Finding Factors Impacting Productivity in Software Development Project Using Structured Equation Modelling

Sanjay Mohapatra, Divya Kumar Gupta
Associate Professor,
Xavier Institute of Management, Bhubaneswar
cooldivay@gmail.com, sanjaym@ximb.ac.in, sanjay_mohapatra@yahoo.com
doi: 10.4156/ijipm.vol2. issue1.10

Abstract

The size of international software industry is over $1000bn. Total revenue earned by the Indian software industry in year 2005 was $30bn and which is expected to grow. The price quoted by vendors to its customers and the rate of the customer service is dependent on the productivity of the project. Given the fact that software development complexity is on an ever-increasing curve, it becomes critical to tackle the issues that influence the productivity levels of the project teams involved. Identification of factors as either aiding or hindering productivity enables management to take steps to encourage the positive influences and to eliminate the negative ones. Hence it is important to understand the factors affecting productivity and how these factors affect productivity when they are present in tandem. Hence this report aims to develop a model using structured equation modelling to find important factors affecting productivity and their impact on productivity. This paper is a further improvement on the paper "Maximizing Productivity by Controlling Influencing Factors and Commercial Software Development", by Mohapatra, S. The authors expect to come out with a ready model for productivity which can help any software development team in allocating resources to achieve maximum output. The Methodology used for this primary research is collection of data from project and delivery managers across three Indian IT majors and analysing that data employing SEM using a computer aided tool. Results showed that the best fit model consisting of important factors like experience of project team in technology and domain, training for the technology platform to be worked on and domain of work, client support and various other software testing tools availability helps in increasing productivity.

Keywords: *Productivity, Development Projects, Software Project Management, Structured Equation Modelling*

1. Literature Review

Boehm (1981), in an earlier study conducted with large government software projects as part of a sponsored Government programme. He found that cost of software projects is inversely proportional to variables that would help improve productivity. He found that high staff capability and low application complexity would improve productivity. However, there was no detail explanation was provided for measuring staff capability. Lawrence (1981) conducted study in medium to large organization in Australia and performed multivariate analysis of productivity variance using different variables related to computing environment and organization. His observation was that productivity increased with simpler application being developed. However interface with database management system decreased productivity. These observations were similar to findings made by Mohanty (1981) and Kemerer (1987). In addition good computing speed from hardware used increased productivity Thadhani (1984), Conte (1986) and lambert (1984).

Vosburg et al. (1984), have made a detail study of large software projects being developed in 17 different ITT subsidiaries in nine different countries. The researchers classified variables that affected productivity into 'product' related variables and 'production processes. The product related variables were found to be 'computing resource constraints', 'program complexity', 'customer participation', and 'size of program or application' being developed. The production process related variables that were found to affect productivity were 'concurrent hardware-software development, 'development computer size' (which is similar to computing speed of hardware), 'stability of requirement and

specification', 'usage of modern programming practices' (meaning usage of top-down design, modular design, and following quality assurance programs) and 'personnel experience'. The researchers concluded that productivity improvement programmes were effective only if all the variables are simultaneously controlled and monitored.

Jeffrey (1987) and Curtis (1981) studied small projects (200 SLOC – Source Lines of Code) and large projects (SLOC more than 100000) and came to conclusion that to achieve maximum productivity there should be an optimum staff size depending on the technology used for development and size of the application developed. They also found that if the staff size is increased beyond the optimum level, it did not improve productivity any more, rather reduced it (productivity). Since there was no empirical study, the findings could not be validated in other organizations. In another study Kemerer (1990) and Jones (1978) studied a number of projects in Japan and U.S. (24 U.S. and 16 Japanese development projects) and found that productivity was influenced by application type, programming language used, hardware platforms used for development, percentage of code reuses per project and number of tools or methods used per project.

Boehm (1981), in his study, developed a model called COCOMO II and reported that productivity of software development projects is mostly affected by who developed the system and how well the team was organized and managed. Scacchi also reported that when projects were poorly managed or organized, productivity was substantially lower than those projects which were well managed. Conditions in workplace and the skills of the developers acted as the project specific drivers for productivity. However, Bhansali et. al. (1991) reported that programmers were two-to-four times more than productive when using languages like Ada compared to languages like Fortan or Pascal. However, it was not clear what was significant in explaining the difference in productivity.

