Using of Web Objects Method in Agile Web Software Projects

Stipe Čelar, *Member, IEEE*, Željko Šeremet, Željko Marušić and Mili Turić

***Abstract* — Paper describes an implementation of Web Objects (WO) methods for measurement of software development results. The method is applied to a data-set of 2 projects obtained by Faculty of Science and Education, University of Mostar, focused on web application projects. The projects are done by internal agile team during 2013 year. The ultimate goal of this approach is using of WO for early estimation in agile projects.**

***Keywords* — Software metrics, web development, function points, web objects, agile projects.**

I. INTRODUCTION

Nowadays the most software projects of the world which are in the form of Total Systems are implementing as a Web application. Depends on what we expect from our product, in this report we want to say that Web application could be considered as a kind of web site or portal or even as an intranet Web application. How to evaluate a Web application is the most important subjects. It has different ideas that we will explain some of them which is related to our topic in this report. [1] [2]

Web technologies, once exploited only for creating hypermedia applications (static web sites), have known a huge development, and now they are one of the cutting-edge technologies for very different kinds of systems, from small informationcentred applications to large enterprise-scale commercial systems. The differences from traditional software, in terms of technology, development model, time-to-market needs, and volatility of requirements and so on, pose serious challenges in adapting traditional size metrics, like Function Points [3], to measure Web applications. [4]

Estimating the size of web applications poses new problems for cost analysts. Because hypertext languages (html, xml, etc.), multi-media files (audio, video, etc.), scripts (for animation, bindings, etc.) and web building blocks (active components like ActiveX and applets, building blocks like buttons and objects like shopping carts, and static components like DCOM and OLE) are employed in such applications, it is difficult to use traditional size metrics like source lines of code and function points. Improved techniques for estimating the size needed to resolve the shortcomings of the Web Object metrics. [5] [6]

This method was applied to a data-set of 2 projects obtained by Faculty of Science and Education, University of Mostar, focused on Web application projects.

The paper is organized as follows: Section 2 discusses the related works in the area of web size measurement, with a detailed description of the Web Object metrics. Section 3 describes our web development projects. Section 4 presents the results of the research. Section 5 presents the conclusions and plans for future work.

II. RELATED WORK

A. *Overview of Web measures*

Nowadays there exist various size measures for Web applications [7]. In 1998, the IFPUG published guidelines to measure Web applications using the Function Point Analysis (FPA) counting rules [8]. Rollo [9], however, identified difficulties when measuring Web applications with these guidelines, such as difficulties encountered in identifying the system's boundary and its logical files. Also, the company Total Metrics [10] recognized that the IFPUG guidelines did not resolve many of the counting issues faced when sizing Web applications. They provided interpretations of the IFPUG guidelines and explained how FPA can be applied to size Web Applications. [11]

A major limitation of Function Points is their restricted applicability being largely applied to Management Information Systems (MIS). To address this shortcoming additional types of functional size measure have been developed: Object oriented Function Points were proposed for Integrated CASE environments [12]; Feature Points were designed to work equally well with MIS applications, and other types of software such as real-time software or embedded software [13]; Full Function Points (FFP) were proposed as a method for measuring the size of real-time systems [14]. [15]

Another proposal is Internet Points [16], which extends FPA to explicitly support Web project estimation. A Web site is sized by counting seven types of functions: files, RDB tables, APIs, messages sent by the system, number of static HTML pages, number of dynamic HTML pages and number of interactive pages. This method was automated in a tool called Cost Xpert.

Cleary [17] proposes the Web-Points as a measure for size and complexity of static Web sites. This method takes into account the complexity of the HTML pages of a web site. The number of Web-Points assigned to a page is a function of the size of the page in words, the number of

Stipe Čelar (*corresponding author*) is with the FESB, University of Split, Ruđera Boškovića 32, Split, Croatia, (phone: +38521305843, e-mail: stipe.celar@fesb.hr).

Željko Šeremet is with the FSR, University of Mostar, Matica Hrvatska b.b., Mostar, Bosnia and Herzegovina, (phone: +38736337015, e-mail: zeljko.seremet@fsr.ba)

Željko Marušić is with Faculty of Science and Education, University of Mostar, Matica Hrvatska b.b., Mostar, Bosnia and Herzegovina, (phone: +38736355474, email: zeljkomarusic@fpmoz.ba).

