International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 39808-39816
© Research India Publications. http://www.ripublication.com

Relative Analysis of Sizing Methods in the sense of E-Commerce system

John T Mesia Dhas,

Research Scholar, Vel Tech University, Assistant Professor, Department of C S E,
RVS Padhmavathy College of Engineering and Technology, Chennai, Tamil Nadu jtmdhas@ymail.com

Dr. C.R. Bharathi

Associate Professor, Department of Electronics and Communication Engineering
Vel Tech University, Chennai, Tamil Nadu crbharathi@veltechuniv.edu.in

Abstract
The estimation of size of software is great solution for determining the amount of time and effort needed for developing software systems. The success of any software project mainly depends on best estimation of project effort, time and cost. Estimation widely used in setting rational objectives for completing the project. The basic element for estimating all is size. The software industry uses various sizing techniques they are Lines of code, Function points, Feature points, Use Case points, Object points, Internet points, etc are not effectively supported for determining the size of E-Commerce systems. The wrong estimates lead imperfectness, loss and customer dissatisfaction. This paper explains the leading sizing techniques and their inabilities in sizing and also the necessities of new sizing approach for E-Commerce systems.

Keywords: LOC-Line Of Code, FPA-Function Point Analysis, UCP-Use Case Points, OP-Object Points, E-Commerce-Electronic Commerce

1. Introduction

Sizing is the estimation of coding needed to complete requirements. Every entity can be measured in terms of some units. Software size can be measured in terms of number of lines, counting functions, counting features, number of pages of user documentation, etc. The techniques supported are LOC, FPA, Feature Points, OP, Back firing, UCP, Internet points etc. This paper describe about the popular sizing approaches and their weakness in the sense of sizing E-Commerce system.

1.1. About E-Commerce System

Information Technology has transformed the way people work. Electronic Commerce (e-commerce) has unleashed yet another revolution, which is changing the way businesses buy and sell product and services. Associated to buying and selling of information, products and services over computer communication networks, e-commerce helps conduct traditional commerce through new ways of transferring and processing information, since it is information which is at the heart of any commercial activity. Information is electronically transferred from computer to computer, in an automated way.
E-Commerce refers to the paperless exchange of business information using electronic data interchange, electronic mail, electronic bulletin boards, electronic fund transfer, World Wide Web, and other network based technologies. E-Commerce not only automates manual processes and paper transactions, but also helps organizations move to a fully electronic environment and change the way they operate. After the e-commerce framework was announced by the US Government in 1997-at the time when the internet was allowed to be used by commercial organizations-it was the US Government's announcement that all federal purchases would be made paperless that gave an impetus to this new way of conducting trade and commerce. Surprisingly, it is an application that is today associated with e-governance, namely e-procurement. The internet gave yet another boost to e-commerce because it is a low cost alternative to proprietary networks. The more well known Electronic Data Interchange (EDI), the inter-organisational exchange of business documentation in structured, machine processable form over computer communication networks, is still the dominant part of e-commerce.

1.2. Basic Process to estimate the size

The following steps needed for estimating the size of software in linear way.

1. Define your size measure.
2. Identify all items to be built.
3. Estimate the size of each items using sizing approaches
4. Add up sizes of each item.
5. Validate the results
6. Repeat steps 2-4, if appropriate.

1.3. Evaluation of Sizing Techniques

The following figure 1.3.1 states the technique and the year in which it was developed. The software era functioning from 1950's, early stage executable instructions are stated in punched cards, size of program is assessed by counting punched cards. Afterwards various sizing techniques were introduced one by one based on the necessities.

