CHAPTER 2

REVIEW OF LITERATURE

2.1 INTRODUCTION

For the purpose of the study – ‘An Empirical Study on the Software Metrics Management Practices in In-House Development Centres’, the Researcher reviewed related books, international and national magazines published and unpublished articles, opinions expressed by senior professionals of IT industry and materials from related web sites. All these were reviewed in context, chronological order, content, sectors, and other relevant groups and clusters.

2.2 SOFTWARE ENGINEERING AND LIFE CYCLE MODELS

Jalote, Pankaj (2008) in his book – “A Concise Introduction to Software Engineering” identifies that industrial strength software is the actual area of the problem with respect to Software Engineering field and goes on to include factors such as schedule, cost and quality as the most important forces that address the problem area. Such a software is also expected to cater to the needs of users who expect quality in software. Besides the aforementioned factors, productivity which is a ratio between input and output, is another factor that plays an important role. There are several quality components such as usability, portability, maintainability, efficiency, reliability and functionality when quality is addressed. The author proposes to state that there are fundamental metrics that contribute to the development of industrial
strengths software and these metrics are very essential and are driving the development activity, besides mentioning the importance of handling changes during software development.

The author in his article on “Development Life Cycle Models” states that software developers follow some core Software Engineering principles to deal with the complex nature of software under consideration. Most importantly, according to the author, life cycle model forms the basis of any software development although many different life cycle models exist in reality. Any life cycle model deals with right from initial concept to software release, maintenance and even upgradation. The author is of the view that the models can be customised to suit the requirements of the client. The author insists on identifying and following a life cycle model to be used as a reference model for successful development of software, besides addressing key issues such as change management, jumping the deadline due to delays caused during development. The author advocates a cautious approach to development (http://zone.ni.com 2014).

The author in his article “SDLC Iterative Model” discusses about the properties of an iterative model that starts with an implementation of a minimum set of requirements done in an iterative way until the final product is made ready. This marks the absence of full and complete specifications about the requirements, but the solution implemented in several iterations addressing essential set of small requirements with each iteration ending with a newer version of the software. The author reveals that software development is an iterative process and it happens in small chunks over a period until it is fully developed. He introduces the concept of versioning through iterative development process (www.tutorialspoint.com 2014).

Yau, & Collofello, (1980), in their article “Some Stability Measures for Software Maintenance” state that major cost of software is the cost of
maintenance. Cost of maintenance is influenced by an important quality factor namely stability of software which is the resistance exerted by the software when there is a ripple effect occurring due to the introduction of changes in the software. The authors insist on the importance of stability of the program as a contributing factor during maintenance. As maintenance cost is more than the development cost, the stability metrics become more relevant and important according to the authors.

The authors Munassar & Govardhan (2010) in their article “A Comparison between Five Models of Software Engineering” opine that although in reality software development does not happen like in pure waterfall model, one can use the waterfall models as reference model for both development and maintenance in order to avoid some obvious issues that would come up as major issues in later stages. They discuss about the merits and demerits of the waterfall model such as the instructive nature of the waterfall model as one goes through it, although in practice, it is followed with appropriate modifications. The authors show their concern about increased cost of corrections when they are carried out at later stages, instead of correcting at the appropriate stages of development. The non-overlap feature of the waterfall model is only theoretical although in reality the stages of development keep overlapping. Another concern expressed by the authors is the long term nature of waterfall model as the final solution is seen only after a long time as the development takes place only in the later stages. Documentation and watertight nature of the waterfall model are the other important limitations according to the authors.

The author in his article “SDLC Agile Model” mentions the use of agile model which gives importance to rapid and frequent delivery with iteration and increments (www.tutorialspoint.com 2014). The strength of this approach lies in breaking a large problem into smaller problems and evolving
solution builds for the smaller increments as per the importance and relevance, with every build undergoing several iterations, with each iteration running from one to three weeks. In every iteration, there can be activities such as planning, analysis, design, implementation and testing. The important feature of agile model is that it believes that every project needs to be handled differently with customisation wherever required. Time boxing is another concept followed in agile methodology where time is given more preference over the features of the product. Pair programming, less documentation, frequent client interaction and delivery in increments are the other features of the agile methodology. The author advocates iterative and incremental model with time boxing concept that delivers the software functionality in several steps within a few weeks. Although there are many approaches to this kind of flexible, practical, and delivery oriented development strategy, the agile methodology gains its importance due to the flexibility and result oriented approach.

