

Empirical investigation of Systems Cost Estimation Models in the Limpopo province of South Africa: A requirements modelling problem

Benson Moyo¹, Magda Huisman²

¹Computer Science and Information Systems Department, University of Venda, Thohoyandou, Limpopo, South Africa;

²School of Computer, Statistical and Mathematical Sciences, North-West University, Potchefstroom Campus, South Africa

Abstract - *There are many factors believed to be important to systems development cost estimation. However an in-depth analysis demonstrates requirements as central cost drivers. The various transformations requirements go through from candidate requirements to released response is the most intricate part of systems development cost estimation. Requirements exist independent of systems development methodologies. Requirements may be viewed from bespoke or market driven perspectives. The former assumes a traditional economic agent theory view where a client organisation requests for a service from the systems development organisation. The later, market-driven requirements elicitation entails predicting requirements by the systems development organisation based on market research output. Irrespective of the perspective the systems development cost estimation is imperative. The study investigates adoption and usage of cost estimation models by the systems development companies in the Limpopo province of South Africa. The paper introduces a requirements transition state diagram and pinpoints informal cost estimation models as predominant. In this article we also present the results of our survey findings and the discussion of those results as well as the recommendations for further work*

Keywords: *System development, requirements, cost estimation*

1 Introduction

Requirements exist independent of systems development methodologies. Requirements form the basis for the contract among the developer, the client and the user in a traditional economic agent theory based development. In market-driven systems development, requirements are predicted by the development organisation from market research output. This gives two ramifications of requirements; one based on a particular organisation where client and users are accessible and the other where clients and the users are the universe market or a market segment hard or expensive to access if not impossible. The nature of requirements influences the selection of systems development methodologies. The purpose of systems development

methodologies is to guide the development team successfully translate prioritised set of requirements into systems solution. They facilitate the development of technological frames to align expectations of technology and minimise incongruence, in systems development and reduce uncertainty in requirements determination [18]. Systems development methodologies are prospected to define work breakdown structure [19], provide support for improved product quality, productivity, human resource control and cost control [18]. Within the systems development methodology adopted by an organisation, forecasting and controlling systems development costs is crucial. Cost estimation is a systems development methodology activity that allows management to justify the relevance of a systems project in terms of reputation, social responsibility and more importantly financial value.

Research in systems development is not short of findings on software crisis that emerged from the 1960s. The crisis is claimed to stem from deficiencies in requirement determination emanating from inconsistencies, omissions, errors and ambiguities associated with requirements management. Systems development is basically governed by the requirements of the target system. The aforesaid requirement deficiencies are viewed as the key cause of systems development project failure. Failure means that the systems neither performs to specifications, nor meets budgetary constraints, nor is delivered within specified schedule, nor satisfies user needs and expectations. Cost estimation is primarily based on requirements. Inaccurate requirements identification, analysis, prioritisation, abstraction, triage, tracking and specification constitute the main causes for failure. In order to derive cost and schedule estimates requirements are the main input into the estimation process.

Cost estimate entails effort and schedule cost required to complete the system that meet client expectations, and satisfy developer starting from consultation to deployment. The client might be the universal market or market segment or a particular organisation (economic agent theory [30]). Requirements elicitation, analysis and management are

complex processes that inject complexity to software cost estimation.

