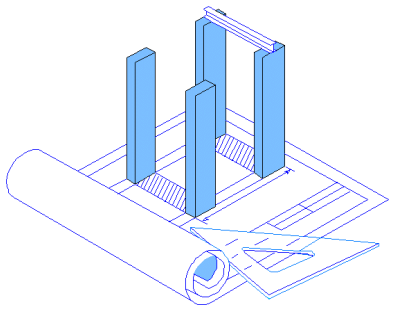
TASSC

technical paper



Estimating Web-Based Application Development

Robert Armstrong

Tassc Limited
www.tassc-solutions.com

First Published: February 2000
Last Updated: January 2008

Web-based applications represent the future of Information Technology. For those businesses, large and small, with an established presence on the World Wide Web, that future is already here. It is a connected world and few would argue that, despite the recent dot.com collapse, the 'e' revolution is now unstoppable.

Organisations embracing e-commerce demand a secure, reliable, efficient and flexible infrastructure across a range of hardware and software platforms. Successful delivery of mission-critical web-based applications and their integration into the business depends as much on project management best practice as it does on choosing the most appropriate technology.

In this paper, we examine the issues specific to estimating web-based application development and discuss extensions to ObjectMetrix, an emerging standard for estimating and forecasting effort, duration and cost of UML-based software development projects.

What is a Web-Based Application?

browser interface to business logic

In all but the most trivial of cases, a web-based application is essentially a client-server system, which combines traditional business logic and functionality, usually server-based, with the hypermedia navigation and data entry facilities provided by browser technology running on the client. Through a client web browser, users are able to invoke business operations and subsequently to change the state of business data on the server.

The range of web-based applications varies enormously, from simple web sites that are essentially hypertext document presentation applications, to sophisticated high-volume e-commerce applications often involving supply, sourcing, ordering, payment, tracking and dispatch of goods or the provision of services.

Web-Based Application Development

implications for project management and development processes

The unique nature of many web-based applications broadens the role of traditional project management and adds a new dimension to the software development process. In addition to the participation and contribution of analysts, designers, programmers, architects, managers, domain experts and so on, web-based applications often contain significant multimedia content (images, movie clips, sound clips and text) requiring specialist resources for their development.

multimedia development

Groups separate from the software engineers usually produce multimedia web content in parallel, in a similar way that on-line help is typically produced by a technical writer rather than a software developer. Some recent studies indicate that specialist resources needed to develop multimedia content may outnumber software engineers.

For the purposes of estimating software development effort, multimedia content is assumed to exist and the effort required for their production is outside the scope of the software engineering process. However, the effort of integrating these elements needs to taken into account. Furthermore the effort to create the multimedia content itself must be incorporated to produce an overall project estimate.

focus on developing business logic

The novelty of developing web-based applications can be captivating and may obscure the fact that modelling an application's business logic, whether it is being implemented in a web page, a component or an object remains the central focus of the project.

Web-based application development requires project management best practice as much as any other type of application development.

Web-Based Application Architecture

a client-server architecture

The architecture of a web-based application is essentially client-server, comprising a client browser, a web server and a communications network.

A principal consideration in client-server application architecture, and therefore web-based application architecture, is partitioning functionality between the client and the server. This is often reduced to a choice between thin client versus thick client (or something in between) and the extent to which a distributed object infrastructure is required.

These decisions are dependent on a number of factors specific to the nature of the application under consideration and the capabilities of the underlying technology.

Only when the high-level application architecture has been defined is it possible to specify which architecturally significant software artifacts need to exist on the client or on the server, and those that need to exist on both. Software artifacts can then be identified, enumerated and used as detailed input to the software estimation process.

Web-Based Application Concepts

web pages are new abstractions

Although web-based applications are essentially client-server systems that can be modelled by standard UML types, some of their novel features require the introduction of some new concepts necessary to model the web-specific aspects of the system.

