering will continue to be sticky for a long time to come. One can expect the human race to continue attempting systems just within or just beyond our reach; and software systems are perhaps the most intricate and complex of man's handiwork. The management of this complex craft will demand our best use of new languages and systems, our best adaptation of proven engineering management methods, liberal doses of common sense, and a God-given humility to recognize our fallibility and limitations."

F.P. Brooks, Jr.
The Mythical Man-month (Broo75)
Our investigations during the last couple of years show very conclusively that the software development process exhibits a fundamental characteristic behaviour. The process is characterised by complex human inter-communication necessary for transforming real-life concepts into abstract machine instructions - to be accepted and executed by computer to display effectively the required performance.

This requirement or expectation from software is increasing very fast while the software technology is not developing that fast resulting in software exigency. In a way situation is similar to the times of Charles Babbage!

People have been trying to plan and manage software development process deterministically assuming it to be an ordinary linear process. In fact, the process is stochastic in nature because of changes in requirements, fluctuations in management parameters and inherent entropy in the process.

The phenomenon of software development is quite complex. It is a dynamic process and there are considerable 'noise' components present. The observed parameters - manpower, money and time - are subjected to management perturbation due to indecision and otherwise. Also, the problem of imprecise and continually changing requirements and specifications further complicate the situation. Our Integrated Life Cycle Methodology is significant from two perspectives (Fig. 9.1).
Firstly, user (customer) point of view - normally a users group/organization contracts out the task of software development as a project to a software development group/organization (contractor) for a set of functional requirements (RFP), so it must know the ILC estimates for the effort, cost and time required so that it can justify funding the project in order to get the best product for the money and that remains operational throughout its life time.

Secondly, development organization or contractor point of view - in the beginning of the project i.e. RFP stage, developer/contractor needs ILC estimates for planning and costing purposes so that when the project is underway, these estimates provide the project profile and trend of manpower, cost and time which is required for dynamic monitoring at the management level.

The work breakdown structure (WBS) (Taus80), daily reporting of minute details, Gnat charts (Shoo83), PERT/CPM analysis etc. for project execution and control purpose are outside the scope of ILC Methodology.

ILC Methodology deals with the aggregate behaviour of software development process exhibited by cumulative effort of people applied for developing software products. It is not meant for the analysis of behaviour of individual programmers effort to produce small programs. It becomes valid when the individuals are assigned work and brought together into a group or team effort to carry out the total task, then the group dynamics, group progress and stochastic behaviour comes into play. As such
there are no rigid rules but when two or more of the following criteria are satisfied by a project ILC Methodology may be applied:

1. Project team consists of \( \geq 4 \) people
2. Project duration \( \geq 8 \) months
3. Project budget \( \geq \text{Rs. } 1.0 \) lac
4. Product size \( \geq 1000 \) pages (code and documentation)

This methodology is applicable to all types of software products - new stand-alone systems, rebuild of old systems, embedded real-time or control systems, scientific or commercial systems. However, this is not applicable to software porting from one system to another or conversion from one language to another because this does not require phasing of effort in integrated life cycle mode – connectivity and homogeneity of effort.

Application of ILC Methodology to software systems explains the behaviour of software development process and we have found good pragmatic ways to valid system characteristics so that management questions are answerable quantitatively. Summarily our 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 ($\phi$) and development time ($\tau$), linear ILC equation, Iso-tech lines, ILC throughput equation in terms of $\phi$, $\tau$, and $T$ (Software Engineering Environment Factor).

6. Dynamic ILC equation, analogous models and experimental analogs for software engineering laboratory.

7. Simulation of software projects using CSMP.


It is to be emphasised that understanding of the software development process is improving and data is becoming available which permits application of ILC Methodology. However, it is also necessary to understand the significance of accurate and adequate input to the estimation process. This input consists of the specifications available at the various phases of the project. These specs must be formalized (Kita82). Estimation is an interactive and iterative process. It improves with the progress of the project. It is, therefore, necessary to check the estimates at each defined stage. An estimate found at a preliminary stage may have $\pm$ 80% error. At this stage answers to the management questions - feasibility, cost, duration, manpower loading, risk & tradeoffs are sought. These are answered adequately by ILC Methodology. Having achieved proper documentation and estimation it is necessary to set clearly defined milestones to monitor the progress of the project with the help of ILC Methodology.
FIG. 9.1 - TWO PERSPECTIVES AND ILC METHODOLOGY