Journal of Science and Technology (JST)
ISSN : 2456-5660 Volume 4 Issue 1, Jan-February 2019, PP 40-52
www.jst.org.in

Significance of Software Layered Technology on Size of Projects

Jogannagari Malla Reddy[1], S.V.A.V. Prasad[2], B.V. Ramana Murthy[3]

[1](Research Scholar, Department of CSE, Lingaya's University, Haryana State, India)
[2](Prof & Dean (R & D), Department of CSE, Lingaya's University, Haryana State, India)
[3](Professor. Department of CSE, Mahaveer Engineering College, Hyderabad)

Abstract : *The objective of the software engineering is committed to build software projects within the budget, time and required quality. Software engineering is a layered paradigm comprised of process, methods, tools and quality focus as bedrock to develop the product. Software firms build software projects of varying sizes constrained on resources, time and functional requirement. Impact of software engineering layered technology may vary according to the size of the projects during their development. Quantitative evaluation of layer significance on size of the software project could be categorized as a complex task because it involves a collective decision on multiple criteria. Analytic Hierarchy Process (AHP) provides an effective quantitative approach for finding the significance of software layered technology on size of the projects. This paper presents the estimations through quantitative approach on real time data collected from several software firms. These findings help for a better project management with respect to the cost, time and resources during building a software project.*

Keywords - *Software Layered Technology, Agile and Non Agile Projects, Analytical Hierarchy Process (AHP)*

I. INTRODUCTION

Software industry anxiously looks for effective methods to improve the software product quality. Project management has been base for the progress of any engineering discipline to maintain the quality of a product and software engineering is no exception. The lack of quality has significant costs to developers in terms of wastage of effort; loss of market place, dissatisfaction from customers with faulty and rejected systems which fail to meet their goals. High maturity organizations expect to use metrics heavily for process and project management. The quality goals are defined though various measurement and metrics. Metrics have been used in quantitative decision making for risk assessment and optimizing the size of software projects.

Software engineering is a layered technology, encompasses various like quality focus, process, technical methods and use of tools to develop the software products [1]. Each layer has its own significance on different size of projects under development. The complexity of a software project is to be continuously measured, tracked and controlled. Software metrics measure different aspects of complexity, which plays the crucial role in analyzing and enhance the software development process. Literature reported research has stressed the importance of external quality aspects of software like maintainability, portability, reusability and reliability. Software process metrics measures the significance of the layers implementation. This paper presents the empirical evaluation of the role and significance of layered technology on size of the projects using Analytic Hierarchy Process. The work presents the relationship between quality and success rate in correlation with variables reflecting the organization and aspects of project's governance. The work infers that quantitative information provides the most effective solution to the problem of evaluating the significance of layered technology. The organization of the rest of the paper is as follows. Section 2 describes the related work focusing on literature reported work on metrics. Section 3 states the role of metrics in software engineering. Section 4 describes the evaluation of software layered technology significance on size projects with AHP mathematical derivations. Finally a discussion about future scope and inferences is given in the Section 5.

II. RELATED WORK

Over the years many theorists and researchers have worked on software engineering on the domain of quality metrics. The developed taxonomy can be benefited to extend the knowledge in improving the software quality culture.

- Organizations adapted ISO stands of quality in software development to excel their performance.

ISO/ICE 9126 quality model have various internal and external quality factors [2].

- Sadia Rehman et al[3] described the role of software metrics in the global software development with systematic literature for data search

- Kunal Chopra, et al [4] analyzed and evaluated the various aspects of software metrics to improve the quality culture of software development.

- W.K. S.D. Fernando, et. al [5] applied the questionnaire to find the importance of software metrics in software development projects of Sri Lanka.

- Fenton discussed the information system reliability of execution probability on given environment for a given period of time [6].

- Rawat, et.al [7], highlighted the various views on software quality and developed many metrics and models which are resulted as remarkable success.