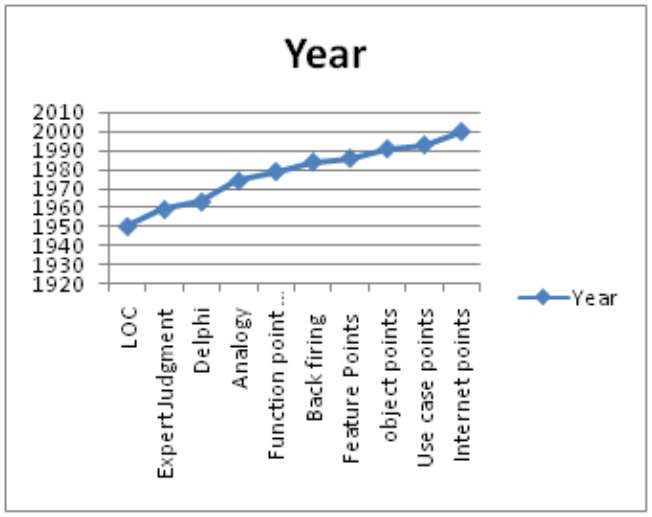

Figure 1.3.1 Evaluation of Sizing Techniques

The above diagram stated that every estimation techniques were introduced after some interval of timing. In the mean time various high level languages were introduced. Already available sizing techniques are helpless to estimate accurate size in some sense so the new techniques were introduced. Now a day's commerce sector uses various online applications to promote trade and commerce. E-Commerce is one among them. Sizing E-Commerce system is a difficult process because available estimation techniques are not supported to predict the size of software. The following section described about the sizing techniques and the reason in which they produce inaccurate size.

2. Classification of Sizing Techniques

Sizing Techniques are classified in to two broad categories based on their characteristics. They are linear method and parametric methods. Linear methods follow Expert oriented sizing techniques. They are Expert judgment, Delphi and Estimation by analogy. Parametric method assumes that the relationships among the different project parameters can be modeled using suitable mathematical expressions. The techniques under this method are LOC, Function point analysis, Feature Points, object points, Back firing, Use case points and Internet points. The following figure 2.1 shows the classification.

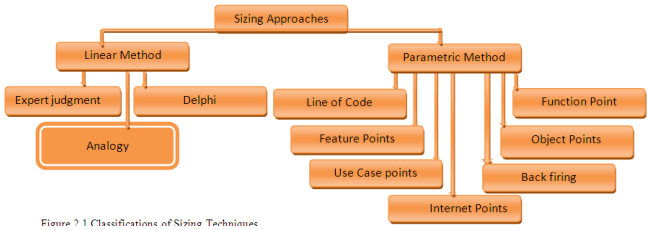

Figure 2.1 Classifications of Sizing Techniques

2.1. Expert judgment:

Expert or group of experts uses their experience and understanding of the proposed project they make estimation. Original technique arose from work done at the RAND corporation in 1950's [Helmer-1959] and matured in the following decade [Dalkey-1963].The following steps are used for estimation [1].

Steps:

- Coordinator presents each expert with a specification and estimation form.
- Coordinator calls a group meeting in which the experts discuss estimation issues with the coordinator and each other.. Experts fill out forms anonymously.
- Coordinator prepares and distributes a summary of the estimation on an iteration form.
- Coordinator calls a group meeting specially focusing on having the experts discuss points where there estimates varied widely.
- Experts fill out forms again anonymously and steps 4-6 are repeated many round s appropriate

Advantages:

- New project is similar to the old project they accessed then their estimates are good.
- Experts are good in the problem domain then we can receive accurate estimates from them.
- Good for level 3 and level 4 organizations.

Limitations of Expert judgment in the sense of E-Commerce system

- This method cannot be quantified.
- Human errors are possible
- Time consuming one.
- Spend more money for experts
- Estimating each part of E-Commerce system required different experts such as networking, security processes, transactions, authentications, etc. that lead more expensive

2.2. Wideband Delphi

Delphi technique originated at Rand Corporation. It is expert oriented; experts conduct review process based on application and provide estimates like cost, duration, effort, etc. For providing accurate estimates they have taken multiple iterations

Advantages

- Implementation is easy and in expensive in case of simple systems.
- It takes the expertise of several people.
- All participants become better educated about the software.
- It does not require historical data.
- It used for high level and detailed estimation.
- Results are more accurate and less dangerous than LOC

Limitations of Wideband Delphi in the sense of E-Commerce system

- Estimating each part of E-Commerce system required different experts such as networking, security processes, transactions, authentications, etc. that lead more expensive
- Single expert not having expected skill set.
- Expert may not provide 100% confidence regarding sizing.