Costing is governed by requirement based on both internal and external factors. For example from internal factors client and user requirements, systems quality, and contractual obligations whereas on the external factors we have the legal, statutory and regulatory requirements. The dilemma is on deciding which factors to include and which transformation to apply in order to map the factors into a single financial value. The growth in size, importance and complexity of software has exacerbated cost estimation [1]. No one cost model can address all the cost estimation situations of systems development. Literature reviewed reveals that there is still a gap between the use of systems cost estimation models and the actual systems costing practices in organisations. Logically systems cost estimation models should permit systems project managers reliably approximate systems cost. However, systems projects are complex in nature and it is challenging to accurately estimate their costs. From a causal wisdom perspective, a miniature perturbation in the development process for instance, may cause a huge change in systems costs. A small perturbation on the systems development methodology may also result in the deterioration of communication which in turn might lead to conceptual incongruence within the development team. Systems development conceptual incongruence degenerate into assignment scope challenges, quality issues and schedule slippages which in most cases, trigger considerable costs. Project managers have their fears on two extremes on systems project cost estimation: over costing or under costing. Over costing may damage the company's reputation and lead to failure to win systems development contracts (or fail to penetrate the market in case of a market driven systems development). On the other hand under estimating the cost may lead to loss of money and decrease in organisational profitability.

Cost estimation is a multidimensional construct that require identification and definition of numerous attributes. These attributes are attached to a weight which is then converted into cost drivers which in turn are transformed into financial values. In systems cost estimation there is no uniform set of attributes or parameters for every project. Each project is unique and may require contextual consideration to define the parameters. Developing a mobile phone game application does not exert the same demands as an online banking application, nor a nuclear reactor safety monitoring systems. Failure of each of the three mentioned systems has different consequences. For instance mobile phone game system failure may lead to disappointment; the banking system malfunction may culminate in financial loss, whereas the nuclear safety monitoring system failure may be catastrophic. Requirements important in one system application may not be so important in another system application. Even if the same requirements are found in different systems, their weight intensities may differ. The different importance levels, rejection and

prioritisation from one systems project to another makes it difficult to develop generic cost estimation models. Each cost estimation model is based on researcher's assumptions on a particular problem domain. Research has proposed a number of parameters but the underlying factors are requirements.

There is no simple way to make an accurate estimate of the effort required to develop systems [1]. Estimates are generated from requirements definitions or market research. The estimates define the effort, duration and staffing and other resources Alterations can be done to attain a trade-off between effort and duration. At the same time the systems product should respond to the requirements otherwise it might neither penetrate the market nor pass acceptance test even if the release date is met. Focusing on release date may have an implicit maintenance burden as some of the features may not be developed properly.

There are many requirements that need to be met by systems cost estimation models, in order to be adopted by an organisation as satisfactory in capturing the systems project costs measures. Systems cost estimation model maps various systems metrics and measurement into a single financial value. As the systems development process evolves so cost estimation should improve. There are many systems cost estimation models in existence, and companies in Limpopo are presented with the challenge to find the appropriate costing models, practices, techniques and tools. Systems cost estimation include specialised planning such as: quality plan which measures quality procedures and levels that will be used in a systems development, verification and validation plan which caters for the relevance degree of solution approximation by the system developed, configuration management plan which focuses on the alignment of management procedures and structures to be used, maintenance plan which predicts the maintenance burden, defect density which forecast the cost of defect discovery and removal, reviews and inspections, and change management plan that shows how the skills, experience, fears, and perceptions of project team members and users will be considered.

Systems cost estimation models strive to approximate the solution of systems cost estimation problem. Systems cost estimation is prediction of the cost of the resources that will be required to complete all of the work of a systems project [1]. It is important for project managers to use the appropriate model that will enable a successful completion of a systems project. In this work we investigate the actual adoption of the systems cost models in practice.

Khativi and Jawawi[8] state that the main reason for project failure is imprecision in cost estimation. However this is a high level of abstraction as one might ask for the causes of this imprecision. Systems project managers strive to acquire as much information on existing cost estimation models as possible. There is no exhaustive repository of projects that can

enable success and failure comparisons of cost estimation models, despite claims by cost estimation model authors that such repositories exist. The failure and success histories of cost models have limited scope as each project responds to specific requirements. Specific situational characteristics make it so difficult to design a 'one size fits all' costing model [13], though model developers believe the use of templates and frameworks can solve the problem.