- According to the Total Quality Management [TQM] literature and measurable criteria for assessing quality are necessary to avoid "arguments of style" (Deming, 1986). Measurable quality criteria are also essential for implementing statistical process control.

- Paulk et al. 1993 defined the intent of software process improvement for improving product quality, increasing productivity, and reducing the cycle time for product development.

- Farooq et,al [8], presented paper on software measurement and metrics in software development life cycle. The paper mainly highlighted on software test metrics and its role in software testing process.

- Thomas L Satty [9], [10], [11], [12] defined the principles and philosophy of Analytical Hierarchy Process for Multi criteria decision making approach.

- Evangelos Triantaphyllou and Stuart H. Mann [13] presented paper on Analytical Hierarchy Process is an effective approach for decision making. The paper examines the some of practical and computational issues involved in the AHP method used in the engineering applications.

- Gurdev Singh, et al [14] discussed the relationship between the software complexity metrics and various attributes of software system, advantages of Software metrics and classification of software metrics with more elaboration.

- Norita Ahmad presented paper on AHP for tool selection in software project management. The work establishes a framework for comparing individual product decisions across projects [15].

- Chandni and Parminder Kaur [16] evaluated the software architecture quantitatively under agile environment based on determined parameters.

- Amit Verma and Iqbaldeep Kaur [17] presented comparative Analysis of Software Engineering Paradigms according to the requirement of the specific Applications.

- Parita Jain and Laxmi Ahuja reviewed various success factors of agile software development and challenges faced in terms of assuring quality in agile [18].

- Ruchika Malhotra and Anuradha Chug [19] developed comparative Analysis of Agile methods for dynamic development and facilitates quick delivery with a scope of flexibility in continuous enhancement.

III. THE ROLE OF METRICS IN SOFTWARE ENGINEERING

In this section, we would first look the overview taxonomy of quality Metrics and its role in software engineering for developing the quality products.

A. Need of Measurement in Software Engineering

Measurement is necessary in our daily life. Economic measurements indicates the country financial strength, a patient blood samples help the doctors to diagnose the patient health condition. Humidity and temperature measurements indicates the whether forecasting for the geographical scientists. Without measurement, the technology cannot operate and control the situation. The measurement is a key element of any engineering process and there is no exception for software engineering.

In the nature most of the products are physical which consist of direct measurements, but software is logical product measured with indirect measurement. “What is not measurable and make into measurable” [4]. Measurement is for better understanding of the attributes and to assess the quality of software engineering projects/products as powerful as other engineering disciplines that we build.

The software organizations are using metrics in the project management for evaluation and conformation of effective software projects. Software engineering construct and maintain the software projects which includes activities like planning, managing and costing in various stages of software development life cycle. The metrics can be used in different phases of the software development life cycle. The metrics continuously observe, understand, controlled and measure software complexities. The objective of software metrics is that applicable to both process, project and product metrics.

“Metrics don’t solve problem, Metrics provide information so that people can solve problems” [20].

Metrics is essential tool for software measurement in development of quality products in software engineering. Measurement is a quantitative indication of extent, amount, dimension, capacity or size of attribute of product or process. It is act of determining a measure. The metric is a “quantitative measure of degree to which a system, component or process a given attributes” [IEEE].

The need and importance of metrics in software engineering is as follows.

- Determine the quality of the current product or process, what we developing.
- Predict & assess the quality of product or process.
- Improve and enhance the quality of product or process.

B. Characteristics

The quality factors are well defined in measurable or quantifiable. These metrics helps to indicate whether an organization or product is achieving significance of software goals. The following various characteristics are associated with metrics

- Metrics are simple, precise, well defined and easy to understand.
- Metrics are measurable, quantifiable and cost effective.
- Metrics are timely usable in robust and reliable.
- Metrics are more consistent used over the time.
- They are independent and accountable.

C. Software Metrics Goals

Metrics must have certain goals in order to facilitate certain features in the software industry. Metrics are quantifiable the product by accomplishing the following goals.