2.3. Estimating size by analogy

Based on the size of similar projects developed in the past is supported for estimating the size of new software. For this estimation historical data and experts are needed. Sometimes scaling concept also used. This kid of guessing's not supported for E-Commerce system sizing because of complex parameters.

2.4. Lines of code

The computer era starts from the mid of 20th century. It is used from the beginning stage of evolution of programming languages. Main objective of LOC is count each executable instruction including data definition once and produce the size.

Advantages

- It is widely used and universally accepted.
- It measure software from the developer point of view.

Limitations of LOC in the sense of E-Commerce System

- E-Commerce system is a economic based web application hold huge different types of transactions.
- The worth of one line of a process is not equal to the one line of another transaction.
- The worth of one transaction that may contain authentication, banking processes, money transactions, etc is multiple times greater than LOC.
- Mapping of transaction and authentication part of E-Commerce system is difficult to the sizing concept behind LOC.

2.5. Function point analysis

A function point is a unit of measurement to express the amount of business functionality and information system provides to a user. It is a standard metric for the relative size and complexity of a software system, originally developed by Alan Albrecht of IBM in the late 1970s. Function points (FPs) can be used to estimate the relative size and complexity of software in the early stages of development-analysis and design. The size is determined by identifying the components of the system as seen by the end-user: the inputs, outputs, inquiries, interfaces to other systems, and logical internal files. The components are classified as simple, average, or complex. All of these values are then scored and the total is expressed in Unadjusted FPs (UFPs). Complexity factors described by 14 general systems characteristics, such as reusability, performance, and complexity of processing can be used to weight the UFP. Factors are also weighted on a scale of 0-not present 1-minor influence, to 5-strong influence. The result of these computations is a number that correlates to system size.

Advantages

- It calculates size in the user perspective.
- Although the FP metric doesn't correspond to any actual physical attribute of a software system (such as lines of code or the number of subroutines) it is useful as a relative measure for comparing projects, measuring productivity, and estimating the amount a development effort and time needed for a project.
- FP can be applied early in the software development lifecycle.
- It is independent of programming languages.
- It is a good sizing technique for the application programs in 1980's.

Limitations o FPA in the sense of E-Commerce system

- FPA focus on the computation part of an application. In 1980's, an application system has fully computational part. So that is focused on external inputs, outputs, inquires, internal logical files and External interface files. But E-Commerce system has different types of files, software, rules and regulations, economic transactions etc. Sizing of this part was not good in FPA.
- It is a count based method. The count of each component is high then it states the complexity is high. But it not considered the worthy of each component.
- It is not well suited to non-MIS applications especially E-commerce system like web applications.

2.6. Feature Point

It was the extension of FPA designated to deal with different kind of applications such as embedded system, real time system, system software, etc. In 1986, software productivity research developed feature point analysis. FPA never consider the complexity of algorithms involved in each application. To overcome that problem feature point method was introduced. The complexity of algorithms defined in terms of the number of rules required to express that algorithm. The formula for calculating Feature point FuP is

$$FuP = Raw\ Feature\ Point * CAF$$

Where FuP denotes Feature Point and CAF denotes Complexity adjustment Factor.

Determination of Raw Feature point:

Count inputs, outputs, files, inquiries, algorithms and interfaces required for a system and multiply with average weighting factor. All of these values are then scored and the total is expressed in raw feature point. The following table 2.6.1 assists for calculating raw feature point.