Mili Turić is with Venio Indicium d.o.o., Doverska 19, Split, Croatia, (phone: +385 21 674287, email: mili.turic@venio.hr).

existing links, and the number of non-textual elements of the web page. Cleary's metric was used with productivity data to determine the effort required for the development or improvement of static Web sites. Cleary's proposal focuses on static Web sites and therefore does not consider behavioural and navigational properties of Web applications. [11]

However, to date, there are no widely accepted Web measures in industry. Among these measures, a functional size measure could provide a good solution to the problem of sizing Web applications since:

a) it is based on the functionality that will be provided to the users and not on the artefacts that are produced when the Web application is delivered (e.g., number of Web pages, multimedia files);
b) it can be measured in early stages of the Web development process;
c) it is based on standards that define the concept of size and the requirements for software sizing (e.g., [18], [19]); and
d) it is widely accepted and used in industry [20], [2].

Widely adopted functional sized measurement (FSM) methods such as FPA and COSMIC have some important limitations. The first problem is that the application of the rules requires human interpretation (e.g., it is not always obvious how to classify and to count every element of the requirements). Consequently, a function point expert on applying the method is required (e.g., a certified specialist). The second problem is that none of the ISO-standardized FSM methods have been designed to take the particular features of Web applications into account.

To address these limitations, 10 size measurement procedures for Web applications have been proposed in the last years [2] [21]. Four of these procedures were proposed by practitioners (IFPUG guidelines for Web applications [8], internet points [16], Web-points [17], Web Objects [22]). The remaining six were proposed by researchers (OOmFPWeb [11], [23], data web points [24], [25], [26], [27] and OO-HFP [27]).

B. Web Objects

Using software science as basis, it has developed a new metrics, Web Objects, to represent the size of such web applications. Web Objects are an extension of function points that take predictors that web applications are sensitive to into account as the size of such applications are being estimated. As conceptually illustrated in Figure 1, Web Objects extend traditional function points to take the following four additional types of objects into account because they require additional effort to incorporate them into web applications:

- **Multi-media files** – size predictors developed to take the effort required to incorporate audio, video and images into applications. Such effort includes the work involved in creating web pages; creating video for web (MPEG-files); creating publishable documents for the web; and creating, editing and enhancing complex images for both clients and servers.
- **Web building blocks** – size predictors developed to take the effort required to develop web-enabled fine-grained component and building block libraries and any wrapper code required to either instantiate or integrate them. Such predictors do not count the standard libraries that come as part of your web environment and typically include both Windows and Java components. Instead, they take only the additional active (ActiveX, applets, agents, guards, etc.), fine-grained static (COM, DCOM, OLE, etc.) and course-grain reusable (shopping carts, buttons, logos, etc.) building blocks that you acquire or develop to incorporate into web applications into account for both your client and server.
- **Scripts** – size predictors developed to take the effort required to link html/xml data and generate reports automatically; query ODBC-compliant databases via prompts; integrate and animate applications via predefined logic (via GIF); and direct dynamic web content per customizable pallets, masks, windows and commands (streaming video, real-time 3D, special effects, motion, guided workflow, batch capture, etc.) for both clients and servers.
- **Links** (xml, html and query language lines) – size predictors developed to take the effort required to link applications, integrate them together dynamically and bind them to the database and other applications in a persistent manner. [6]

To count web objects, it evaluates the following nine components of a web system based upon user requirements and page layouts:

- **Internal Logical Files** – logical, persistent entities maintained by the web application to store information of interest.
- **Multi-Media Files** – physical, persistent entities used by the web application to generate output in multi-media format.
- **Web Building Blocks** – logical, persistent entities used to build the web applications and automate their functionality.
- **Scripts** – logical, persistent entities used by the web application to link internal files and building blocks together in predefined patterns.
- **Links** – logical, persistent entities maintained by the web application to find links of interest to external applications
- **External Interface Files** – logical, persistent entities that are referenced by the web application, but are maintained by another software application.
- **External Inputs** – logical, elementary business processes that cross into the application boundary to maintain the data on an Internal Logical File, access a Multi-Media File, invoke a Script, access a Link or ensure compliance with user requirements.
- **External Outputs** – logical, elementary business processes that result in data leaving the application boundary to meet a user requirements (e.g., reports, screens).
- **External Queries** – logical, elementary business processes that consist of a data "trigger" followed by a retrieval of data that leaves the application boundary (e.g., browsing of data). [6], [25]

A list of operands (the objects) and operators (actions that can be done to the object) that contributes to the application complexity is given in [21].

