Towards Better Accuracy and Risk Handling In Cost Estimation Using Assessment Values

Renu Goswami[1], Ajay Kumar Sharma[2]

M.Tech Scholar[1], Associate Professor[2]

Geetanjali Institute of Technical Studies, Udaipur, Rajasthan

Abstract— **Software product development is one of the well known fields which provide effective and timely development strategies. Among its various process of SDLC, cost estimation financial decisions with their usages for the featured development. It identifies the expected resources applied on the basis of some predefined models which apply the attributable analysis for cost estimation. Primary goal is towards getting the accuracy and optimality in estimation. This paper deals with study of various cost models along with the well known Constructive Cost Model. There are several different models available for the estimation of software project. One of the most popular cost estimation models is COCOMO. In intermediate COCOMO, software estimation is done based on scaling of fifteen cost drivers from very low to extra high. There is a lack of approach available for the selection of this scaling and also no effective estimation approach available for the cloud based application development.**

So this work formulates the factor using estimated monetary value (EAV) for considering the impact of risk in cost estimation as a quantitative approach. The EAV process is concerned with analyzing, identifying and responding to particular risks in its overall lifecycle. It actually applies the tolerance limits along with the uncertainties to the prior estimation. Here measuring the uncertainty at the initial stages of project lifecycle or estimation, some contingency plan might be developed. With the help of quantitative analysis of cost drivers, more accurate scaling in terms of risk can be done and effort adjustment factor calculated so that software project estimation must be done more persistently.

Keywords— Software Engineering, Cost Estimation, COCOMO, Risk Assessment, Estimated Assessment Value (EAV);

I. INTRODUCTION

Software development lifecycle is one of the important models applied for getting the quality development of software's. It includes the various process used to measure the resources, development time, requirements analysis, project evaluation and scheduling, cost calculation etc. Among them the cost estimation is the indirect measure for predicting the cost of the project. Such estimation depends upon the environment and the other derivative which affects the cost of product or project. Product having some previous knowledge about their features is easy to predict with their cost but for additional efforts cost needs to be added into it.

The existing estimation models serve the purpose of managing the tradeoffs between the client and the developer. A firm needs to analyse the things in earlier phases that what is easily deliverable to the client at affordable prices [1]. Thus to design effective and efficient cost estimation models some of the enlisted points are:

- *Identifies the critical cost drivers*
- *Apply the proper scaling factor required for variations in the cost drivers.*
- *Correlate the project with the previously developed features and codes*
- *Evaluate the problems and cost derivatives which raise the costs*
- *Apply the budget constraints and their impact towards development and organization policies*
- *Incorporate the critical features in an iterative manner.*

While designing the new cost model, critical analysis should be performed based on the features. These impact analysis will be able to predict the financial burdensome on the client and their respective market. Traditionally the COCOMO model is totally based on the cost derivers which were extended later with the number of attributes and size of the project. Among all the drivers the most important measure is its size units into Kilo lines of code (KLOC). A function point (FP) is one of the well known empirical measurement model for size prediction. This model might get affected with the size, development environment and the complexity of the complete project or product. Some of the dependent quantifiable entities of FP estimation for size are:

- *No. of Inputs*
- *No. of Outputs*
- *No. of Enquires*
- *No. of Files*
- *No. of Interfaces*

Constructive Cost Model (COCOMO)

Barry W Boehm in 1981[2] developed a effective model named as Constructive Cost model. It is an algorithmic cost model. Algorithmic cost model is developed based on relating the current project to previous projects. It is based on historical information [3]. Cocomo is based on size of the project. The size of the project may vary depending upon the function points. Types of COCOMO Models:

Intermediate COCOMO

- It is used for medium sized projects.
- The cost drivers are intermediate to basic and advanced Cocomo.
- Cost drivers depend upon product reliability, database size, execution and storage.[2]
- Team size is medium.

Limitations of COCOMO

- COCOMO model is applied to evaluate the effort required to fulfilled the development of given project including the design and development. Each phase is processed using the different models.
- Traditional model is not showing the perfect results as the attributes used for calculation might get affected by some uncertain conditions.
- The model is not capable of handling the over budget and under budget issues as the things are taken as constant during the complete development lifecycle of the project. It also affects the partial development of the system such as prototype model.
- The upgraded model of COCOMO is constructed with the maintenance cost and quality values which gives better accuracy in estimation.
- Traditional COCOMO is not appropriate for non-sequential, faster development and reusability cases [3].

