EVOLVING PREDICTOR VARIABLE ESTIMATION MODEL FOR WEB ENGINEERING PROJECTS

Sanjay Kumar Srivastava,

Poonam Prasad &

S.P. Varma

Department of Mathematics and Computer
Science,B.R.A. Bihar University,
Muzaffarpur, India
srivastavasanjaykumar@yahoo.co.in,
poonamprasadmca@yahoo.co.in,
spvarma2shailesh@gmail.com

Abstract: The field of web cost estimation is an important area which has not received much attention and has been the reason of failure of a number of web projects. Early web cost estimation can save disastrous situation. In this paper a review of previous work done in the field of web estimation like Size measures of Cowderoy, Mendes et al., Rollo and Cleary has been made. It further narrates Mendes Web Cost estimation Model, Tukutuku Project and derivation of effort equation. The paper investigates the applicability verification survey of web projects developed by Indian companies. The verification results being favorable Evolving Predictor Variable Estimation Model for Web Engineering Projects has been framed on the basis of curve fitting by sums of exponentials using Froberg's method and Moore's method. The model takes into the strength of development team as predictor variable and gives the gross effort in hours worked by taking into account the additional issues of contingencies, risk management and profitability issues which were neglected by Mendes et al. while creating their Early Web Cost Estimation Model. It further calculates the Current Sale Price of the web based project based on the development team in Indian Rupees (INR) at the 2014 price level. The early web effort and cost prediction has thus become more accurate and shall be advantageous at the project enquiry and bidding stage.

Keywords/Index Terms: Size Measures, Metrics, Early Web Cost Estimation.

1. Introduction

The Internet, it has been said, is the greatest invention of the twentieth century. Internet is an acronym for internetwork and is in fact a network of computer networks that may be dissimilar and are joined together by means of gateways that handle data transfer and conversion of messages from the sending networks' protocol to those of the receiving ne

World Wide Web (www) or simply the Web is the total set of interlinked hypertext documents residing on the Hyper Text Transfer Protocol (HTTP) servers around the world of Internet. The documents on the Web, called Web pages, are written in Hyper Text Markup Language (HTML) and are identified by Uniform Resource Locaters (URLs).

Web Engineering can be defined as the application of the principles of mathematics and science in order to create and utilize the web pages efficiently and effectively. Web Engineering Projects are concerned with the Web Applications (web apps) development, economically, timely and with least efforts as far as possible.

The software industry is facing greatly enhanced competitiveness due to the emergence of market leaders from the third world economies like India, Korea, Mexico, Malaysia etc. and China is the latest addition to the list. This has added fuel to the fire in the global recession that has severely affected the IT sector worldwide. In fact the need of the hour is to evolve cutting edge estimation techniques which deliver accurate estimates of size, effort, schedule and cost of software development projects. There has been rising trends towards the development of the network compatible web engineering projects catering to the needs of the multinational corporations in this era of business globalization. The time has come for an insight and in depth study and analysis into the factors affecting the web based projects' estimation [1]. The emergence of web as a delivery environment has catapulted both the commercial and educational web application development. Although a variety of development tools are available to the web developers yet the industry lacks a uniform and standardized development technology. The scarcity of data sets relating to the web projects development history and the lackadaisical approach of the web development industry has been a major repulsion for researchers in this area. The problem is further compounded in India by fierce competition in the software industry where the web development companies has been often trying every trick and resorting to every technique. They view with suspicion when asked for their past development data sets relating to records of project development effort, cost and time schedule etc. mistaking researchers with persons spying on behalf of rival companies eliciting information to be misused against them in forthcoming tenders and rate quotations. Under such challenges an investigation into virgin areas of Web Cost Estimation Techniques was undertaken. The finding of the investigation is organized into following four parts:

Part I, described in Section 2 presents an overview of Web Estimation Methods based on metrics like size, effort, cost and schedule already known.

Part II, described in Section 3 presents the Early Web Cost Estimation Model by Mendes et al. which was the first major industrial scale investigation on the basis of huge data sets culminating into validated formula to estimate the development effort.

Part III, described in Section 4 presents the Applicability Verification Survey of the Mendes Early Web Cost Estimation Model into the Indian Context.

Part IV, described in Section 5 presents an extension of the Mendes Early Web Cost Estimation Model by considering the contingency and profitability issues and the outcome being validated have been christened into Evolving Predictor Variable Estimation Model for Web Engineering Projects.

2. SURVEY OF PREVIOUS WORKS DONE IN THE FIELD OF WEB ESTIMATON

There are two categories of applications which broadly represent the applications delivered using the Web: Web hypermedia applications and Web software applications (Christodoulou et al., 2000). A Web hypermedia application is a non-conventional application characterized by the authoring of information using nodes (chunks of information), links (relations between nodes), anchors and access structures (for navigation) and its delivery over the Web. Technologies commonly used for developing such applications are HTML, JavaScript and multimedia. These applications have great potential in areas such as software engineering (Fielding and Taylor, 2000), literature (Tosca, 1999), education (Michau et al., 2001), and training (Ranwez et al., 2000). Web software application, conversely, represents more conventional software applications that depend on the Web or use the Web's infrastructure for execution. Typical applications include legacy information systems such as databases, booking systems, knowledge bases etc. Many e-commerce applications fall into this category. Typically they employ development technologies (e.g., DCOM, ActiveX etc), database systems, and development solutions (e.g. J2EE).