Cost estimation is critical for organisations both that specialise on systems and those that outsource systems development. The purpose of this research is to investigate existing cost estimation models and their adoption in the systems development industry in Limpopo province of South Africa.

A survey on companies is carried out and the results show low usage of different systems project costing models in existence. Mostly formal systems project costing models are used when dealing with government systems projects. This may be that in order to win a tender from the government there is need to have rigorous cost estimation algorithms. The paper is organised into five sections. The first provides an overview of cost estimation characteristics. The second outlines the rationale behind cost estimation. The third describes briefly our research approach, the fourth section gives a discussion of the findings and finally, the fifth section provides conclusions and recommendations for further work.

2 Systems Cost estimation

Systems cost estimation is embedded into systems development. In this section we present justification, need and examples of systems costing models.

2.1 The rationale behind cost estimation

In order to make strategic decisions managers need some information about the resources required for the project. However this information is usually not available at the time it is needed. Estimates provide an approximation on the effort, schedule and other constraints needed. With estimates decisions on whether to proceed with a project can be made. Estimates serve at the intelligence and choice phases of the decision making process. One of the purposes of performing a cost estimate is to have a means by which the development costs can be monitored and controlled. Monitoring and control may be performed either at micro level or macro level. At micro level progress on addressing requirements is checked and at macro level progress is assessed by checking feature developed.

Costs estimates make the basis for the management and the development company to approve a project proposal or reject it. It is a crucial factor in determining when and how the project should be carried out. The project planning, controlling, resource allocation and roles in the project and overall activities of the project are linked to the cost estimate. A system is an investment and therefore it should demonstrate

financial, technical and social feasibility. Systems cost estimates are critical to developers, clients and users [3]. In a bespoke development environment they can be used for generating request for proposals so that the client can have the clue of the approximate amount required for the whole system that is contract negotiations between the client and the developer, scheduling between the programmers and the project managers, monitoring and control. On a market-driven systems development the developer makes assumptions on client and user requirements based on market research and work out the justification to commit resources.

The reasons for performing a cost estimate dictate what to estimate, how to estimate, when to carry out the estimation and the degree of accuracy. In principle cost estimation is iterative in nature in order to continuously update management on the project status. Suri and Ranjan[7] assert that small projects can be easily estimated and accuracy is not very important. But as the size of project increases, requirements become hard to elicit and analyse. The increases and dynamism of requirements lead to complex dependences and complicated relationships among them.

2.2 Requirements in cost estimation

Requirements present challenges as they come from different stakeholders that may even have conflicting objectives. The huge volume of requirements inflow and the need to select and reject some is a challenge. For example a selected requirement can be dependent on a rejected requirement. On another hand considering all requirements is not feasible as some may be contradictory and conflicting. Inconsistency, errors, ambiguity, incompleteness and different levels of abstraction are some of the challenges. Despite all these challenges cost estimates are based on requirements as they form part of the contract and the main link between the client and the developer.

Systems cost estimation involves measurement. Like any other measurement it depends on the perspective about the phenomenon to measure in this case requirements. The perspective is dependent on the operational definition of the concept. However, requirements as aforementioned are presented at different levels of abstraction making it difficult to calibrate them. The operational definition compounds the difficulties of uniform measurement of requirements as they may fall on nominal scale or ordinal scale or interval scale, or ratio scale. They may also be presented in varying degrees of abstraction of course this is due to different stakeholder linguistic capabilities. Requirements are a unifying concept. They go through a series of stages and sometimes are rejected due to change of environment. This has implications on cost estimation. Estimators model requirements into higher level features and functions and during this process of translation some requirements may be rejected without visualising dependencies. Unimportant requirements may be included. In some cases a complete misinterpretation during translation

from requirements to model systems artefacts may result. Figure 1 shows a proposal of a highly simplified view of requirement transition state diagram. During the initial phases of a systems project, large number of candidate requirements will be collected. These would be subjected to a selection and rejection processes. In order to cost selected requirements there should be cost elements and unit measures. The cost unit measures constitute a complex set of factors that affect cost estimation. The difficulty and the level of accuracy of systems cost estimation are shaped by the different categories of requirements which are considered as cost factors.