The fundamental abstraction in a web-based application is the web page. Web pages can range from simple documents that may contain images, movie clips, sound and text, that can be rendered by a browser; to pages containing complex scripts that interact with server resources such as external systems, databases and business logic.

For the purposes of estimation, these distinctions are critical. Clearly we need to be able to determine if a particular web page is a simple HTML document for presentation in a browser, or is a complex collection of scripts, logic and data. This distinction is necessary to accurately estimate development effort for the web-specific aspects of the system.

web pages are first class abstractions like classes and components

Web pages are considered first class abstractions, as significant as traditional software artifact types used to model the rest of the system. A sophisticated web page is in effect an executable system element, similar to an object (class instance) or a component.

web pages are different from classes and components

Although we can view web pages in a similar way to classes or components, there are some significant differences. It is relatively straightforward to model a web page that resides exclusively on the client or on the server, though either may contain significant levels of sophistication in their own right.

The real difference arises in cases where a single web page may contain scripts that interact with resources on the server, and a different set of scripts that execute on the client, leading to different behaviour. From an estimation point of view we need to be able to make a clear distinction between these different kinds of web pages because each will require significantly different effort to develop.

scripts are also first class abstractions

The effort required to develop a web page is a function of the effort required developing any scripts it contains, and the effort required to include any multimedia content. As already discussed, multimedia creation and development is considered as separate from the development of the software, but integrating these elements and testing their operation requires considerable effort.

By far the most significant aspect of developing web pages are the features that they provide through the scripts that they contain. Therefore scripts must also be considered as first class abstractions in a web-based application.

Estimating UML-Based Applications

Application scope is a measure of the scale of the software under development, and is represented by the organisation of architecturally significant software artifacts. ObjectMetrix defines software artifact types (classifiers) that are fundamental to object-oriented and component-based applications, which correspond to definitions in the UML.

In specifying the scope of object-oriented and component-based applications for estimation purposes it is necessary to identify the following key abstractions:

subsystem

A set of application features geared to support a specific area of the business. A subsystem is a logical grouping of classes that implement a related set of use cases.

class

An abstraction that is fundamental to the problem domain, responsible for encapsulating business information or the provision of specialised system services.

use case

A feature of the application visible to system actors (typically end users or other systems) that describe functional and behavioural characteristics.

component

A replaceable and executable unit of software built from classes to implement a specified set of interfaces.

interface

Well-defined contractual specification of the services offered by a component within an application.

Estimating Web-Based Applications

Web pages and scripts often perform important business operations and act as real objects in an application. They need to co-exist with traditional UML abstractions in an application and need to be modelled and estimated in an equivalent and comparable way. ObjectMetrix has been extended to include the definition of software artifacts that are fundamental to web-based applications.

Specifying the functional and behavioural requirements of a web application is captured and expressed by use cases. Use cases provide a consistent approach to how end users and other systems interact with the application as a whole, including the web-specific aspects.

In specifying the scope of web-based applications for estimation purposes it is also necessary to identify the following new abstractions:

web page

A web page can be considered a generic container or an encapsulation that may exhibit behaviour and may contain data, and can exist as a client web page, a server web page or a combination of the two.

A client web page executes only on the client, and typically contains text, data and presentation information. Client web pages may contain forms (collections of input boxes, text areas, radio buttons, check boxes etc.) as well as business logic. A server web page executes only on the server. Server web pages typically contain business logic and interact with server resources.

script

The business logic and functionality of a web page is implemented by scripts, which may be interpreted by a scripting engine or may be loaded and executed as pre-compiled modules.

An interpreted script exists as a separate file of text statements in a scripting language (e.g. JavaScript, VBScript, or PERL) that can be included in a web page.

A compiled module is a separate binary file or library of subroutines or functions (e.g. .dll and .exe) that can be included in a web page. Compiled modules produce HTML output at the request of a client browser and typically contain high business logic content.