2.1 Web Size Metrics for Web Cost Estimation

To date few papers have proposed Web size metrics aimed at Web cost estimation (Cowderoy, 1998; Mendes et al., 1999; Cowderoy, 2000; Mendes et al., 2000; Reifer, 2000; Rollo, 2000; Cleary, 2000; Mendes et al., 2001). Cowderoy (1998;2000), Reifer (2000) and Cleary (2000) have used industrial data sets of Web projects to justify their size metrics and to generate corresponding cost models, each collecting their data from just one Web company, possibly affecting the external validity of their results. Mendes et al. (2001) proposes size metrics for static and dynamic Web applications and Mendes et al. (2000) proposes size metrics for

Web hypermedia applications. However the data sets employed in these studies are based on Web applications developed by students, which may also affect the external validity of their results. Each of these papers is reviewed in the following sub-Sections chronologically [5][6][7][8].

2.1.1. Size Measures by Cowderoy (1998; 2000)

Cowderoy (1998; 2000) recommends several size metrics for cost estimation and risk assessment of Web application development projects. Metrics were organized by the Entities to which they apply[2].

2.1.1.1. Web application

They include *Web pages* (WP), *Home pages* (HP), *Leaf nodes* (LN), *Hidden nodes* (HN), *Depth* (DE), *Application Paragraph count* (APC), *Delivered images* (DI), *Audio files* (AF), *Application movies* (AM), *3d objects* (3DO), *Virtual worlds* (VW) and *External hyperlinks* (EH).

2.1.1.2. Web page

They include *Actions* (AC), *Page paragraph count* (PPC), *Navigational structure*s (NS), *Page movies* (PM), and *Interconnectivity* (IN).

2.1.1.3. Media

It includes *Image size* (IS), *Image composites* (ICS), *Language versions* (LV), *Duration* (DU), *Audio sequences* (AS) and *Imported images* (IMI).

2.1.1.4. Program

It includes *Lines of source code* (LOC) and *McCabe cyclomatic complexity* (MCC) (Fenton and Pfleeger, 1997).

2.1.2. Size Measures by Mendes et al.

Mendes et al. (1999; 2000; 2001) proposed size metrics to be used to predict authoring effort for hypermedia applications and then for Web applications. All metrics are presented organized by Entities to which they apply [3].

2.4.4.2.1. Hypermedia application

Hyper document size (HS): the number of documents that the hypermedia application has. Documents are considered here to be either HTML files or any kind of file that is defined as a document in the hypermedia systems used in the evaluation.

Connectivity (CON): the number of links that the hypermedia application has. These links can be internal or external. Dynamically generated links are excluded.

Compactness (Botafogo et al., 1992) (COM): measures how inter-connected the nodes are.

S*tratum* (Botafogo et al., 1992) (STR): measures to what degree the hypermedia application is organized for directed reading.

Link Generality (LG): measures if the link applies to a single instance, for example point-to-point links, or whether it applies (or can be applied) to multiple instances.

2.4.4.2.2. Web application

Later Mendes et al. (2000) proposed a new set of size metrics, all targeting at Web applications:

Page count (PAC): the number of HTML or SHTML files used in a Web application.

Media count (MEC): the number of unique media files used in a Web application.

Program count (POC): the number of CGI scripts, JavaScript files, Java applets used in a Web application.

Total page allocation (TPA): the total space allocated for all HTML or SHTML pages (Mbytes) used in a Web application.

Total media allocation (TMA): total space allocated for all media files (Mbytes) used in a Web application.

Total code length (TCL): total number of lines of code for all programs used in a Web application.

Reused media count (RMC): the number of reused or modified media files used in a Web application.

Reused program count (RPC): the number of reused or modified programs used in a Web application.

Total reused media allocation (TRM): total space allocated for all reused media files used in a Web application (Mbytes).

Total reused code length (TRC): total number of lines of code for all programs reused by a Web application.

Code comment length (CCL): total number of comment lines in all programs in a Web application.

Reused code length (RCL): total number of reused lines of code in all programs in a Web application.

Reused comment length (ROL): total number of reused comment lines in all programs in a Web application.

Total page complexity (TPC): the average number of different types of media used in the Web application, excluding text.

Connectivity (CON): measures the total number of internal links, not including dynamically generated links.

Connectivity density (COD): computed as *Connectivity* divided by *page coun*t.

Cyclomatic complexity (Fenton and Pfleeger, 1997) (CCO): computed as *Connectivity -page coun*t) + 2.

2.4.4.2.3. The Revised List

This list was revised to include also bottom-up metrics (Mendes et al., 2001): Web page

Page allocation (PAL): measures the allocated space of a HTML or SHTML file (Kbytes).

Page complexity (PCO): the number of different types of media used on a page, not including text.

Graphic complexity (GRC): the number of graphics media used in a page.

Audio complexity (AUC): the number of audio media used in a page.

Video complexity (VIC): the number of video media used in a page.

Animation complexity (ANC): the

number of animations used in a page.

Scanned image complexity (SIC): the number of scanned images used in a page.

Page linking complexity (PLC): the number of links per page.

Media

Media duration (MED): the duration of audio, video, and animation (minutes).

Media allocation (MEA): The sizes of a media file (Kbytes).

