Web Project Management

Sunita Suralkar, Department of Computer Technology, Veermata Jijabai Technological Institute, Mumbai
Nilambari Joshi, Department of Computer Technology, Veermata Jijabai Technological Institute, Mumbai
Dr. B.B.Meshram, Department of Computer Technology, Veermata Jijabai Technological Institute, Mumbai

ABSTRACT
This paper describes about the need for Web project management, fundamentals of project management for web projects: what it is, why projects go wrong, and what's different about web projects. We also discuss Cost Estimation Techniques based on Size Metrics. Though Web project development is similar to traditional software development applications, the special characteristics of Web Application development requires adaption of many software engineering approaches or even development of completely new approach to plan and iterate Web application development process. Web application projects have their unique requirements. Web project management lacks process models for web application development. Effective project management for Web Application development is required.

Keywords: Web project management, web size metrics, cost estimation.

Introduction
The web has become the main platform for set up business and social application and organizational information systems. Many organizations have drawn-out the scope of their web-based systems. They have also begun to provide mobile and wireless access to them. Therefore, web-based applications now present an array of content and functionality to a huge number of users and carry out many different tenacities. Because of its global presence, the expectations and demands placed on the web applications have increased significantly over the years.

Many developers of web applications projects do not take into concern complex, unique requirements of web applications. They also fail to realize the features and requirements of web-based systems considerably different from that of traditional software, and so does their growth. They need to identify these differences and take suitable actions to perform the unique requirements of web applications. Hence, many developers and maintainers conduct web applications in an unplanned manner, and fail to adopt wide-ranging design methodologies, resulting in poor quality web systems and applications.

Managing web projects is difficult, the technology requirements, experts from different domains and stakeholders that frequently pull you in different directions, with constricted deadlines adding additional pressures.

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Traditional development</th>
<th>Evolving Web Development</th>
</tr>
</thead>
</table>

Despite some similarities to traditional applications, the unique characteristics of Web applications require the revision of many software engineering approaches or even the development of completely new approaches to make it possible to plan and reiterate Web application development processes.

Web project development is still in its early stages and as such, absences for process models that can serve as a guideline for the development of Web based applications.

Web developments are tough to estimate. With firms having limited resources in terms of people, time, cost, schedule and infrastructure, software developers need to calculate the time and effort required to complete such projects successfully.

Web Projects differ from traditional software project development as can be seen from Table 1.1
Effective management of risks, quality and usability currently appears to be the prime area of Web project management. The major concern estimators’ face is in estimating size, because size drives most of their models.

<table>
<thead>
<tr>
<th></th>
<th>Traditional approach</th>
<th>Web-based challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimating process</td>
<td>Most use analogy supplemented by lessons gleaned from past experience</td>
<td>Job costing done ad hoc based on inputs from the developers (often too optimistic)</td>
</tr>
<tr>
<td>Size estimation</td>
<td>Because systems are built to requirements, SLOC or function points are used. Separate models are used for COTS and reused software (generate equivalent new lines that are merged into the estimates).</td>
<td>Applications are built using templates and a variety of Web-based objects (html, applets, components, building blocks). No agreement on a size measure for Web applications has yet been reached within the community.</td>
</tr>
<tr>
<td>Effort estimation</td>
<td>Effort is estimated via regression formulas modified by cost drivers (plot project data to develop relationships between variables).</td>
<td>Effort is estimated by breaking the job down into tasks and identifying what is needed to do the work. Little history is available.</td>
</tr>
</tbody>
</table>
Developers involved in most working Web projects agree that SLOC might not be suitable for primary estimation because they are design-based, while FPs might be untimely because applications do more than just transform inputs to outputs. In response, dozens of size metrics have developed for Web development (object points, application points, and multimedia points, for example). The proposed metric, Web Objects, calculates size by considering each of the many elements that involve the Web application.

For Web development, cost is difficult to estimate because:

1. There is no standard to sizing Web applications. Each can be created using diverse technologies such as several forms of Java (Java, Servlets, Enterprise Java Beans, Applets, and Java Server Pages), HTML, JavaScript, XML, XSL, and so on. Attempts have been made to apply Function points principles to sizing Web applications (Rollo, 2000; Mendes et al., 2002), however Mendes et al. did not find any improvement in estimation accuracy using Function points-based size metrics, when compared to other size metrics (e.g. number of Web pages, number of images).

2. Web development processes differ substantially from traditional approaches (Reifer, 2002; Baskerville et al., 2003). A recent survey of Web development practices by Barry and Lang (2001) showed that practitioners are not using development processes proposed in the literature (e.g. UML). They often develop their own in-house methods, which of ten include prototyping.

