

WebComposition/EMS: A Value-Driven Approach to Evolution

Stefan Wild¹, Martin Gaedke²

^{1,2} Chemnitz University of Technology
09111 Chemnitz, Germany

¹ swild83@gmail.com, ² martin.gaedke@cs.tu-chemnitz.de

Abstract. The evolution of Web applications is the continuous process of maintaining available and developing new functionalities. Already implemented functionalities are provided by the units of the distributed Web system, i.e. its components and services, which have to be maintained to assure a stable operation. Furthermore, to react on upcoming needs existing units are adapted or new ones are developed and integrated. Depending on the unit of evolution, more or less efforts are invested. Alas, the benefit caused by units evolved becomes only apparent after some time and usually solely with regard to the entire product. However, the values of units of evolution are subject to change during their life cycles because they are influenced by several parameters like audience acceptance, development and maintenance costs. That is, before a positive or negative impact on the product revenue can really be perceived, the values of Web application services and components already underlie changes. So, it would be profitable to take notice of such value alternations to be able to anticipate revenue tendencies as early as possible and consequently to initiate appropriate actions on time.

Keywords: evolution, portfolio management, maintenance, agile development, components, services

1 Introduction

Today's modern Web applications are compositions of components and services that are responsible for providing various functionalities [1]. Once a Web application has been deployed it is in global competition with other worldwide available products that offer similar functionalities. To keep up with competitors, existing functionalities are maintained and promising functionalities are added to the distributed Web system during its life cycle. The units of a Web application, i.e. its components and services, are adapted or created to react on new user needs and enhanced technologies [2]. A Web application and hence its units are evolved to fulfill changing and emerging requirements on content, user experience, and infrastructure while ensuring that the acceptance remains constant or increases.

However, it is a complex business to perform the evolution of a Web application. The evolution is very difficult, error-prone, and expensive as maintenance and new design decisions have to be drawn in an agile manner. Due to the limited time frame for introducing new functionalities and accordingly having a lead over the competitors, the evolution of a Web application must always be driven in the right direction. This direction might change frequently through various influences, like the market or additional user expectations. On the other hand, the effects of actions during the evolution of a Web application often become obvious too late and are too general. Only after some time, the consequences of an evolution, i.e. its success indicators, are reflected in the profit or loss of a product. Unfortunately, these indicators are solely related to the Web application at all and not apportioned to each of its units. Thus, it is hard to distinguish valuable from loss-generating units. As a result of that, further investments of resources are not made on a sound basis.

In this paper, we introduce WebComposition/EMS, which is a value-driven approach to evolution addressing these problems with concepts borrowed from portfolio management. This agile approach continuously monitors each component and service of a Web application and records each such unit's values of the composition over time. As such, maintenance decisions can be supported by visualizing the history of past values of components or services or even the Web application's overall composition. By applying techniques from finance mathematics it might also be possible to estimate trends towards positive or negative tendencies of the Web application.

In the following section the process of managing the portfolio is described. The elements managed in the portfolio of a Web application, i.e. the units of evolution, are discussed in Section 3. Finally, in Section 4, our idea is summarized and an outlook on further research in the topic "WebComposition/EMS" is given.

2 Treating Web applications as Portfolios

A Web application is a superset of all its units which, in turn, are supersets of the demands they fulfill. By regarding the functionality being provided by such a unit similar to a stock, a value could be assigned to it. This value is limited to a specific point in time, i.e. it is time-dependent. Therefore, the values of units can be seen similar to stock prices. When adopting this picture to the entire Web application, a Web application's value could be determined as the sum of the values of its units. In the life cycle of a distributed Web system, this value would rise or fall depending on the progression of the values of all its units. Many stocks of different types might be hosted and managed under one product umbrella. So, the entire application could be imagined as a stock portfolio, a portfolio of units which make certain functionalities available.

A portfolio is defined in business as a collection of assets and investments all owned by the same individual or organization [3][4]. When a Web application is regarded as a portfolio, its assets – the stocks – are its functionalities. To manage the portfolio by analyzing, selecting, monitoring, and measuring the performance of assets that have been placed together, we have at first to find out what investments are currently available [5]. Applying this management aspect to our case, then we have to investigate which components and services are provided by a Web application.

Functionalities have been realized to fulfill certain demands. These requirements and expectations are formulated as user stories. By examining all user stories, all demands, and, therefore, all functionalities can be identified. Usually, the services and components of a Web application accommodate these functionalities and make them available through well-defined interfaces. In our context we call such providers of functionality *units*. To simplify further consideration, we assume that only one functionality is provided per unit. That is, the portfolio is composed of various units representing all available functionalities. To evaluate the quality of the evolution of a distributed Web system, the portfolio and consequently all its units have to be analyzed.

Performing a quantifiable analysis requires the specification of a measure. Therefore, a time-limited value is assigned to the portfolio and to all its units. The value of a portfolio $p(t)$ can be calculated by summing the values of its units of evolution $u_i(t)$ as shown in (1).

$$p(t) = \sum_{i=1}^n u_i(t) \quad (1)$$

Since the fundamental idea of portfolio management is to maximize the group's return, given a certain level of risk, $p(t)$ has to remain constant or grow in the course of time [5]. The same would be optimal for each unit's value $u_i(t)$, but this is rather uncommon. In contrast to it, after the introduction of units – their initial public offering – trends, unbalances, and flows between them might now be recognized and can be analyzed. Raw indications about conflicts between units in form of value movements are perceivable already at the portfolio level. In this manner, the non-beneficial units or units that bar the way for new ones can be identified. The success of newly integrated functionality becomes measurable and comparable.

