Tying Web Project Estimation with Risk Management

Pradnya B. Rane[1], Suchita Patil[2], Pallavi Kulkarni[3] and B.B.Meshram[4]

**[1] Computer Department, Veermata Jijabai Technological Institute,
Mumbai, Maharashtra, India**
pradnyarane@gmail.com

**[2] Computer Department, Veermata Jijabai Technological Institute,
Mumbai, Maharashtra, India**
suchitapatil26@gmail.com

**[3] Computer Department, Veermata Jijabai Technological Institute,
Mumbai, Maharashtra, India**
pskulkarni77@gmail.com

**[4] Computer Department, Veermata Jijabai Technological Institute,
Mumbai, Maharashtra, India**
bbmeshram@vjti.org.in

Abstract

Web project management is about using web services that are geographically distributed, where project managers and team members are located in different cities, states and countries. It plays an important role in any organization for saving money and utilizing time efficiently. The project goals must be followed in terms of costs, time, scope, risk, effort and quality. Also, Modern software development, with an emphasis on web and distributed development, with limited face-to-face interaction among team members presents specific challenges and risk areas to the software industry which need to be considered and managed. On this background, this paper focuses on tying different web project estimation methods, with different risks involved and risk resolution techniques into an integrative framework for managing risks in distributed contexts. We can also combine these methods to minimize the impact of risk involved.

Keywords: *Agile software process, metrics, risk, estimation*.

1. Introduction

Today's complex environments require ongoing implementation. Web Project management provides a framework for working amidst persistent change. Manage the work involved in a project in terms of competing demands for scope, time, cost, risk, and quality, Stakeholders with different needs and expectations etc. As explained by Khaled El Emam, A.Gunes Koru[1], Project cancellations waste corporate resources and they are often difficult to deal with because they require special management skills and critical business decisions. Project cancellations aren't always a bad thing. Cancelled projects could lead to substantial learning or produce artifacts applicable to future projects. We also need to consider that project cancellation and performance might depend on size. Smaller projects tend to have lower cancellation rates, and of projects that deliver, smaller projects tend to perform better in terms of quality, being on budget, and being on schedule. Some of the project cancellation reasons are given below:

1. Failure to align project with organizational objectives
2. Poor scope
3. Unrealistic expectations
4. Lack of executive sponsorship
5. Lack of project management
6. Inability to move beyond individual and personality conflicts
7. Politics

On the basis of these background, web project management plays an important role. The paper starts with introduction; gives information about some related work on software metrics, web measures and risk management in the section II. Section III discusses various estimation methods, limitations of estimation and challenges. The paper concludes with section IV.

2. Related Work

Several techniques for cost and effort estimation have been proposed falling into three general categories[16]:

1) Expert judgment (EJ): Expert judgment has been widely used and can be an effective estimating tool on its own or as an adjusting factor for algorithmic models. When applying EJ, the means of deriving an estimate are not explicit and therefore difficult to repeat[20].

2) Algorithmic models (AM) : Algorithmic models represent the relationship between effort and one or more project characteristics. The main "cost driver" used in such a model is usually some notion of size (e.g. the number of lines of source code, number of pages, number of links). AM need calibration or to be adjusted to local circumstances. Example of AM is the COCOMO model[19].

3) Machine learning (ML): Machine learning techniques have in recent years been used as a complement or an alternative to the previous two categories. Examples include fuzzy logic models, regression trees, neural networks, and case based reasoning[19].

Hence the web project management includes software estimation along with risk management which can be explained as follows.

2.1 Software metrics

As explained by Rob Pooley, Dave Senior and Duncan Christie [4], A software metric can define a standard way of measuring some attribute of the software development process, such as size, cost, defects, communications, difficulty, or environment. Typically, the attributes measured fall into the following core categories:

- Cost (staff effort, phase effort, total effort);
- Errors (date found or corrected, effort required, error source and class);
- Process characteristics (development language, process model, technology);
- Project dynamics (changes or growth in requirements or code);
- Project characteristics (development dates, project size, total effort);
- Software structure (size, complexity).

2.2 Web size measures

Software practitioners recognise the importance of realistic estimates of effort for the successful management of software projects, the Web being no exception. Prediction is a necessary part of an effective process , be it authoring, design, testing or Web development as a whole.

