

## Estimating Maintenance Cost for Web Applications

Ion IVAN, Mihai Liviu DESPA  
 Bucharest University of Economic Studies, Romania  
 ionivan@ase.ro, mihai.despa@yahoo.com

*The current paper tackles the issue of determining a method for estimating maintenance costs for web applications. The current state of research in the field of web application maintenance is summarized and leading theories and results are highlighted. The cost of web maintenance is determined by the number of man-hours invested in maintenance tasks. Web maintenance tasks are categorized into content maintenance and technical maintenance. Research is centered on analyzing technical maintenance tasks. The research hypothesis is formulated on the assumption that the number of man-hours invested in maintenance tasks can be assessed based on the web application's user interaction level, complexity and content update effort. Data regarding the costs of maintenance tasks is collected from 24 maintenance projects implemented by a web development company that tackles a wide area of web applications. Homogeneity and diversity of collected data is submitted for debate by presenting a sample of the data and depicting the overall size and comprehensive nature of the entire dataset. A set of metrics dedicated to estimating maintenance costs in web applications is defined based on conclusions formulated by analyzing the collected data and the theories and practices dominating the current state of research. Metrics are validated with regards to the initial research hypothesis. Research hypothesis are validated and conclusions are formulated on the topic of estimating the maintenance cost of web applications. The limits of the research process which represented the basis for the current paper are enunciated. Future research topics are submitted for debate.*

**Keywords:** Maintenance Cost, Web Development, Web Application Maintenance, Software Maintenance

### 1 Research Premises and Hypothesis

Maintenance costs have a consistent weight in the total costs generated by an application during its entire lifecycle. Previously conducted research and studies that focus on software maintenance support this claim. In [2], the 60/60 rule of software is depicted. The rule claims that maintenance accounts for 60% of the total costs generated by a software in its entire lifecycle. Enhancement or perfective [3] maintenance tasks account for 60% of all the costs generated by maintenance [2], whereas error correction or corrective [3] maintenance tasks account only for 17% of total maintenance costs [2]. In [4] it's stated that software applications evolve over time to meet the changing needs which leads to maintenance generating 67% of the applications total lifecycle costs. In [5] it is argued that maintenance typically accounts for 75% or more of the total software workload. Other relevant research place maintenance costs at 90% [6], 75% [7], 90% [8] or at 60% to 70%

[9] of the overall application's lifecycle cost. It is therefore of paramount importance to accurately estimate maintenance cost before implementing a software project in order to better comprehend the total costs associated with developing and owning a software application. Better understanding maintenance costs will provide valuable information regarding resource planning and the appropriate time to decommission the application.

The cost of maintenance is directly determined by the number of people involved in the maintenance process and the number of hours each person invests in the maintenance tasks. Thus in order to estimate maintenance cost one needs to estimate the effort invested in the maintenance process. The maintenance effort is expressed in man-hours which entails the number of hours dedicated by the maintenance team to implementing maintenance tasks. Maintenance in web applications has two main components: content maintenance and technical maintenance.

Content maintenance requires performing tasks of a non-technical nature like updating product inventory and information, adding news or articles, managing blog posts, updat-

ing contact or brand and visual identity information and moderating user comments or messages. Content maintenance team staffing and therefore content maintenance effort is determined by the web application's size [1].

**Table 1.** Content team staffing based on web application size [1]

Web application type	Man-Hours per year	Staffing Levels for Content Publishing and Quality Assurance
Small	1,500–4,000	1–2 people
Medium	4,000–10,000	1–2 people
Large	10,000+	From 2-3 people upwards

According to [1] and as presented in Table 1, content maintenance cost is determined by the number of hours required to keep the web application up to date in terms of content. The larger the application the more hours need to be invested in keeping it up to date in terms of content. The more content changes in an application the more technical maintenance tasks are generated. Technical maintenance tasks are generated because content managers spot bugs or request new functionality to help keep the content up to date. Therefore, content

update frequency is a factor that impacts maintenance effort and cost.

Technical maintenance requires adding new functionality, performing regular back-ups, monitoring website outages, checking IP reputation, checking DNS configuration, test website speed, check for broken links, perform software updates if third party software was used when developing the web application, monitor traffic statistics and review search engine indexing.

