ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Efficient Cost Estimation Techniques for Web Applications

Manpreet Kaur

Asst. Professor, Dept. of Computer Science, SRGCW, Amritsar, India

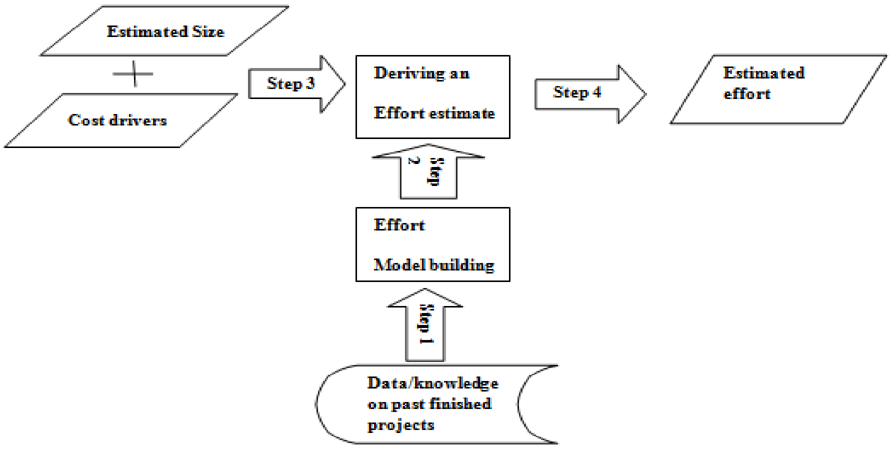
ABSTRACT: Web effort models and effort estimates help project managers allocate resources, control costs, and schedule and improve current practices, which allow projects to be finished on time and within budget. While web development and maintenance, these issues are also crucial and very challenging as Web projects have short schedules. The objective of this paper is to discuss the concepts related to Effort Estimation techniques.

KEYWORDS: Regression Analysis, Linear Regression, Stepwise Regression, CART, Case-Based Reasoning.

I. INTRODUCTION

Web is used as a delivery platform for numerous types of Web applications, ranging from complex e-commerce solutions with back-end databases to online personal static web pages and blogs[1]. With the sheer diversity of types of Web applications and technologies employed, there is ever- growing number of Web companies bidding for as many web applications as they can accommodate. Web Applications have evolved and became more complicated from time to time. Now a days, every web application can have different complexities. This becomes a problem to estimate how much effort is needed to finish a web application project. False effort estimation can lead to a delayed project, because there is not enough time to finish the project with the estimated effort. Technologies used for such applications are HTML, JavaScript and multimedia. It is developed to publish information over the web. A Web software application , is a conventional software application that depends on the web or use the web's infrastructure for execution. Such applications include databases, booking system, knowledge bases etc [1].
The purpose of estimating effort is to predict the amount of effort required to accomplish a given task based on the knowledge of previous similar project characteristics that are believed to be related to the effort. The project characteristics are the input, and effort is the output we wish to predict [1].

Figure 1: Steps used to obtain an effort estimate



DOI: 10.15680/IJIRCCE.2016. 0407101

Several techniques for effort estimation have been proposed over the past 30 years in software engineering. These fall into three broad categories [1].

Expert judgment (EJ) – EJ has been widely used. However, the means of deriving an estimate are not explicit and therefore not repeatable. Expert opinion, although always difficult to quantify, can be an effective estimating tool on its own or as an adjusting factor for algorithmic models. In Delphi method, no direct intervention is among the experts. Coordinator looks after the whole process.

Algorithmic models (AM) – AM to date the most popular in the literature, attempt to represent the relationship between effort and one or more project characteristics. The main "cost driver" used in such a model is usually taken to be some notion of software size (e.g. the number of lines of source code, number of pages, number of links). Algorithmic models need calibration or adjustment to local circumstances. Examples of algorithmic models are the COCOMO model (Boehm, 1981), the SLIM model (Putnam, 1978).The need for calibration of a model for each individual measurement environment. The variable accuracy level achieved even after calibration.

Machine learning (ML) - Machine learning techniques have in the last decade been used as a complement or alternative to the previous two categories. Examples include fuzzy logic models , regression trees , neural networks, and case-based reasoning.