In case of Software industry output is in the form of software application which is a solution to a customer problem The output is measured as number of lines of code (NLOC) or number of function points and the input is measured in the terms of man hours spent in the preparing that output. There is lack of standard coding practices in different technologies. For example, coding practices and syntax in Java is different than that of COBOL; as a result similar applications will have different NLOC in two different technologies. Hence, productivity will differ for different projects giving similar output taking similar input but using different programming languages. Industry came out with the concept of Function points (FP) where output is measured in the terms of FP rather than NLOC, which is non dependant on the programming language used. Input for all projects are measured in man-months which is the total numbers of Man months spent in the development of all life cycle of the project. Life cycle of a project consists of stages such as Design, Test plan preparation, Coding, Testing and Maintenance or Warranty. The size of the output is dependent on the customer requirements and cannot be changed by the service provider. Hence, to improve productivity, the strategy that can be adopted by the service provider is to reduce total project effort without compromising the quality of work. Training is provided to team members to improve their skills and reduce effort related to rework. Training also improves domain knowledge which helps in effective requirement gathering and analysis. Experienced team members play a vital role in increasing productivity as the quality of output will be higher and they can finish the work faster compared to non-experienced team members. Hence, it is quite important to understand the factors that help improve productivity. This understanding will help management to prioritize and allocate resources for these factors so that higher productivity can be achieved. Also these factors not only influence productivity but also impact each other. These aspects also needed to be studied.

As per the part of the research practitioners from different software organizations were contacted; these practitioners, have had worked in the industry for long time and have had rich and variety of experience in project management. As per the Interviews conducted with these Project managers and Delivery managers of different IT majors situated in Bhubaneswar, India, the following factors emerged as the basic and important reasons affecting the Software productivity for a commercial project. In the opinion of these practitioners, factors such as application complexity, client support, experience in technology and domain, training in technology and domain, computing speed of processors, availability of software testing tools, document management system have noticeable impact on productivity. Details of the impact of these variables on productivity are discussed in the next section.

2. Objective

The objective of this research was to find out different important factors affecting the productivity of any development project. To reach the higher efficiency level it is important to look into the factors affecting productivity and also it is beneficial to have a statistical model that tells the project manager in advance the degree of impact that these factors will have on productivity.

Based on different literature available and the perceived gap in those readings, the major objectives of the project were defined as follows:

- To find factors that affect productivity.
- To develop a model for predicting productivity value for given project variables.

Application complexity:

If the project is complex, then it will take more time in project completion. In complex projects, team members will take more time to understand the project and its requirements. Similarly more time will be required for resolving issues and clarifying doubts raised by team members. Because of this the progress of application development goes slow taking more time to complete the development process and effectively productivity decreases.

Hypothesis for application complexity

H1: If application complexity is increased the Software productivity will decrease.

Client Support:

Client support is one of the key requirements for successful closure of project. Interaction with client is required right from the requirement analysis phase till the end of the project. Hence more the client supports the easier will it be for the project team to move ahead faster and execute the project faster.

Hypothesis for Client Support

H2: Productivity would tend to increase as the client support increases.

Experience in Technology:

Past experience of the project team in the same technology will help in decreasing the time required for the project execution and hence increase the productivity.

Hypothesis for Experience in Technology

H3: If Experience in Technology is increased the Software productivity will increase too.

Experience in Domain:

For doing any IT project it is absolutely necessary for the project team members to understand the domain. This will help the project team to understand all the needs and develop the application accordingly.

Hypothesis for Experience in Domain

H4: If Experience in Domain is increased the Software productivity will increase too.

Technology Training:

Training helps in increasing the skill level of the team members and the efficiency of the project should increase with the involvement of more number of trained people.

Hypothesis for Technology Training

H5: If Technology training is increased the Software productivity is expected to increase too.

Training in Domain:

With increase in effort spent in domain training, the team members can understand the requirements of the client better and can develop the application faster.

Hypothesis for Domain Training

H6: If Domain Training is increased the software productivity is expected to decrease.

Computation Speed:

Hardware speed and performance of system can very well affect the time taken by the project in completion. Slow performance will not only increase the execution time but also will de-motivate programmer. Same is highlighted in the reports prepared by the various industry experts like Boehm[1981] and Lambert[1984]. COCOMO II model also gives a very high importance to the computation speed which basically talks about the time taken by the system to show the results to developer after executing command.

Hypothesis for Computation Speed

H7: Higher the Computation speed the Software productivity will tend increase.

Software Testing tools:

There are many Software testing tools that are used by different project teams. Using such tools reduces testing time drastically and increases productivity.