The author in his article makes an in depth analysis of yet another successful software development strategy known as “Prototyping Approach” (www.tutorialspoint.com 2014). The author classifies different prototyping approaches and their applicability, suitability and relevance during software development. He also cautions the developer that when internal computations are heavy in nature, the user interface design need not be seriously considered for prototyping as the requirements are fairly understood by the developers. According to the author, the prototypes can be broadly classified into Vertical and Horizontal prototypes. The Horizontal prototypes are used for understanding user requirements and user interface design whereas the Vertical prototypes are for understanding the functionality of the software and it is more technical in nature. Besides these, there are other types such as Throwaway or Rapid Prototypes, Evolutionary Prototypes, Incremental
Prototypes and Extreme Prototypes. Where there is too much of processing on data, prototyping is not advocated.

The author in his article mentions about the use of RAD - Rapid Application Development model which is based on development of prototypes in parallel by many groups, use of components, frequent feedbacks from the user, iterative and incremental approach (www.tutorialspoint.com 2014). The prototypes developed are never discarded in this approach as these are integrated and delivered rapidly. The author reveals the importance of RAD approach as one of the successful approaches to software development. This model essentially supports iterative and incremental approach for faster development of reusable prototype that ensures understanding of almost all user requirements both stated and hidden, besides being flexible to accommodate aspects that are encountered as the prototype is developed.

Boehm (1988) in his article “A Spiral Model of Software Development and Enhancement” maintains that the spiral model addresses various key concepts and critical factors such as prototyping as a means to evolve proof of concept and risk analysis, besides advocating suitability and applicability of the model for software development. This model essentially follows a risk driven approach which is the main framework on which the process is built.

Jalote, Pankaj (1991) in his book “An Integrated Approach to Software Engineering” mentions that major cost component of a computer system is the cost of software which is unreliable and very costly to develop. The author is of the opinion that the process is important while developing SW products. Quality and productivity can be achieved if appropriate processes are followed for development of software. The author also advocates the use of software metrics for achieving high level of quality and productivity besides mentioning that the focus must be laid on the software
development process as the quality and productivity depend so much on the process. The author also mentions that metrics need to be used proactively to develop a high quality and high productive software.

Basili & Weiss (1984) in their article "A Methodology For Collecting Valid Software Engineering Data" propose a data collection strategy for the study of software process. Data collection process was tested for feasibility and was found to be both useful and feasible. The data collection is based on the changes being made to the software during development and once the changes are implemented the data is collected. For ensuring accuracy of the data, concurrent validation is done which is based on the data collected through interview. The authors laid focus on data collection, data analysis, categorization and designing of data forms. The authors are of the opinion that change control, change management are essential activities to understand the nature of software development methodologies.

2.3 SOFTWARE PROJECT MANAGEMENT

The authors Pressman, Roger & Maxim Bruce (2015) mention that the Software Project Management is an all season activity followed in the entire life cycle within the Software Engineering framework. The all season activity otherwise known as umbrella activity begins before any technical activity starts and goes on happening throughout the modelling, construction, and deployment of computer software. The factors that have a great impact on Software Project Management are – people, product, process and project. According to the authors, focus must be given to people organization, team formation, motivation for high quality work and for effective communication. The authors clearly explain the difference between Software Engineering and Software Project Management and proceed to give details of activities to be performed in Software Project Management.
The authors treat the Software Project Management activity as an umbrella activity that happens throughout the development cycle of the software. The authors stress on the role played by people in Software Project Management, besides mentioning about metrics and measurement.

The authors Sonchan & Ramingwon (2014) in their article "Top Twenty Risks in Software Projects: A Content Analysis and Delphi Study" point out about the importance of risk management. The authors claim that it is a very essential process in the management of software project. The major activities in Risk Management are the identification and analysis. However, the authors caution that this is an activity to be performed only by expert developers as it needs a high level of knowledge and experience. The authors give an in-depth analysis of risk identification, control and management after identifying top risks from around 30 research publications with respective Software Project Management, besides the authors made use of delphi method for Risk Management which can be potentially used as a reference material by project managers.

The authors Pettersson et al (2008) in their article "A Practitioner's Guide to Light Weight Software Process Assessment and Improvement Planning" advocate iFLAP, a light weight framework on software process improvement (SPI) that performs better over most of the SPI frameworks which are either too costly or do not address the requirement of the user. The authors, besides giving guidance materials, also provide two industrial case applications using which the learning becomes easy for the users. The iFLAP framework is advocated to be the most suitable SPI framework for small and medium sized corporate users.

According to the author Fairley, Richard (1997), organizational structure, assigning work activities, and evaluating work activities are important aspects of Software Project Management. The author claims that
planning is required for both the process and product, and it is one of the neglected aspects. The author discusses all project management issues and goes on giving a number of examples, calculation of various project management factors such as cost estimation, time estimation and the like. The author brings in all the concepts together in a single place that addresses both the Software Project Management and Software Engineering. The author Boehm (1981) proposes a structured discussion on a few software metrics with respect to COCOMO, a particular model on software estimation.