II. SURVEY OF WEB COST ESTIMATION MODELS: A LITERATURE REVIEW

This section presents a survey of Web cost estimation models proposed in the literature.

First Study: Measuring Functionality and Productivity in Web-Based Applications: A Case Study [7]

The aim of this study was to build cost estimation models to help Web development companies predict and track development costs when new technologies are employed. This work gathered data on five Web applications developed using an object-oriented framework. Their cost estimation model takes into account the learning factor associated with writing new code that uses a framework, when this framework is used for the first time. They considered different types of reuse and the size measures employed were object-oriented function points and lines of code. Their cost models were generated using linear regression. Their results showed that the cost of writing new code, represented by calls to components provided by the framework, is reduced as developers are more experienced in using the framework.

Second Study: Measurement and Effort Prediction for Web Applications [11]

Mendes et al. (2000) investigated the use of case-based reasoning, linear regression, and stepwise regression techniques to estimate development effort for Web applications developed by experienced or inexperienced students. The case-based reasoning estimations were generated using a freeware tool called ANGEL developed at the University of Bournemouth, UK. The most similar Web projects were retrieved using the unweighted Euclidean distance using the "leave one out" cross-validation. Estimated effort was generated using either the closest analogue or the mean of two or three analogues. The two datasets (HEL and LEL) employed had data on Web applications developed by second-year computer science students and had 29 and 41 data points respectively. HEL represented data from students with high experience in Web development, whereas LEL had data from inexperienced students. Prediction accuracy was measured using MMRE and MdMRE. Results for the HEL group were statistically significantly better than those for the LEL group. In addition, case-based reasoning showed the best results overall.

Third Study: Web development: Estimating Quick-to-Market Software [9]

Reifer (2000) proposed a Web cost estimation model, WEBMO, which is an extension of the COCOMO II model. The WEBMO model has nine cost drivers and a fixed effort power law, instead of seven cost drivers and variable effort power law as used in the COCOMO II model. Size is measured in Web objects, which are calculated by applying Halstead's formula for volume. They are based on sub-components such as: number of building blocks (Active X, DCOM, OLE, etc.), number of COTS components (includes any wrapper code), number of multimedia files, except graphics files (text, video, sound, etc.), number of object or application points (Cowderoy, 2000) or others proposed (number server data tables, number client data tables etc.), number of xml, sgml, html and query language lines (number lines including links to data attributes), number of Web components (applets, agents, etc.), number of graphics files (templates, images, pictures, etc.), number of scripts (visual language, audio, motion, etc.) and any other measures that companies find suitable. Reifer allegedly used data on 46 finished industrial Web projects and obtained predictions which are "repeatable and robust."

Fourth Study: Web Metrics - Estimating Design and Authoring Effort [2]

Mendes et al. (2001) investigated the prediction accuracy of top-down and bottom-up Web cost estimation models, generated using linear and stepwise multiple regression models. They employed one dataset with data on 37 Web applications developed by honours and postgraduate computer science students. Gathered measures were organized into five categories: length size, reusability, complexity size, effort, and confounding factors (factors that, if not controlled, could influence the validity of the evaluation), and are associated to one of the following entities: application, page, media and program. Prediction models were generated for each entity and prediction accuracy was measured using the MMRE metric. Results showed that the best predictions were obtained for the entity program, based on non-reused program measures (code length and code comment length).

Fifth Study: Measurement, Prediction, and Risk Analysis for Web Applications [5]

Fester and Mendes (2001) investigated the used of proposed a generalized linear model (GLM) for Web cost estimation. Generalized linear models provide a flexible regression framework for predictive modeling of effort. The models allow non-linear relationships between response and predictor variables, and they allow for a wide range of choices for the distribution of the response variable (e.g., effort). Fester and Mendes (2001) employed the same dataset used in Mendes et al. (2001), however they reduced the number of size measures targeting at only the entity type application. These measures were organized into five categories: effort metrics, structure metrics, complexity metrics, reuse metrics and size metrics. In addition to proposing a prediction model, they also investigate the use of the GLM model as a framework for risk management. They did not measure prediction accuracy but relied on the model fit produced for the model. However a model with a good fit to the data is not the same as a good prediction model.