COCOMO II

In way to achieve better accuracy towards estimation of the resources the extended model of COCOMO II is suggested in 1995. It overcomes the cost calculation problems associated with the no sequential rapid developments. Mainly the model is parted into three major areas:

- Application composition is used for projects having rapid development requirements and richness with GUI interfaces.
- Early design phase is applied to get an overview of the design pictures and done by considering the architectural perspectives.
- Post architectural designing is used to validate the given design on the users constructs.

In COCOMO II the constant value b is replaced by 5 scale factors. [5]
Effort (E) is calculated as follows

$$E = a * (KDSI)^{sf} * \pi (EM)$$

Where a is constant, sf is scaling factor, EM is Effort Multiplier (7 for Early design, 17 for Post architecture).

- Helps in making decisions based on business and financial calculations of the project.
- Establishes the cost and schedule of the project under development, this provides a plan for the project.
- Provides a more reliable cost and schedule, hence the risk mitigation is easy to accomplish.
- It overcomes the problem of reengineering and reuse of software modules.
- Develops a process at each level. Hence takes care of the capability maturity model.

II. WORK MOTIVATION

Project development starts with considering all the factors discussed at the time of requirement gathering. Later on the systematic SDLC process is formally applied for undergoing development task. Initially the lifecycle architecture is prepared to have accurate information on the cost driver inputs. Primary COCOMO was not able to cover the post architectural elements of the projects which were later on extended with COCOMO-II. This model considers the current hypothesis about the most effective and dynamic attributes of development stages based on subsequent data analysis. This analysis deals with the relationships of point's objects, function points and the line of codes. The same process is applicable for the various languages and composition systems, enabling flexibility in the choice of sizing parameters.

The data analysis involves the dynamic handling of various attributes which is suggested only by the COCOMO II like reliability, granularity, reusability, quality, schedule constraints, risk, overruns, resource management, change handling etc. This model presents a broader view of prior and posterior factors which affects the estimation.

III. LITERATURE SURVEY

During the last few years various approaches had been suggested for improving the traditional approaches of cost estimation. The aim is focusing towards getting more accurate results in the form of effective estimations. Among them some of the article which supports this research to be carrying forwards is taken here as surveyed papers.

In the paper [8] focus is made over improving the accuracy of schedule, effort and cost estimates. The estimation techniques can primarily be subdivided into two major categories: formal methods (parametric models such as COCOMO) and expert-judgment based methods. This research focuses on this question by analyzing when formal estimation methods are more useful than expert-based estimation given specific software quality requirements. Quality attributes such as usability, maintainability or portability are different in nature. They all have a particular influence on the software and system architecture and subsequently on the effort, schedule and cost necessary to develop the particular system. As the need for high quality software increases, it becomes more and more important to analyze the influence of quality requirements on costs. The goal is to more accurately predict the costs originating from high quality requirements. It is first important to understand those implications so that the inclusion of quality costs into project estimates will allow software companies to make better predictions. This will avoid big cost overruns because of unrealistic expectations and allow companies to justify quality costs towards their customers.

The cost, effort estimates of software projects done by the various companies. Out of those the effective model selections have to be made by getting the MRE (Mean Relative Error). *The paper [9]* have administered the historical data to COCOMO 81, COCOMOII model and

identified that the stellar predicament is that no cost model gives the exact estimate of a software project. This is due to the fact that a lot of productivity factors are not contemplated in estimation process. Along with the other element of the cost estimation the reusability is figured out for reducing the cost of already developed modules. It also evaluates the impact of the object orientation such as inheritance and polymorphism. It gives great return of investments using most of the legacy system reused by the developers. So further research exposure is in "software Reuse" and Reuse software cost estimation model. The paper also focuses on some of the reusable code study.

In this work [10], it is investigated the precision of size and cost drivers in the estimation of effort using Constructive Cost Model (COCOMO). All the factors applied to the cost derivative inputs will somewhere affect the certainty of output and the accuracy of the gross estimation. The size metrics can be used to represents the fuzzy data when the input fields are having unclear data for the defined attributes. It is always applied to the subjective category data with symmetrical triangles data and trapezoidal membership functions. This limitation can be resolved using Gaussian membership function along with previously selected COCOMO parameters. Apart from this improvement the paper also include the mechanism to join the benefits of both size and the cost attributes. While evaluating the proposed model of COCOMO we found that the suggested model is outperforming it and results were closer to the actual effort.