- Software metrics can measure the product for characterization, comparison, tracking, predication enhancement, validation and evaluation.
- Software metrics should provide information to take the decisions on quantitative base during the software lifecycle.

- The metrics used for decision making at risk assessment, reduction at evaluation stage.
  Software metrics measure the product defect predication, cost estimation and forecasting.
- The weight of the person is known attribute can be measured. But other attributes of person, such as knowledge creates a fuss or confusion.

- The length or height of the person measured with different scales like meters, inches, feet of the same attribute. The distance or height of the satellite from earth measured with miles but not with height of the person which again makes measurement definition far from accurate.

- The accuracy of the measurement depends on the instrument or metrics what we measured in proper. For example length can measured with accuracy as long as the ruler is accurate and when used in proper way.

Once we apply the measurements for different aspects of real world, we need to analyze the results and define the conclusions about the attributes which are derived. Results show what sort of changes or manipulations have to apply for the process to attain quality of the product.

IV. SIGNIFICANCE OF SOFTWARE LAYERED TECHNOLOGY ON SIZE OF THE OBJECTS

The objective of this research is to find the significance of process, methods and tools of software layered technology on size of projects quantitatively using Analytic Hierarchy Process. The research helps to estimate the significance of layers in the in the agile and non agile projects for comparison, tracking, evaluation and forecasting. The out comes of the research will useful for project management in the software industry.

The Project management methodology scales the projects into different sizes like small, medium, large size projects based on the following criteria:

- Availability of Financial resources
- The no. of team members participated
- Required time schedule
- Number and size of deliverables produced & its complexity

Financial resources, time frame and team size depends on the project size its complexity. Every project starts with initiation progresses with planning, execution and ended with closure regardless with its size and complexity.

Small Scale Projects: Small projects are agile and light weight projects. They have limited impact and scope with single goal. Agile projects are used quick plan, design and implement than non agile projects, which are manageable and produce immediate results. Once problem is solved, the project team disbands. Small projects tend to have limited time and resources. Agile projects require less effort in analysis and communication than non agile projects.

Medium Scale Projects The Medium Scale projects are moderate compare to small and large scale projects The little bit agile and non agile methodology applicable for the these projects. The resources and time also required according to the project size.

Large Scale projects: Large projects are non-agile and heavy weight projects. Large projects are complex which requires more effort, resources and time to develop the deliverables. Large project plans with multiple objectives, business needs with interdependent requirements. The development of large scale projects needs the set framework activities like requirement analysis, design, construction and deployment. There is need of extensive experience in developing heavy weight projects compare to light weight projects. The numerous resources are required at the project life cycle to initiate, plan, execute, control and deploy the project. The large projects are learning systems tend to have a large impact on the business, community and employees.

The software engineering is layered technology which encompasses the layers of process, methods and tools for development software projects regardless its size and complexity with intent of software quality. The software quality based on the remaining three layers which is base for software layered technology.

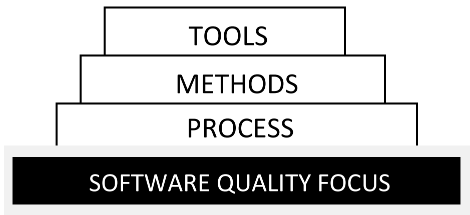

Figure. 1. Software Layered Technology

Quality Focus layer

The quality focus is the bedrock and backbone for software engineering. The various Total Quality Management, Six Sigma, Statistical analysis philosophies defined and targeted for improvement of quality focus. The software product should meet the stakeholder satisfaction.

Process layer

Organizational activities have prime relevance with the process layer. Process layer is the basic resource of organization in developing the software projects. It is the framework for timely delivery of software. Process layer determines deliverables, feasibility study, establish the milestones, software change and configuration management and software quality assurance and risk management.