The authors Thayer et al (1981) identified potential twenty problems from the survey they conducted on 249 people on topics such as problem importance, problem nature, problem solutions and proceed to suggest solutions to some of them. Among the problems mentioned were planning for project management, selection criteria for project managers, lack of standards and so on. They advocated the use of standards, procedures, documentation, training and educating top management.

The author Basili (1980) suggests a quantitative approach for the evaluation, control and prediction of cost of software development and maintenance. The author Cooper (1978) advocates solutions to deal with software management obstacles and proposes 'Standardization' as the most important management device for an effective corporate management of software development.

The author Jalote (2002) in his book “Software Project Management in Practice” discusses various software project practices followed in the Indian software development companies particularly Infosys. The statistics revealed by him is that the total value of software projects is in the order of US$ 600 billion by way of around a million projects being executed by half a million software project managers globally. Surprisingly most of these projects fail to deliver and address their goals. The reasons for
the failure are many but the most important one is the lack of Software Project Management. The author advocates the use of accumulated project experience in Software Project Management areas. The author advocates formality and rigor into Software Project Management practices, failing which he predicts that default result of software project failure. The author identifies some key areas of Software Project Management like Process Database, Process Capability Baseline, Process Planning, Efforts Estimation and Scheduling, Quality Planning, Risk Management, Project Monitoring and Control, and Project Closure.

Walker Royce (2008) in his book “Software Project Management – A Unified Framework” states that with the experience gained in Software Engineering, new approach to Software Project Management has emerged after replacing some of the conventional principles with the new ones. Competitive environment has emerged as a result of demand on features of software, finally resulting into reduced cost. The author opines that there have been many changes in conventional Software Engineering although some of the principles are still valid. Modern Software Engineering has undergone numerous changes from the old conventional approaches and principles due to various factors such as advancement in technology, changing customer needs, increased levels of competition and so on.

The author Ramesh, Gopalaswamy (2005) in his book “Managing Global Software Projects” proposes a number of metrics to be practiced at different stages in any software project and also explains how the metrics play a role in making the software a successful one. The author advocates a roadmap be formed for a goal oriented Software Project Management with techniques for measurement of metrics to evaluate the progress made in the project. The author details that there has to be a balance between Human finesse and Engineering expertise. The author concludes by saying data
captured and recorded in Project Closure brings about a formal framework to success in the project being managed. The author lays a foundation for Software Project Management and emphasizes its importance in software development. Besides detailing various areas under SWPM, the author gives a lot of focus on metrics and its uses. The author also insists on role of Process Database and Project Closure metrics and their importance in the management of global software project.

The authors Bloch et al (2012) in their article "Delivering Large-Scale IT Projects on Time, on Budget and on Value" narrate that out of the study McKinsey did research along with University of Oxford, the IT projects drive many organizations critically that if something goes wrong, the organization goes into risk as technology is spread almost to the entire organization. The authors report that around 50% of IT projects worth above 15 million US$ exceed their budgets. Overall more than 45% of the large IT projects exceed their budgeted cost, around 7% exceed their scheduled delivery time and deliver 56% less than the expected value. Software projects shoot beyond the cost and time estimates. The authors claim that out of 5400 IT projects surveyed by them, most of the projects had a cost overrun and longer duration projects invariably ran over time and budget with every additional year spent. The authors also claims that IT projects delivered 56% less value than what was expected of them.

The authors Jones & Mortensen (1995) narrate "As per IEEE standards, Software Project Management is the process of planning, organising, staffing, monitoring, controlling and leading a software project.” The authors mention that there must be a project manager to head every project and lead the team during development. The manager has to act as an interface between the developers and the users, management representatives, suppliers and other stakeholders. The job of the project manager is to draw
Software Project Management Plan (SPMP) and execute the project by organising, staffing, controlling reporting. Project manager is the force that drives and controls (http://microelectronics.esa.int/ 1995) the project and takes the project forward towards success. The authors stress on the importance of having competent project managers and the role to be played by them in managing software projects. At the same time the authors insist on having Software Project Management plan and various project management reports that give lots of feedback about the progress being made in the software development.

The author Sarfraz (2009) in his article “Managing for a Successful Project Closure” narrates that any project which is supposed to be a successful one will have its closure done well and effectively. Various problems, causes, situations, solutions are often encountered as the project gets executed and this information may be really a very valuable one as far as the present and future projects are concerned. Project Closure gives an opportunity to formally record such critical information so that in future projects can be executed carefully taking cues from the information recorded during Project Closure. The author mentions that a successful project needs to have a Project Closure. The author reiterates the need for a project manager who needs to play multiple roles in successfully managing software projects. The author is of the strong opinion that a winning project is aptly to be called as a well closed project.

