



*Ph.D. in Electronic and Computer Engineering
Dept. of Electrical and Electronic Engineering
University of Cagliari*



Web Framework Points: an Effort Estimation Methodology for Web Application Development

Erika Corona

Advisor: Prof. Michele Marchesi

Curriculum: ING-INF/05 SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

XXV Cycle
April 2013



*Ph.D. in Electronic and Computer Engineering
Dept. of Electrical and Electronic Engineering
University of Cagliari*



Web Framework Points: an Effort Estimation Methodology for Web Application Development

Erika Corona

Advisor: Prof. Michele Marchesi

Curriculum: ING-INF/05 SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

XXV Cycle
April 2013

Dedicated to my family

Abstract

Software effort estimation is one of the most critical components of a successful software project: “*Completing the project on time and within budget*” is the classic challenge for all project managers. However, predictions made by project managers about their project are often inexact: software projects need, on average, 30-40% more effort than estimated. Research on software development effort and cost estimation has been abundant and diversified since the end of the Seventies. The topic is still very much alive, as shown by the numerous works existing in the literature.

During these three years of research activity, I had the opportunity to go into the knowledge and to experiment some of the main software effort estimation methodologies existing in literature. In particular, I focused my research on Web effort estimation. As stated by many authors, the existing models for classic software applications are not well suited to measure the effort of Web applications, that unfortunately are not exempt from cost and time overruns, as traditional software projects.

Initially, I compared the effectiveness of Albrecht’s classic Function Points (FP) and Reifer’s Web Objects (WO) metrics in estimating development effort for Web applications, in the context of an Italian software company. I tested these metrics on a dataset made of 24 projects provided by the software company between 2003 and 2010. I compared the estimate data with the real effort of each project completely developed, using the MRE (Magnitude of Relative Error) method. The experimental results showed a high error in estimates when using WO metric, which proved to be more effective than the FP metric in only two occurrences. In the context of this first work, it appeared evident that effort estimation depends not only on functional size measures, but other factors had to be considered, such as model accuracy and other challenges specific to Web applications; though the former represent the input that influences most the final results. For this reason, I revised the WO methodology, creating the RWO methodology. I applied this methodology to the same dataset of projects, comparing the results to those gathered by applying the FP and WO methods. The experimental results showed that the RWO method reached effort prediction results that are comparable to – and in 4 cases even better than – the FP method.

Motivated by the dominant use of Content Management Framework (CMF) in Web application development and the inadequacy of the RWO method when used with the latest Web application development tools, I finally chose to focus my research on the study of a new Web effort estimation methodology for Web applications developed with a CMF. I proposed a new methodology for effort estimation: the *Web CMF Objects* one. In this methodology, new key elements for analysis and planning were identified; they allow to define every important step in the development of a Web application using a CMF. Following the RWO method approach, the estimated effort of a Web project stems from the sum of all elements, each of them weighted with its own complexity. I tested the whole methodology on 9 projects provided by three different Italian software companies, comparing the value of the effort estimate to the actual, final effort of each project, in man-days. I then compared the effort estimate both with values obtained from the Web CMF Objects methodology and with those obtained from the respective effort estimation methodologies of the three companies, getting excellent results: a value of $Pred(0.25)$ equal to 100% for the Web CMF Objects methodology.

Recently, I completed the presentation and assessment of Web CMF Objects methodology, upgrading the cost model for the calculation of effort estimation. I named it again **Web Framework Points methodology**. I tested the updated methodology on 19 projects

provided by three software companies, getting good results: a value of $Pred(0.25)$ equal to 79%.

The aim of my research is to contribute to reducing the estimation error in software development projects developed through Content Management Frameworks, with the purpose to make the **Web Framework Points methodology** a useful tool for software companies.

Contents

1	Introduction	1
1.1	Thesis overview	2
2	Estimation Methods in Software Engineering	5
2.1	Software Engineering Experimentation	5
2.2	Psychology of prediction process	9
2.3	Expert-based vs formal model	10
2.4	Effectiveness of forecasting	11
2.5	Cost Model Overview	11
3	A Revised Web Objects Method to Estimate Web Application Development Effort	13
3.1	The proposed approach: RWO	14
3.2	Experimental Results	17
3.2.1	Dataset	18
3.2.2	Effort Prediction and Evaluation Method	18
3.2.3	Results	19
3.3	Conclusions	19
4	Adoption and use of Open-Source CMF in Italy	21
4.1	Research method and gathered data	21
4.2	Data analysis and results	22
4.3	Summary of the results	26
5	The Web Framework Points Methodology	29
5.1	Content Management Framework	29
5.2	The Web Framework Points estimation approach	30
5.3	Size Estimation of a Web Application	32
5.3.1	General Elements	32
5.3.2	Specific Functionalities	33
5.4	Complexity Degree	35
5.5	Cost Model	35
5.6	Calculation of the Estimation	36
6	Experimental Results	39
6.1	Dataset	39
6.2	Effort Prediction and Evaluation Method	41