III. Development projects at University of Mostar

Faculty of Science and Education, University of Mostar, started two strategic, interconnected web software projects – "Nastava" (eng. *"Teaching"*) and "Raspored" (eng *"Schedule Classes"*) in 2013. Internal agile development team consisting of three experts decided to introduce measurement and estimation techniques in their Web development process. As basic size and complexity metrics during these two development projects they chose the WO metrics and predictors from [21].

The team used its own, customized agile development methodology. In conjunction with records of team's spent working hours (*Date; Work description; Work category; Work duration; Developer; Use Case; Project*), WO metrics will be used to improve the process of planning and control of the future, similar development projects.

A. Project "Teaching"

The curriculum is the foundation for any educational institution. In the project "Teaching" the web application is developed, that:

a) Provide detailed information for students on individual courses, and courses in one place.
b) Provide accurate and timely information about changes in the curriculum for the needs of the referent for teaching and information system for exam.
c) Provide possibility of tracking history of changes in the curriculum.
d) Provide students with information about consultations with professors and teaching assistants (terms, office number, phone number, e-mails,…).
e) Integrate the application "Teaching" with the e-learning platform.

The application consists of the informative part that is displayed to end users (front-end) and the other one that is intended for content editors (back-end). One of the important requirements was to enable each editor separate rights and privileges of use.

Based on a detailed analysis, it was decided that technology will be based on PHP. Interactive components are implemented with AJAX. On the client side JavaScript code is running and CSS was used for the design. To manage the data itself a MySQL database server is used.

B. Project "Schedule Classes"

The smaller project "Schedule Classes" provides students information about the terms of teaching, halls and running of exams on individual study groups, venues or lecturer. This application relies on the administrative part of the editing that has been implemented in the project "Teaching". A team of people and technology that will be used remains the same as for the project "Teaching". This application can be considered, for its properties as an extension of the project "Teaching" because it will use part of the back-end interface and is related to the curriculum. The application consists also of parts that belong to the standard desktop applications and services and therefore is quite complex and heterogeneous in nature. So it is listed as a separate project.

IV. Measurement results

Using the predictors listed in [21] to compute the number of Web Objects, we have been able to predict a Web application's size repeatable and robustly. These predictors let us consider the elements that contribute to the Web application's size. We can represent each predictor by the unique number of operands and operators that they contribute to the application. Like function points, the key to developing repeatable predictor counts is a well-defined set of counting conventions. This approach lets us to achieve consistency across organizations and resolve conflicts, because size estimates are formulated using such standards.

By their very nature, such counts must clearly separate operands from operators because the latter represent what we do to an object, not what the object does. Table 1 also provides examples of operands and operators to clarify what is counted as Web Object estimates.

Table 1 shows the objects we developed in web applications "Teaching" and "Schedule Classes".

TABLE 1 NUMBER OF DEVELOPED WEB OBJECTS

Web Object predictors	***Number of WO***	
	"Teaching"	**"Schedule Classes"**
Number of building blocks	18	10
Number of COTS components	38	5
Number of multimedia files	33	14
Number of object or application points	36	3
Number of Web components	6	4
Number of graphics files	50	10
Number of scripts	31	5
Other	2	0
Sum	**214**	**51**
Total	**265**	

In WO counting process we first identified the Web elements that contribute to the job we are counting. We started by selecting the applicable items listed in the predictor column and counted them across each application. Finally, we summed the columns to compute the number of Web Objects for both applications. The sums of 214 and 55 WO in Table 1 represent the overall size of both applications. Such a trivial Web elements summing, without taking into account the coefficients with which they contribute to the application complexity is not in accordance with the Halstead's theory [6].