Web objects, computes size by considering each of the many elements that comprise the Web application. The metric computes size using Halstead's equation for volume (that is, a proposed measure of size that is language independent and related to the vocabulary used to describe it in terms of operands and operators) given in (1) as follows[15]:

$$V^* = N \log2(n) = (N1^* + N2^*) \log2 (n1^* + n2^*) \qquad (1)$$

Where,
N = number of total occurrences of operands and operators
n = number of distinct operands and operators
N1* = total occurrences of operand estimator
N2* = total occurrences of operator
n2* = number of unique operators estimators
n1* = number of unique operands estimator
V* = volume of work involved represented as Web Objects.

As explained by Emilia Mendes [2], estimation can be done for entities involved in the estimation, length size metrics, reusability metrics, complexity metrics, effort metrics, functionality metrics etc.

2.3 Risk management

Software Engineering Institute (SEI) defines risk as "the possibility of suffering loss" and it defines loss in a development project, as " the impact to the project which could be in the form of diminished quality of the end product, increased costs, delayed completion, loss of market share, or failure." For each risk there are two aspects: risk probability and risk loss. As explained by John Stouby Persson these aspects are used to estimate the impact or Risk Exposure (RE) as given in (2)[9]:

$$RE=P(UO).L(UO) \qquad (2)$$

Where,
-RE is the Risk Exposure (or risk impact)
-P(UO) is the probability of an unsatisfactory outcome
-L(UO) is the loss associated to unsatisfactory outcome.

Risk probability estimation is not a straightforward task and can not be 100% accurate (as otherwise there is no risk).Risk resolution techniques are divided into four categories: planning, control, social integration, and technical integration as explained in [9].Finally we can tie Risk Assessment to Resource Estimation as explained by Scott E. [12]where we can assess the risk of accomplishing the work to be done, allocating appropriate resources to mitigate the identified or anticipated risks, monitoring the risks throughout the project, and deciding how to deal with the risks.

Shan Liu[14] explained the use of balanced score card for evaluating and mitigating information systems development risk. Risks in the systems development process are identified and evaluated through BSC. Risk mitigation strategies are also put forward to transform the risk into strategic execution. BSC can be a tool for reducing the IS development risks and improving development performance while guarantee the realization of target. On one hand, BSC can balance the short time target (risk management) and long time target (strategy). On the other hand, it can also balance the expected result and real executions.

2.4 Tying risk assessment to resource estimation

If the organization is to produce products that the customer wants and can use, individuals from three groups of related disciplines management, development, and product assurance should be involved in its software systems engineering projects. Product assurance is the integrated application of four disciplines: quality assurance, verification and validation, test and evaluation, and configuration management. Product assurance provides management with visibility and traceability into the development process, thereby helping to reduce risks.
Successful software systems engineering project execution thus involves

- Assessing the risk of accomplishing the work to be done,
- Allocating appropriate resources to mitigate the identified or anticipated risks,
- Monitoring the risks throughout the project, and
- Deciding how to deal with the risks[12].

Developing a project plan includes assessing the risk of accomplishing the work. In this article, author refer to the work to be done as a statement of work (SOW) or a statement of objectives (SOO) such as found in a request for proposal(RFP), which specifies the work to be accomplished. There is a five-step risk-assessment process:
1. Decide the number of risk levels.
2. For each risk level, define risk criteria.
3. For each risk level, define the number of matches required to assign the project to the risk level.
4. If the matches are insufficient to assign a risk level, define the default risk level.
5. For each risk level, decide on recommended resource allocation percentages for management, development, and product assurance.
In the Figure 1 we can see an example set of risk criteria for the three tiers such as high risk, medium risk and low risk and the resources are are applied on the basis of risk levels as explained in [12].Organization will have its own set of tiers and corresponding risk criteria. It’s important to establish a database of risk criteria for your projects.