Program

Program Code length (POL): the number of lines of code in program.

2.4.4.3 Size Metrics by Rollo (2000)

Rollo (2000) did not suggest any new size metric; however, he was the first researcher to investigate the issues of measuring the size of Web hypermedia and Web software applications, aiming at cost estimation, using several function point analysis methods. He measures the size of two applications in IFPUG, *MKII*, and *COSMIC-FFP* (Common Software Measurement International Consortium-Full Function Points) methods. Rollo (2000) concludes that *COSMIC-FFP* proved to be the most flexible approach for counting the functional size of Web hypermedia and Web software applications and can be applied to any Web application.

2.4.4.4. Size Metrics by Cleary (2000)

Cleary (2000) proposes size metrics for Web cost estimation dividing them into two types: size metrics for Web hypermedia applications and size metrics for Web software applications.

2.4.4.4.1. Web hypermedia application

Non-textual elements (NTE): the number of unique non-textual elements within an application.

Externally sourced elements (ESE): the number of externally sourced elements.

Customized infra-structure components (CIC): the number of customized infra-structure components.

Total Web points (TWP): the total size of a Web hypermedia application in Web points by adding: number of Web pages of “Low” complexity multiplied by the value for “Low” Web points; with number of Web pages of “Medium” complexity multiplied by the value for “Medium” Web points; with number of Web pages of “High” complexity multiplied by the value for “High” Web points.

2.4.4.4.2. Web software application

Function points (FPS): the functionality of a Web software application. Does not specify any particular method.

2.4.4.4.3 Web page

Non-textual elements page (NTP):

the number of non-textual elements within a Web page.

Words Page (WOP): measures the number of words in a Web page.

Web points (WPP): the length of a Web page. Scale points are “Low”, “Medium” and “High”. Each scale point is attributed a number of Web points, previously calibrated to a specific Web projects data set.

Number of links into a Web page (NIL): the number of incoming links; can be internal or external links.

Number of links out of a Web page (NOL): the number of outgoing links; can be internal or external links.

Web page complexity (WPC): the complexity of a Web page based upon its *number of words*, and combined *number of incoming* and *outgoing links*, plus the *number of non-textual elements*. The scale points are “Low”, “Medium” and “High”. Value ranges are provided for each scale point, for number of words and combination of *incoming links* + *outgoing links* + *non-textual elements*. These values have been calibrated based on a specific Web projects data set.

2.4.4.5. Size Measures by Reifer (2000)

Reifer (2000) proposes a size metric called Web Objects, which measures the number of Web Objects. Size is measured using an adaptation of Halstead’s equation for volume, tuned for Web applications. The equation is as follows:

$V = N \log_2(n) = (N1 + N2) \log2 (n1 + n2)$ (1)

Where:

N = number of total occurrences of operand and operators

n = number of distinct operands and operators

$N1$ = total occurrences of operand estimator

$N2$ = total occurrences of operator estimators

$n1$ = number of unique operands estimator

$n2$ = number of unique operators estimators

V = volume of work involved represented as Web Objects

Operands are comprised of the following metrics:

Number of building blocks (NBB): number of components, e.g., Active X, DCOM, OLE.

Number of COTS (NCO): number of COTS components (including any wrapper code).

Number of multimedia files (NMM): number of multimedia files, except graphics files (text, video, sound etc).

Number of object or application points (Cowderoy et al., 1998; 2000) (NOA): the number of object or application points or others proposed (# server data tables, # client data tables etc).

Number of Lines (NLI): number of xml, sgml, html and query language lines (# lines including links to data attributes).

Number of Web components (NCM): number of applets, agents etc.

Number of graphics files (NGR): number of templates, images, pictures etc.

Number of scripts (NSC): number of scripts for visual language, audio, motion etc.

3. MENDES EARLY WEB COST ESTIMATION MODEL

All size metrics presented in the Section 2 were invariably related to implemented Web applications. Even when targeted at measuring functionality based on function point analysis, researchers only considered the final Web application, rather than requirements documentation generated using any existing Web development methods. This makes their usefulness as early effort predictors questionable. Mendes et al. (2006) conducted surveys and case study to bring light to this issue, not only by identifying early size metrics and cost drivers based on current practices of several Web companies worldwide, but also by comparing these identified metrics to those that have been proposed in the past, looking for possible convergence[4].

3.1. Mendes First Survey: Using on-line web project price quote forms

The purpose of this survey (S1) was to identify early Web size metrics and factors used by Web companies to estimate Effort for Web projects early on in the development cycle. The target population was that of Web companies that offer online Web project price quotes to customers. There was no need to contact Web companies directly, only to download their on-line Web project price quote forms from the Web. To obtain sample population a number of questions were asked by sending web forms and getting replies online.

3.1.1. Survey Results

The data collected from 133 on-line quotes was organized into six categories: Web application static metrics, Web application dynamic metrics, Cost Drivers, Web project metrics, Web company metrics, Web interface style metrics. The survey showed that out of these metrics two metrics stood out, ***total number of Web pages*** *(70% companies)* and ***features/functionality*** *(66% companies)*. Both can be taken as size metrics where the first is a typical **length size metric** and the second an abstract measure of **functionality**. Seventy four (74) Web companies were also asked for the available Web project budget. Mendes et al. believe this metric can have a bearing on the contingency and/or profit costs that are provided in a price quote. Project estimated end date, project estimated start date and application type also were important. Mendes et al. believe these help set priorities and perhaps decide on what skills are necessary and available to the project.