Methods Layer

Methods layer of the software layered technology provides the technical knowledge (i.e. how to's") for building software. It comprises the array of tasks like Requirement Analysis, Design, program construction and deployment.

Tools layer

This layer provides support for process and methods by providing the automated and semi automated tools. These tools bring the software development process as rapid with automation. For Example the CASE (Computer Aided Software Engineering) tools may also include editors, database, test case generators and code generators etc.

A Sampling

The Sampling of this study comprised with various public, private sector software houses. The targeted population comprised of IT Practitioners with much minimum 15 years experience in software development. Besides, the IT Practitioners have various roles in the software development which includes Project manager, System analysts, Software Engineers, Tester and programmers. These Experts were selected in this study because they have much experience and involvement in software development life cycle of various types of projects. The reason why various roles were involved was do the diverse practice implemented by different organizations with standard job title or role is pertinent to handle software project development. For example, in an extreme case, a software engineer of agile project could be responsible to elicit requirements, design software, code and run the test case himself. Among the 55 respondents, 40 of them claim to be familiar and involved in producing inputs for the development.

B The AHP Methodology

The Analytic Hierarchy Process multi-criteria decision making method invented by ***Saaty*** in 1980 and improved by Vargas in 2001. AHP practically used in management science by Anderson et al., in 2000. It is a powerful and flexible tool for providing solutions for complex multi criteria decision-making applications. This tool is developed to solve complex problems selecting the alternatives based on multi criteria. AHP deals to measure intangible criteria and how to interpret in measurement of tangibles: so they can be combined with those of intangibles to yield sensible, not arbitrary numerical results (Satty, 2005).

C Structure of AHP method

Analytic hierarchy process divides the main problem into smaller and more detailed elements. The Analytical Hierarchy Process problem hierarchy organized into top level as goal, intermediate level as criteria's and lower level as alternatives. Figure 2 shows the hierarchical decomposition of criteria and alternatives.

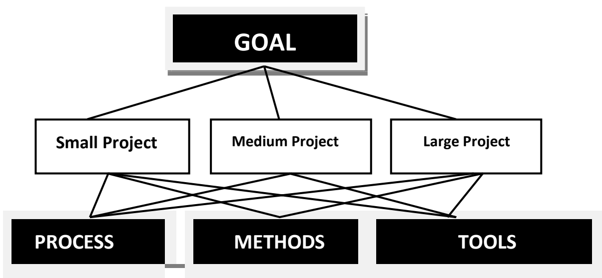

Figure 2: Hierarchical decomposition of Criteria's & alternatives

The pair wise comparison matrix represents the corresponding judgment of experts on the scale of relative importance (Saaty, 2008) of the following.

Table 1: Scale Of Relative Importance (Saaty ,1990)

Weight	**Definition**	**Explanation**
1	Equal importance	Two activities in *equal* importance
3	Moderate	One activity *moderate* over

	importance	another
5	Strong importance	One activity *strong* over another
7	Very strong importance	One activity *very strong* in practice over another
9	Extreme importance	One activity *Extreme* over another.
2,4,6,8	Intermediate values between two activities	When compromise is needed.
Reciprocals of above non Zero : If activity I has of above non nonzero numbers assigned to it when compared with activity j, then j has the reciprocal value when compared with it		

Pair-wise comparison is the next step to be carried in which the corresponding maximum left eigenvector is evaluated by geometric means of each row (Triantaphyllou and Mann, 1990). That is n-th root of product of all the elements in each row. Next the numbers are normalized by dividing them with their sum [13]. Initially the consistency index CI to be estimated. This is done by sum of columns in the judgment matrix and multiply the resulting vector by the vector of priorities (i.e approximated eigenvector) obtained earlier. This results the approximation of the maximum eigenvalue denoted by λ_{max}. Then, the C.I calculated with the formula as:

$$CI = (\lambda_{max} - n) / (n-1)$$

Then consistency ratio measured with formula,

$$CR = CI / RCI$$

RCI stands for Random Consistency index defined by the Saaty, 2000).