Qualifiers

The characteristics, relationships and number of each type of software artifacts are quantitative and are therefore the principal factors that determine the resulting estimate. In ObjectMetrix, applying a set of qualifiers can further refine individual software artifacts.

complexity

Complexity is a measure of the degree of difficulty in developing a software artifact. The more complex a software artifact, the more effort will be required to develop it.

size

Size is a measure of the quantity of the content of a software artifact. The more content, the bigger the software artifact and the more effort will be required to develop it.

reuse

Reuse is a measure of the extent to which pre-existing software can be utilised in the development of a particular software artifact. The more reuse that can be achieved the less effort will be required to develop the software artifact.

genericity

Genericity is a measure of how much a particular software artifact is required or intended to be reused. The more re-usable the software artifact, the more robust and efficient it needs to be, and therefore the more effort will be required to develop it.

The Impact of Technology

bewildering array of existing and emerging technologies

There is a bewildering array of existing and emerging technologies available for and used in the development of web-based applications. Some of the areas web technologies encompass include: presentation (e.g. HTML, DHTML, XML); scripting and programming languages (e.g. VBScript, JavaScript, PERL ActiveX, Java); network protocols and distributed computing technologies (e.g. DOM, DCOM, HTTP, TCP/IP, IIOP, CORBA, RMI, JavaBeans).

dominant technology is HTML

In web-based applications the unique and dominant technology is HTML, (HyperText Markup Language), and more recently Dynamic HTML and XML (Extensible Markup Language), used to construct web pages. Web pages may or may not include scripts, modules, multimedia, or text content, but almost always comprise a proportion of HTML or DHTML, which specifies how a (client) web page should be rendered in a browser.

In estimation, technology is the choice of scripting and programming languages, environments and network protocols used in developing a web-based application. The main impact of technology is on the programming activity. However, the choice of technology may also impact design, testing and integration.

technology choice has a small effect on the estimate

The overall effect of technology choice on the estimate is typically less than the impact of qualifiers or the skill level of the development team. In ObjectMetrix, an estimate for web-based application development can be calculated that is independent of technology, where the choice of, for example scripting or programming language has not yet been made. This is possible because there is an intrinsic effort required to carry out development activities, which is irrespective of any specific technology.

Productivity Metrics

metric (n) – a standard of measurement

In ObjectMetrix, a clear distinction is made between the process of estimation (calculating) and metrics (standards of measurement). ObjectMetrix uses a set of base productivity metrics for the effort required to develop UML software artifacts, which have been refined from data gathered from hundreds of software development projects. These are an essential start point in calculating the estimate.

Separating metrics from the estimation model in this way allows the metrics to be calibrated to take account of the particular circumstances of each project, and even allows ObjectMetrix to incorporate third party metric data.

web page and script metrics are provided in ObjectMetrix, but they can be calibrated

Unlike object-oriented and component-based applications, the relatively recent mainstream emergence and adoption of web-based applications has not yet allowed enough time for a sufficient body of reliable metric data to be established. Only recently has any attention been given to productivity metrics for web-based development.

However, Tassc in collaboration with web-based development groups in the UK and the US has developed a series of base productivity metrics for web-based application development. ObjectMetrix uses these base productivity metrics as an initial start point for the effort required to develop web-based software artifacts, and like the other base productivity metric data sets within ObjectMetrix, they can be calibrated.

Conclusion

'e' is for estimation

The technology available for the creation of web-based applications is becoming easier and easier to use, leading many organisations to develop and introduce web applications as attractive web sites without carrying out sufficient business analysis. Too often the result is an incomplete, unreliable and difficult to maintain application that fails to meet the needs of the business, and costs more and takes longer to develop than expected.

Through our involvement in hundreds of software projects we have recognised that perhaps the most important factor in delivering on time and within budget is the preparation of realistic and reliable software estimates.

ObjectMetrix has proven to be an accurate, measurable and repeatable technique for estimating and forecasting duration, cost and resource requirements for object-oriented and component-based software development projects. The extensions to ObjectMetrix described in this paper now offer a reliable, measurable and repeatable estimation technique for the new world of web-based application development.