**Table 2.** Technical team staffing based on web application complexity [1]

Web application type	Web application complexity	Total Team Staffing:
Basic	Plain content (HTML/XHTML)	1 person (for a small to medium sized site).
Dynamic	Dynamically generated from a database	1 or 2 people (or more on a very large or busy site).
Transactional	Fully transactional content	From 1 or 2 people upwards (many more on a large or busy site).

According to [1] and as presented in Table 2, technical team staffing is determined by the level of complexity built into the web application. Thus, a Basic web application which is characterized by static content build mainly using front-end technologies like HTML, CSS and JavaScript will require no more than 1 person to perform maintenance tasks. A Dynamic web application which stores its content in a database and has functionality which is powered by PHP, Java, ASP.NET, Python,

Ruby or any other high-level programming language will require 1 to 2 persons to perform maintenance tasks. A Transactional web application which entails a content management system, an ecommerce engine and often a complex user role architecture will require 2 or more persons to perform maintenance tasks. Therefore, application complexity is a factor that impacts maintenance effort and cost.

**Table 3.** Technical team staffing based on web application estimated traffic [2]

Web application type	Page Impressions per month	Total Team Staffing:
Quiet	0–100,000	Between 1 and 3 people.
Intermediate	100,000–1,000,000	About 2-3 people.
Busy	1,000,000+	From 3 people upwards

According to [2] and as presented in Table 3, technical maintenance effort is influenced by the web application's estimated traffic. Thus, a *Quiet* web application with an estimated number of page impressions between 0 and 100.000 per month will require between 1 and 3 persons to perform maintenance tasks. An *Intermediate* web application with an estimated number of page impressions between 100.000 and 1.000.000 per month will require between 2 and 3 persons to perform maintenance tasks. A *Busy* web application with an estimated number of page impressions higher than 1.000.000 per month will require between more than 3 persons to perform maintenance tasks. The more users interact with the web application the more likely it is for them to cause outages, spot bugs or request new features. Therefore, user interaction level is a factor that impacts maintenance effort and cost. By analyzing data and information presented

in previously conducted research and studies we can conclude that maintenance costs are determined by the application's content update effort, complexity and user interaction level. Thus, the research hypothesis that supports the current paper is formulated as: a web application's maintenance cost can be accurately estimated by accounting for the application's complexity, content update effort and user interaction level.

## 2 Acquiring Data on Maintenance Cost

In order to test the research hypothesis data from real life web maintenance projects was used. All the data was collected from the same web development company. Data totals a number of 586 maintenance tasks from 24 different maintenance projects collected within a four-month timespan. A sample of the data is presented in Table 4.

**Table 4.** Maintenance tasks data sample

Task Name	Client	Project	Type	Duration (hours)	Cost (\$)
PAT Interview Questions Guide	David	PAT	perfective	3	105
Pop up box when certain email addresses sign up	David	PAT	perfective	1	35
Update Value Story completion when a Task is marked as completed on the One Page Plan	Paul	CT	perfective	4	140
SOK Registration process 2	Bjorn	SOK	perfective	2	70
Monitor is UP: saleoot	Greg	SOOT	corrective	2	70
Custom content preview plugin	Erik	CNGHM	corrective	3	105
Cancel profile Louise Robinson/Taylor	Alex	TLT	perfective	0.5	17
Add Houslow as the location of the job as one of the sub locations for London	Alex	TLT	perfective	1	35
Exhibitors list-SOK2016	Bjorn	SOK	perfective	1	35

Order form	Karl	MLTK	corrective	9	315
Saleoot feed changes	Greg	SOOT	adaptive	0.5	17
Infospace one off task	Greg	OSN	perfective	3	105
Medical malpractice guide template	Erik	CNGHM	perfective	6	210
Sabres - add checkbox to products that appear on homepage	Sely	SBRS	perfective	2	70

Out of the total 586 tasks, 8% are adaptive maintenance tasks, 10 are corrective maintenance tasks, 76% are perfective maintenance tasks and 6% are preventive maintenance tasks. The vast majority, 76%, are tasks that are generated by requests of enhancing current functionality or develop new functionality. The projects were developed for clients from United Kingdom, United States, Romania, Norway, Australia and Germany. The column Task Name displays data that represents the code used to identify each maintenance task when exchanging information with the client. Task Name is usually generated based on the subject of the email sent by the client, containing a specific maintenance request. Duration column displays data that represents the time

frame need to implement a particular maintenance task. Cost column displays data that represents the amount billed to the client for implementing a particular maintenance task. Different clients are billed at different rates depending on the complexity of the web application and on the team member assigned to implement the task.