Table 2: Random Conistency Index Based On Matrix Size (Adopted From Saaty, 2000)

Matrix Size (n)	**Random Consistency Index**
1	0
2	0
3	0.58
4	0.90
5	1.12
6	1.24
7	1.32

8	1.41
9	1.45

If the problem has M alternatives and N criteria, then the decision maker is required to construct N judgment matrices (each criteria) of order M*M and one judgment matrix of order N*N (for N criteria). Finally, the decision matrix the final priorities denoted as A^i_{AHP} .

$$A^i_{AHP} = \sum_{j=1}^{N} a_{ij} w_j, \text{ for } i = 1, 2 \ldots M \text{ --------- (1)}$$

D. Algorithm

AHP methodology is as follows:

1. Define the problem and the prime objective to be decided.
2. Organize the problem into a hierarchical structure as shown in figure 2. Make the root node as goal of the problem, Intermediate level as criteria's and lower levels as alternatives. Construct a set of pair-wise comparison matrices. Compare each upper level element with corresponding lower level element and calibrate on numerical scale (i.e 1 to 9). In the matrix diagonal elements are equal or 1 and other elements will simply the reciprocals of earlier comparisons.
3. Calculate and find the maximum Eigen value (λ-max), consistency index (CI), consistency ratio (CR), and normalized values for each criteria/alternatives.
4. Evaluate the CR, if the CR value less than or equal to 0.10 which indicate a good level of consistency for decision making otherwise inconsistency of judgments within the respective matrix. Then the evaluation process should be reviewed, reconsidered and improved. The acceptable consistency helps to ensure decision making with more reliability.
5. The problem has M alternatives and N criteria, then the decision maker is required to construct N judgment matrices (for each criteria) of order M*M and one judgment matrix of order N*N (for N criteria). Finally, the decision matrix the final priorities denoted as A^i_{AHP} .

E. Mathematical derivatives

Step 1:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,n} \\ a_{21} & a_{22} & \cdots & a_{2,n} \\ a_{31} & a_{32} & \cdots & a_{3,n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{n,n} \end{bmatrix} a_{i,j} = 1, for\ i = j, a_{i,j} = \frac{1}{a_{j,i}} for a_{i,j} \neq 0$$

Step 2: Find the n^{th} root of product of each row.

Step 3: Derivation of Priority (pk). The numbers are normalized (each row n^{th} -root value) by dividing them with their sum.

Normalized Matrix

$$P = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ . \\ . \\ . \\ . \\ p_n \end{bmatrix} \qquad A = \frac{(AXP)}{P} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ . \\ . \\ \lambda_n \end{bmatrix}$$

$$\lambda_{\max} = \frac{\lambda_1 + \lambda_2 + \lambda_3 + \cdots + \lambda_n}{n}$$

Consistency Index $C_I = \frac{\lambda_{\max} - n}{(n-1)}$

Consistency Ratio $= \frac{C_I}{R_I}$ *R_I is Random Index.*

Step 4:

AHP formula for decision making

$A^i_{AHP} = \sum_{j=1}^{N} a_{ij} w_j$, for i = 1, 2 … M----------(1)

Weights of alternatives with respect to each of the criteria mentioned in the tables 3 to 5 and its priority vectors represented in pie graphs from figures 4 to 6.