Requirements can be grouped under two broad classes; the functional and non-functional. Requirements can determine the following factors: systems quality, duration, team size, number of consultants, number and characteristics of stakeholders, legal statutes, regulatory statutes, mandatory statutes, organisational policy, criticality and complexity of the system, project risk factor, team expertise and experience, development platform, systems development methodology, techniques, and tools adopted, contingency plan and defect density.

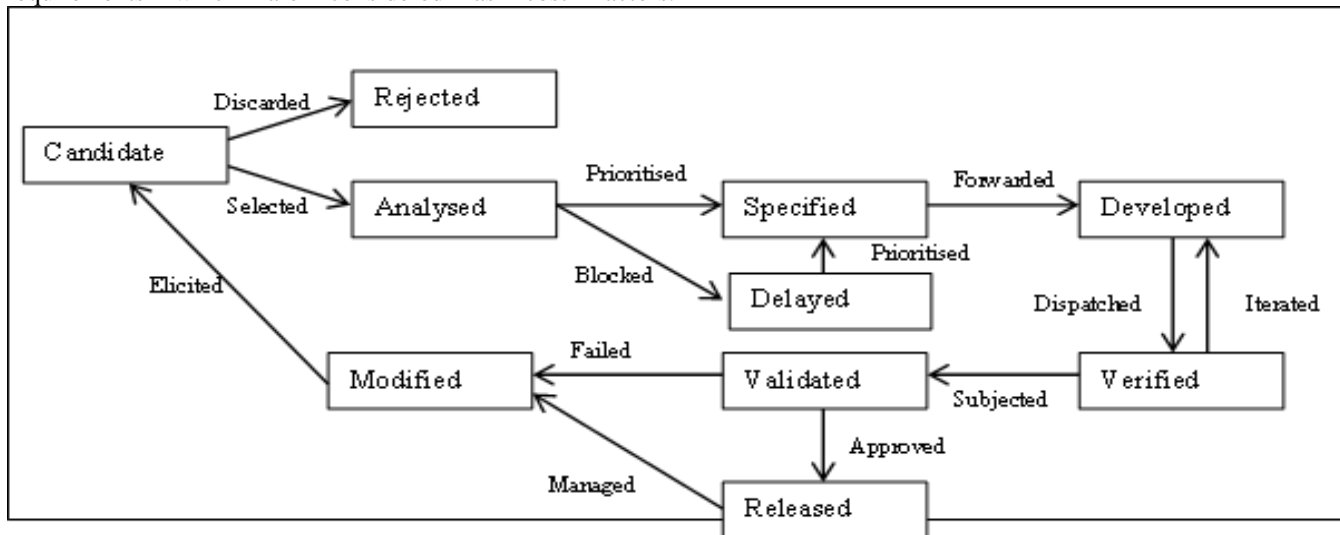


Figure 1: Requirements state transition diagram

The estimator works out the duration of the project in person-week or person-month based on the requirements analysis. Systems cost is directly proportional to project duration. The longer the duration between initial selected requirements and release, the more likely that there will be significant changes to the initial requirements resulting in more inaccurate cost estimates. This may occur due to continuous inflow of new requirements, modification of already known requirements, remodelling of requirements, change in user expectations, changes in the environment in which the system is to be installed, or change of technology.

Each systems project has risks associated with it. These risks may be associated with the integration of tools that offer source code generation, debugging, tests, document generation, diagramming, version controlling and other programming related services into the development process. Risks can also be due to lack of experience in the application domain, omitted requirements, misinterpreted requirements, mismanaged requirements, staff turnover and change of environment. The level of requirement uncertainty is used to establish the level of tolerance and estimate and allocate an estimate contingency fund.