3.1.2 Case Study for validating the results obtained from Survey

The survey identified metrics related to a project's price quote. Mendes et al. applied to their work the same model employed in Kitchenham et al., 2003, where price is the results of three components: estimated effort, contingency and profit. Since their objective was to identify only those metrics specifically targeted at effort estimation, they employed a case study and a second survey to identify the subset of metrics obtained in first survey directly related to effort estimation. The case study consisted of contacting an experienced Web company to confirm/deny, based on the ranking provided, those metrics they consider important for early Web cost estimation. The Web Company contacted is based in Rio de Janeiro, Brazil and five people work in Web design and development within the company and has to date a portfolio of more than 50 Web applications. They document their development process and record size and effort metrics from past projects. Depending on the type of project, one out of two types of process models is used: prototyping or waterfall. The choice depends on the Company's familiarity with the application domain. The Company's effort estimation practices are based solely on expert opinion, where the average estimation accuracy for their Web projects, based on effort estimates obtained early in the life cycle, is 10%.

The company's director was asked to help validate those metrics obtained from first Survey and validation here represents identifying size metrics and cost factors important to be used in the Web cost estimation process early in the development life cycle. Therefore all metrics from earlier survey selected by the Company director were actually employed on their Web cost estimation process. This person has worked in software development for more than 20 years and is experienced in management of large projects, conventional or Web-based. For Web application static metrics agreement was reached for most metrics. For Web application dynamic metrics more features/functionality were added to the list and the director suggested that adding a complexity level to each feature/functionality would help discern more difficult implementations. More specifically, this Company groups functions/features within three groups: simple, complex and very complex. Each has an associated baseline, which represents a percentage to be added to estimated effort. The Company's baselines reflect average percentages based on past experience. These metrics were also confirmed as suitable for early Web cost estimation. The metrics Mendes et al. have obtained as a result of their first survey and validated by the mature Web Company corroborate their findings.

3.2 **Second Survey for validating the results obtained from First Survey**

The purpose of this second survey was also to validate the results that have been obtained from first survey. Mendes et al. have also considered in this survey some of the results obtained from the case study, more specifically regarding Web application dynamic metrics. The target population was that of Web companies in New Zealand that estimate effort for their Web projects. The survey instrument was a questionnaire with nine questions was prepared by one of the team members, and the method of gathering data was via interviews over the phone. The results they obtained validated to a large extent the results obtained from the first survey.

3.3. **Tukutuku Benchmarking Project**

The feedback obtained from the first and second Surveys and the case study was used by Mendes et al. to prepare Web forms to gather data on Web projects worldwide. This data gathering initiative was called the Tukutuku benchmarking project11. The Tukutuku project aimed at gathering data on Web projects worldwide to be used to develop Web cost estimation models based on early effort predictors and to benchmark productivity across and within Web Companies. While preparing the project data entry forms used in Tukutuku, careful consideration was given to differentiate more complex features/functions from less complex ones, as this was then current practice of some of the Web companies surveyed in the second survey and also by the mature Web Company from their case study. They had a detailed list of features/function obtained from the first survey and the case study. Although some certainly seemed more complex than others they did not want to suggest or impose any complexity, leaving it for each Web company to decide. The solution was devised as follows:

- Companies contributing Web project data to Tukutuku were asked to indicate (tick) all the features/functions that the application had.
- For each feature/function they would also indicate if it was a black box reuse, reuse with adaptation, or new development.

- They were also asked to indicate if a given feature/function employed high effort to be developed / adapted.
- To be familiar with what each Company understood by high effort they also asked them to indicate the effort in person hours that would be automatically representative of high effort to develop or adapt a feature/function.
- Companies would be able to provide features/functionality that they had not considered.

The Tukutuku Benchmarking project started in 2006 and till 2006 had received 67 Web projects from 25 Web companies in 9 different countries. 27 projects came from two companies (13 and 14 respectively). Each Web project in the database provided 43 variables to characterize a Web application and its development process. Mendes et al. were aware that the data obtained was a result of a self-selected sample. However they believed the data in the Tukutuku database can be very useful as an indicator provided one is aware of the limitations. No automated measurement tools were used by the Web companies that volunteered data for the Tukutuku database. Therefore the accuracy of their data could not be determined. In order to identify guesstimates from more accurate effort data, they asked companies how their effort data was collected. They found that in at least 77.6% of Web projects in the Tukutuku database effort values were based on more than guesstimates. However, Mendes et al. are also aware that the use of timesheets does not guarantee 100% accuracy in the effort values recorded. The data collected to date for the Tukutuku project has not followed rigorous quality assurance procedures to validate the data and the projects' application domains are mixed.

3.4. **Using Multivariate Regression to Identify Early Web Size Metrics and Cost Factors**

All the variable data were analyzed using multivariate forward stepwise regression. The set of variables used for building the cost models is shown in Table 1. This is a subset of the Tukukuku data set since several variables had to be excluded if they were within the constraints of most instances of a variable being zero, the variable was categorical or the variable was related to another variable, in which case both could not be included in the same model. This was investigated using a Spearman's rank correlation analysis ($\alpha = 0.05$). Whenever variables were highly skewed they were transformed to a natural logarithmic scale to approximate a normal distribution (Maxwell, 2002). In addition, whenever a variable needed to be transformed but had zero values, the natural logarithmic transformation was applied to the variable's value after adding 1.