Although the contribution of Web elements in the overall complexity and size is not linear, such linear dependency can be assumed with reasonable confidence for a small development team and its agile planning process. In these two projects it is confirmed by the records of project effort that the developers recorded daily. Fig. 1. and Fig. 2. show the project activities in these two projects (item "*work category*" from the work evidence).

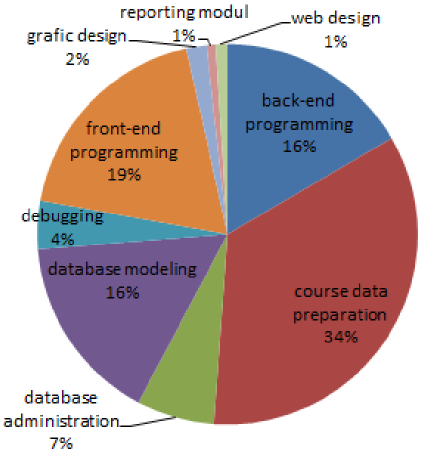

Fig. 1. Project effort ("Teaching")

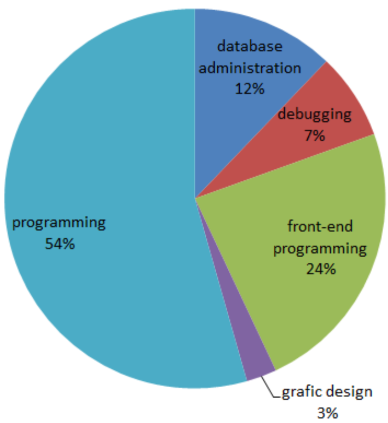

Fig. 2. Project effort ("Schedule Classes")

V. CONCLUSION

The main aim of the work described in this paper is to present implementation of Web Object method into small development team as a step to agile estimation method. We presented an empirical study of software development effort measurement using Web Objects method, performed on a set of two web projects carried on at Faculty of Science and Education, University of Mostar.

The smaller project "Schedule Classes" (Fig. 2) recorded 30.1% of the project effort "Teaching" while their ratio in WO is very similar – 25.7%. These results encourage us to use WO as an agile estimation method for similar projects even in an early project stages, especially as some works refer to errors of even 600% [7].

A further step in our research will consist in recognizing the complexity contribution of each recognized Web elements.

REFERENCES

[1] A. T. Tabrizi and M. S. Moshkenani, "Evaluation of Web applications according to intelligence parameters", *Proc. of the 2008 International Conf. on Computational Intelligence for Modelling Control & Automation – CIMCA '08*, IEEE, p. 698-702, 2008

[2] S. Abrahão, J. Gómez and E. Insfran, „Validating a size measure for effort estimation in model-driven Web development", *Information Sciences*, Elsevier, vol 180(20), pp. 3932–3954 2010

[3] A. J. Albrecht, "*Measuring application development productivity*," in Proc. of IBM App. Develop. Symp., Oct. 1979, pp. 83–92.

[4] G. Barabino, G. Porruvecchio, G. Concas, M. Marchesi, R. De Lorenzi and M. Giaccardi, „ *An Empirical Comparison of Function Points and Web Objects*", Computational Intelligence and Software Engineering, 2009. CiSE 2009. Intern. Conf. on 11-13 Dec. 2009

[5] R. Folgieri, G. Barabino, G. Concas, E. Corona, R. De Lorenzi, M. Marchesi and A. Segni, "*A Revised Web Objects Method to Estimate Web Application Development Effort",* WETSoM'11, May 24, 2011, Waikiki, Honolulu, HI, USA

[6] D. Reifer, "*Web objects counting conventions*", 2002, Available: http://www.reifer.com/docs/WOCounting.doc (retreived at: 14.2.13.)

[7] E. Mendes, N. Mosley and S. Counsell, „*Comparison of Web size measures for predicting Web design and authoring effort*", IEE Proceedings - Software, vol. 149, no. 3, 2002, pp. 86-92.

[8] IFPUG, „*Hints to Counting Web Sites*", IFPUG White Paper, 1998.

[9] T. Rollo, 'Sizing E-commerce', *Proceedings of the ACOSM 2000 – Australian conference on Software measurement*, Sydney, 2000

[10] Total Metrics, „*Web Based Applications: How is FPA applied to sizing Web Based applications?*", 2001.