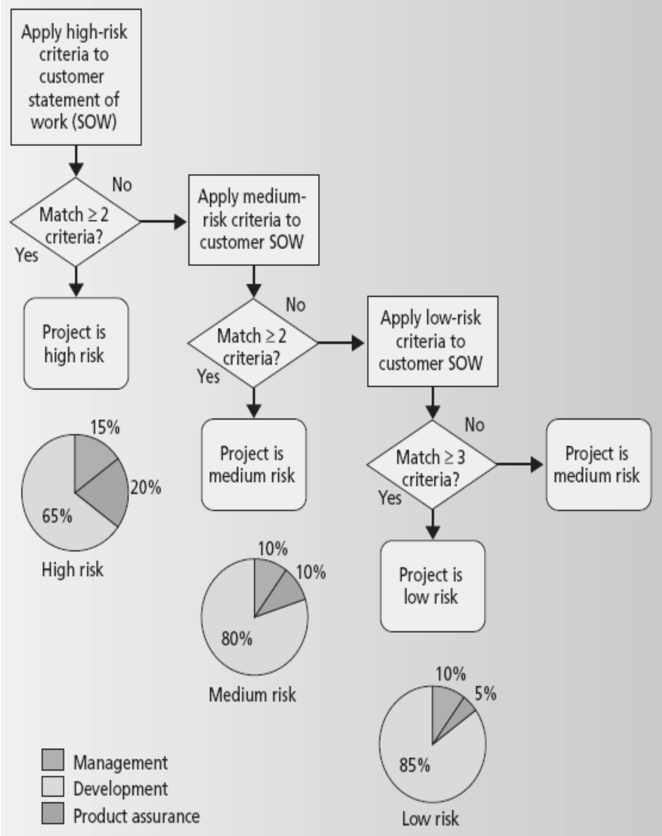

Figure 1. Logic for applying risk criteria to a statement of work (SOW)[12].

3. Discussion

Estimation is a crucial element of software project planning. Unfortunately, there are inherent limitations in the ability to estimate projects accurately due to the inherent uncertainties in software projects.

3.1 Estimation methods

1) Agile Software Process (ASP): As explained by Mikio Aoya[3] ASP aims to develop software quickly while maintaining the flexibility needed to respond to changing requirements. Agility in software development means not only quick delivery of software products but also quick adaptation to changing requirements. To be agile, the process must be flexible enough to adapt smoothly to changes in requirements and delivery schedule. ASP alters traditional management principles as follows:
-the process architecture shifts from monolithic to modular, and
- the process dynamics shift from volume-based to time-based.

Requirements management, one of the most collaboration-intensive activities in software development, presents significant difficulties when stakeholders are distributed, as in today's global projects. Communicating and managing requirements in a distributed setting was one of the concerns the practitioners expressed most often.

2) Bayesian Network Models for Web Effort Prediction: A cornerstone of Web project management is sound effort estimation, the process by which effort is predicted and used to determine costs and allocate resources effectively, enabling projects to be delivered on time and within budget as explained in Emilia Mendes[5]. A BN is a model which supports reasoning with uncertainty due to the way in which it incorporates existing complex domain knowledge.

3) Analogy-Based Cost Estimation: Cost estimation is a vital task in most important software project decisions such as resource allocation and bidding. Analogy-based cost estimation is particularly transparent, as it relies on historical information from similar past projects, whereby similarities are determined by comparing the projects' key attributes and features However, one crucial aspect of the analogy-based method is not yet fully accounted for: the different impact or weighting of a project's various features [10].

4) Use – Case Based Project Estimation: Iterative development offers several opportunities to apply a simple use-case-based measure of project performance. Each iteration's structure follows the familiar waterfall disciplines of requirements, analysis, design, code, and test. With the project decomposed into iterations, it is much easier to determine progress and tolerate change[13].

A use case records a functional requirement in the form of a dialogue between the user and the application. Each use case focuses on the business, covers functional requirements, is easily read by the business user (because it is written in business terms), and has a generally accepted format. The use-case-based estimation model aims to capture the experience of the specialists doing the estimation, include disparate stakeholders in the decision making, and enable a consensus on the agreed timescale. This method not only increases the accuracy of measurement but also produces a sense of shared ownership of the model results.

5) Measuring the model's predictive power: After obtaining the prediction models, we can assess each model's accuracy by measuring their prediction power. A small MMRE indicates a good prediction model.