The paper [11] presents a meta-model that combines enterprise architecture modeling concepts with the COCOMO II estimation model. One of such model is ArchiMate which is used to describe the layers in enterprises architectures and to for example show what applications are used in what business processes. This paper also presents a specialization of ArchiMate that handles project specific factors. The project specific meta-model elements are then combined with the regular ArchiMate meta-model classes to calculate the migration cost estimate. The combined meta-model contains the seventeen effort multipliers as well as the five scale factors in a combination. The meta-model differentiates between the three ArchiMate layers as well as the new project specific meta-model classes. Our study proposes a method combining expert estimation with the meta-model-based approach to raise the accuracy of estimation process. The integration was verified for the combination of different categories of projects which resulted relative error of only 10%.

The paper [12] analyzes the process maturity (CMM) impacts on SDLC phases using some of the recently suggested attributes of COCOMO II. Analysis considers the 40 older projects with different maturity levels ranging from 1 to 4. Here the developed model is applied with the PRED and PMAT scale factors. It will also measure the prediction accuracy using the given PRED model. Finally the goal is to increase the strength of estimations in person per month. A common criterion for the evaluation of cost estimation models is the Relative Error (RE) or the Magnitude of Relative Error (MRE).The study showed that

the proposed model (with the new PMAT rating values) yielded better estimates as compared to the generic, constructive cost model II model's estimates.

Software metric and evaluation is base on measure of software factors which are typically related to the product, the process and the resources of software development. One of the greatest challenges for software developers is predicting the development effort for a software system based on some metrics for the last decades. One of the new approaches that called soft computing techniques may offer an unusual for this confront. This paper described an enhanced soft computing model for the estimation of software cost and time estimation [13]. The proposed model base on COCOMO II has two input's group from COCOMO II cost drivers and scale factors and one output, effort estimation. This model covers those three fuzzy steps, fuzzification process, inference from fuzzy rules and defuzzification process. A result show that the values of MMRE (Mean of Magnitude of Relative Error) apply soft computing was considerably lower than MMRE applying by algorithmic models.

In the paper [14] a new fuzzy based estimation system is proposed using realistic models with higher accuracy and reduced estimation efforts. It also analyzes the role of fuzziness and applies the special two sided Gaussian function. Here the Gaussian function which gave superior transition from one interval to another. After applying the process the results are measured by means of applying COCOMO II and proposed model based on fuzzy logic to the NASA dataset and created an artificial dataset, it had been found that proposed model was performing better than ordinal COCOMO II and the achieved results were closer to the actual effort. The relative error for proposed model using two-side Gaussian membership functions is lower than that of the error obtained using ordinal COCOMO II.

The paper [15] extends the fuzzy based estimation process using two newly suggested metrics for COCOMO. The first model uses SLOC as input variable and the effort (E) is calculated and the second model uses FP approach. The proposed fuzzy models show better estimation capabilities compared to other reported models in the literature and better assist the project manager in computing the software required development effort. The validation results are carried out using Albrecht data set.

IV. PROBLEM SCENARIO

Recent studies had suggested that the estimation had always shows a varying nature due to dynamic factors. Risk is one of that factors which modifies the project development constraints like schedule, budget, operational and technical. The mitigation strategies are developed for handling them but their impact on the budget is never analyzed effectively. Thus the risk must be quantized in smaller problems that are solved easily with less efforts and accuracy. Previously the estimated effort by the basic COCOMO and COCOMO II is not considering the change in expected condition. It should be compared with the all the previously measured estimates and from which the

forecasted direction and the current position of estimates must be calculated. Briefly the problem is described as:

"The risk factors are not present in basic COCOMO but with the later version COCOMO II, it is represented in the cost driver. This cost is evaluated once at the starting of the project and somewhere at the intermediate pivot points. There is no mechanism available which analyses the impact of dynamic risk evolutions on project factors. Partial values or smaller change could also be affected in cost measurement as it affects as a real time conditions."

In absence of the risk handling in cost the estimation could not be measured accurately.