Requirements also drive the size of the team and the level of expertise needed for the systems project. Cost increases with the size and composition of the team in other words the cost is directly proportional to the team size, expertise and experience. The more the team increases the more complex team coordination and communication will be among the members. Communication becomes less effective with the increase of the team and the project manager has to possess good management skills to keep the team productive. It is mostly the project manager's task to know the effort each team member and the capability of each member when scheduling the work that is meant to address the systems requirements. Therefore the increase in the team does not mean that the work is going to be completed earlier or best done.

Requirements leads to the selection of systems development methodology and in reverse the methodology addresses the requirements. The methods, techniques, tools, programming languages, programming paradigm, team skills, expertise, and experience are all unified by the methodology and a cost factors. Estimation is not a task done only once, at the project inception; it is a process where estimates and re-estimates are undertaken throughout the lifecycle of a project. The relevance of an estimator is not necessarily the accuracy of the initial estimates, but rather the degree to which the estimates converge towards the actual costs.

2.3 Existing systems cost estimation models

Organisations use different models to calculate cost estimates. These cost estimation models can be classified under two main categories: the formal mathematical models and informal experience based models. The formal models endeavour to quantify the cost factors and apply a set of relations that describe the mapping between the cost factors and the cost values. The mapping functions are formulated through analysis of historical data, assumptions and may be adjusted to each individual development context. On the other hand informal models are used by highly skilled experienced developers, expert or /and managers who have gained sufficient knowledge from previous systems development projects. The informal models rely on the history of past projects. A repository of detailed metrics and descriptions of characteristics recorded for each project. This repository may be a computerised database or manual record or simple organisational memory. The estimator can query the database searching for projects with similar characteristics and then benchmark the estimate on actual costs and process of the previous projects. Informal models are hard to understand as the experienced estimator may rely on tacit knowledge to obtain the estimate.

2.3.1 Formal models

A formal model may transform the systems requirements into a measure of the “size” of the systems in the form of Source Lines of Code (SLOC) as the basis for creating the cost estimates. Source Line of Code is an estimation parameter that illustrates the number of all commands, control structures, variable declarations, assignments, compiler directives, variable method definition and declarations excluding comments, blanks, and continuation lines. The advantage of SLOC is that estimating line of code seems intuitive. The lines of code are a parameter commonly used in formal model cost estimation models. It is also straightforward to count the lines of code in finished product which make it easy to compare the cost estimate and the actual cost. The disadvantage surfaces from deciding what to include as a line of code. The next challenge is when different programming languages are used it becomes difficult to use the SLOC model for cost estimation. Each language has its own number of lines of code to accomplish the same task. Line of code is not appropriate in a multiple programming language environment characterised by the current trends in systems development.

Instead of using the line of code, function points metrics can be used. It is a measurement based on functionality of systems[9]. It measures the amount of functionality in a system by counting and weighting inputs, outputs, queries, and logical flow, interfaces, files handling and device manipulation. Grouping the functions gives another measure referred to as the feature points. The feature points also consider algorithms as parameter and encapsulate control structures.

A function point based cost estimate known as systems functional size measurement is recognised by the International Organization for Standardization (ISO) and (International Electrotechnical Commission (IEC) as standard for measuring systems size. For example International Function Point Users Group (IFPUG) Function Point (FP) method [14] and the Common Software Measurement International Consortium (COSMIC) function point method [16]).

One most commonly known, well publicised and taught formal cost estimation models is Boehm’s Constructive Cost Model (COCOMO) [15]. COCOMO is widely practiced and popular among the systems development community because of its adaptability to different development environments. It predicts the length and effort of a project by drawing an association between the size of the systems and various cost drivers. The factors are assigned weights based on modelled requirements cost factors, project’s domain, environment, and constraints. The drawback on this model is that it requires too many parameters and simply selecting different values for the multipliers can vary the minimum and maximum estimates by a very high margin. Without historical data, it is difficult for an organization to determine the approximate values for these multipliers. The whole model fails when an organization is developing a system outside its immediate domain of expertise.