6.3	Results	41
7	Web Framework Points Tool	45
7.1	Technology used	45
7.2	Functioning	46
7.3	Architecture	47
7.4	Home Page	48
7.5	Add Project form	49
7.6	Activities required to implement the project	50
7.7	Project DB	52
8	Validity Analysis	55
8.1	Internal validity	55
8.2	External validity	56
8.3	Construct validity	56
8.4	Conclusion validity	57
8.5	How to reduce threats to validity identified	57
9	Conclusions	59
	Bibliography	61

List of Figures

2.1	Three-steps approach according to Juristo and Moreno [18].	7
2.2	Components of theories and experiments scheme, according to Hannay J. E. et al.[17].	8
2.3	Variables in experiments	8
4.1	Kind of belonging company	22
4.2	Number of employees	23
4.3	Number of developers	23
4.4	CMF usually adopted	23
4.5	CMF used	24
4.6	Adoption of the library percentage	24
4.7	Library editing percentage	25
4.8	Modules editing percentage	25
4.9	Necessary time for editing	26
4.10	Ready modules buying percentage	26
5.1	Web Framework Points methodology scheme	31
6.1	WFP qualitative judgment	42
6.2	Companies qualitative judgment	43
7.1	MVC Pattern	46
7.2	WFP architecture	47
7.3	Registration Form	48
7.4	Add Project Form	49
7.5	Single Elements	50
7.6	Multiple Elements	51
7.7	Project DB	52
7.8	Copy Form	53
7.9	A project report	54

List of Tables

2.1	Cost Drivers of COCOMO II model	12
3.1	Taxonomy of the RWO model	17
3.2	Web projects descriptive statistics (pers./hours)	18
3.3	MRE values	19
5.1	Software Companies Main Features	36
5.2	Cost Model of the Team A	37
6.1	Software Companies estimation methods and technologies used	40
6.2	Effort Estimate on 19 Datasets Pertaining to Real Projects	42
6.3	MRE Statistics of WFP methodology	42
6.4	Companies Effort Estimate on 19 Datasets Pertaining to Real Projects	43
6.5	MRE Statistics of Companies Effort Estimates.	43
7.1	Technology used in the WFP application	46

Chapter 1

Introduction

Nowadays, Web sites and Web portals are more and more complex, and have to manage and convey to their visitors huge amounts of information. When developing these applications, programmers typically use a Content Management Framework (CMF), a software that provides most of what is needed to develop Web applications, and that is easily extensible through proper add-ons and plugins ¹.

There are several CMFs, like the open source CMFs Joomla! [1], Drupal [2], and WordPress [3], that have been created to help the management of those large amounts of content and to develop both simple and complex Web applications. Because of the capability of handling and editing heterogeneous data sources, an increasing number of organizations and corporations turned to CMFs to fulfil their need to publish data and provide services – such as business intelligence, GIS, e-Business – in their websites and portals.

Unfortunately, developing Web applications through CMFs is not exempt from cost and time overruns, as in traditional software projects. Estimation is one of the most critical components of a successful software project: “*Completing the project on time and within budget*” is the classic challenge for all project managers [4]. However, predictions made by project managers about their project are often inexact: software projects need, on average, 30-40% more effort ² than estimated [5].

In spite of the many estimation models available, currently there is no model able to adequately measure the effort of a Web application [6, 7, 8, 9, 10]. For this reason, my research has been focused on the study of a new methodology for estimating the effort of Web applications developed with a CMF.

I concerned myself with effort estimation for Web applications three times, in 2011 [11] and in 2012 [12, 13]. In my 2011 paper [11], I compared the effectiveness of Albrecht’s classic Function Points (FP) metric [14] and Reifer’s Web Objects (WO) one [15] in estimating development effort for Web applications. I tested these metrics on a dataset made of 24 projects provided by a software company between 2003 and 2010. The experimental results showed a high error in estimates when using WO metric, which proved to be more effective than FP metric in only two occurrences. However, neither of the metrics passed Conte’s criterion [16] of having at least 75% of the estimates with an error less than or equal to 25%, although the FP metric was the closest to its satisfaction. In the context of this first work, it appeared evi-

¹For more details, see Chapter 4

²Effort= resources, time and cost required to develop a software project

dent that effort estimation depends not only on functional size measures, but other factors had to be considered, such as model accuracy and other challenges specific to Web applications, though the former represent the input that influences most the final results. For this reason, I revised the WO methodology, creating the RWO model. This model estimates the effort required to develop a Web project in terms of man-days, using a combination of two metrics: Albrecht's classic FP metric and Reifer's WO metric. I applied the RWO method to the same dataset made of 24 projects, comparing the results to those gathered by applying FP and WO methods. The experimental results showed that the RWO method reached effort prediction results that are comparable to – and in 4 cases even better than – the FP method.