Table 5 presents data that has a relevant impact on maintenance effort and subsequently on maintenance cost. Data was collected for the 24 analysed web applications over a time span of 4 months. As identified in the research premises and hypothesis state, the factors that have a significant impact on maintenance costs are the application’s user interaction level, content update frequency and application complexity.

**Table 5.** Data influencing maintenance effort

Task Name	Maintenance Time (hours)	User Interaction Level (minutes)	Content Update Frequency (hours)	Application Complexity (hours)
AVZ	25	2.6	160	440
BRK	22.5	4.5	140	250
SRBS	5	3.2	30	60
BD	19	3.4	128	210
CLBT	3	2	24	40
CNGHM	85	7.2	225	520
DQS	22	2.2	148	235
DSF	55	5.2	340	340
GTW	12.5	1.3	80	160
HRGVS	9	0.9	50	120
MC	26.5	2.8	140	240
MLTK	10	1.2	80	130
OSN	69.5	6.3	378	440
PARTS	17	2	135	265
PAT	94.5	5.3	532	630
RMG	4.5	0.6	36	75
SOOT	10	1.6	88	130

SOK	55.5	2	336	370
TC	74.5	1.4	424	490
TLT	41.5	6.3	310	310
TRCT	4	1	36	40
WWS	31	4.9	208	260
YWA	2	0.4	18	30
CT	127	5.2	648	1140

*Maintenance Time* column displays data regarding the total time allocated, within the 4 months analysed timespan, to implement maintenance tasks for a particular project. Data displayed in the *Maintenance time* column was collected automatically as time spent on implementing each maintenance tasks was tracked by each developer using a time tracking application. *User Interaction Level* column displays data regarding the average time spent by a user interacting with the application within the 4 months analysed timespan. Data displayed in the *User Interaction Level* column was automatically collected using Google Analytics statistics. *Content Update Frequency* column displays data regarding the number of hours spent within the 4 months analysed timespan for updating or adding content on a particular application. Data displayed in the *Content Update Frequency* column was manually collected. Column *Application Complexity* displays data regarding the number of hours invested in developing a particular application. Data displayed in the *Application Complexity* column was manually collected.

### 3 Defining Maintenance Costs Metrics

The software industry relies excessively on expert estimations when it comes to providing estimates related to software development costs. Previously published research states that expert estimation based on previous experience is the basis for 72% to 86% of all estimations related to software development effort [10]. Though there are metrics and models designed for estimating software maintenance costs, they are seldom used as they did not manage to convince the software community of their effectiveness. According to [11] the most popular and widely used cost estimating models are:

- COCOMO uses a size input measured in terms of application points and a productivity to calculate effort; it accounts for person-months; a multiplicative constant, an estimated size of the software application, an exponential constant and specifies effort multipliers [11];
- KnowledgePlan is a commercial software estimation tool developed and first released by Software Productivity Research [11];
- PRICE-S is an activity based estimation model that estimates the effort required for each activity; it accounts for software size, baseline productivity of an industry or an application domain and productivity factor [11];
- SEER-SEM uses the Rayleigh probability distribution of staffing profile versus time to determine development effort [11];
- SLIM also assumes that the staffing profile follows a form of Rayleigh probability distribution of project staff buildup over time [11].

The most popular maintenance cost estimating models use different structures for estimating effort but it is common to employ an exponential type structure as depicted in the below template cost estimation model.

$$Effort = A * C * D^{Exp} \quad (1)$$

where:

*Effort* - the effort required to implement tasks;  
*A* – size or complexity of the application being analyzed;

*D* – effort determining variable;

*Exp* – effort exponential multiplier.