TABLE1: VARIABLES USED IN STEPWISE REGRESSION

Variable	Meaning
lntoteff	Natural log. Of the total effort to develop a Web application.
nlang	Number of different languages used on the project
devteam	The number of people who worked on the project
teamexp	Average team experience with the development language(s) employed
lnnewwp	Natural log. of (1+ number of new Web pages)
lnimgnew	Natural log. of (1+ number of new images in the applications)
lnimglib	Natural log. of (1+ total number of images reused from a library)
lnimg3p	Natural log. of (1+ total number of images developed by a third party)
hfotsa	Total number of adapted high effort functions.
lntoth	Natural log. of (1+ total number of high effort functions).
totnhigh	total number of low effort functions
Natural log. = Natural logarithm	

The final Mendes Early Web Cost Model is presented in Table 2 below.

TABLE 2 : BEST FITTING MODEL TO CALCULATE lntoteffor

Independent Variables	Coefficient	Std. Error	t	p>|t|	95% Confidence Interval
(constant)	2.154	0.260	8.281	0.000	1.634 – 2.674
lnnewwp	0.435	0.061	7.184	0.000	0.314 – 0.556
lntoth	0.671	0.160	4.198	0.000	0.352 – 0.991
devteam	0.239	0.083	2.876	0.005	0.073 – 0.406

The equation as read from the final model's output is:

$$\mathbf{\ln(toteffor) = 2.154 + 0.435 \times \ln(newWP+1) + 0.671 \times \ln(tothigh+1) + 0.239 \times devteam\ . \quad (1)}$$

Which, when transformed back to the raw data scale, gives the equation:

$$\mathbf{toteffor = 8.619 \times (newWP+1)0.435\times(tothigh+1)}\ x\ 0\mathbf{.671 \times \mathit{e}0.239 \times devteam \quad (2)}$$

Where, toteffor is total effort, newWP is new web page, tothigh is total high functions and devteam is size of the development team.

When there is no reuse, Webpages is the same as newWP.

Despite this cost model not presenting good estimation accuracy, its main objective is to indicate that two of the variables selected by the best fit model are a very close match to the two variables ranked highest in the first survey – number of Web pages and number of high effort features/functions. Mendes et al. believe this result to be very promising, suggesting that these metrics can be estimated by customers early in the development life cycle and are suitable for building Web cost models at, for example, the bidding stage. It is to be noted that to obtain the number of high effort functions it is necessary to provide the client with a list of high effort functions and to count those that have been selected. In addition, the development team size must be already known by the Web Company.

4. APPLIABILITY VERIFIATION SURVEY OF THE MEDES EARLY WEB COST ESTIMATION MODEL INTO THE INDIAN CONTEXT

The Indian Software industry started flourishing in the last decade of twentieth century thanks to easy availability of software personnel and outsourcing of software development work from the United States. In the first decade of the present century it emerged as a major player with entry of Multinational Corporations into Indian soil and diversification of the domestic Indian Software industrial houses into multinational operations. Today India is recognized as a Software Giant all around the world and its software exports has expanded beyond imagination. We now produce software relating to business applications and insurance, banking and financial applications, multimedia, avionics and space research, educational and instructional, e-governance and service delivery, enterprise information management and ERP solution, e-commerce and m-commerce, mobile computing and cloud computing, database management and system development etc. and a vast majority of them are web application development. They cater not only to the domestic market but also to international trade requirements. No doubt some of the Indian software companies that started from scratch about 25 years ago have now turnover exceeding billions of $. It was therefore necessary to streamline the Indian software development industry and this could be done by providing some help to them by evolving a cost estimation model based on more realistic variables than Mendes prescribed and more valid in the Indian context.

4.1. Limitations of Mendes Early Web Cost Estimation Model

Mendes et al. have only hinted at the contingency and profitability as cost variable but neglected them in their calculations. This has made their work a little bit approximate than would have been if both of them were also taken into account. It was felt necessary to investigate further into the matter to ameliorate the results obtained by them.

4.2. Verification Survey of Indian Software Projects

A survey was felt necessary to be conducted with Software projects completed by Indian Companies. In the beginning India's Top Twenty Software Companies were contacted in the year 2011 by means of formal request letters over speed post. Only two companies replied to cooperate in this survey. It was then decided to follow up with telephone and mobile over which they replied either to contact their software vendors or that they are simply not interested and started questioning the very intent of this survey. When contacted with their vendors two more vendors agreed to provide the data sets of their completed projects. Two more small local companies were also persuaded to part with such data sets such that the company size becomes homogenous. The following data were obtained with much persuasion:

TABLE3: DATA GATHERED DURING THE APPLICABILITY VERIFICATION SURVEY OF INDIAN PROJECTS.