2.4 SOFTWARE METRICS

The author Kan, Stephen (2003) in his book Metrics and Models in Software Quality Engineering” states that for a software to be successful, effective defect removal must be done during development. The experience and literature agree to take a stand that early defect detection and correction leads to improved quality of the software product. The author advocates
matrix approach to handle the activities related to defect removal effectiveness. The author also advocates the use of Rayleigh model for quality assessment and reliability. The Rayleigh model is most suitable particularly when the product is ready for shipment as it is used to make a projection of latent defects once the development work is over. The Rayleigh model addresses both prevention and removal of defects, besides being a proven framework on quality management. The author points out at the importance of defect removal at an early stage instead of discovering it during testing phase. Early detection improves defect removal effectiveness and the author establishes his claim. But, the author mentions the defect removal models are not predictive models. According to the author, the Rayleigh model which is a formal parametric model is a widely used model for reliability studies for quality management.

Arthur (1985) in his book on “Measuring Programmer Productivity and Software Quality” points out that a set of software quality metrics like efficiency, correctness, reliability and maintainability as a function of several software quality criteria. The author proceeds further by identifying the set of quality factors which is a function of nearly twenty two quality criteria. The author discusses in detail the quality metrics and the role played by them in software quality.

Woodfield, et al (1981) in their article “A Study of Several Metrics for Programming Effort” suggest that by managing programming complexity properly, it is possible to reduce the cost of development and cost of maintenance. In general, the measures created to address the complexity issues were based on experience and also on intuition of people with rich experience. The popular metrics in this connection are McCabe’s cyclomatic number v(G) and Halstead’s effort measure E. The authors compare these two metrics and go on to evolve another metric based model which is found to be
better than the available ones. The authors deal with programming complexity metrics and the impact it has on the cost of software. Whereas the complexity factor was measured using experience and intuition, the authors made use of four metrics namely McCabe’s cyclomatic number \( v(G) \), Lines of code (LOC), Halstead’s effort measures \( E \), and the author defined \( e \) metric.

Putnam (1978) in his article “A General Empirical Solution to the Macro Software Sizing and Estimating Problem.” advocates four important parameters to address the software development challenges namely delayed delivery and cost overruns. The four parameters are development time, technology, effort and elapsed time. These parameters are used to arrive at a macro methodology to evolve more accurate estimates in terms of time and cost. This model helps managers to take a rational decision based on a calculated risk. The calculation is so simple that it takes only a few hours to complete the estimates. The author comes to the rescue of project managers and estimators of application software development by introducing easy to find factors to be used in an estimation model to fairly estimate the cost and time of application development in a short period of few hours. However he cautions the project managers to use these factors with great care because of the differences prevailing in procedures and also the standards. The development of this model eventually led the author to come out with a model known as ‘Software Life Cycle Methodology (SLIM)’.

Henry & Kafura (1984) in their article “The Evaluation of Software Systems’ Structure using Quantitative Software Metrics.” claim to have had a positive experience by following a quantitative approach for the software structure. However, the design and analysis mostly depends on a qualitative approach. The design and implementation of a software system is based on the flow of information going in all possible paths through an identified component. According to the authors, the analysis and design is primarily a
qualitative work that could be quantified by working on quantitative structure portraying interconnection of components used in software development.

The authors Bouwers et al (2013) in their article “Software Metrics: Pitfalls and Best Practices” mention about their experience about the benefits of following metrics and also the disadvantages of following a wrong set of metrics. The authors also emphasize on doing research to improve the practices of metrics besides simply using the metrics as a routine practice. According to the authors metrics need to be placed in a context to reap maximum benefits. Additionally, the authors propose a model to define the quality of a software system. The authors caution against the wrong usage of metrics used out of context and the devastating side effects it can cause. The authors impress upon the challenges and pitfalls about using the metrics during software development, besides mentioning the advantages metrics generate.

The authors Khuttan, Anuj et al (2014) in their article “A Survey of Effort Estimation Techniques for the Software Development” discuss about the importance of estimating the efforts particularly at the early stage of software development. The authors have made a comparative study of Putnam’s COCOMO, and ANN-COCOMO II models that are used for software estimation and conclude that with development time, the Putnam model is very sensitive, the cost drivers are important for accurate predictions using COCOMO and the ANN-COCOMO delivers a more accurate estimate nearly close to actual software cost.

Chowdhury, Somak Roy (2014) in his article “Tips for Defining and Collecting IT Process Metrics” discusses the metrics program in the context of strategic, tactical and operational management. The ‘Why’ aspect is mapped to strategic management, the ‘What’ aspect is mapped to tactical management and the ‘How’ aspect is mapped to operational management by
the author. The author advocates the use of metrics created by one process owner by another process owner which according to the author creates enthusiasm rather than resistance.