Previously published maintenance cost estimating models overwhelmingly rely on estimating the effort invested into the maintenance process and then converting that effort into actual costs. SLIM and SEER-SEM estimate effort by determining the development team size and evolution over time. PRICE-S estimates effort by breaking down the maintenance effort into activities and accounting for application complexity and development team productivity. COCOMO uses an estimated size of the software application collaborated with a system of multiplicative constants that are empirically determined in order to estimate effort.

The cost estimating model developed within the research process follows a similar pattern to that presented in the template cost estimation model however, previously developed cost estimating models do not account for characteristics that are proprietary to web applications. Such characteristics are user interaction and content update procedures. For web applications the user traffic and therefore the user interaction level are very important as they determine the strain generated on the hosting environment. Also in web application content is updated and made available for all users. These characteristics have a considerable impact on the maintenance process and influence costs.

In the cost estimating model proposed in the current paper, the cost of monthly maintenance is estimated using the CMM indicator which is defined as:

$$\text{CMM} = \text{CMTM} + \text{CMCM} \quad (2)$$

where:

CMTM – effort of monthly technical maintenance; expressed in hours;

CMCM – effort of monthly content maintenance; expressed in hours.

Cost of monthly content maintenance is determined by estimating the volume of content needed to be uploaded on the website and the time needed to upload it. A basic estimation can be performed using data from Table 1.

Thus, in order to determine the CMCM value templates can be used in conjunction with estimated volume of content to be uploaded.

Estimating effort of monthly technical maintenance and therefore calculating CMTM is considerably more difficult and constitutes the main focus of the current paper. As identified in the state of the research stage, technical maintenance is influenced by:

- user interaction level which entails that the more users interact with the application the higher the probability of discovering bugs or faults within the application; once a bug is identified and reported it generates a maintenance tasks; also, the more users interact with the application the higher the strain on the hosting resources thus increasing the likelihood of generating adaptive and preventive maintenance task;
- content update frequency which entails that the higher the frequency of changing or adding content on the site the higher the probability of discovering bugs or faults or generating perfective task requirements; by adding various content, multiple use case scenarios are employed thus increasing the likelihood of covering scenarios that were not account for in the development process;
- application complexity which entails that the more complex an application the more maintenance work will be required; any enhancement or upgrade within a complex application can impact multiple components thus increasing the effort for properly implementing the change; also, complex application are likely to require additional preventive maintenance operations.

An important factor when considering a maintenance cost and effort estimating metric is weather the maintenance team is the same as development team. Most maintenance tasks are very similar or have considerable redundancy however when a maintenance team must work with a code that has been developed by a different team an additional maintenance tasks is generated. This tasks is often

the most difficult and consists of understanding the exiting product from a functional point of view and from a technical point of view. According to [2] this task is the dominant maintenance activity, consuming roughly 30% of maintenance time. In this regard one could claim that the maintenance process is more difficult than development itself.

Compiling all the gathered research evidence a maintenance indicator for estimating technical maintenance needs to account for user interaction level, content update frequency, application complexity and whether maintenance will be performed by the same team that developed the application or not. Thus, the effort of monthly technical maintenance is estimated using the CMTM indicator which is defined as:

$$CMTM = T_m * (0.85 U_i + 0.09 C_{uf} + 0.05 A_c) \tag{3}$$

where:

- U<sub>i</sub> – the level of user interaction; variable expressed in hours;
- C<sub>uf</sub> – content update effort; variable expressed in hours;
- A<sub>c</sub> – application complexity; variable expressed in hours
- T<sub>m</sub>- team coefficient; binary variable.

$$T_m = \begin{cases} 1, & \text{if maintenance is performed by the same team that developed the application} \\ 1,3, & \text{if maintenance is performed by a diffren team than the one that developed the application} \end{cases}$$

By performing a dimensional analysis on the CMTM we can determine that it is an indicator expressed in hours. Thus, it is used to estimate the number of monthly maintenance hours required for a web application by accounting for level of user interaction, content update effort application complexity and whether the application is being maintained or not by the same team the developed it. Thus CMTM is an effort indicator and in order to obtain an estimate of the actual cost one just needs to factor in the average man-hour cost that will be used in the maintenance process. The coefficients for the CMTM indicator were determined by employing a statistical method known as ordinary least squares or OLS. The model was applied on a number of 20 observation which consist of maintenance data collected from the 24 web application analysed within the research process. Data from 4 web application was not used in the statistical analysis as it was saved in order to validate the CMTM indicator.