Sixth Study: The Application of Case-Based Reasoning to Early Web Project Cost Estimation [4]

Most work on Web cost estimation proposes models based on late product size measures, such as number of HTML pages, number of images etc. However, for the successful management of software/Web projects, estimates are necessary throughout the whole development life cycle. Preliminary (early) effort estimates in particular are essential when bidding for a contract or when determining a project's feasibility in terms of cost-benefit analysis. Mendes et al. (2002) focus on the harvesting of size measures at different points in the Web development life cycle, to estimate development effort, and their comparison based on several predictions accuracy indicators. Their aim was to investigate how different cost predictors are, and if there are any statistically significant differences between them. Their effort estimation models were generated using case-based reasoning. Their study was based on data from 25 Web applications developed by pairs of postgraduate computer science students.

Seventh Study: A Comparison of Development Effort Estimation Techniques for Web Hypermedia Applications [13]

An in depth comparison of Web cost estimation models is presented in Mendes et al. (2002b), where they: (1) compare the prediction accuracy of three CBR techniques to estimate the effort to develop Web applications and (2) compare the prediction accuracy of the best CBR technique, according to our findings, against three commonly used prediction models, namely multiple linear regression, stepwise regression, and regression trees. They employed one dataset of 37 Web applications developed by honours and postgraduate computer science students. Prediction accuracy was measured using MMRE, MdMRE, Pred(25) and box plots of residuals. Their results showed that different measures of prediction accuracy gave different results. MMRE and MdMRE showed better prediction accuracy for Multiple regression models whereas box plots showed better accuracy for CBR.

Eighth Study: An Empirical Study on the Design Effort of Web Applications [1]

Baresi et al. (2002) conduct an exploratory study, where, using an experiment, several hypotheses are tested. Amongst these, the study investigates whether estimated effort provided by students can be used to estimate actual effort. Their results show that it is possible to use the estimated values as predictors for the actual ones, however other variables, such as size, also need to be incorporated to the model to make it more realistic and meaningful. The dataset they used had data on 39 Web applications developed by computer science students.

Ninth Study: Cost Estimation for Web Applications [14]

The aim of Ruhe et al.'s study (2003) was to assess whether the COBRATM4 (Cost Estimation Benchmarking and Risk Analysis) method was adequate for estimating Web development effort accurately using data from a small Web company. COBRA is a method that aims to develop an understandable cost estimation model based on a company specific dataset. It uses expert opinion and data on past projects to estimate development effort and risks for a new project. The size measure employed was Web objects, measured for each one of the 12 finished Web applications used in this study. The prediction accuracy obtained using COBRA was compared to those attained employing expert opinion and linear regression, all measured using MMRE and Pred(25), giving COBRA the most accurate results.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Tenth Study: Do Adaptation Rules Improve Web Cost Estimation? [6]

This study compared several methods of CBR-based effort estimation, investigating the use of adaptation rules as a contributing factor for better estimation accuracy. They used two datasets, where the difference between these datasets was the level of "messiness" each had. "Messiness" was evaluated by the number of outliers and the amount of co linearity. The dataset which was less "messy" than the other presented a continuous "cost" function, translated as a strong linear relationship between size and effort. The "messiest" dataset, on the other hand, presented a discontinuous "cost" function, where there was no linear or log-linear relationship between size and effort. Both datasets represented data on Web applications developed by students. Two types of adaptation were used, one with weights and another without weights (Mendes et al. 2003). None of the adaptation rules gave better predictions for the "messier" data set , however for the less "messy" dataset one type of adaptation rule (no weights) gave good prediction accuracy. Prediction accuracy was measured using MMRE, Pred(25) and box plots of absolute residuals.

Eleventh Study: Estimating the Design Effort of Web Applications [2]

This study investigated the relationship between a number of size measures obtained from W2000 design artifacts and the total effort needed to design web applications. The categories employed were information model, navigation model and presentation model. They identified a few attributes that may be related to the total design effort. In addition, they also carried out a finer-grain analysis, studying which of the used measures have an impact on the design effort when using W2000. Their dataset comprised 30 Web applications developed by students.

III. TECHNIQUES FOR BUILDING WEB COST MODELS

This section details how to apply three different techniques to estimate costs for Web projects. The previous studies that include the use of these three techniques for building Web Cost Models are also explained.