S .N.	Project type	New WP (No.)	Tothigh (No.)	Dev Team (No.)	Toteffor (hrs-worked)	Crim (hrs-worked)	Grosseffor (hrs-worked)
1	Business app.	5	11	3	215	60	275
2	Business app.	18	61	6	2138	438	2576
3	Busin.ess app	12	16	8	1192	132	1324
4	Educational	17	22	9	2308	378	2686
5	multimedia.	28	48	10	5792	1043	6835
6	Financial	17	54	11	6308	625	6933
7	e-governance	22	59	12	9420	1188	10608
8	Banking	34	89	14	22920	3811	26731
9	Share market	47	126	15	44570	5432	50002

Where newWP, Tothigh, devteam, toteffor have usual meanings as per equation (2) and crim is contingency and risk management effort and grosseffor is gross effort of development including crim.In the table3 above projects have been arranged in the increasing order of development team size. Further data were gathered relating to development time; cost of development and profitability etc .which has been arranged in table 4 as follows:

TABLE 4: COST AND PROFITABILITY DATA

S.N.	Project Id	Hrs. per day	No. of days	Dev. Cost in INR (C1)	Crim cost in INR (C2)	Cost price CP in INR (C1+C2)	Sale price SP, in INR	Profit in INR	Profit in %
1	P1	10	9	17080	4805	21885	25000	3115	14.2
2	P2	10	43	116667	21020	137687	163699	26012	18.8
3	P3	10	17	47212	5115	52327	61540	9213	17.6
4	P4	14	21	110210	18851	129061	143379	14318	11.1
5	P5	8	83	214502	31892	246394	273710	27316	11.1
6	P6	12	58	382120	78210	460330	505000	44670	9.7
7	P7	11	80	603500	76500	680000	742125	62125	9.1
8	P8	12	140	1232808	283360	1516168	1736013	219845	14.5
9	P9	12	248	2645330	320000	2965330	3298900	333570	11.2

4.3.1. Verification

Calculation

The above data were analyzed further in the light of Mendes Early Web Cost Model as per the following:

4.3.1.1. Project P1

As per Mendes Eqn.(2),

$$toteffor = 8.619 \times (newWP+1)^{0.435} \times (tothigh+1)^{0.671} \times e^{0.239 \times devteam}$$

$$=8.619*(5+1)^{0.435}*(11+1)^{0.671}*e^{0.239*3}$$
$$=8.619*2.181*5.298*2.048$$
=203.96 hours worked.

As per observed data,

Grosseffor= 9 days of 3 persons working @10 hrs/day
=9*3*10 hrs
= 270 hrs.

Mean Payment/ hr. = CP/grosseffor
=21885/275
=89.58

Development time =C1/mean payment rate
=17080/89.58
=190.67 hrs.

For 3 persons@10 hrs./day, dev. time in No. of days

=190.67/3*10

=6.35 days

Additional time for contingency and risk management, crim,

=crim cost/mean payment rate

=4805/89.58

=53.64 hrs.

For 3 persons@10 hrs./day, dev. time in No. of days

=53.64/3*10

=1.79 days

Total deveop. time =6.35+1.79

=8.14 days

4.3.1.2. Project P4

As per Mendes Eqn.(2),

$$\text{toteffor} = 8.619 \times (\text{newWP}+1)^{0.435} \times (\text{tothigh}+1)^{0.671} \times e^{0.239 \times \text{devteam}}$$

$$=8.619*(17+1)^{0.435}*(22+1)^{0.671}*e^{0.239*9}$$

=8.619*3.516*8.198*8.593

=2134.8 hours worked.

As per observed data,

Grosseffor= 21 days of 9persons working @14 hrs/day

=21*9*14 hrs

= 2646 hrs.

Mean Payment/ hr. = CP/grosseffor

=129061/2646

=48.78

Development time =C1/mean payment rate

=110210/48.78

=2259.33 hrs.

For 9 persons@14 hrs./day, dev. time in No. of days

=2259.33/9*14

=17.93 days

Additional time for contingency and risk management, crim,

=crim cost/mean payment rate

=18851/48.78

=386.45 hrs.

For 9 persons@114 hrs./day, dev. time in No. of days

=386.45/9*14

=3.07 days

Total deveop. time =17.93+3.07

=21 days

4.3.1.3. Project P9

As per Mendes Eqn.(2),

$$toteffor = 8.619 \times (newWP+1)^{0.435} \times (tothigh+1)^{0.671} \times e^{0.239 \times devteam}$$

$$=8.619*(47+1)^{0.435}*(126+1)^{0.671}*e^{0.239*15}$$

=8.619*5.387*25.802*36.053

=43191.53 hours worked.

As per observed data,

Grosseffor= 248 days of 15 persons working @12 hrs/day

=248*15*12 hrs

= 44640 hrs.

Mean Payment/ hr. = CP/grosseffor

=2965330/44640

=66.43

Development time =C1/mean payment rate

=2645330/66.43

=39821.31 hrs.

For 15persons@12 hrs./day, dev. time in No. of days

=39821.31/15*12

=221.23 days

Additional time for contingency and risk management, crim,

=crim cost/mean payment rate

=320000/66.43

=4817.1 hrs.

For 15 persons@12 hrs./day, dev. time in No. of days

=4817.1/15*12

=26.76 days

Total develop. time =221.23+26.76

=247.9

9 days

TABLE 5: PREDICTION ACCURACY OF DATA ANALYZED

S.N.	Project Id	Prediction			Observation			Accuracy %
		Toteffor in hrs. worked	Crim in hrs. worked	Grosseffor In hrs. worked	Toteffor in hrs. worked	Crim in hrs. worked	Grosseffor In hrs. worked	
1	P1	203.96	53.64	257.6	215	60	275	6.32
2	P4	2134.80	386.45	2521.25	2306	378	2686	6.13
3	P9	43191.53	4817.1	48008.63	44570	5432	50002	3.99

In the above analysis the initial assumption that contingency and risk management is a cost variable has been proved to be valid as the errors are well within the permissible limits and are arising because of various assumptions made so as to simplify web cost effort estimation.