2.3.2 Informal models

An organisation may have a multi-stage estimation process. The contractor may present initial estimates. The estimate will be presented just to win the contract. The strategy is ‘pricing to win’. The estimate is made as low as possible so as to win the contract. Often times the estimate is done based on ambiguous and sometimes contradictory requirement. Once a company has been awarded the contract, it may then perform another more detailed estimate. The post contract winning estimate is done looking at the real systems problem to be solved. In most cases the posts-contract estimate is higher than the pre-contract estimate. The contractor may present this updated estimate to the client, and if management reject it there are a number of ways in which the contractor can go around the problem. They can develop a system with less functionality and suggest enhancements. If all fails then the contractor may accept the price to win estimate. Costs do not accurately reflect the work required and the rejection on acceptance test is common.

Estimation based on expert judgment is done by considering advice given by experts who have more experience in similar projects[8]. The work breakdown structure (WBS) organises activity patterns that vary from project to project, by defining assignment scope. It identifies activities needed to complete project development and the effort, staffing, duration of each task and the skills required. The size of activities defined within the WBS is dependent on the level of detail of the estimate and size of the project. A

cost estimate frequently includes a complete work breakdown structure (WBS), which project team members use as a basis for understanding their roles and avoid bumping into each other's way. The sum of the direct individual activity costs and other indirect cost such as travel costs gives the overall project cost. The Delphi technique constitutes a team of experts tasked to generate systems cost estimates of a project given all available factors and constraints. Each expert provides an estimate without consulting other team members. The second iteration allows each expert to have access to the

estimation information provided by other experts during the first iteration. The process continues until the expert estimates converge to a point.

The following table 1 shows a historical perspective of the formal models and indicates the modelling level of requirements abstraction that gives the cost unit measure. Avoid using too many capital letters. All section headings including the subsection headings should be flushed left.

Table 1: Trends in cost estimation

Year	Authors	Model	Cost drivers
1970	Boehm[1]	Rule of thumb	no specific factors
1975	Albrecht and Gaffhey[20]	Function Point Analysis	external input, external output, external inquiries, external, interfaces, internal files
1977	Park[1988]	PRICE-S	source lines of code, function points, predictive object points, defect prediction
1979	Putnam[21]	Putnam Model	manpower distribution, environment indicator, duration
1979	Albrecht[24]	Function Point	interfaces, forms, reports, database tables
1980	Jensen[25]	SEER-SEM	source line of code, effort, schedule, defect predict, risk, reliability
1981	Boehm[17]	COCOMO	source lines of code, effort
1983	Rubin[27]	ESTIMACS	function point, effort, staff count and deployment, risk, portfolio impact, customer complexity
1983	Symons[22]	Mark II function points	inputs, outputs, queries, and logical flow, interfaces, files handling, data processing
1986	ISO/IEC 20926[14]	IFPUG	function point
1992	Bergeron and St-Armaud[23]	Mark II Function Point	inputs, outputs, queries, and logical flow, interfaces, files handling, data processing
1995	Boehm[15], Boehm et al[26]	COCOMO II	object points, function points, lines of code, effort
1997	Jones[32]	Checkpoint	activity, task, estimates, deliverables ,defects, schedules
1998	Chatzoglou and Macaulay [28]	MARCS	time, effort, staff size

3 Methodology

The approach is essentially qualitative research based and uses instruments of interviews and questionnaire. Intensive and extensive literature survey is conducted to establish the status of systems cost estimation in industry. A questionnaire is administered to the project managers and freelancers in the Limpopo province of South Africa. It is important to acknowledge that the sample was purposeful and non-probabilistic. The guiding selection principle was to increase the probability of the presence of the phenomenon of study interest in the sample [13]. Participants were recruited from the Limpopo province systems development companies.