Table 2.6.1 Raw Feature Point Calculation

Feature Type	Average	Total
No. of Inputs	___ * 4 =	
No. of Outputs	___ * 5 =	
No. of Files	___ * 7 =	
No. of Inquiries	___ * 4 =	
No. of Interfaces	___ * 7 =	
Count the number of Algorithms	___ * 3 =	
Total Raw Feature Point _____		

Determination of CAF:
The complexity adjustment factor is calculated based on the two environmental factors. The range of influence of each factor is from 0 to 5. The environmental factors are logic values and data values. Logical value is assessed based on the complexity of algorithm or logics used in application..Data value is assessed based on the complexity of data used in algorithm or logics used in application. Following table 2.6.2 assist for finding the environmental factors of an application. Choose any one from each factor category.

Table 2.6.2. environmental factors and range of influence

Factors and values	
Logic Values (select one)	
Simple algorithms and calculations	-1
Majority of simple algorithms	-2
Average complexity of algorithms	-3
Some difficult algorithms	-4
Many difficult algorithms	-5
Data values (select one)	
Simple Data	-1
Numerous variables but simple relationships	-2
Multiple Fields, Files and Interactions	-3
Complex file structures	-4
Very complex files and data relationships	-5

The sum of logical value and data value provide environmental factor. Environmental factor ranges from 2 to 10. For each range of environmental factor specific CAF is assigned. The following table2.6.3 shows the CAF value for each range of environmental factor.

Table2.6.3 complexity adjustment factor and values

Environmental factor	CAF
2	0.6
3	0.7
4	0.8
5	0.9
6	1.0
7	1.1
8	1.2
9	1.3
10	1.4

Multiply the raw feature point with CAF provides the exact feature point of the system.

Advantages:
1. It is an excellent approach to size the algorithmically intensive system.
2. FP can be applied early in the software development lifecycle.
3. It is independent of programming languages. But naturally it is good for embedded system and real time system sizing.

Limitations of Feature points in the sense of E-Commerce system
1. It never considers the database and networking support needed for application.
2. It never considers other technical factors influenced the execution of e-commerce system.

2.7. Use case points
Use case point was introduced in the year of 1993 by Gustav Karner of objectory. It is an extension of FPA. It support for sizing in the early stage itself. The formula for calculating Use case point is
UCP = UUCP * TCF * EF
Where UCP = Use Case points.
UUCP = Unadjusted Use Case points.
TCF = Technical complexity factor
EF = Environmental Factor

Determination of Unadjusted Use Case points
UUCP can be calculated based on the unadjusted actor weight and unadjusted use case weight. Identify actors and its complexity from each use case of an application system.Find the weight. Weight may be 1, 2 or 3 based on the actor complexity that is simple average or complex. Sum the weight for the actors in all use cases to obtain the unadjusted actor weight. Similarly identify the use cases and assign weight 5, 10, 15 based on complexity. Sum the weight for all use cases to obtain the unadjusted use case weight. So the formula for calculating the unadjusted use case points UUCP is
UUCP = Unadjusted actor weight(UAW) + Unadjusted use case weight(UUCW)

Determination of technical complexity factor
Technical complexity of the product can be calculated based on the degree of influence of each of 13 factors. The Following table 2.7.1 describes the technical factors and their weight. It is similar to the CAF of FPA.

Table 2.7.1 Technical factors and their weight

Technical factor	weight
Distributed system	2
Response or throughput performance objectives	2
End user efficiency	1
Complex internal processing	1
Reusable code	1
Easy to install	0.5
Easy to use	0.5
Portable	2
Easy to change	1
Concurrent Processing	1
Include security features	1
Provide access for third parties	1
Special user training facilities are required	1

The degree of influence of each factor range from 0 to 5. For each factor, multiply the degree of influence by the weight, and sum the products to obtain the technical complexity sum TSUM. The formula for computing TCF is
TCF = 0.6+0.01*TSUM

Determination of environmental factor
It is calculated based on the 8 environmental factors which addresses the skills and training of the staff and requirement stability. The rating of influence range from 0 to 5. Multiply the rate of influence with weight and sum them to obtain environment sum Esum. The following table 2.7.2 shows the environmental factors and weight.