3. Web project’s primary goal is to bring quality applications to market as quickly as possible, varying from a few weeks to six months (Pressman, 2000; Reifer, 2002; Offutt, 2002).

4. People involved in Web development are represented by less experienced programmers, users as developers, graphic designers and new hires straight from university (Reifer, 2002; Standing, 2002).

5. Processes employed are in general ad hoc (Standing, 2002), although some organizations are starting to look into the use of agile methods (Ambler, 2002).

Table 1.2: Web Estimating Challenges. [3]

<table>
<thead>
<tr>
<th>Schedule estimation</th>
<th>Schedule is estimated using a cube root relationship with effort</th>
<th>Schedule is estimated based upon analogy. Models typically estimate schedules high because cube root relationship doesn’t hold.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality estimation</td>
<td>Quality is measurable from internal metrics like defect rates and system properties</td>
<td>Quality is hard to measure. New metrics are needed to assess “quality” of multimedia.</td>
</tr>
<tr>
<td>Model calibration</td>
<td>Measurements from past projects are used to calibrate models to improve accuracy</td>
<td>Measurements from past projects are used to identify folklore (too few to be used yet)</td>
</tr>
<tr>
<td>“What if?” analysis</td>
<td>Estimating models are used to perform quantitative “what if” and risk analysis. They are also used to compute return on investment (ROI) and cost/benefits.</td>
<td>Most “what if” and risk analysis is qualitative because models do not exist. ROI and cost–benefit analysis for electronic commerce applications remain an open challenge.</td>
</tr>
</tbody>
</table>

Related Work

Web metrics for web cost estimation are proposed by Cowderoy et al., 1998; Mendes et al., 1999, Cowderoy, 2000; Mendes and Mosley, 2000; Reifer, 2000; Rollo, 2000; Cleary, 2000; Mendes et al., 2001.

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.

Mendes et al. (1999, 2001), Mendes and Mosley (2000) propose size metrics to be used to predict authoring effort for hypermedia applications and then for Web applications.

Rollo (2000) did not suggest any new size metric, however, as far as we know, 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 both IFPUG, MKII, and COSMIC-FFP 2 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.

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.

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.

All size metrics presented in this Section were invariably related to implement 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.

The metrics proposed for early cost estimation web cost estimation were grouped into six categories:
1. Web application static metrics
2. Web application dynamic metrics
3. Cost Drivers
4. Web project metrics
5. Web company metrics
6. Web Interface style metrics

Web application static metrics correspond to attributes that have been commonly used to size Web hypermedia applications (e.g., number of Web pages, number of graphics etc.). Web application dynamic metrics refer to any features or functionality mentioned (e.g., database integration, on-line secure order form etc.), which is taken as an abstract measure of functionality and therefore size (Fenton and Pfleeger, 1997). Cost drivers correspond to attributes that may influence effort but do not characterize application size. Web project metrics incorporated metrics we believed may have a bearing on the contingency and/or profit costs that are provided in a price quote (Kitchenham et al., 2003) (e.g., project budget). Web company metrics are attributes that characterize a Web company (e.g., target audience), and Web interface style metrics are attributes that characterize the final appearance (style) of the Web application (e.g., background colour, style etc.).

Discussion

Many developers of web applications projects do not take into concern many-sided, unique requirements of web applications. They also fail to realize that characteristics and requirements of web-based systems considerably different from that of traditional software, and so does their development. They need to identify these differences and take suitable actions to perform the unique requirements of web applications. Software estimation applied at early stage of software development, so capturing the estimation from the known characteristics of software will be a useful approach for estimation, many tools used to represent the requirement and the design of any software such as UML diagrams, Z specification language, state transition diagrams, data flow diagrams.

The suggested web metrics are for early web cost estimation here the requirements are not gathered and many details are not known, this leads to an approximate estimation.

In web applications we have mobile codes that are downloaded at client side and are executed, none of the web metrics for calculating the size metrics consider this, the code size at client side is not considered.

For calculating the size metrics we can using the work breakdown structures and divide the major projects into smaller subparts and then calculate size for each them, adding them we can get the cost of the size metrics too.

Web Projects are different than traditional software development in terms of return of investment factor is not considered in calculating the web cost metrics this results in software failures.

There is a need to develop a model which able to estimate the efforts, cost for OSS, hence the normal cost, efforts models inadequate for this purpose, such kind of models should care for the nature of OSS environment such that the development of OSS can be achieved almost 24x7 between the volunteers around the world, also how to measure the impact of the community surrounding the core development team, rather than the normal effects of a normal software wanted to be estimate.
Conclusion
For effective web project management, the cost estimation is still done in unplanned manner; some benchmarking techniques have to be devised. The size metrics studied 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 made using any existing Web development methods. This makes their usefulness as primary effort predictors uncertain.

References