Hypothesis for Technology Training

H8: Availability of any kind of good testing tool will tend to increase the productivity.

Document management system:

Document management system helps in retrieval of a document that was stored at a time of any previous project and is required later in life. Any such management system should follow standards like ISO, CMMi etc. This system is very much important for knowledge management and presence of such systems improves productivity.

Hypothesis for Technology Training

H9: More the level of compliance to document management system it would tend to increase the productivity.

3. Research Methodology

For objective one, secondary sources (literature review) were used to find the factors that affect productivity.

For objective two, data were obtained from a local software organization executing projects in the categories of development, maintenance and products. Roughly 33% of the projects were development projects and revenue from these projects was more than 40% of total revenue in 2006 -2007. This is one of the fastest growing organizations in India. Having been set up in early ‘80s, it has grown at the rate of more than 100% per year for the initial 15 years. It has worked only on overseas projects during its two decades of existence. Its equity share of face value of Rs. 2/- each had gone up to Rs. 3000/- in the stock market raising its market capitalization to more than Rs. 50,00,000 crores (www.nseindia.com, 2007). It is one of the companies which had achieved CMM and CMMi level 5

certifications for the entire organization. The management of the company is well respected both in India and abroad and is also known to be one of the best employers (www.hewittassociates.com, 2002). The company also has a social face, having been extremely active in social work and social projects throughout India. This organization is considered as a model organization in terms of its work ethics, governance practices and various other ideals. Practices followed in this company are used as benchmarks by many other companies in the IT industry. These were the added reasons for taking this company for the study.

Sample selection and Sample Size

As stated earlier, only development projects have been studied here and while selecting the samples, the following criteria were used:

1. The projects were executed in its local development centre so that data could be easily available. Besides, this criterion was to ensure that differences in environment and location did not distort the comparability.
2. The projects were fairly large and the size of the application developed was more than 2000 Function Points and less than 4000 function points. This criterion was essential to ensure that projects were characteristically similar and handled in the same way (not in the way mega or mini projects were handled). Thus differential features of mega or mini projects making differences in productivity and in defect density were unlikely.
3. The projects were completed in April 2008. This criterion was used as it ensured that workload pressure was similar for the organization as a whole and time differences did not create any additional factor which could distort the results.

In April, 2008, 128 projects were completed and out of these, 81 were development projects and the rest were maintenance projects and products. Out of 81 development projects, 63 met the above criteria and complete data were available for these projects. Hence 63 projects were chosen for this part of the study.

Analysis Plan

Structured equation modelling using AMOS 5.0 was used for developing the model. The software was available in the institute and accessible to the researchers.

Definition of variables

1. Application Complexity: - To measure complexity of the applications, cyclometric complexity was proposed. However, the researchers and project managers felt that it was not feasible to use as a metric. Hence, using Likert scale, application complexity was measured by finding perception of the project managers. Based on perception of a project manager, an application was ranked on a scale of 1 to 7, with 7 being the most complex and 1 being the least complex programme.
2. Client Support: - This defines the degree of involvement of client in requirement analysis and design phases of the project and in resolving queries during project execution. It was measured on Likert scale. This was a perception measure as responded by project managers. If the client was fast in resolving the issues and queries raised by the team members, then project manger rated the client at 7. A very slow response from client will fetch 1.
3. Computation Speed:- Various desktops and other terminals present with the project team can very a lot over computation speed and hardware present. Based on the Project Managers response this was also managed on the scale of 1 to 7, where higher the value better was the computation speed.
4. Software Testing Tools: - There are different tools available that were used for automating execution of test cases. These tools could easily record Test scripts and results related to it. If project managers believed that available tool could automate all the test scripts value was 7 on scale and no presence of such tool fetched value as 1.

5. Availability of modules:- This variable explains the presence of any such code or module that can be reused in the exciting projects. Let say if code or module from a similar project that was executed in the past can be reused, will definitely help in increasing productivity. Presence of any such module definitely saves time and its presence is measured on the scale of 1 to 7; here 7 meaning that there is maximum availability of reusable code or module (as perceived by project manager) and 1 meaning least or no availability of reusable code or modules.
6. Productivity: Productivity of the software project is defined as output in size divided by effort; this is measured by size of application (in function points) divided by the total efforts spent in completing the project measured in man-months.
7. Experience in Domain and Technology: These are measured in man months of experience and denote the availability of team members with experience in domain and technology.
8. Training in Domain and Technology: These are measure in man months of effort spent in the training for the particular project.