5. Evolving Predictor Variable Estimation Model for Web Engineering Projects

With a view to accommodate the contingency and risk management cost into the Mendes Equation it was decided to investigate further into the data observed and predicted and evolve a predictor variable estimation model of web engineering projects by using statistics and calculus.

5.1 Using Multivariate Regression to derive Predictor Variable Web Estimation Model

5.1.1 Curve Fitting by Sum of Exponentials

In the analysis of web development data we can use the fitting of a sum of exponentials of the form,

$$y = A_1e^{\lambda 1x} + A_2e^{\lambda 2x} + A_3e^{\lambda 3x} + \ldots\ldots\ldots\ldots + A_ne^{\lambda n} \qquad (3)$$

Where $A_{1,}$ $A_{2,}$ A_3 ……. A_n and $\lambda_{1,}$ $\lambda_{2,}$ $\lambda_{3,}$ …. λ_n are unknowns.

Eqn. (3) satisfies a differential equation of the type:

$$d^ny/dx^n + a_1\, d^{n-1}y/dx^{n-1} + a_2\, d^{n-2}y/dx^{n-2} + a_ny = 0 \qquad (4)$$

Where $a_{1,}$ $a_{2,}$ a_3 … a_n are unknowns.

5.1.1.1. Froberg's Method

Froberg suggested a method for computing these derivatives numerically in equation (4) at the given points and substituting them in equation (3), thus obtaining a system of n linear equations for n unknowns $a_{1,}$ $a_{2,}$ a_3 … a_n that can be solved.

Again it can be seen that $\lambda_{1,}$ $\lambda_{2,}$ λ_3 … λ_n are the roots of the polynomial equation.

$$\lambda^n + a_1\lambda^{n-1} + a_2\lambda^{n-2} + \ldots\ldots\ldots +a_n = 0 \qquad . (5)$$

Which when solved enables us to determine A_1, A_2, A_3, …. A_n, from equation (3) by method of least squares. An obvious disadvantage is with their increasing order and therefore leading to unreliable results.

5.1.1.2 Moore's Method

Equation (4) can be solved by Moore's method which is described below for n =2.

For n = 2, eqn. (3) becomes,

$$y = A_1e^{\lambda 1x} + A_2e^{\lambda 2x} \qquad (6)$$

This satisfies the differential equation,

$$d^2y/dx^2 = a_1dy/dx + a_2y \qquad (7)$$

Assuming a is the initial value of x and integrate eqn. (7) w.r.t. x we get, x

$y'(x) – y'(a) = a_1y(x) – a_1y(a) + a_2\int y(x)dx$ a . (8)

Where, $y'(x) = dy(x)/dx$. Integrating equation (8) again w.r.t. x from a to x, we will get,

$y(x) – y(a) – (x - a)y'(a) = a_1\int y(x)d(x) – a_1(x-a)y(a) + a_2\int \int y(x)d(x)d(x)$ . . (9)

Using results from calculus, we obtain,

$\int . . . \int f(x)dx . . . dx = \frac{1}{|\_(n-1)} \int (x-t)^{n-1}f(t)d(t)$ . . . . . (10)

Equation (10) simplifies to,

$y(x)-y(a)-(x-a)y'(a) = a_1 \int y(x)dx - a_1(x-a)y(a) + a_2 \int (x-t)y(t)dt$. (11)

In order to use eqn. (11) to set up a system of linear equation in terms of a_1 and a_2, $y'(a)$ need to be eliminated and this is done in following way: Choosing two data points x_1 and x_2 such that $a-x_1=x_2-a$, then from eqn.(11) we will get,

$y(x_1)-y(a)-(x_2-a)y'(a) = a_1 \int y(x)dx - a_1(x_1-a)y(a) + a_2 \int (x_1-t)y(t)dt$ (12)

$y(x_2)-y(a)-(x_2-a)y'(a) = a_1 \int y(x)dx - a_1(x_2-a)y(a) + a_2\int (x_2-t)y(t)dt$ (13)

Again simplifying the eqns. (12) and (13) by using $a-x_1=x_2-a$, we get,

$y(x_1)+y(x_2)-2y(a)=a_1[\int y(x)dx + \int y(x)dx] +a_2[\int (x_1-t)y(t)dt+\int (x_2-t)y(t)dt]$ . . (14)

Now eqn.(14) can be used to setup a system of linear equations for a_1 and a_2 and then we obtain λ_1 and λ_2 from characteristic equation.