Singh Kuldeep & Dwivedi Upendra (2014) in their article “A Survey of Various Cost & Effort Estimation Models” compare various popular estimation models on project costing besides addressing advantages and disadvantages of the models. The authors are of the opinion that well computed estimates lead to less error prone projects in terms of cost overruns and time overruns. Their contribution would help many researchers for developing more efficient and better models. They surveyed various estimations models to conclude that error rates are minimized if an accurate estimation is done, besides claiming that the effort estimation process is a real indicator of progress of a software project.

Jones, Capers (2014) in the article "Evaluating Software Metrics and Software Measurement Practices" evaluate a set of existing software metrics and practices. The author claims that in Software Engineering for more than 50 years or so, ineffective and inaccurate software metrics were practised. The author gives importance and focus on the productivity aspect and quality aspect while considering to analyse and evaluate the software metrics practices. The author advocates function point metrics for economics aspects and productivity purposes besides recommending defect removal efficiency as the very best suitable metric for quality aspect. The author emphasises on productivity and quality factors as contributing economic factors to software development in the modern world. The author points out that function points clubbed with defect removal efficiency stand to be one of the best choices of analysis of software quality.

Jose Gregory & Joseph Jusha (2014) in their article “Test Metrics and KPI’s” define what software metrics is all about and go on to describe
about the Key Performance Indicators (KPIs) as indicators to the state of critical goals of strategic importance. KPIs are computed using several metrics to measure the performance so as to take a stand on how things progress towards the attainment of long term goals of the organization. The authors describe about key performance indicators (KPIs) that stand to represent strategic goals of the organization. The KPIs draw inputs from various combinations of metrics enabling decision making process involved during software development life cycle.

One of the well known models of cost estimation is Wolverton’s model which is founded on past data taken from earlier projects. The model stands on three steps namely - identifying the module type, identifying the complexity level over a six point scale and finally estimation based on lines of code. The author Wolverton, (1974) in his article “The Cost of Developing Large Scale Software.” mentions that software forecasting on cost estimation can be thought of as a two step process. The first one is to make structural forecasts and the next one is to compute the forecast on money-volume aspect. The structural forecast is based on technology and function and it is not concerned with the project size whereas the money-volume process depends on either explicitly stated algorithm or a process that is fully mind oriented. It also depends on estimation algorithms on costs while considering the traditional forecasting methods such as bottom-up approach, top-down approach, standards approach and similarities and differences in the approaches. Two case studies are considered to show the forecasting method and the author identifies and suggests topics for further research.

Jalote (1991) in his book “An Integrated Approach to Software Engineering” mentions that all management activities depend on plans and therefore planning is supposed to be one of the most important activities in any Software Project Management. Any project that has no plan is most
likely to fail. Information is required to take managerial decisions, particularly during software development process and such information need to be quantified rather than being subjective. In order to quantify the data, software metrics are to be used in two types. The first type is process metrics and the other one is product metrics. Process metrics are for defining the characteristics of the process whereas product metrics are for defining the characteristics of the product. The author is of the opinion that a key to success in any project is to have a metrics based information and management. The author is of the opinion that subjective data must be replaced by objective data to evolve appropriate metrics for the effective management of software process and product. The author insists on metrics on having necessary metrics which is an important factor in Software Engineering activity. Any project not having appropriate metrics shall only march against the fundamental goals of Software Engineering.

Rubin (1987) in the article “A Comparison of Software Cost Estimation Tools.” claims that there are only a few cost estimation tools practised in the industry. The author identifies a common issue to be solved by popular estimation tools for easy comparison. The author makes an attempt to compare several software estimation tools such as GECOMO, SLIM, JS-2. The author concludes the article by mentioning solutions to a selected problem by applying different estimation tools.

The cooperative programming model (COPMO), as proposed by the authors Thebaut & Shen (1984), takes into account programming efforts spent and the coordinating efforts spent by all the members of the team. From the article, it is understood that following the empirical data, validations done point out that the model compares very positively with reference to factors of complexities with other models.
The authors Li & Cheung (1987) in their article “An Empirical Study of Software Metrics.” analysed 255 students’ fortran programs to study thirty one metrics through a fortran static source code analyzer known as FORTRANL. The authors concluded to do away with the metrics and proposed to have a new hybrid metric as the old metrics had problems of incompleteness.