To do this, we can use the Mean Magnitude of Relative Error (MMRE),which is given in (3):

$$\frac{\sum_{i=1}^{i=n}\left(\frac{|\mathrm{Epred}_i - \mathrm{Eact}_i|}{\mathrm{Eact}_i}\right)}{n} * 100 \qquad (3)$$

Where –

Eact is the actual effort, and Epred is the estimated effort.

The BRE (Balanced Relative Error) is, as its name indicates, a more balanced measure than MRE. It is calculated from (4):

$$BRE = \frac{|x-y|}{\min(x,y)}, \quad \text{x = actual and y = estimate.} \qquad (4)$$

6) Measuring productivity: PRODUCTIVITY is the amount of output (what is produced) per unit of input used. In general, productivity is difficult to measure because outputs and inputs are typically quite diverse and are often themselves difficult to measure. If we can measure the size of the software product and the effort required to develop the product, we have:[11]

$$Productivity = Size/Effort. \qquad (5)$$

Equation (5) assumes that size is the output of the software production process and effort is the input to the process. Equation (5) is simple to operationalize if we have a single dominant size measure. For example, product size measured in lines of code.

A strong understanding of software productivity, coupled with a good estimate of software size, is key to predicting project effort and, ultimately, producing reliable project duration estimates, schedules, and resource needs. Project managers and engineers often measure or predict the size of released software—the volume of software in the marketed product. However, the final release doesn't include reworked code—code that was changed or deleted during development. Hence, a simple method measuring new effective lines of code is explained by Edmund[22].

3.2 Limitations of estimation

Estimation is a crucial element of software project planning. Unfortunately, there are inherent limitations in the ability to estimate projects accurately due to the inherent uncertainties in software projects. In software, we primarily want to estimate three aspects of a project: effort, schedule, and cost. Most methods start by estimating size—as thousands of lines of code (KLOC), function points—or some other proxy point and using that to estimate effort (that is, staff months).

The literature[11] lists several common causes for overruns:

- Specifying incomplete or unclear requirements (not knowing what to do),

- Failing to adjust schedules when scope changes (too much work),
- Setting overly aggressive development schedules (too little time), and
- Insufficient resources (not enough people or equipment)

These reasons are generic. Some more reasons are also explained by Linda[11].

3.3 Challenges

1) Distributed development challenges: Developing software across distributed sites presents many challenges which are summarized in the following points[21]:

- Inadequate informal communications
- Lack of trust
- Culture differences (e.g. different language, different corporate culture and developers' background)
- Time-zone difference (leading to ineffective synchronous communication)
- Development process differences
- Knowledge management challenges (most of the existing management approaches are designed for co-located teams).
- Technical issue: Incompatible data formats and exchanges.
- Security issue (Ensuring electronic transmissions confidentiality and privacy).

2) Web development challenges: Web development need to be considered carefully as well as the significance of associated challenges and risks. Ideally, assessment and management of web development risks should be performed during the whole life cycle of the projects.

The importance of web risks is different from others in a number of ways[21]:

- Their impact and significance are different. For example the exposure to security threats is higher in the web.
- As web applications may be deployed instantly worldwide their risks can affect wider range of components and applications simultaneously in very short period of time.
- Additional risk sources include communication, culture, diversity and difference in geographical locations .
- Estimation of risk probability and loss is more difficult because of the involved challenges and relative lack of experience with them.

4. Conclusions

Project estimates gives the opportunity to adjust project parameters to meet budgets and deadlines. The attributes of Web applications corresponding to three size categories, namely length, complexity and functionality. Try estimating previous (completed) projects to validate and tune the methods. Accuracy is correlated with training and the ability to see results, not development experience. There's an urgent need for adequate, early stage effort prediction for Web development. As the use of the Web as a resource delivery environment increases, effort estimation can contribute significantly to the reduction of costs and time involved in developing web applications.

References

[1] Khaled El Emam, A.Gunes Koru,"Replicated Servey of IT Software Proiect Failures" IEEE/September-October 2008.

[2] Emilia Mendes University of Auckland,Nile Mosle Auckland University of Technology Steve Counsell Birkbeck College, University of London "Web Metrics Estimating Design and Authoring Effort" IEEE/2001 .