Table3 shows the weights of three layers with respect to the Small Projects

Table 3: Weights Of Alternatives With Respect To Small Scale Projects

Criteria [C1] : Small Scale projects				
Small Project	**Process**	**Methods**	**Tools**	**Priority Vector (P)**
Process	1	1/2	1/4	0.132
Methods	2	1	1/4	0.208
Tools	4	4	1	0 .660
Total Priority				1.000
λ_{max} = 3.054, **CI** = 0.027, **CR** = 0.046				

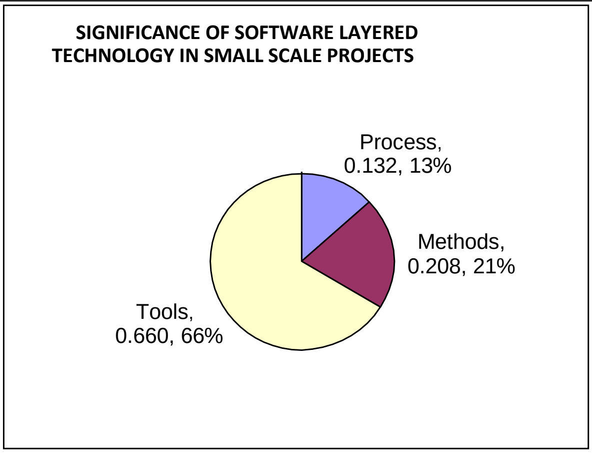

Figure 4: Weights of alternatives with respect to the Small Scale Projects

The next two matrices are respectively judgments of the relative merits of process, methods and tools of software layered technology with respect to Medium and Large Scale projects.

Table 4: Weights Of Alternatives With Respect To Medium Scale Projects

Criteria [C2] : Medium Scale projects				
Medium Projects	**Process**	**Methods**	**Tools**	**Priority Vector (P)**
Process	1	3	4	0.625
Methods	1/3	1	2	0.239
Tools	1/4	1/2	1	0.136
Total Priority				1.000
λmax. = 3.018, **CI** = 0.009 **CR** = 0.016				

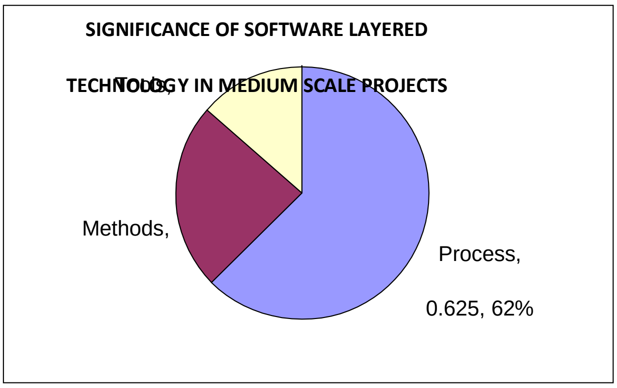

Figure 5: Weights of alternatives with respect to Medium Scale Projects

Table 5: Weights Of Alternatives With Respect To Large Scale Projects

Criteria [C3] : Large Scale projects				
Large Projects	**Process**	**Methods**	**Tools**	**Priority Vector (P)**
Process	1	2	9	0.655
Methods	1/2	1	2	0.250
Tools	1/9	1/2	1	0.095
Total Priority				1.000
λmax. = 3.074, **CI** = 0.037, **CR** = 0.063				

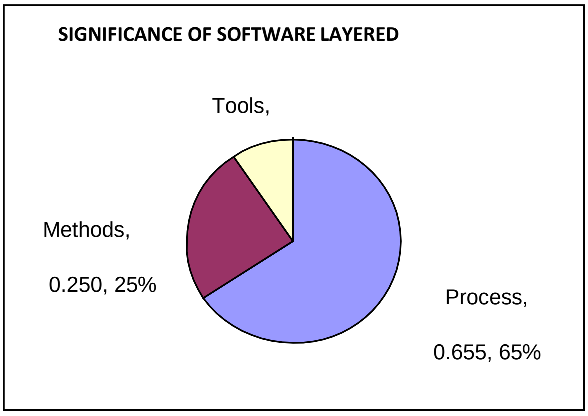

Figure 6: Weights of alternatives with respect to Large Scale Projects

In this problem, the work finds significance of software layered technology rather than best project, so all the projects have same significance. There is no need of criteria importance of Projects. Earlier computed priority vectors are used to form the entries of the decision matrix for this problem. The decision matrix and the resulted final priorities are as follows:

Table 7: Significance Of The Software Layered Technology On Size Of Projects

Layers of Software Layered Technology	Small Scale Projects	Medium Scale Projects	Large Scale Projects
PROCESS	0.132	0.625	0.655
METHODS	0.208	0.239	0.250
TOOLS	0.660	0.136	0.095
Total Priority	1.000	1.000	1.000

The significance of the software layered technology on size of projects shown in the figure 8.