1. Web Effort Estimation Using Regression Analysis

Investigating Web size metrics for early Web cost estimation [15]

This paper's aim is to bring light to the issue by identifying early size metrics and cost drivers for Web cost estimation based on Multivariate Regression Analysis. This is achieved using two surveys and a case study. This study identifies size metrics and cost factors important to be used in the Web cost estimation process early in the development life cycle.

2. Web Effort Estimation Using Case-Based Reasoning

This technique, proposed by the machine-learning community, uses the following claim as its basis: Similar problems provide similar solutions. CBR provides effort estimates for new projects by comparing the characteristics of the current project to be estimated against a library of historical data from completed projects with known effort (case base).

Measurement and Effort Prediction for Web Applications [11]

This study investigated the use of case-based reasoning, linear regression, and stepwise regression techniques to estimate development effort for Web applications developed by experienced or inexperienced students. The case-based reasoning estimations were generated using a freeware tool called ANGEL developed at the University of Bournemouth, UK. The most similar Web projects were retrieved using the unweighted Euclidean distance using the "leave one out" cross-validation. Estimated effort was generated using either the closest analogue or the mean of two or three analogues. The two datasets (HEL and LEL) employed had data on Web applications developed by second-year computer science students and had 29 and 41 data points respectively. HEL represented data from students with high experience in Web development, whereas LEL had data from inexperienced students. The size measures collected were page count (total number of HTML pages created from scratch), reused page count (total number of reused HTML pages), connectivity (total number of links in the application), compactness (scale from one to five indicating the level of inter-connectedness in the application. One represents no connections and five represented a totally connected application), Stratum, (scale from zero to one indicating how "linear" the application is. One represents no sequential navigation and five represents totally sequential navigation) and structure (topology of the application's backbone, being either sequential, hierarchical or network). Prediction accuracy was measured using MMRE and MdMRE.

DOI: 10.15680/IJIRCCE.2016. 0407101

Results for the HEL group were statistically significantly better than those for the LEL group. In addition, case-based reasoning showed the best results overall.

The Application of Case-Based Reasoning to Early Web Project Cost Estimation [4]

Most work on Web cost estimation proposes models based on late product size measures, such as number of HTML pages, number of images etc. However, for the successful management of software/Web projects, estimates are necessary throughout the whole development life cycle. Preliminary (early) effort estimates in particular are essential when bidding for a contract or when determining a project's feasibility in terms of cost-benefit analysis. This work focus on the harvesting of size measures at different points in the Web development life cycle, to estimate development effort, and their comparison based on several predictions accuracy indicators. Their aim was to investigate how different cost predictors are, and if there are any statistically significant differences between them. Their effort estimation models were generated using case-based reasoning, where several different parameters were used: similarity measure; scaling; number of closest analogues; analogy adaptation; and feature subset selection. Their study was based on data from 25 Web applications developed by pairs of postgraduate computer science students.

A Comparison of Development Effort Estimation Techniques for Web Hypermedia Applications [13]

An in depth comparison of Web cost estimation models is presented in Mendes et al. (2002), where they: (1) compare the prediction accuracy of three CBR techniques to estimate the effort to develop Web applications and (2) compare the prediction accuracy of the best CBR technique, according to our findings, against three commonly used prediction models, namely multiple linear regression, stepwise regression, and regression trees. They employed one dataset of 37 Web applications developed by honours and postgraduate computer science students. Prediction accuracy was measured using MMRE, MdMRE, Pred(25) and box plots of residuals. Their results showed that different measures of prediction accuracy gave different results. MMRE and MdMRE showed better prediction accuracy for Multiple regression models whereas box plots showed better accuracy for CBR.

Do Adaptation Rules Improve Web Cost Estimation? [6]

This study compared several methods of CBR-based effort estimation, investigating the use of adaptation rules as a contributing factor for better estimation accuracy. They used two datasets, where the difference between these datasets was the level of "messiness" each had. "Messiness" was evaluated by the number of outliers and the amount of co linearity. The dataset which was less "messy" than the other presented a continuous "cost" function, translated as a strong linear relationship between size and effort. The "messiest" dataset, on the other hand, presented a discontinuous "cost" function, where there was no linear or log-linear relationship between size and effort. Both datasets represented data on Web applications developed by students. Two types of adaptation were used, one with weights and another without weights. None of the adaptation rules gave better predictions for the "messier" data set, however for the less "messy" dataset one type of adaptation rule (no weights) gave good prediction accuracy. Prediction accuracy was measured using MMRE, Pred(25) and box plots of absolute residuals.