[11] S. Abrahão, G. Poels and O. Pastor, „*Evaluating a Functional Size Measurement Method for Web Applications: An Empirical Analysis*", Proceedings of the 10th International Symposium on Software Metrics (METRICS'04), 2004

[12] R. D. Banker, R. J. Kaufman and R. Kumar, „An empirical test of object-oriented output measurement metrics in a computer aided software engineering environment." *Journal of Management Information Systems*, vol. 8, no. 3, Winter 1991-92.

[13] T. C. Jones, „*Estimating Software Costs*". McGraw-Hill, New York, 1998.

[14] A. Abran, M. Maya, J.-M. Desharnais and D. St-Pierre, „Adapting Function Points to Real-Time Software", *American Programmer*, November 1997, pp. 32-42.

[15] M. Ruhe, R. Jeffery and I. Wieczorek, "Using Web Objects for Estimating Software Development Effort for Web Applications", *Proceedings of the Ninth International Software Metrics Symposium (METRICS'03)*, IEEE, pp. 30-37, 2003

[16] Cost xpert group inc., Estimating internet development, Available: http://www.costxpert.com, (retrieved at: 14.2.13)

[17] D. Cleary, „Web-Based Development and Functional Size Measurement," Proc. of IFPUG Annual Conference, San Diego, USA, 2000, Available: http://citeseerx.ist.psu.edu/showciting;jsessionid=382D3DE894C4784F688EBAA7C8980321?cid=900878, (retrieved at: 03.06.09)

[18] ISO/IEC 20926: Software engineering – IFPUG 4.1, Unadjusted functional size measurement method – counting practices manual, 2003.

[19] ISO/IEC 19761, Software engineering – COSMIC-FFP, A functional size measurement method, 2003.

[20] International Software Benchmarking Standards Group (ISBSG), Available: http://www.isbsg.org, (retrieved at: 03.06.09)

[21] D. Čeke, *"Mjerenje funkcionalne veličine web aplikacija bazirane na konceptualnom modelu"*, PhD qualification exam, Faculty of Electrical Engineering, Zagreb, Croatia, 2012, Available: http://bts.fer.hr/_download/repository/KDI_Denis_Ceke.pdf (retrieved at: 14.2.13.)

[22] D. J. Reifer, *"Web Development: Estimating Quick-to-Market Software"* Available: http://www.inf.utfsm.cl/~visconti/iswav/documentos/WebDevelopment.pdf (retrieved at: 14.2.13.)

[23] S. Abrahão and O. Pastor, „*Measuring the functional size of web applications*", International Journal of Web Engineering and Technology 1 (1) (2003) 5–16.

[24] S. F. Ochoa, M. C. Bastarrica and G. Parra, "Estimating the Development Effort of Web Projects in Chile", *LA-WEB '03 Proc. of the First Conference on Latin American Web Congress*, Page 114, IEEE Computer Society Washington, DC, USA ©2003

[25] E.J.D. Cândido and R. Sanchez, "Estimating the size of web applications by using a simplified function point method," in *Proceedings of the WebMedia/LAWEB 2004*, 2004, pp. 98–105

[26] G. Costagliola , S. Di Martino, F. Ferrucci, C. Gravino , G. Tortora, G. Vitiello, "*A COSMIC-FFP: Approach to Predict Web Application Development Effort*", J. of Web Engineering, vol.5, no.2, pp. 93-120, 2006

[27] P. Fraternali, M. Tisi, A. Bongio, "Automating function point analysis with model driven development," in *Proc. of the Conf. of the Centre for Advanced Studies on Collaborative Research* (*CASCON'06*), Markham, Ontario, Canada, 2006, pp. 33–247.

[28] S. Abrahão, E. Mendes, J. Gomez, E. Insfran, "A model-driven measurement procedure for sizing Web applications: design, automation and validation", in *Proc. of the ACM/IEEE 10th Intern. Conf. on Model-Driven Engineering Languages and Systems* (*MODELS'07*), Nashville, TN, USA, LNCS, vol. 4735, pp. 467–481, Springer, 2007