4. Data Collection

The data were collected from three different organizations which have development centres in Bhubaneswar (India). All these companies are internationally well known Indian IT service providers and execute projects for customers across geographies. These companies are well respected in the arena of clients and these companies are performing work for many domains. These organizations are certified at CMMi level 5 and all projects executed here are compliant to MMI processes. Personal interview method was followed to collect all this data from project and delivery managers. We were able to collect information for 63 projects. These organizations were contacted in two phases. First phase was after the secondary research (literature review). In this phase project managers and delivery managers were interviewed to know the practical view related to factors affecting productivity. In the second phase data were collected from the same organizations for the factors that were perceived to be affecting productivity.

5. Model Development

Based on analysis, construct design and information collected by interviewing different practitioners, 12 different models were selected to be tested on AMOS 5.0 for finding the best fit model. Using the variables, the software was executed several times to find the best fit model with lowest CMIN value and p value of 0.05. Also for the RMR GFI test the above mentioned model had lowest RMR value and GFI value of .877 and AGFI value of above 7. Parsimony-Adjusted Measures with PNFI value of .077 also suggested that this is a good model to be selected. Here in this is model productivity which is dependant variable is placed in centre taking inputs from various independent variables. A very important thing to notice is the interdependence that is shown in the model (Figure 1); here we see there are 5 bidirectional arrows that depicts interdependence between various factors eventually affecting productivity. Like here Experience in technology and experience in domain is mutually dependant. Similarly ‘Application complexity and Client support’, ‘Technology training and Technology experience’, ‘Domain training and Domain experience’ and ‘Technology training and Domain training’ are mutually dependant.

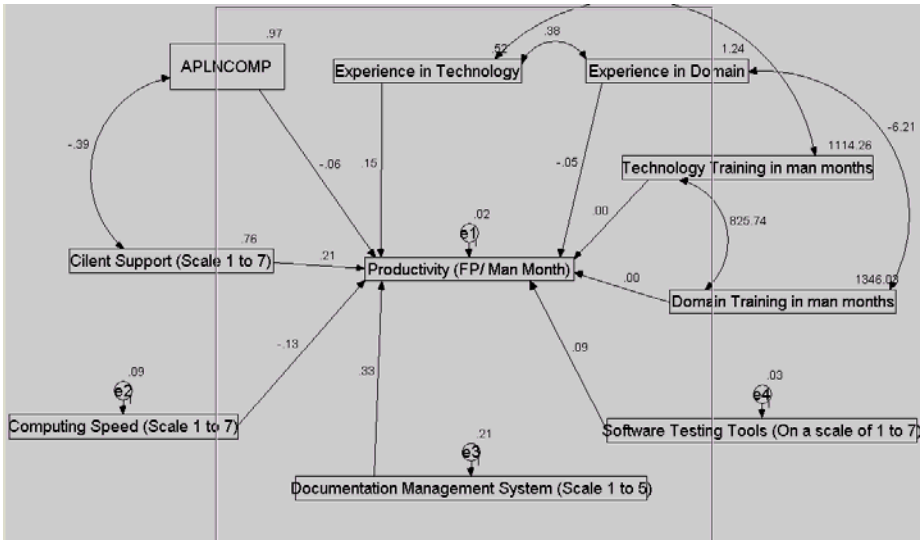

Figure 1: Model Generated using AMOS

6. Results and Discussion

This model and the regression and correlation result which were performed along with model verification provided us with various results and conclusion. In this model all the variables were observed variables except for the error terms e1,e2,e3,e4. Productivity was treated as the main Dependent variable as all other factors were presumed to affect productivity directly. Hence productivity was kept in the centre and it was found that all the variables directly affected productivity. Along with the one-sided arrows, there were few bi-directional arrows which depicted mutual relation and dependence between two variables, like in the case of 'experience in technology' and 'training in technology' were seen to be mutually dependent; similarly 'domain experience' and 'domain training' were mutually dependent. Also technology training was found to be mutually dependent on domain training and technology training. Application complexity and client support were also dependent and hence a bi-directional arrow was selected between these.

Tables 1 and 2 show the result of **regression and standardized regression coefficients.** Out of the selected factors six (as shown) were found to be very significant; they were application complexity, computing speed, document management system, experience in technology, experience in domain, and domain training. Three of the previously selected factors were rejected (these were dependence of productivity on Client support, technology training and Software testing tools). Here impact of application complexity is significant as estmate value is less than .05.