3. Web Effort Estimation Using Classification and Regression Trees

Classification and regression trees CART uses independent variables (predictors) to build binary trees where each leaf node either represents a category to which an estimate belongs, or a value for an estimate. The former situation occurs with classification trees and the latter occurs with regression trees; that is, whenever predictors are categorical (e.g., *Yes* or *No*), the tree is called a classification tree, and whenever predictors are numerical, the tree is called a regression tree. In order to obtain an estimate, one has to traverse the tree nodes from root to leaf by selecting the nodes that represent the category or value for the independent variables associated with the case to be estimated.

A Comparison of Development Effort Estimation Techniques for Web Hypermedia Applications [13]

An in depth comparison of Web cost estimation models is presented in this paper, where they: (1) compare the prediction accuracy of three CBR techniques to estimate the effort to develop Web applications and (2) compare the prediction accuracy of the best CBR technique, according to our findings, against three commonly used prediction models, namely multiple linear regression, stepwise regression, and regression trees. They employed one dataset of 37 Web applications developed by honours and postgraduate computer science students. Prediction accuracy was measured using MMRE, MdMRE, Pred(25) and box plots of residuals. Their results showed that different measures of prediction accuracy gave different results. MMRE and MdMRE showed better prediction accuracy for multiple regression models whereas box plots showed better accuracy for CBR.

IV COMPARISON OF DIFFERENT TECHNIQUES

Web effort estimation have used different techniques to estimate effort for new projects, each with a varying degree of success. There is no single effort estimation technique that always provides the most accurate prediction, and that the identification of such a technique may perhaps never occur.

Studies Published in 2000 (Mendes, Counsell, & Mosley, 2000) [11]

This study investigated the use of case-based reasoning (CBR), linear regression, and stepwise regression techniques to estimate development effort for Web applications. These applications were developed by both experienced and inexperienced students. The case-based reasoning estimations were generated using a freeware tool, ANGEL, developed at the University of Bournemouth, United Kingdom. The CBR technique used the measure of similarity employed in the study as the unweighted Euclidean distance using a jackknife approach.

The study used two separate data sets (HEL and LEL) containing data respectively on 29 and 41 Web applications developed by second-year computer science students from the University of Southampton, United Kingdom. HEL contains data on projects developed by students who had high experience in Web development previous to participating in the study, whereas LEL contains data on projects developed by students who had very low experience in Web development previous to participating in the study.

The accuracy of the prediction obtained using data on Web projects developed by the HEL group was significantly superior to the accuracy of the predictions obtained using data on Web projects developed by the LEL group. In addition, case-based reasoning showed the best results overall.

Studies Published in 2001 (Mendes, Mosley, & Counsell, 2001) [12]

This study compared the prediction accuracy between several Web effort estimation models using two multivariate regression techniques: linear regression and stepwise regression.

The data set used in the study contained data on 37 Web applications developed by honours and postgraduate computer science students from the University of Auckland, New Zealand. The measures that were gathered were organized into five categories: length size, reusability, complexity size, effort, and confounding factors (factors that, if not controlled, could influence the validity of the evaluation). Each measure was also associated to one of the following entities: application, page, media, and program. Effort estimation models were generated for each entity and prediction accuracy was measured using the MMRE measure. Results showed that the best predictions were obtained for the entity program based on non-reused program measures (code length and code comment length).

Studies Published in 2002 (Mendes, Watson, Triggs, Mosley, & Counsell, 2002) [13]

This study presents an in-depth comparison of Web effort estimation models using four different techniques: CBR, linear regression, stepwise regression, and classification and regression trees (CARTS). This was the first study to compare more than three different effort estimation techniques.

It compared the prediction accuracy between the best CBR configuration and three commonly used prediction models, namely, multiple linear regression, stepwise regression, and classification and regression trees. The data set used contained data on 37 Web hypermedia applications developed by honours and postgraduate computer science students from the University of Auckland, New Zealand. Their results showed that different measures of prediction accuracy gave different results. MMRE and MdMRE showed better prediction accuracy for multiple regression models whereas box plots showed better accuracy for CBR.