Table 2.7.2 Environmental factors and weight

Environmental factors	Weight
Familiar with rational unified process	1.5
Application experience	0.5
Object oriented experience	1
Lead analyst capability	0.5
Motivation	1
Stable requirements	2
Part time workers	-1
Difficult programming languages	-1

The formula for computing Environmental factor EF is
EF= 1.4-0.03*Esum

Advantages
- It support for estimating the size of software in the first phase of development itself.
- It is good for the application generated by using object oriented methodology.

Limitations of Use case points in the sense of E-Commerce System
- It also counts the number of actors and use cases involved in an application system and identify the complexity. But it never identifies the implementation level difficulty.
- Use case provides the initial view of the business model. But it is not much detail. Using this we can't provide exact estimates.
- Use case complexity is assessed based on number of transactions. It never considers the weight of code or inner part of use case.
- Sizing of the different levels of transactions in of E-Commerce system is difficult.
- Simulation, animation, video and audio specifications and their complexity not assessed.

2.8. Object points
Object point was introduced by Banker in 1991. It was object count instead of function count. Here the objects denote rule set, 3GL module, screens and reports. These objects are closer to the work done by developers. This approach meshes well with projects that use integrated computer aided software engineering environments to develop software.

Determination of object points
Count all instances of each object type. Each object is assessed with complexity weight. Sum up complexity weight of all objects to get the objects point (OP). Multiply OP by a reuse factor (RF) and find the new object point (NOP) using the following formula.
NOP = OP*(1-RF)

Advantages
- Good for GUI based applications.
- It highly considered reusability.

Limitations of object point in the sense of E-Commerce system
- E-Commerce system is also a FUI based application. Instead of screens, reports, and code lit of special objects are there, such as security based transactions etc. Object point suggested no way for sizing those items.
- E-Commerce system is a web based application. Accessed by variety of customers from geographically distributed area. So multiple system characteristics to be considered. But object point considered only reusability out of all technical and environmental factors influenced the system.

2.9. Other estimation methods
The above stated sizing techniques are popular in software industry. But the following sizing techniques are not much popular but some companies introduced them based on the need.

2.9.1. Web points
Assessing the size of web pages David Clary introduced this method in 2000. The size is assessed based on the complexity of web page. Complexity of each web page is considered based on the count of words and number of hyperlinks. Count the size of each page and summing them gives the size of application. E-Commerce system has multiple algorithms,

produce multiple reports, database involved and different media files involved. So this sizing technique is not suited for E-Commerce system. It supports for assessing the size of small web site.

2.9.2. Web Objects

It was introduced by Donald Reifer in 2000. It considers multiple objects of web pages like building blocks, web components, COTS components, graphic files, multimedia files and scripts. Count all objects and as like FPA web objects are sized. It is good for assessing the size of web site but E-Commerce system is highly more than a website. It is a secured transaction rich web application. Each of the module should be generated using different business and financial rules. So it is not sufficient for sizing E-Commerce system[2].

2.9.3. Halstead's Software science:

The Software Science developed by M.H.Halstead principally attempts to estimate the programming effort.
The measurable and countable properties are :

- n_1 = number of unique or distinct operators appearing in that implementation
- n_2 = number of unique or distinct operands appearing in that implementation
- N_1 = total usage of all of the operators appearing in that implementation
- N_2 = total usage of all of the operands appearing in that implementation

From these metrics Halstead defines :

i. the vocabulary n as $n = n_1 + n_2$
ii. the implementation length N as $N = N_1 + N_2$

Operators can be "+" and "*" but also an index "[...]" or a statement separation "..;..". The number of operands consists of the numbers of literal expressions, constants and variables

Length Equation:

It may be necessary to know about the relationship between length N and vocabulary n. **Length Equation** is as follows. " ' " on N means it is calculated rather than counted :

$N' = n_1 log_2 n_1 + n_2 log_2 n_2$

It is experimentally observed that N ' gives a rather close agreement to program length. Like length effort and duration also calculated.