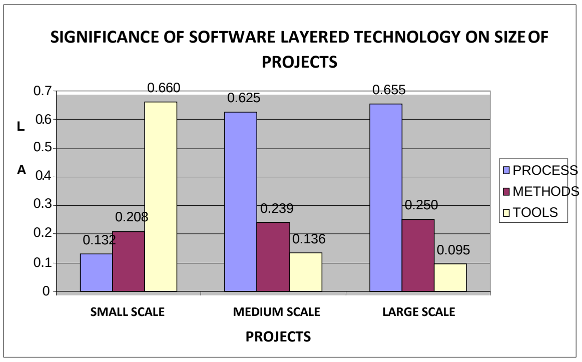

Figure 8: Significance of the software layered technology on size of projects

Therefore, the layer significance of software layered technology differs in agile and non agile projects.

F. Observations & Findings

From the research findings we can conclude that in the software industry under analysis the implementation of the small scale, medium scale and large scale projects have identifiable variance of layer significance visible in the above bar graph.

I. In Small Scale projects, the organization process involvement is very less compare to the Medium and Large Scale Projects. There moderate methods will be used in these projects. But utilization of automatic and semi-automatics tools involvements is more compare to the non agile projects.

People and their directions seems more important the processes and procedures.
Developing the software with help of tools is more in practice than technical documentation, budgets and plans.
Co-operation and direct interaction is more important than process, methods to quick plan, design and implement.
Since the usage of tools is much significant than other two aspects, therefore, it increases the rapid development time and more refactorization.
Finally it is also inferred that there is a need of light weight methodologies to develop these projects.

II. Medium and Large Scale projects have approximately similar characteristics. These projects are belonging to the non- agile projects.
Both projects heavily depend on organization processes and technical methods and tools.
These projects require more budgets, time compare to the small scale projects.

G. Observation

Process increasing from Small Scale projects to Large Scale Projects. But there is large variation between small and large scale projects and a small variation between Medium and Large scale projects as given in table 8.

Table 8: Process With Reference To Small, Medium And Large Scale Projects

PROCESS		
Small Scale	Medium Scale	Large Scale
13%	62%	65%

Table 9 gives the scope of technical methods with reference to the three cases. Here the leverage of using technical methods from small scale projects through large scale projects is increasing consistently.

Table 9: Methods With Reference To Small, Medium And Large Scale Projects

METHODS		
Small Scale	Medium Scale	Large Scale
21%	24%	25%

Table 10 gives an observation that the utilization of automatic and semi-automatic tools is reducing from small Scale projects to Large Scale projects.

Table 10: Tools With Reference To Small, Medium And Large Scale Projects

TOOLS		
Small Scale	Medium Scale	Large Scale
66%	14%	10%

Table 11 presents the relation between process and tools. Finally, the work infers that, when the process, methods increases then the use of tools reduces in the projects based on size. The work also strengthened the hypothesis that significance of software layered technology concept changes based the size of projects.

Table 11: Relation Between Process And Tools

RELATION BETWEEN PROCESS & TOOLS		
Projects	Process	Tools
Small	13%	66%
Medium	62%	14%
Large	65%	10%

V. CONCLUSION

AHP provides a convenient approach for solving complex Multi Criteria Decision Making problems in software engineering. Irrespective of size of the project, if the three aspects namely process, methods and tools have equal role during development then development time will be reduced without any compromise of product quality. However due to lack of economic resources and human effort, small projects tend to be developed more in automated environment. This leads to certain conflicts which results in poor performance of the developing product. However, the improved expertise of using automated tools may reduce such conflicts. The work has found that significance of software layered technology differs with the size of the projects.