Studies Published in 2003 (Ruhe, Jeffery, & Wieczorek, 2003) [14]

This study investigated whether the COBRATM (Cost Estimation Benchmarking and Risk Analysis) method, previously employed in software engineering, was adequate to obtain accurate effort estimates for Web applications. It is a hybrid effort estimation technique that mixes expert judgment and multiple regressions, building a productivity estimation model.

Ruhe et al. adapted COBRA to Web projects using a small data set of 12 industrial projects developed by an Australian Web company. The size measure employed was Web objects , measured for each one of the 12 finished Web applications used in this study. The effort estimates obtained using COBRA were compared to those using expert opinion and linear regression. Prediction accuracy was measured using MMRE and Pred(25). COBRA provided the most accurate results.

Table 1 : Summary of literature on Web cost estimation where several estimation techniques are compared [19]

Study	Prediction techniques	Best Technique
Mendes et al. (2000)	Case-based reasoning, Linear regression, Stepwise regression	Case-based reasoning for high-experience group
Mendes et al. (2001)	Linear regression, Stepwise regression	Linear Regression
Mendes et al. (2003)	Case-based reasoning, Linear regression, Stepwise regression, Classification and regression Trees	Linear and Stepwise Regression or Case-based reasoning
Ruhe et al. (2003)	COBRA, expert opinion, Linear Regression	COBRA

V. CONCLUSION

Software practitioners recognize the importance of realistic estimates of effort to the successful management of software projects, the Web being no exception. Having realistic estimates at an early stage in a project's life cycle allow project managers and development organizations to manage resources effectively.

The literature shows that a minimum of two and a maximum of four different effort estimation techniques have been used in previous studies. It is learnt that multiple techniques can be implemented on single data to estimate the cost of web projects. This paper has presented a survey of previous work in Web cost estimation and has summarized their findings. This helps organize the body of knowledge in Web cost estimation and also helps those who wish to research or to know more about the field.

REFERNCES

[1] Baresi, "An Empirical Study on the Design Effort of Web Applications", 2002.
[2] Baresi, " Estimating the Design Effort of Web Applications", 2003.
[3] Emilia Mendes, "Cost Estimation Techniques for Web Projects", IGI Publishing, Hershey, New York, 2008.
[4] Mendes, " The Application of Case-Based Reasoning to Early Web Project Cost Estimation, 2002.
[5] Mendes, Fewster, "Measurement, Prediction, and Risk Analysis for Web Applications, 2001.
[6] Mendes," Do Adaptation Rules Improve Web Cost Estimation?", 2003.
[7]Morisio, "Measuring Functionality and Productivity in Web-Based Applications: A Case Study", 1999.
[8] Pressman R.S., "Software Engineering: A Practioner's Perspective", 5th ed., McGraw- Hill, New York, 2000, pp. 769-798.
[9] Reifer, D.J., "Web development: estimating quick-to-market software", Software, IEEE, vol. 17, No. 6, pp. 57-64, Nov/Dec 2000.
[10] Somerville, Software engineering, 6th Edition, Addison-Wesley, 2000.
[11]Mendes, E., Counsell, S., & Mosley, N.,"Measurement and effort prediction of Web applications", *2000, Proceedings of the 2nd ICSE Workshop on Web Engineering* (pp. 57-74).
[12]Mendes, E., Mosley, N., & Counsell, S. (2001), "Web metrics: Estimating design and authoring effort", *IEEE Multimedia, 8*(1), 50-57.
[13]Mendes, E., Watson, I., Triggs, C., Mosley, N., & Counsell, S. (2002, june), "A comparison of development effort estimation techniques for Web hypermedia applications", *Proceedings IEEE Metrics Symposium* (pp. 141-151).
[14]Ruhe, M., jeffery, R., & Wieczorek, I. (2003), "Cost estimation for Web applications" *Proceedings of ICSE 2003* (pp. 285-294).
[15] Mendes, Mosley & Counsell (2005), "Investigating Web size metrics for early Web cost estimation", Journal of Systems and Software, 77(2), 157-172.

BIOGRAPHY

Manpreet kaur is a Research Scholar in I.K. Gujral Punjab Technical University, Jalandhar. She received M.Tech (Comp. Sc. And Engg.) degree in 2010 from Guru Nanak Dev University, Punjab, India. Her research interests are Software Engineering, Effort Estimation of software development and Web-based Applications.