Advantages of Halstead metrics :

- Do not require in-depth analysis of programming structure.
- Predicts rate of error.
- Predicts maintenance effort.
- Useful in scheduling and reporting projects.
- Measure overall quality of programs.
- Simple to calculate.
- Can be used for any programming language.
- Numerous industry studies support the use of Halstead in predicting programming effort and mean number of programming bugs.

Limitations of Halstead Software science in the sense of E-Commerce system

- It depends on completed code.
- Good only for the scientific language like FORTRAN.
- E-Commerce System is a secured economic transaction system. Sizing these elements are not specified in Halstead Software science.

2.9.4. Backfiring

Capers Jones of software productivity research developed a technique in 1984 called "Backfire" counting to estimate the size of existing legacy systems by counting the lines of code in the software product and then multiplying by a language specific conversion factor. This technique provides moderate accuracy. It is LOC based, so no way it is supported for assessing the size of E-Commerce system.

2.9.5. Object Oriented Size measures

Entities that persist in the world are modeled by a software program, which include both the application domain and solution domain. Application objects can be physical things, roles and events. Solution objects may be architecture elements and software components. The trick to obtain useful size measure is to stay near the application side. But application object provides limited information for sizing. So it mostly provide inaccurate estimate in early stage.

2.9.6. Model Blitz

This method suggested that based on the requirements; construct a model and this model support for sizing. It is a simple and quick sizing technique but sizing happened after design phase only. For E-Commerce system, model itself take more cost and time.

3. Comparative analysis of sizing techniques in the sense of E-Commerce system sizing

E-Commerce refers to the paperless exchange of business information using electronic data interchange, electronic mail, electronic bulletin boards, electronic fund transfer, World Wide Web, and other network based technologies. E-Commerce system is a web application so number of Web pages associated. Screens and reports are associated with E-Commerce system. Screens are act as a user interface and reports are the great output expected by the stakeholders of the system. The comparison table prepared based on the features of e-commerce system. Table 3.1 provides the details of comparison. This analysis stated that the most needful characteristics of E-Commerce system are not considered by each sizing techniques. So a new sizing technique required for estimating the size of E-Commerce system.

Table 3.1 Comparative analysis of sizing techniques in the sense of E-Commerce system sizing

Techniques/Features of E-Commerce System	**LOC**	**FPA**	**Feature Point**	**Use case point**	**Object point**	**Internet points (Web count, Web objects)**
Input and output consideration	Nil	Inputs and outputs are the two separate components for calculating Unadjusted function points	Inputs and outputs are the two separate components for calculating Raw feature points	Inputs are passed from actor and outputs are passed to actors. It count the number of actors but it never care input and output transactions	Nil	Nil
Logical files involvement	Count the number of executable codes of each file for calculating LOC	Count all files for calculating Unadjusted function points	Count all files for calculating Raw feature points	Mostly the logical files are the use cases. Use case modeling shows the major use cases. So it may not consider all the logical files involved in application	In the form of rule set or 3GL module, all the logical files are considered	Little bit considered.
Interface files	Nil	Count all interface files for calculating Unadjusted function points	Count all interface files for calculating Raw feature points	Nil	Nil	Considered in the form of hyperlinks
Web pages	Nil-it may count the lines of script but it is not useful because single executable instruction is not equal to single line of textual information in the script.	Nil	nil	nil	nil	Good for sizing web pages
Screens and reports (GUI Support)	Nil	Calculate inputs and outputs but it never identify the worth of screens and reports	Nil	Nil	Major components to calculate OP	Nil
Multimedia files-video, audio, simulation, animation	Nil	Nil	Nil	Nil	Nil	Little bit considered. Because the sizing is suitable for small website not for transaction rich E-Commerce system