Acknowledgements

We sincerely acknowledge to all the faculty members of the Department of Computer Science & Engineering, Lingaya's University, Faridabad and development software experts of various government and non government firms of software industry for their motivation and support in the period of research.

REFERENCES

[1] Pressman. R.S, "Software Engineering: Practitioner's Approach", Mc Graw-Hill, inc 1977.

[2] Manuscript , ISO, "ISO 8402," 1994.

[3] Sadia Rehman et al, "SWOT Analysis of Software Quality Metrics for Global Software Development: A Systematic Literature Review Protocol", *IOSR Journal of Computer Engineering*, Vol.2. Issue 1, 2012

[4] Kunal Chopra and Monika Sachdeva, "Classification of Software Projects Based on Software Metrics: A Review", *International Conference On Communication, Computing & Systems*, 2014.

[5] W.K.S.D. Fernando, et al, " The Importance of Software Metrics : Perspective of Software Development Projects in Sri Lanka",

SAITM Research Symposium on Engineering Advancements,2014

[6] N. Fenton and M. Neil, "Software Metrics and Risk", 2nd *European Software Measurement Conference*, 8 October, 1999.

[7] M.S.Rawat, A. Mittal, S.K. Debey , "Survey on Impact of Software Metrics on Software Quality", Vol.3, *International Journal of Advancement Computer Science and Applications*, 2012

[8] S.U. Farooq, S,M,K. Quadri , N.Ahmad, "Software Measurements and Metrics: role in Effective Software Testing", *International Journal of Engineering Science and Technology(IJEST),* Vol,3, No. 1,2011.

[9] Saaty, T.L (2000), "The Analytic Hierarchy Process", *Mc Graw-Hill International*, New York, NY, 2000.

[10] Saaty, T.L, "Multi criteria Decision making: The Analytic Hierarchy Process", McGraw Hill, NY, 1980.

[11] Satty, T.L,"Scaling method for priorities in hierarchical structures", *Journal of Mathematical Psychology*, pp 234 – 281, 1977

[12] Saaty, T.L: " How to Make a Decision : The Analytic Hierarchical Process*", European Journal of Operational Research*, Vol 48, pp 9 – 26, 1990

[13] Evangelos Triantaphyllou, et al, "Using the Analytic Hierarchy Process for Decision Making in Engineering Applications: Some Challenges", *International Journal of Industrial Engineering Applications and Practice,* Vol.2.No.1, PP. 35-44, 1995.

[14] Gurdev Singh, et al, " A Study of Software Metrics", *International Journal of Computational Engineering & Management*, Vol. 11, p 22-26, Jan 2011.

[15] Norita Ahmad, " Software Project Management Tools : Making a Practical Decision Using AHP", *Proceedings of the 30th Annual IEEE/NASA Software Engineering Workshop*, IEEE Xplore, 2006

[16] Chandni, et al, " Quantitative Evaluation of Software Architecture", *Proceeding of the INDIACom-2016, IEEE Xplore* ,2016

[17] Amit Verma, et al, Comparative Analysis of Software Engineering Paradigms ", *Proceeding of the INDIACom-2016,* IEEE Xplore, 2016.

[18] Parita Jain, et al, "Current State of the Research in Agile Quality Development", *Proceeding of the INDIACom-2016,* IEEE Xplore, 2016.

[19] Ruchika Malhotra, et. al., "Comparative Analysis of Agile Methods and Iterative Enhancement Model in Assessment of Software Maintenance *", Proceeding of the INDIACom-2016*, IEEE Xplore, 2016.

[20] Singh Yogesh & Pradeep Bhatia, "Module Weakness- New Measures", *ACM SIGSOFT Software Engineering Notes,* 81, July,1998.