4 Results and discussion

Table 2 shows the usage trends of different costing models found in systems development houses in Limpopo. Experience based costing is more pronounced despite the dissemination of models, tools and techniques from the research community. For freelancers their cost estimates are based on the feature points. Cost estimation is allocated an average time of three weeks per project.

Table 2: Cost model usage

Costing Model used	Project Managers (N=12)	(%)
COCOMO	2	16.6
Expert Judgement	5	41.2
SLOC	0	00.0
Past Experience	9	75.0
Feature points	3	25.0
Average time taken per project on cost estimation		
Less than 1 week	2	16.6
1- 3 weeks	7	58.3
4- 5 weeks	3	25.0
More than 5 weeks	1	16.0

Table 3 shows the probability of approximating costs in different development phases of the development process. Estimates in the initial stages of the project are not reliable as the level of requirement understanding is low. Sometimes what may be established as feasible might not be as feasible as earlier thought. Development process iterates over requirements several times leading to the establishment of factors governing requirement dynamics. Requirements are regarded as the most difficult part of a project.

Table 3 *SDLC phase cost estimates accuracy*

Phase	Estimate accuracy (%)
Feasibility study	3.4
System analysis	10.8
Requirements analysis	21.6
System design	87
System construction	88
Defect detection and removal	94.1
Continuous enhancements	96

5 Conclusions

We found that systems cost estimation is based on and governed by the requirements of the target system. Requirements determination is an integral part of systems cost estimation and the basis for systems costing in Limpopo province. However, before cost estimation requirements are modelled into cost drivers in the form of function points, feature points or object points. The main challenges emerge during the requirement transformation into cost factors. In order to minimise the impact of error in cost estimation the estimators carry out the estimation process throughout the life cycle of a project. However, with requirements poorly identified, specified, and modelled failure probability is high irrespective of the number of times cost estimation is done. Timing of estimates, estimation constraints, systems development methodology used, experience and expertise are the basis for cost estimation in Limpopo, but all these are dependent on requirements.

Despite availability and publicising of formal cost estimation models, informal models are most preferred by the systems development industry in the province of Limpopo in South Africa. As further work we would replicate the research in the remaining eight provinces of South Africa and endeavour to find out the reasons for not adopting the formal cost models.