[3] Mikio Aoyama, Niigata Institute of Technology," Web-Based Agile Software Developement", IEEE/November/ December 1998.

[4] Rob Pooley,Dave Senior and Duncan Christie, "Collectin and Analyzing Web-Based Project Metrics" IEEE/Jan-Feb 2002 .

[5] Emilia Mendes and Nile Mosley," Bayesian Network Models for Web Effort Prediction: A Comparative Study", IEEE TRANSACTIONS , VOL. 34, NO. 6, NOVEMBER/DECEMBER 2008.

[6] E. Mendes, N. Mosley and S. Counsell , "Comparison of Web size measures for predicting Web design and authoring effort", IEEE,Vol. 149, No. 3, June 2002.

[7] Barbara Kitchenham and Emilia Mendes, "Software Productivity Measurement Using Multiple Size Measures", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 30, NO. 12, DECEMBER 2004.

[8] Vibha Sinha and Bikram Sengupta, Satish Chandra, "Enabling Collaboration in Distributed Requirements Management", IEEE/2006.

[9] John Stouby Persson,Lars Matheassen,Jesper Boeg, Thomas Stenskrog, Flemming Steinson "Managing Risks in Distributed software Projects: An Intergrative Framework "IEEE/ August 2009.

[10] Martin Auer,Adam Trendowicz,Bernhard Graser, Ern Haunschmid, stefan biffl, "Optimal Project Feature Weights in Analogy- Based Cost Estimation Improvement and Limitations" IEEE/FEBRUARY 2006.

[11] Linda M. Laird, "Limitations of Estimation" IEEE/Nov-Dec 2006.

[12] Scott E. Donaldson and Stanley G. Siegel," Enriching Your Project Planning: Tying Risk Assessment to Resource Estimation",IEEE/2007.

[13] Ray Ashman, “Project Estimation: A Simple Use- Case-based Model” IEEE/July- August 2004 .

[14] Shan Liu, Jinlong Zhang, Yuqing Liu, Tao Chen “Evaluating and Mitigating Information Systems Development Risk through Balanced Score Card”, IEEE/2009.

[15] Donald J. Reifer,” Web Development:Estimating Quick-to-Market Software”,IEEE/2000.

[16] Majid Javid Moayed, Abdul Azim Abdul Ghani, Mojtaba

Seyedzadegan,” Comparing Between Web Application Effort Estimation Methods”.IEEE.

[17] Barry W. Boehm,” Achievements and Challenges in Cocomo-Based Software Resource Estimation”,IEEE/2008.

[18] Magne Jørgensen, “Practical Guidelines forExpert-Judgment-Based Software Effort Estimation”,IEEE/2005.

[19] Mendes, E., Mosely N., Counsell, S.,”A Comparision of Development Effort Estimation Techniques for Web Hypermedia

Applications”,IEEE Symposium on software Metrics, 2002.

[20] R. Gray, S. G. MacDonell, and M. J. Shepperd, "Factors

Systematically associated with errors in subjective estimates of software development effort: the stability of expert judgement", IEEE 6th International Metrics Symposium, Boca Raton, November 5-6, 1999.

[21] Ayad Ali Keshlaf,” Risk Management for Web and Distributed Software Development Projects”, The Fifth International Conference on Internet Monitoring and Protection,IEEE.

[22] Edmund P. Morozoff, ”Using a line-of-code metric to understand software rework”, IEEE/2010.

Pradnya B. Rane is a post graduate student in computer department in VJTI, Matunga, Mumbai.

Suchita Patil is a post graduate student in computer department in VJTI, Matunga, Mumbai.

Pallavi Kulkarni is a post graduate student in computer department in VJTI, Matunga, Mumbai.

Dr. B. B. Meshram is working as Professor in Computer Technology Dept., VJTI, Matunga, Mumbai. He is Ph.D. in Computer Engineering and has 35 papers to his credit at National and International level. He has taught various subjects such as Object Oriented Software Engg., Network Security, Advanced Databases, Advanced Computer Network (TCP/IP), Data warehouse and Data mining, etc at Post Graduate Level. He has guided several projects at graduate and post graduate level. He is the life member of CSI and Institute of Engineers, etc.