$\lambda^2=a_1 \lambda+a_2$ . . . . (15)

Finally, A_1 and A_2 can be obtained by the method of least squares. Thus we obtain the required form of the equation as,

$y=A_1e^{\lambda 1x} +A_2e^{\lambda 2x}$ . . . (16)

5.2 Results Obtained by Using Curve fitting by Sums of Exponentials

In order to use curve fitting by sums of exponentials let us neglect the profitability issue and concentrate on total development effort and contingency and risk management effort. Thus the order of the differential equation reduces to 2 and then Moore,s Method can be applied on the data sets. By using curve fitting and solving differential equation on the observed data set we get the following equation of gross effort and current sale price of the web projects,

$$\text{grosseffort} = 102.74\,(e^{0.41*devteam} – e^{-3.34*devteam}) \quad . \quad . \quad . \quad (17)$$

and,

$$\text{C.S.P.} = 10043*\mathbf{(e^{0.41*devteam} – e^{-3.34*devteam})} \quad . \quad . \quad (18)$$

5.3 *Validating Evolving Predictor Variable Web Estimation Model*

To validate the estimation model derived using the predictor variable equation (17) and (18) for independent project P10 with following observed parameters:

<u>*Test Case 1*</u>

Project Id = P_{10}

Development team in no., (devteam) = 8,

Gross effort = 2548 hrs. worked.

CSP in INR = 246500 at the current(2014) price level.

Comparing the validation set with the results obtained from the proposed model:

Project Id = P_{10}

Development team in no., (devteam) = 8,

Predicted Gross effort = $102.74*(e^{0.41x8} – e^{-3.34x8})$ = 2732.9 hrs. worked.

CSP in INR = 266900 at the current (2014) price level.

Gross effort = $102.749*(e^{0,41*3}$-6185e-$3.34^{*3})$

= 5696*1.481-6185*1.318

= 8435.78-8151.83

= 283.95 hrs. worked.

Accuracy = 275-283.95/275

= 3.25% hence well within tolerable limits.

On an average it was found that,

The gross effort prediction accuracy is 1.7%.

The current sale price prediction accuracy is 7.6%.

As shown in graph below:

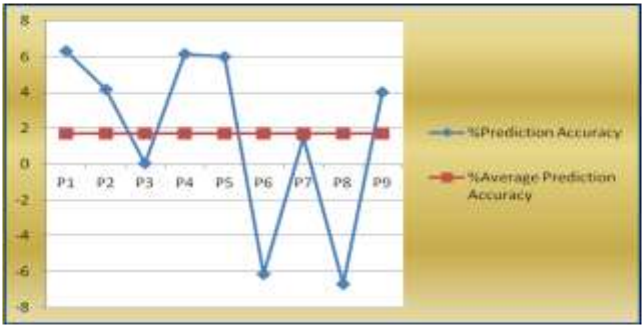

Fig.1: Line Chart Showing the variation of prediction accuracy in % of various projects as compared to the Average prediction accuracy.

6. Summaries and Conclusion

The paper highlighted the prevalent software estimation models and Mendes Early Web Cost Estimation Model in detail and the need to extend it by considering the additional issues of contingency, risk management and profitability issues so as to be more accurate and realistic. Thus by using tools like regression and calculus and it derived the Evolving Predictor Variable Estimation Model for Web Engineering Projects on the basis of the following equations:

$$\textbf{grosseffort} = \textbf{102.74}\,(e^{0.41*devteam} - e^{-3.34*devteam})$$

in hours worked.

$$\textbf{C.S.P.} = \textbf{10043}*(e^{0.41*devteam} - e^{-3.34*devteam})$$

in INR(at 2014 price).

Where devteam is the number of team members.

It would be very useful for web application development projects in early gross effort and cost prediction [9].

7. ACKNOWLEDGEMENT

The authors wish to acknowledge all the companies, software professionals, friends and faculties who generously contributed to the research work failing which this project would not have been possible.

REFERENCES

Srivastava, Sanjay Kumar & Varma, S.P. (2012). Software Projects Estimation Models, *IT Floor, Vol.1, No.2, pp.134-140,* B.R.A. Bihar University, Muzaffarpur, Bihar, PIN-842001

Cowderoy, A.J.C. (2000), Measures of Size and Complexity for web site content, *Proc. Combined 11th ESCOM conference and the 3rd SCOPE Conference on Software Product Quality, Munich, Germany, pp.423-431.*

Mendes, E., Mosley N. & Counsell, S. (2002) Comparison of Length, Complexity and functionality as Size Measures for Predicting Web Design and Authoring Effort, *IEEE Proc. Software pp. 149.3.86…92.*

Mendes, E., Mosley, N. & Counsell, S. (2007) Investigating Early Web Size Measures for Web Cost Estimation, *IEEE Proc. Software.*

Mendes, E. & Kitchenham, B.A. (2004) Further Comparison of Cross-company and within-company Effort estimation Models for Web Applications, *submitted manuscripts.*

Mendes, E. & Mosely, N. (2000) Web Metrics and Development Effort Prediction, *Proc. ACOSM, 2000, Sydney Australia.*

Mendes, E., Hall, W. & Harrison, R. (1999) Applying measurements principles to improve hypermedia authoring, *New Review of Hypermedia and Multimedia, Taylor Graham Publishers, Chapter 5, pp.105-132.*

Mendes, E., Mosely, N. & Counsell, S. (2001) Web Metrics – Estimating Design and Authoring Effort, *IEEE Multimedia, Special Issue on Web Engineering, Jan.-Mar., 2001, pp.50-57.*

Srivastava, Sanjay Kumar & Varma, S.P. (2014) Evolving Predictor Variable Estimation Model for Web Engineering Projects, *Proc. of the 1st international Conference on Recent Trends in Computer Science and Engineering, Central University of Bihar, Patna, 08-09 Feb. 2014, pp.335-346.*