Table 1. Regression Coefficients (Red highlighted ones have been found to be statistically insignificant)

			Estimate
PRODUCTI	<---	APLNCOMP	-.189
PRODUCTI	<---	CILENT_S	.005

			Estimate
PRODUCTI	<---	COMPUTIN	.122
PRODUCTI	<---	DOCUMENT	.457
PRODUCTI	<---	EXPRINTE	.327
PRODUCTI	<---	EXPRINDO	-.162
PRODUCTI	<---	TECHNOLO	-.014
PRODUCTI	<---	DOMAIN_T	-.279
PRODUCTI	<---	SOFTWARE	.048

Out of six factors which have significant effect on the Productivity, Document management system is most important as it affects the software productivity the most. It implies that with the increase in this factor, productivity will increase in other words productivity and this factor is directly proportional. With Unit change in the application complexity the productivity decreases by the factor of .189; better computing devices increases the productivity by the factor of .122. Unit improvement in documentation increases software productivity by the factor of .457 which is most significant. Similarly experience in Technology and experience in domain effects inversely with technology experience helps in productivity where as domain experience decreases productivity as against our initial assumption. Domain training improvement decreases productivity by .279.

Standard Regressions Weights considering all the factors:

Table 2. Standardized regression coefficients

	Estimate	S.E.	C.R.	P	Label
PRODUCTI <--- APLNCOMP	-.063	.035	-1.772	.076	par_1
PRODUCTI <--- CILENT_S	.214	.052	4.120	***	par_2
PRODUCTI <--- COMPUTIN	.130	.162	-.803	.422	par_3
PRODUCTI <--- DOCUMENT	.328	.136	2.419	.016	par_4
PRODUCTI <--- EXPRINTE	.148	.085	1.737	.082	par_5
PRODUCTI <--- EXPRINDO	-.047	.045	-1.047	.295	par_6
PRODUCTI <--- TECHNOLO	.000	.002	-.080	.936	par_7
PRODUCTI <--- DOMAIN_T	-.002	.001	-2.231	.026	par_8
PRODUCTI <--- SOFTWARE	.091	.843	.108	.914	par_9

Correlation between mutually related factors:

Tables 3 and 4 show the ‘correlation’ and ‘covariance’ between two mutually related factors. It says experience in technology and experience in domain is very much significant and the 1 unit change in the standardized value of experience in technology leads to change in experience in domain by .473 standard unit of experience in domain. Experience in technology and Technology training are inversely related as implied by the factor of -0.276, Technology and domain training are again highly related as per their nature and as it emerged out of primary research. Similarly experience in Domain and domain Training are inversely related as more is the experience in domain less is the training required in the domain area of the project. Another very important deduction from correlation is the relation among application complexity and the client support that is though client support is not directly affecting productivity but it has a significant correlation with application complexity; lesser the client support more is the application complexity and hence less is the productivity.

Table 3. Correlation between related factors

			Estimate
EXPRINTE	<-->	EXPRINDO	.473
EXPRINTE	<-->	TECHNOLO	-.276

			Estimate
TECHNOLO	<-->	DOMAIN_T	.674
EXPRINDO	<-->	DOMAIN_T	-.152
APLNCOMP	<-->	CILENT_S	-.454

Covariance between mutually related factors:

This table 4 is crucial to depict the effect of factors over each other and can be treated as one of the major output of the project. All correlations shown are significant except for the one for which P value is more than 0.25 i.e. between the Experience in domain and Domain Training. As per this finding we can say there is the high covariance between technology Experience and Domain Experience, Technology Experience and Technology Training, Technology training and Domain Training and application complexity and client support.

Table 4. Covariance between related factors

			Estimate	S.E.	C.R.	P	Label
EXPRINTE	<-->	EXPRINDO	.379	.194	1.957	.050	par_10
EXPRINTE	<-->	TECHNOLO	-6.612	4.037	-1.638	.101	par_11
TECHNOLO	<-->	DOMAIN_T	825.737	347.141	2.379	.017	par_12
EXPRINDO	<-->	DOMAIN_T	-6.211	5.757	-1.079	.281	par_13
APLNCOMP	<-->	CILENT_S	-.389	.181	-2.150	.032	par_14

7. Hypothesis Analysis

H1: If application complexity is increased the Software productivity will decrease.