Henry et al (1981) in their article “On the Relationships among Three Software Metrics.” claim that in the design, construction and maintenance of software systems, automatable metrics on software quality seem to have a number of benefits. The authors used three metrics namely the Halstead’s effort, the McCabe’s cyclomatic complexity and the Henry and Kafura’s information flow complexity. UNIX was the common software system chosen. The initial result showed that there was high level of correlation between the Halstead’s and the McCabe’s metrics, whereas the information flow metric seems an independent complexity measure. The authors in their attempt to study the relationship among the three complexity metrics namely Halstead’s effort, Henry and Kafura information flow complexity, and McCabe’s cyclomatic complexity finally found out that McCabe’s metrics had a high correlation with Halstead’s metrics whereas the third metric was an independent complexity measure.

Mohanty (1981) reviewed more than fifteen models that included models of Walston/Felix, Wolverton and Price-S, and identified around 49 project factors that had influence on the cost of software development. The author in his article “Software Cost Estimation: Present and Future.” points out that the estimation varies with the models used. Presence and absence of certain quality factors which are measures of goodness in design, development, testing and integration, also contribute to the variation. The
author strongly suggests to relate quality to the cost and proposes to use metrics of quality factors for estimating the software cost.

2.5 SOFTWARE METRICS CATEGORIES

Jalote (2005) in his book on “Software Project Management in Practice” discusses in greater detail various Metrics Management Practices as followed in the Indian software organization “Infosys”. The author mentions many metrics and metric categories of which some important metric categories such as Process Database, Process Capability Baseline, Process Planning, Quality Planning, Estimation & Scheduling, Risk Management, Project Monitoring & Control and Project Closure have been identified. The Process Data or Process Database reflects the experience gained in each and every project. This contains information about the project characteristics such as name of the project, names of all concerned project managers, domain type of the application, HW platform, languages used in the project and the like. Process Capability Baseline is a repository of information gained by recording experiences gained from the earlier projects done and evolving metrics out of them. Process Planning is about what processes or model should be followed for the development of software. Also includes critical activities such as process tailoring and change management. Efforts estimation is about calculating how much programmer hours are required for the execution of the project. Schedule metrics deal with the delivery time related metrics of the project. Quality planning metrics deal with the quality metrics parameters such as defects injected, defects unearthed during review, defect removal efficiency, and the like all in a quantifiable format. Risk Management metrics deal with such metrics as probability of a risk happening quantified and the impact it will have on the time and cost of the project. Project Monitoring & Control metrics are the factors or metrics that will give indication to the manager as to how the project is happening with respect to the planned
activity. And if any deviation is found, corrective action is to be initiated. Project Closure metrics are the collection of metrics gained by executing the project. This gets recorded in the Process Database for future reference and for forming baselines.

2.6 SECTORS UNDER CONSIDERATION

In this empirical research six Sectors namely Education, Engineering, Finance, Health, Logistics and Tourism were considered so as to select Respondents from these sectors. Information technology is all pervasive in nature and no sector is an exception to it. However considering the nature of the research, constraints and the research design chosen, it was decided to focus on these six Sectors for doing the empirical study. The following sections provide details about each Sector.

2.6.1 Sector Type - Education

According to the author, Education is offered by both the public and private sector with local funding, state level funding and central level funding and control (en.wikipedia.org 2014). As per the Indian Constitution, free education is provided to the children of age 6 to 14. India’s education system is a major contributing factor to the economic development of the country. More than 75% of the children with age group between 7 and 10 get the benefits of primary education. Scientific research and Higher education are enabled by public institutions. Private school system in India is very large that complements the government schools. Private institutions play a large and important role in India. The private education market in India reached around 450 million US$ in the year 2008, and is estimated to reach around 40 billion US$ market.
The Annual Status of Education Report (ASER) 2012 states that the enrolment percentage of all rural children in the age group 6-14 was around 96.5%. Although India is marching forward to match universal education, the quality of school education, particularly in the government schools, is under question. The private schools in India are highly regulated as to what they should teach and what type of institutions they need to be. There were 677 Universities as of 2014 as against 20 in 1950. There were 45 Central Universities, 318 State Universities, 185 State Private universities, 129 Deemed Universities, 16 IITs, 30 NITs, 5 IISERs, and 4 Institutions started under State legislations. There were 37204 colleges as of Mar 2013 as against 500 in 1950 (http://mhrd.gov.in 2014).