Regression Statistics								
Multiple R	0.976251113							
R Square	0.953066236							
Adjusted R Square	0.888721087							
Standard Error	10.21861205							
Observations	20							
ANOVA								
	df	SS	MS	F	Significance F			
Regression	3	36047.10945	12015.70315	115.0708622	4.84932E-11			
Residual	17	1775.140549	104.4200323					
Total	20	37822.25						
	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
Intercept	0	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
User Interaction Level	0.8567114	1.362479976	0.62878825	0.537845585	-2.017870078	3.731292879	-2.017870078	3.731292879
Content Update Frequency	0.099903134	0.033443836	2.987191253	0.008277607	0.029342808	0.170463459	0.029342808	0.170463459
Application Complexity	0.05151646	0.028634905	1.799079093	0.089782896	-0.00889791	0.111930829	-0.00889791	0.111930829

Fig. 1. OLS for determining CMTM coefficients

The CMTM indicator was designed for estimating not for calculating and should be used in order to assemble an overview regarding

the effort need to properly implement the maintenance process.

#### 4 Validating Maintenance Cost Metrics

In order to validate the CMTM indicator, maintenance data from the 4 of the analyzed web applications was used. The 4 web applications were not used in the statistical analysis performed to determine the model on which CMTM was developed. Data from the 4 web applications was collected regarding actual time invested in maintenance tasks, level of user interaction, content update effort application complexity and whether the application is

being maintained or not by the same team the developed it. Estimated monthly maintenance effort, determined using the CMTM indicator was compared with actual maintenance effort. Data in Table 6 depicts level of user interaction, content update effort application complexity and whether the application is being maintained or not by the same team the developed it, for each of the 4 analyzed web applications.

**Table 6.** CMTM indicator validation data

Web application	Maintenance time (hours)	User Interaction Level (minutes)	Content Update Frequency (hours)	Application Complexity (hours)	Maintenance Team (binary)
TRCT	4	0.3	16	40	0
WWS	31	2.9	188	140	0
YWA	2	0.3	4	30	1
CT	127	5.2	648	1140	1

*Maintenance Time* column displays data regarding the total time allocated, within the 4 months analysed timespan, to implement maintenance tasks for a particular web application. *User Interaction Level* column displays data regarding the average time spent by a user interacting with the application within the 4 months analysed timespan. *Content Update Frequency* column displays data regarding the number of hours spent within the 4 months analysed timespan for updating or adding content on a particular application. Column *Application Complexity* displays data

regarding the number of hours invested in developing a particular application. Column *Maintenance team* displays data regarding whether the application was maintained or not by the same team the developed it. If the application was developed by the same team that is in charge of maintenance the value in column ‘Maintenance team’ is 1 otherwise is 0. Two of the application used for validating the indicator are maintained by the same team that developed them and 2 are maintained by a different team.

**Table 7.** CMTM indicator error estimation

Web application	Actual maintenance time (hours)	Estimated maintenance time (hours)	Estimation error (%)
TRCT	4	4.7	17
WWS	31	34	9
YWA	2	2.1	5
CT	127	119.7	-5

Column *Actual maintenance time* displays the actual time that was allocate to maintenance tasks within the 4 months analysed timespan for each web application. Column *Estimated maintenance time* displays the time estimated

for maintenance for each web application using the CMTM indicator. Column *Estimation Error* displays the relative difference between estimated maintenance effort and actual maintenance effort. As depicted in Table 7 for



application TRCT the estimation error of the CMTM indicator on determining the maintenance effort was 17%. For application WWS the estimation error was 9%, for YWA the estimation error was 5% and for CT application the estimation error was -5%. As the difference between estimated maintenance effort and actual maintenance effort relatively small, the CMTM indicator is validated.

As the CMTM indicator is validated also the research hypothesis, stating that maintenance cost can be accurately estimated by accounting for the application's complexity, content update effort and user interaction level, is validated. In addition to the initial hypothesis maintenance cost has been proven to be heavily influenced by the variable determining whether the web application is being maintained by the same team that developed it or by a different team.