Accepted : The negative relation between Productivity and Application complexity was found and hence the above stated hypothesis was accepted with significant P value.

H2: Productivity would tend to increase as the client support increases.

Rejected : The significance level estimate of this Hypothesis was less and hence the mentioned hypothesis was rejected.

H3: If Experience in Technology is increased the Software productivity will increase too.

Accepted : As the P value was significant and positive the hypothesis mentioned was accepted i.e. there was positive relation between Experience in technology and Software productivity

H4: If Experience in Domain is increased the Software productivity will increase too.

Rejected : The hypothesis mentioned here was not valid even though the P value was significant but sign was negative which means Domain Experience and Productivity were inversely proportional, which might be because of loss of time by Project team by high involvement in trivial issues of the projects.

H5: If Technology training is increased the Software productivity is expected to increase too.

Rejected : The estimate value for the same was very low and very less significant hence the assumed hypothesis was rejected.

H6: If Domain Training is increased the software productivity is expected to decrease.

Accepted : More the time spent in the Domain training more effort was gone without output hence less productivity and as expected the estimate value was significant and negative for the same.

H7: Higher the Computation speed the Software productivity will tend increase.

Accepted: Better the Computation speed, less effort was wasted in waiting for execution of syntax commands and hence the productivity increased as shown by the estimate value which is positive and significant as assumed.

H8: Availability of any kind of good testing tool will tend to increase the productivity.

Rejected : Because estimate value for Software testing tool was not significant though very near to 0.048 the hypothesis was rejected.

H9: More the level of compliance to document management system it would tend to increase the productivity.

Accepted : Presence of good Document management system increased the productivity of the Software and same was supported by the estimate value analysis

8. Conclusions

1. Productivity is significantly dependent on Technology training. Hence more time can be spent on effective technology training.
2. Project team that comprises of more experienced people, leads to the better productivity.
3. Though not evident on first instance but Application complexity and client support are mutually dependent on each other and affects each other.
4. Model prepared can be used in future commercial software development which will help project manager in scheduling and resource allocation.

9. Acknowledgement

We specially want to thank Prof. Subhajyoti Ray and Prof. Dipak Misra for their help and guidance at various point in time during the course of this project. We also thank Fr. George Joseph, S.J. for helping us and teaching us the usage of the structured modelling tool used in for the Project Research.

10. References

[1] Boehm, B., Software Engineering Economics, Prentice-hall, 1981
[2] Boehm, B., Penedo, M.,Stuckle, E.D.,Williams, R.D. and Pyster,A.B., A Softwar Development Environment for improving Productivity, 1984
[3] Mohapatra S., Maximizing Productivity by controlling influencing factors in commercial software development,Vol.1,issue-3, International journal of Information and communication technology,2008
[4] Mohapatra S., Improving Cost of Quality through bench marking exercise, International Conference on Principles of Software Evolution, 2002
[5] Mohapatra S, 'Quantitative Project Management through Metrics Analysis – A Case study at Infosys', Metrics 2002 8th International Symposium on software metrics, 2002

[6] Mohapatra S., Mohanty B., 'Achieving Quality in Software projects through Requirement Management', IECON'01 Special session on Software Engineering – IEEE, 2001.
[7] Mohapatra S., Mohanty B., 'Quantitative Software management an experience report at Infosys',The 15th Brazillian Syposium on software Engineering, 2003
[8] Mohapatra S., Mohanty B., 'Return on Investment through monitoring Cost of Quality at Infosys', WCRE,2003
[9] Mohapatra S., Mohanty B., 'Distribution Center Management Model – as as integral part of supply Chain Management' 2nd international Workshop on conceptual modeling approaches for e-business, 2001
[10] Mohapatra S., Mohanty B., 'Software Process Model for predicting defects – an experience report',2002
[11] Mohapatra S., Ravi M., 'Business benefits by monitoring cost of quality' International Conference by QAI,India , 2003
[12] Mohapatra S., Srinivas P., 'Defect Prevention through Defect prediction – A case study at Infosys', IEEE Conference, 2002
[13] Scacchi W., 'Understanding software Productivity', Advances in Software engineering and Knowledge engineering, 1995
[14] Scacchi W., 'Understanding and improving Software Productivity', 2005
[15] Scacchi W., ' Managing Software Enginnering Projects',IEEE Trans. Software Engineer, 1984 pp 49-59
[16] Symons, Charles, 'Software Sizing and Estimation : Mark II Function Point analysis',John Willey.1991