Graphic files	nil	Nil	Nil	Nil	Nil	Little bit considered Because the sizing is suitable for small website not for transaction rich E-Commerce system
Textual document	It may count the lines of document. But it is not good	nil	nil	nil	nil	Count number of words and number of pages. So estimating the size of small web site it may good
Accuracy in sizing of E-Commerce system	Inaccurate	Inaccurate	Inaccurate	Inaccurate	Inaccurate	Inaccurate
Quality in sizing of E-Commerce system	Nil	Nil	Nil	Nil	Nil	Nil
Reusability	Nil	Considered	Nil	Nil	Highly considered	Nil
Database Support	Nil	Nil	considered	Nil	Nil	Nil
Data communication	Nil	Considered as one of the complexity adjustment factor	Nil	Nil	Nil	Nil
Internet and Network securities	Nil	Nil	Little bit considered	Nil	Nil	Little bit considered

4. Conclusion

The existing estimation techniques are not fully supporting for estimating the size of E-Commerce system. So for E-Commerce system we need a special Sizing technique for estimation which considers all the features of E-Commerce system.

References

[1]. Robert t. Futrell, Donald F. Shafer, Linda I. Shafer, "Quality software project management", Pearson Education, 2008.

[2]. Richard D. Stutzke, "Estimating Software-Intensive systems", Addison Wesley, 2005.

[3]. YinhuanZheng, Beizhan Wang, YilongZheng, Liang Shi, "Estimation of softeare projects effort based on function point", 2009, IEEE

[4] Capers Jones, "Estimating Software costs", 2nd Edition McGraw Hill companie

[5]. Zhang Wen-hua, "Computation methd of function points", Computer research and technolodge.

[6]. Ferchichi, J.P.Bourey, M.Bigand, M.Barron, "Design System engineering of software products" implementation of a software estimation model", IMACS-2006, Beijing, China

[7]. YinhuanZheng, BeizhanWang, "Estimation of software projects effort based on function point", Proceedings of 2009 4th International Conference on Computer Science & Education.

[8]. Liang Tian, AfielNoore, "Multistage Software Estimation" 2003 IEEE

[9]. Hareton Leung, Zhang Fan, "Software Cost Estimation", Department of Computing, The Hong Kong Polytechnic University.

[10]. PetrMusilek, WitoldPedrycz, Nan Sun, "On the Sensitivity of COCOMO II Software Cost Estimation Model", Dept. of Electrical and Computer Engineering, University of Alberta, Proceedings of the Eighth IEEE Symposium on Software Metrics (METRICS.02)

[11]. Rodrigo C. Barros, Duncan D. Ruiz, Nelson N. Tenório Jr., " Issues on Estimating Software Metrics in a Large Software Operation", Faculty of Informatics, Pontifical Catholic University of Rio Grande do SulPortoAlegre, Brazil, 2009 IEEE

[12]. MehwishNasir "A Survey of Software Estimation Techniques and Project Planning Practices", National University of Sciences and Technology, Pakistan, 2006 IEEE

[13] Allen, I. E. and Seaman, J. Staying the Course: Online Education in the United States, 2008 *NeedhamMA: Sloan Consortium*

[14] Allen, I.E. and Seaman, J. (2003) Sizing the Opportunity: The Quality and Extent of Online Education in the United States, 2002 and 2003 *Wellesley, MA: The Sloan Consortium*

[15] William D. Graziadei, Sharon Gallagher, Ronald N. Brown, JosephSasiadekBuilding Asynchronous and Synchronous Teaching-Learning Environments: Exploring a Course/Classroom Management System Solution.

[16] ShukorSanim M. Fauzi, M. HairulNizam M. Nasir, Nuraminah R., KamaruzamanJusoff, N. Azylia A. Azam, M. Hafiz Ismail, "The Implementation of Software Process Improvement Models", *International Review on Computers and Software*, Vol. 4. n. 3, pp. 402-407, May 2009.

[17] MuneerAlsurori, JuhanaSalim, "The Status of Information and Communication Technology in the Higher Education in Yemen", *International Review on Computers and Software*, Vol. 5 N. 6, pp. 712-723, November 2010.

[18] Rajib Mall, "Fundamentals of software Engineering"

[19] www.softwaremetrics.com/... Detailed FPA Concepts

[20]. Kamlesh K Bajaj, Debjani Nag, "E-Commerce", 2nd Edition, TaTaMcGrawHill 2009