## 5 Conclusions

Maintenance costs represent a consistent percentage of the total cost generated by a software application during its entire lifecycle. Previously published research estimate maintenance costs to be between 60% and 90% of total costs generated by a software application during its entire lifecycle. Thus, it is very important to accurately estimate maintenance costs even before starting development on a software application. As with any software development process, the cost of maintenance is directly determined by number of man-hours invested into the process. Web application maintenance is comprised of technical maintenance and content maintenance. Content maintenance is easier to manage and estimate in terms of effort and cost. Cost of monthly content maintenance is determined by estimating the volume of content needed to be uploaded on the website and the time needed to upload it. The challenge is to estimate the effort required for technical maintenance. Technical maintenance is determined by the web application's level of user interaction, content update effort application complexity and whether the application is being maintained or not by the same team the developed it. CMTM indicator is built on a model

that accounts for the above mentioned determining factors and represent a valuable tool in estimating the effort required for monthly maintenance. The paper's research hypothesis is validated thus proving that maintenance cost can be accurately estimated by accounting for the application's complexity, content update effort and user interaction level. The research results have a limited degree of generality as maintenance data was collect from a relatively small number of software development projects, respectively 24. Also, the results are limited to the web development field and have not been tested for other types of software applications. Future research might undertake the effort to retest the research hypothesis and to refine the CMTM indicator by employing a larger set of maintenance data. Also future maintenance should look into automating some of the maintenance tasks by using dedicated software tools.

## References

- [1] S. Diffily, *The website manager's handbook*. Lulu.com, 2006, ISBN: 978-1-4116-8529-1
- [2] R. L. Glass, "Frequently forgotten fundamental facts about software engineering.", *IEEE software*, vol. 18, no. 3, pp.112-111, 2001.
- [3] B. P. Lientz, E. B. Swanson and G. E. Tompkins, "Characteristics of application software maintenance", *Communications of the ACM*, vol. 21, no. 6, pp. 466-471, 1978.
- [4] S. R. Schach, *Software Engineering*, Boston, McGraw-Hill, 1999, pp. 11.
- [5] D. D. Galorath, "Software total ownership costs: development is only job one.", *Softw. Technol. News* no.11, 2008.
- [6] L. Erlikh, "Leveraging legacy system dollars for e-business.", *IT professional*, vol. 2, no. 3 pp. 17-23, 2000.
- [7] A. Eastwood, "Firm fires shots at legacy systems", *Computing Canada*, vol.19, no. 2, pp. 17, 1993.
- [8] J. Moad, "Maintaining the competitive edge", *Datamation*, vol. 36, no.4 pp. 61, 1990.

- [9] S. Huff, "Information systems maintenance", *The Business Quarterly*, no. 55, pp. 30-32, 1990.
- [10] M. Jørgensen, "A review of studies on expert estimation of software development effort.", *Journal of Systems and Software*, vol. 70, no. 1, pp. 37-60, 2004.
- [11] V. Nguyen, Improved size and effort estimation models for software maintenance, Doctoral dissertation, University of Southern California, 2010.



**Ion IVAN** has graduated the Faculty of Economic Computation and Economic Cybernetics in 1970. He holds a PhD diploma in Economics from 1978 and he had gone through all didactic positions since 1970 when he joined the staff of the Bucharest Academy of Economic Studies, teaching assistant in 1970, senior lecturer in 1978, assistant professor in 1991 and full professor in 1993. Currently he is full Professor of Economic Informatics within the Department of Computer Science in Economics at Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies. He is the author of more than 25 books and over 75 journal articles in the field of software quality management, software metrics and informatics audit. His work focuses on the analysis of quality of software applications.



**Mihai Liviu DESPA** has graduated the Faculty of Cybernetics, Statistics and Economic Informatics from the Bucharest Academy of Economic Studies in 2008. He has graduated a Master's Program in Project Management at the Faculty of Management from the Bucharest Academy of Economic Studies in 2010. He has a PHD in managing innovation oriented software development projects obtained in 2015 at the Economic Informatics PHD School and he is currently Project Manager at GDM Webmedia SRL. His main field of interest is project management for software development