2.6.2 Sector Type - Engineering

The author in his article on “Indian Engineering Industry” claims that the Indian Engineering industry has seen an unprecedented growth over the past few years and this has attracted investments into the industry. This sector plays a strategic role in the Indian industrial economics especially its interlinking with sectors such as power, steel, infrastructure, oil and gas, automobiles, refinery, mining, and consumer durables. This sector is closely associated with manufacturing sector in the Indian economy (www.asa.in 2015)

The author in his article on “Manufacturing Sector in India” states that from 1991 onwards private sector took the focus from public sector in the manufacturing industry. There were a number of policies liberalized in favour of private sector. The manufacturing sector accounts for nearly 15% of the gross domestic product of the country and provides employment to 14% of the work force (dhi.nic.in 2014). However, India is surely lagging behind Thailand, China and other Asian countries. Road, Ports, Governance, Power and Labour policies are the deficiency areas for lagging behind. India is
towards building up and scaling up manufacturing capacities and the Prime minister’s efforts towards ‘MAKE IN INDIA’ policy is surely going to add value to the growth in the industry.

2.6.3 Sector Type - Finance

The author in his article on “Indian Finance Sector” notes that while it comprises of commercial banks, cooperatives, insurance organizations, pension funds, mutual funds, non-banking financial entities, it is in the process of massive expansion and other financial institutions. The service sector, of which the financial sector is a part, in the country accounts for 57% of the GDP. The banking sector holds nearly 60% of the total assets of the entire financial system of the country. Presently, India is considered to be one of the vibrant capital markets. Although challenges are ahead, yet the future of this sector remains very promising. The assets of banking sector have grown to 1.80 trillion US$ in India for the financial year 2013 and is expected to reach 28.5 trillion US$ by the financial year 2025. The insurance sector had 66.4 billion US$ as its a market size for the financial year 2013 and it is expected to cross 350 to 400 billion US$ by the year 2020. India’s expected corpus fund from the pension sector could go beyond 1 trillion US$ by the year 2025. In July 2014, the forex reserves of India went upto 321 billion US$ (http://indiainbusiness.nic.in 2014).

In the union budget of 2014-2015, a lot of measures have been taken to revive and accelerate the investments. The ‘MAKE IN INDIA’ programme is likely to attract both foreign and domestic investments. According to the report by CII and KPMG, India is going to become globally important economy with the support of insurance and banking sectors. The country is poised to secure the fifth position in the year 2020 with regard to the banking sector. The author explains about the present status of the Indian financial sector and the role played by it in the Indian economy. The author
also talks about the various measures initiated by the government of India to strengthen this sector, besides mentioning about the role of technology in contributing to faster rate of growth in this sector.

2.6.4 Sector Type – Health

The author in his article on "Healthcare Industry in India" states that in terms of both employment and revenue, the Healthcare sector encompassing hospitals, tele-medicines, medical equipment, medical insurance, medical devices, outsourcing and clinical trials, has become India’s one of the largest sectors (www.ibef.org 2014). The Indian Healthcare sector is one of fast growing sectors in India due to the private and public participation. The Government, in rural areas, maintains primary health centres popularly known as PHCs. Active participation by the private institutions addresses the major healthcare needs of Metropolitan and Tier 1 and Tier 2 cities of India. The strength of the Indian Healthcare sector lies in having well-trained medical professionals as compared to peer nations, besides having an advantage on cost over the Western Europe and U.S.A.

The Indian Healthcare sector is poised to touch 100 billion US$ by the year 2015 and 280 billion US$ by the year 2020 according to Price Waterhouse Cooper, one of the leading management and accounting consultants. The Government has plans to provide free drugs, diagnostics treatment and insurance cover to treat serious problems under the National Health Assurance Mission.

The projection for the medical tourism in India is around 3.9 billion US$ for 2015 and a report by KPMG says that the medical tourist population is expected to hit around 320 million by the year 2015. The author points out how the Healthcare industry contributes to the revenue and employment factors in India. The author mentions about the structure of healthcare systems
prevailing in India, the budget allocation done over the years, the growth achieved and so on. The author also predicts the Indian Healthcare sector to reach a positive high very shortly, besides mentioning about the growing importance of medical tourism happening in India.

2.6.5 Sector Type - Tourism

The author in his article on "Indian Tourism and Hospitality Sector" states that the Tourism and Hospitality sector is one among the largest service sectors of India (www.indiainbusiness.nic.in 2014). It is expected that the Gross Domestic Product (GDP)’s growth is going to be 7.8% p.a. during 2013-2023 by way of direct contribution by the Tourism and Hospitality sector mostly in the form of foreign exchange earnings. The tourism sector flourishes because of two ways – one is the arrival of foreign tourists and the other is the Indians opting to reach domestic destinations thoroughly all over the length and breadth of the country. As per the available statistics from the World Travel and Tourism Council, the sector in its domestic segment revenue gained by 5% during the year 2013 and is poised to grow by 8%. The Hospitality sector has witnessed a cumulative yearly growth of 14%. The Government’s policy and infrastructural support are the other factors that have contributed to the development and growth of this sector.