V. PROPOSED SOLUTION

Cost estimation calculates the effort and resources applied to develop the specific requirements. All the requirements are managed statically without any consideration of dynamic risk associated with them. This work proposes a novel assessment value based estimation scheme for cost calculations. Mainly the risk affects the cost and decreases the profit towards development of product and by overcoming the issue reverse is measured. It

can be treated as threat coming at different situations during SDLC. Handling of risk is applied by mitigating the plans and analyses the overall process for getting the better reliability. Thus, the work develops a modular scheme of estimated assessment values (EAV) for analyzing the risk impact at different phases like requirements, planning, analysis, design and development. Previously the estimation model was not capable of handling the factors of risk coming dynamically.

The process starts with initiation of the project arguments and the requirements. Later on the work and the task from which the risk can affects the system is validated in terms of authorization. Once the work is authorized the cost drivers are selected for monitoring their behaviour. Each factor or stage of SDLC is somewhere affected by the risk thus a risk management module is applied on the basis of The EAV process will specifically analyze, identify and respond to a particular situation and measure the impact of considering them into the cost. It actually applies the tolerance limits along with the uncertainties to the prior estimation. Here measuring the uncertainty at the initial stages of project lifecycle or estimation, some contingency plan might be developed. The process architecture of given model is shown below in figure 1.
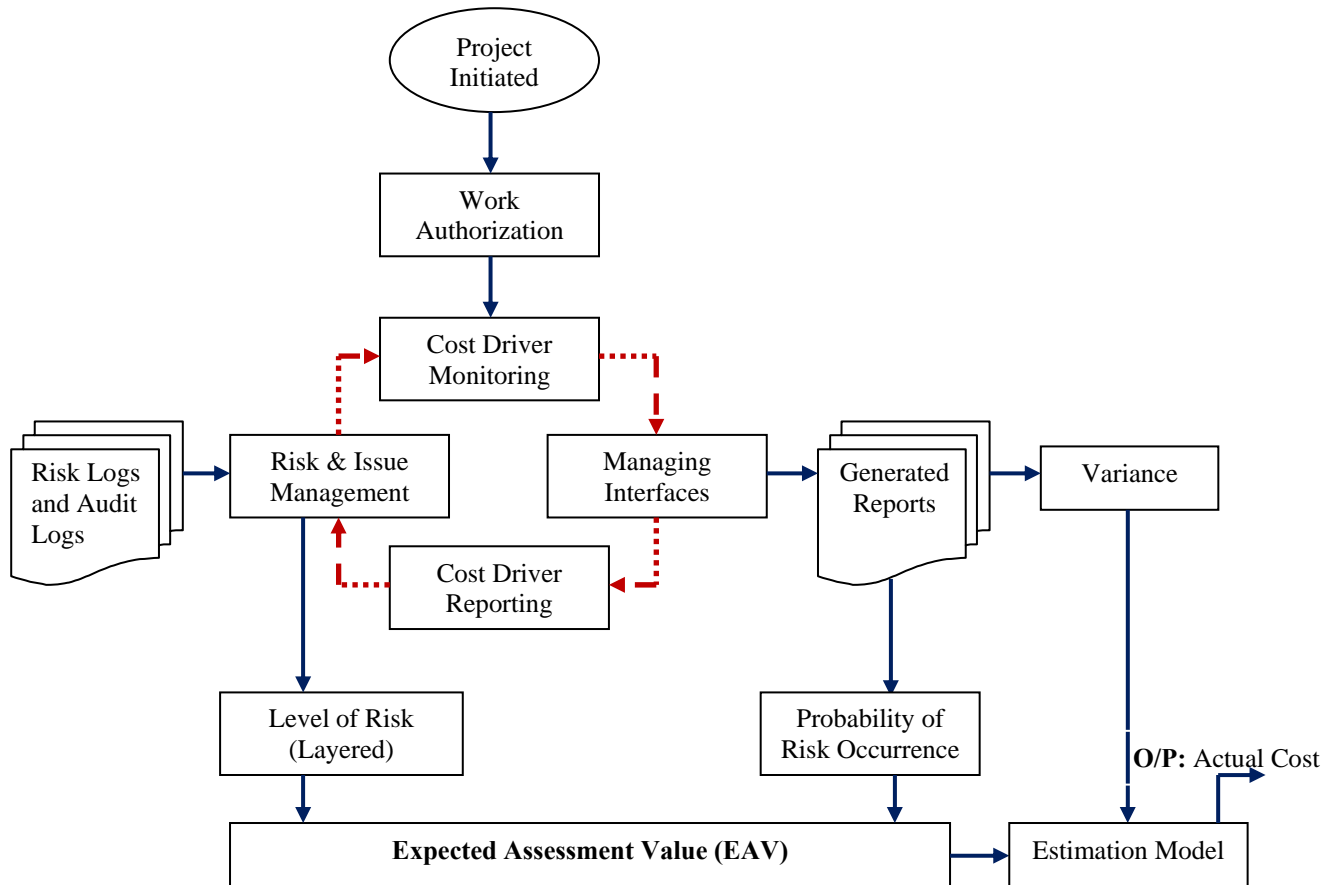
Figure 1: Process Architecture of Expected Assessment Value (EAV)