Resulting from the ongoing analysis of the units of evolution, management consequences on the collection of assets within the portfolio can be drawn. Decisions on the evolution of a Web application will be value-driven, i.e. if a unit is not valuable in various aspects, e.g. usage, need, etc., the investments into it will be reduced or stopped entirely, whereas a beneficial or promising one will be supported and financed further. Management actions will refine the distribution of units within the portfolio to ensure that it is well-positioned. However, to be able to perform such management decisions we need to determine the values of the units of evolution.

3 Units of Evolution

After all units of evolution have been identified and are managed in the portfolio on the basis of a value assigned to them, we describe in this section how such a value $u(t)$ is determined. The value of a unit can be seen as the result of an abstract function f . Parameters of that function are time-dependent values for each dimension of a unit d_i . This means that the function's result is the value of a time-dependent point within an n-dimensional space spanned by vector \vec{D} as shown in (2). The point

will be motion during the evolution of a unit, because of variations within the dimensions. Thus, the calculation result and hence the value assigned to a unit will change, too. Taking the picture on stocks again, the value alternations of units of evolution can be considered as exchange rate fluctuations. In this way trends become visible. But, to get the result of the function we need to know which dimensions have to be regarded.

$$u(t) = f(\vec{D}, t) \quad \vec{D} = (d_1, \dots, d_n) \quad (2)$$

A unit can be described by various attributes, like expensive or seldom used. Most of them are able to be grouped, so that a couple of disjoint sets of attributes can be created. Sets of attributes, which have a direct influence on the value of a unit act as dimensions d_i . So far we have identified these dimensions which have an effect on maintenance and development decisions:

- **Need.** The number and weight of demands that result from user, environmental, and operational influences. The weight could be expressed in terms like “must have”, “nice to have”, etc.
- **Dependency.** Functionalities provided by other services or components a unit depends on. These components or services might be present in the same Web application, but they could also be available in another.
- **Size.** The extent of a unit.
- **Usage.** The frequency and duration of the use of a unit which is directly provided by a Web application or integrated from other sources as done in mashups.
- **Resource Investment.** How much time and man power are spent to maintain a service or component and to develop it further.
- **Risk.** The risk of failure during evolution of a unit, which depends on whether a unit is present in the Web application itself or provided externally.
- **Maintainability.** Grade of ease to service a unit. Functionalities of externally available units have not to be maintained, but their interfaces.
- **Extensibility.** Simplicity to extent a unit by certain functions.
- **Reliability.** Number of issues emerged within a specific period of time.

Although these dimensions are disjoint, they might have a cause-effect-relationship between each other, e.g. if the need of a unit grows its usage might increase, too. A lot of additional aspects could be derived from a unit's dimensions immediately and when regarding the values of dimensions of related units, e.g. the complexity, importance, and overall costs of a unit. By setting values of particular dimensions or aspects of a unit in relation to other components and services a fine-grained comparison and analysis could be performed. In this way, dimension- and aspect-specific statements can be made about units of evolution.

For such an analysis a couple of variables have to be present. To get the values of each dimension they have to be retrieved from reliable sources. In some cases these sources might not be directly accessible, e.g. risk or reliability data. As a consequence, some measurements have to be performed. Due to the fact that the dimensions are different more than one measurement method is required, e.g. for usage and

dependency. Unfortunately, the requirements to perform successful measurements of dimensions are not fulfilled in all cases. That is, for instance, to perform a usage measurement a special infrastructure for data capturing and recording must be provided.

After all relevant target variables have been captured; they can be used as input parameters of f to determine the general value of a unit. Within this function weights have to be assigned to each dimension and relationships between dimensions have to be mapped. The result of the function should deliver a quantifiable indication of the unit's value with regard to other units within the composition of a Web application.

4 Summary and Outlook

An agile approach to drive the evolution of a Web application through managing it as a portfolio of its services and components has been presented. By assigning values to each of them by means of an abstract function over their dimensions, we were able to distinguish profitable units from unprofitable ones. With a continuous analysis of the entire portfolio we could verify whether or not the current distribution of its units and the resources allocated to them for their evolution make economic sense. If necessary, we would be in the position to adjust resource investments immediately.

There is a big need for further research to substantiate this approach and prove it in practice. Therefore, an evolution management system – EMS – is currently realized as part of the WebComposition approach. The suitability of methods and formulas of the portfolio management on the topic of Web engineering will be tested to verify the WebComposition/EMS approach. In this context, further investigations are directed to get an impression on what units are present and how to determine their precise value and hence the value of the entire application. Techniques to measure the value of each unit's dimension will be assessed, applied and possibly developed.

References

1. Gaedke, M., Turowski, K.: Specification of Components Based on the WebComposition Component Model, IRMA International Conference, Toronto (2001)
2. Gaedke, M., Gräf, G.: Development and Evolution of Web-Applications using the WebComposition Process Model. pp. 9--10. WWW9-WebE, Amsterdam (2000)
3. InvestorWords: Portfolio Definition, <http://www.investorwords.com/3741/portfolio.html>
4. Krebs, J.: Agile Portfolio Management, pp. 61--62. Microsoft Press, Redmond (2009)
5. CreditQuest: Portfolio Management, <http://www.creditquest.com/harland-commercial-lending-resource-library/operation-risk-default-probability-harland>