The Tourism and Hospitality sector in India touched 117.7 billion US$ in the year 2011 and is expected to cross 419 billion US$ by the year 2022. The number of foreigners who arrived in India during Jan-Jun 2013 was around 33.6 lakhs. For the period Jan-Jun 2014, foreign exchange earned was around 9.44 billion US$ whereas the corresponding period for the year 2013 stood at 8.58 billion US$, with growth rate registered at 18%. 
The Hospitality sector is expected to add new hotel rooms around 52,000 during 2013-2017 marking an increase of more than 65% capacity in the accommodation. The author states that the Tourism sector is yet another largest segment in the Indian economy. The Government’s policy on Tourism is to promote quick implementation of projects, development of tourism destinations and circuits, promotion of marketing activities and building up of special capacity in the Hospitality sector. The author also mentions about the role played by the Tourism sector in foreign exchange earnings and its contribution to GDP. The author gives details about the future plans and proposed in this sector and various initiatives of the present government including ‘Clean India Campaign’.

2.6.6 Sector Type - Logistics

The author in his article "Indian Logistics Sector to be Worth $385bn by 2015" states that the expected growth of the Logistics sector in India is around 20% and this will trigger a huge job opportunity in the logistics sector in the country (www.newindianexpress.com 2014). The logistics sector is going to be worth around 385 billion US$ during the year 2015. The focus is to be on the creation of warehouses, creation of a centre of excellence on road transport and also cold storage chain for a sustained growth in the sector.

The loss due to the lack of cold storage chain facilities in India amounts to around 44,000 crores of rupees for not being able to keep away vegetables, food grains and fruits from damages and deterioration. The cold chain capacity requirement in the country is around 62 million tons whereas presently it is around 30 million tons only. With the advent of online shopping, there is going to be huge potential for operations related to logistics and great job opportunities for the youth of India. At present, nearly 450 lakh
people have been employed in the sector whereas it is expected to reach a rapid growth very soon.

From the present 890 million tons now, the cargo movement at the Indian ports, both exports and imports, is set to cross 2800 million tons by the year 2020 as the nominal gross domestic product of the country is expected to touch 3.6 trillion US$ by the year 2020 from the current 1.8 trillion US$ (www.ibef.org 2014).

2.7 IN-HOUSE SOFTWARE DEVELOPMENT

Clemons, Eric (2014) in his article on “Why In-House Software Development Matters for Your Company’s Survival” states that in-house development is an important necessity, if the information of an organization and its ability for innovation is critical. Also, for such organizations that are interested to take advantage over competition, the in-house development adds speed and privacy. Such an organization that needs privacy, particularly in the context of competition, would always like to keep it to itself rather than getting its software developed by third party outsourcing. According to the author, a leading credit card company is never in the habit of outsourcing any of its important software to any vendor for if the vendor gets to know the software their edge over the competitors will be lost. Therefore such organizations know what to outsource and what not to outsource to third party vendors. The author says that as per the study conducted by Vanson Bourne and others, it is expected that in-house development in U.S.A. is going to be around 44% in the next three years from its present 33%.

IT operations have evolved over the years from simple data processing to sophisticated software development. Corporates initially put the computers to use for data processing and software had to be written for data processing and it was written mostly by trained in-house resources as the
infrastructure was scarce, machine time was expensive and trained resources were not abundantly available and therefore only corporates could afford and there was no choice but to have in-house teams only. The concept of outsourcing was not in vogue and even if external organizations existed, they could not afford the cost, and hence there was a very limited option for outsourced model of software development. But due to rapid changes in technology both at the hardware and software levels, things have changed. Corporate computing to Home computing could be achieved particularly after the advent of IBM PCs and its clones. Technologies were developed at a faster rate and the availability of trained human resources began to increase. In India, there were corporates such as HCL, Wipro, DCM Data Products, Zenith and such companies. All these were known to sell computer hardware and related software. Their small in-house software teams were engaged in porting operating systems to the new hardware systems they would come out with and they used to train their clients in using the system. But after Y2K, the situation got changed a lot and many of those Indian IT companies ventured into software development and services, and since then there is no looking back. Many popular SW products appeared in the market and in-house development activities were complimented by external software in areas such as ERP, CRM, and SCM.

In spite of the software development explosion that has happened in India, still In-house software development happens particularly in medium and large companies. An important reason to continue with the in-house development activities is for keeping up the business secrecy besides better customisation and product knowledge.
2.8 SUMMARY OF THE CHAPTER

In this chapter, the Researcher presents various important concepts practised and present in the Software Engineering and Software Project Management area particularly with respect to software metrics and Metrics Management. The Researcher also summarizes with all important and relevant information about the sectors considered for the research including the present status of the sectors. The study helped the Researcher to formally approach the research problem and conduct the research in a systematic way.