Some previous experiences with similar projects and types to get the impact in different situations. It will also handle the interfaces and reporting for change detection or dynamic evolution of requirements or any other entity. This process of analysing, identifying and responding to the appeared risk is having great importance in suggested EAV model. Here the objective is to measure impact of risk or level along with the probability of its occurrences. The basic calculation of EAV is represented as:

Expected Assessment Value (EAV) = Risk Occurrence Probability * Layered Risk

The above values can be treated according to situation like sometimes it was added to the project while it can also be subtracted from the overall profits. The complete expense difference with gained amount is considered to be baseline for project. It is the initial approved cost or profit structure for the project. While the risk is affecting the complete project thus the cost drivers are also revaluated to get more accuracy and reliability over the estimation process.

The dynamic risk based cost is measured as:

Cost Driver [1 to m] = Driver Value [Scaling] + Risk with Driver [EAV Value 1 to m]

The above mechanism is calculated for the cost factors having dynamic nature in both basic and COCOMO II. Now the final cost is given as:

Estimated Cost=$\sum_{i=1 \text{ to n, m}}$ [Cost Drivers + EAV Value (m)]* Variance (Person/Month Units)

In the above calculation we are also considering the variance factors which can be used in either cost form or in schedule form. It is measured by detecting the values which is going outside the baseline values. It deals with on time and budget delivery of project, thus if the SDLC is highly affected by dynamic risk then the variance will be higher and if the things are going as they were planned then variance is low.

Interpretation of Work

Since the Project Cost Variance is negative, this means the project is over-budget. Since Schedule Variance is negative, the project is behind schedule. This example project is in major trouble and corrective action needs to be taken to get it back on track. Estimated assessment values using cost variance and schedule variance will help you identify a project in trouble. It will also serve the purpose of effective risk management. Also by the above suggested model the estimation accuracy seems to be improved due to its complete nature covering the dynamic factors along with the adaptability using variance. The above equation also covers the various aspects of early and post architectures. As the estimation is making the individual cost factors multiplied with their EAV and variance, the output will be more precise and accurate cost.

Research Methodology

Considering the various aspects and the deriving primitives of cost estimation, we are using the normalization mechanism as research methodology. This selection was made on the basis of some questionnaires consisted on size of the product, schedule requirements and the effort measured. It also covers the product quality, reliability and maturity levels. By applying the normalization consistency is improved with better projection. It can be applied in three steps: first is to convert the data into equivalent sizes using reuse models. Second is to convert it into the logical countable units as source statements or SLOC. Third is to regroup the results based on their complexity and application types.

VI. EVALUATION PARAMETER

Cost Estimation Accuracy (CEA)

The cost estimation may vary due to changes in the requirements, staff size, and environment in which the software is being developed [4]. The calculation for cost estimation accuracy is given as follows

$$\text{Absolute error} = (E_{pred} - E_{actual})$$
$$\text{Percentage error} = (E_{pred} - E_{actual})/E_{actual}$$
$$\text{Relative error} = 1/n \sum (E_{pred} - E_{actual})/E_{actual}$$

Estimating Quality Factor (EQF)

The above results give a more accurate estimation of costs for future projects.[3] The cost estimation model now becomes more realistic. The EQF compares the actual realized value to different estimates that were made over time. Each estimated value is further weighted with the time the estimate was valid, whereas the realized value is weighted with the total time of the project. The EQF can be expressed as:

EQF= [Area under the Actual Value]/ [Area between the Forecast and Actual Values]

VII. EXPECTED BENEFITS

The potential outcome of the project's results on the software cost estimation is listed below:

- *Quantitative analysis of cost drivers of Intermediate COCOMO gives more accurate results for the estimation of effort and time compare to traditional approach.*
- *In the analysis of cost drivers, we include factors for the cloud based system also. So one can estimate time and cost for cloud based software development.*
- *With the help of system users can compare the results obtained for traditional and proposed system.*

Also it shows that for effort estimation or development time of traditional software development, our proposed strategy for estimating effort and development time using cost drivers gives better accuracy in comparison with the traditional mechanism due to the statistical nature of COCOMO cost drivers estimation.

VIII. CONCLUSION

Traditionally, software estimation with intermediate COCOMO approach is done based on selecting values of 15 cost drivers which ranges from very low to extra high. This approach solely depends on the users experience because currently values of the cost drivers are directly selected and therefore it will be difficult to measure accurate estimation. So we refer the 15 cost drivers of intermediate COCOMO and come out with the quantitative estimation of scaling of all 15 cost drivers by adding an additional impact of risk associated with them. By adding these dynamic values in the system through the estimated monetary value analysis, the accuracy of the estimation gets increased. As a result of work, we conclude that with the proposed approach of analysing each cost drivers value more accurate estimation can be made for traditional software development.

REFERENCES

[1] Module 11 Software Project Planning Version 2 CSE IIT, Kharagpur
[2] Robert W. Zmud Editor An Empirical Validation of Software Cost Estimation Models
[3] Cost Models For Future software life cycle processes: COCOMO 2.0, Barry Boehm, Bradford Clark, Ellis Horowitz and Chris westland, USC Center for Software Engineering, University of Southern California, Los Angeles
[4] The COCOMO II Suite of Software Estimation Models by Dr. Barry Boehm, University of Southern California
[5] 24th International Forum on COCOMO and Systems/Software Cost Modelling
[6] The influence of software quality requirements on the suitability of software cost estimation methods Marc Giombe, Ricardo Valerdi, Stefan Wagner
[7] Identifying the Importance of Software Reuse in COCOMO81, COCOMOII.
[8] By CH.V.M.K.Hari, Prof. Prasad Reddy P.V.G.D, J.N.V.R Swarup Kumar, G.SriRamGanesh, Dept of IT, Gitam University, Visakhapatnam, India. /International Journal on Computer Science and Engineering Vol.1(3), 2009, 142-147
[9] Improving the accuracy of effort estimation through Fuzzy set combination of size and cost drivers CH. SATYANANDA REDDY, KVSVN RAJU Department of Computer Science and Systems Engineering College of Engineering, Andhra University, Visakhapatnam, India
WSEAS TRANSACTIONS on COMPUTERS
[10] Software Migration Project Cost Estimation using COCOMO II and Enterprise Architecture Modeling Alexander Hjalmarsson1, Matus Korman1 and Robert Lagerström1, 1 Royal Institute of Technology, Osquldas v. 10, 10044 Stockholm, Sweden.
[11] Impact of CMMI Based Software Process Maturityon COCOMO II's Effort Estimation Majed Al Yahya, Rodina Ahmad, and Sai Lee Department of Software Engineering, University of Malaya, Malaysia. The International Arab Journal of Information Technology, Vol. 7, No. 2, April 2010
[12] Soft Computing Approach for Software Cost Estimation Iman Attarzadeh and Siew Hock Ow, Department of Software Engineering Faculty of Computer Science & Information Technology, University of Malaya, Kuala Lumpur, MALAYSIA, Soft Computing Approach for Software Cost Estimation
[13] A Novel Algorithmic Cost Estimation Model Based on Soft Computing Technique Iman Attarzadeh and Siew Hock Ow, Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia, Journal of Computer Science 6 (2): 117-125, 2010
[14] Software Effort Estimation Inspired by COCOMO and FP Models: A Fuzzy Logic Approach Alaa F. Sheta and Sultan Aljahdali, Computer Science Department, College of Computers and Information Technology Taif University, Taif, Saudi Arabia, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No. 11, 2013