6 References

- [1] I. Sommerville, in *Software Engineering: Software Cost Estimation*, pp. 1–58, 2004.
- [2] T. N. Sharma, A. Bhardwaj, and A. Sharma, “A Comparative study of COCOMO II and Putnam models of Software Cost Estimation,” vol. 2, no. 11, pp. 1–3, 2011.
- [3] J. Kaur, S. Singh, and K. S. Kahlon, “Comparative Analysis of the Software Effort Estimation Models,” pp. 485–487, 2008.
- [4] M. M. Albakri and R. J. Qureshi, “Empirical Estimation of COCOMO I and COCOMO II Using a Case Study.” pp. 265–270, 2012.
- [5] I. Myrtveit, E. Stensrud, and M. Shepperd, “Reliability and validity in comparative studies of software prediction models,” *IEEE Trans. Software Engineering*, vol. 31, no. 5, pp. 380–391, 2005.
- [6] B. Boehm and C. Abts, “Software Development Cost Estimation Approaches”, 1998.
- [7] P. K. Suri and P. Ranjan, “Comparative Analysis of Software Effort Estimation Techniques,” vol. 48, no. 21, pp. 12–19, 2012.
- [8] V. Khativi and D. N. A. Jawawi, “Software Cost Estimation Methods : A Review,” vol. 2, no. 1, pp. 21–29, 2010.
- [9] S. L. Pfleeger, F. Wu, and R. Lewis, *Software Cost Estimation and Sizing Methods: Issues, and Guidelines (Google eBook)*. Rand Corporation, pp. 97, 2005.
- [10] H.M Haddad, N.R Ross, and W. Kaensaksiri, “Software Reuse Cost Factors” Int’l Conf. Software Engineering and Practice, pp.284-290, 2012
- [11] B. Boehm, C. Abts, and S. Chulani, *Software development cost estimation approaches - A survey*. Annals of Software Engineering, Vol 10, No1-4. Springer, Netherlands, 2000.
- [12] C.F. Kemerer, An empirical validation of software cost estimation models. *Communication of the ACM*, Vol 30, No 5, 416-429, 1987.
- [13] K. Eisenhardt, Building theory from case study research, *Academy of management review*, Vol 14, No 4, pp 532-550, 1989.
- [14] ISO/IEC 20926 Software and systems engineering -Software measurement - IFPUG functional size measurement method 2009, 2nd, ISO, Geneva, 2009
- [15] B. Boehm, *Software Cost Estimation with COCOMO II*, Prentice Hall PTR, Upper Saddle River, NJ, 2000.
- [16] ISO/IEC 19761 Software engineering – COSMIC: a functional size measurement method, 2nd edition, ISO, Geneva, 2011.
- [17] B. Boehm, *Software Engineering Economics*, Englewood Cliffs N.J., Prentice-Hall Inc. 1981.
- [18] M. Huisman and J. Iivari, Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers, *Information & Management* , Vol. 43, No. 1, pp. 29-49, 2006.
- [19] N.Jayaratra, *Understanding and evaluating methodologies: A systemic framework*. McGraw Hill, UK, 1994.
- [20] A.J Albrecht and J. E Gaffhey, Software function, source lines of code and development effort prediction: A software science validation, *IEEE transactions on Software Engineering*.
- [21] L. H Putnam, A general Empirical Solution to the macro software sizing and estimating problem, *IEEE transaction on Software Engineering*, Vol 4, No 4, pp345-361, 1981.
- [22] C. Symons, *Software sizing and estimation Mark II* function points, Wiley, 1991.
- [23] F. Bergeron and J. Y. St-Arnaud, Estimation of information systems development effort: a pilot study, *Information and Management*, Vol 22, No 4, pp 239-254, 1992.
- [24] A. J. Albrecht, “Measuring Application Development Productivity”, Joint SHARE, GUIDE, and IBM Application Development Symposium, pp. 83-92, Monterey, 1979.
- [25] R. Jensen, “An Improved Macrolevel Software Development Resource Estimation Model”, *Proceedings of 5th ISPA Conference*, pp. 88-92, April 1983.
- [26] B. Boehm, B. Clark. E. Horowitz, C. Westland, R. Madachy, and R. Selby, “Cost Models for Future Software Life Cycle Processes: COCOMO 2.0”, *Annals of Software Engineering, Software Process and Product Measurement*, Science Publishers, Amsterdam, The Netherlands, Vol. 1, pp.57-94, 1995.
- [27] H. Rubin, “ESTIMACS”, *IEEE*, 1983.
- [28] P. D. Chatzoglou and L. A Macaulay, A rule based approach to the developing software prediction, *Automated Software Engineering*, Vol 5 No 2, pp211-243.
- [29] M. Shin and A. L. Goel, “Emprirical data modeling in software engineering using radial basis functions”, *IEEE Transactions on Software Engineering*, pp. 567-576, 2000.
- [30] C. Toffolon, S. Dakhli, “A Framework for Software Engineering Inconsistencies Analysis and Reduction”, In *Proceedings 22nd Annual International Computer Software & Applications Conference (COMPSAC 98)*, IEEE Computer Society Press, pp. 270- 277.
- [31] R. Park., “The Central Equations of the PRICE Software Cost Model,” Park R., 4th COCOMO Users’Group Meeting, 1988.
- [32] C. Jones, *Applied Software Measurement*, McGraw Hill, 1997.