The reason for proposing – in 2012 [12] – a new methodology for size estimation was to counteract the inadequacy of the RWO method when used with the latest Web application development tools. The size metric used in the RWO method was found not to be well suited for Web applications developed through a CMF. In particular, operands and operators used in Reifer's metric relate to elements that can be quickly and easily created by modern programming technologies, and whose weight appears to be irrelevant in terms of size calculation for a Web project. I identified new key elements for analysis and planning, allowing for the definition of every important step in the development of a Web application using a CMF. Each considered element contribute to the size estimation through its different degree of complexity. I tested the size estimation ability of my methodology on 7 projects provided by the same company of the previous studies. I compared the value of the size estimate yielded using original requirements to the final size of each project, as measured on the developed Web application, with very low MRE³ values on estimated sizes: the Web CMF Objects methodology has a value of $Pred(0.25)$ equal to 85.7%, so it satisfies the acceptance criterion by Conte et al. [16].

Recently, I completed the presentation and assessment of my methodology [13], suggesting a new cost model for the calculation of effort estimation. I tested the whole methodology on 9 projects provided by three different Italian software companies, comparing the value of the effort estimate to the actual, final effort of each project, in man-days. I then compared the effort estimate both with values obtained from the Web CMF Objects methodology and with those obtained from the respective effort estimation methodologies of the three companies, getting excellent results: a value of $Pred(0.25)$ equal to 100% for the Web CMF Objects methodology.

1.1 Thesis overview

This thesis is organized as follows:

- Chapter 2 presents an overview of main software estimation methods used in software engineering and some experiments conducted in empirical software engineering. These experiments analyse the influence of psychological factors of the development team about the effectiveness of forecasting of used effort estimate methods. The usefulness of experimentations and the approaches to the empirical research will be also introduced.
- In Chapter 3, the Revised Web Objects (RWO) methodology is described. This is a Web application effort estimation methodology, based on a reinterpretation of the WO

³For more details, see Section 6.2

Reifer's methodology. The study and the experimental validation of this methodology can be considered as the preliminary work of the study and the realization of the Web Framework Points (WFP) methodology, the most important part of my thesis. The RWO approach and the results of the experiments performed applying the method will be here described.

- Chapter 4 presents the survey about the adoption and use of Open-Source CMF in Italy. The RWO methodology highlighted the need of a methodology more strictly bounded to the company context and more adaptable to different kinds of projects and technologies used by developers. For this reason, I believed it was appropriate to identify what were the last technology trends, before starting a new methodology experimentation. For lack of this kind of research in literature, I set up a survey with the main questions of interest of my research. The main objective of the survey was to detect what were the most used frameworks for the development of Web applications and also how the development methodology of these applications had recently evolved.
- Chapter 5 proposes the Web Framework Points methodology, the most important part of my thesis. The proposed methodology is meant for Web applications developed with CMFs, regardless of the specific technology used to implement the frameworks. This methodology includes three forecasting models (expert-based, analogy-based and regression-based) within a basic mathematical model, in order to improve the accuracy of prediction. The methodology is made in a way that is as far as possible free from anchor-effect.
- Chapter 6 describes the results of the experiments performed applying the Web Framework Points methodology to obtain its validation. The WFP methodology has been the subject of experimentation on real projects developed by Italian software companies. I evaluated the effectiveness of the methodology in predicting the effort of the analyzed applications through the calculation of the MRE (Magnitude of Relative Error) factor for each project. The WFP methodology has a value of $Pred(0.25)$ equal to 79%, so it fully satisfies Conte's criterion. This means, for my methodology, a good estimation power regarding the effort needed to build an application.
- In Chapter 7 the main features of the interactive application implementing the Web Framework Points methodology will be presented. I used this application both to collect projects data from the companies and to estimate the effort.
- Chapter 8 explains the four types of validity that contribute to judge the overall validity of a research, i.e. internal, construct, external and conclusion validity. Possible threats to validity regarding the obtained experimental results of Web Framework Points methodology will be identified.
- Chapter 9 presents the conclusions and plans for future work.

Chapter 2

Estimation Methods in Software Engineering

This Chapter presents an overview of main software estimation methods used in software engineering and some experiments conducted in empirical software engineering.

2.1 Software Engineering Experimentation

Software engineering is a young and practical discipline, that needs observation, measurement, theories, methodologies and field testing organized in a systematic manner; in one word, it can be considered an empirical discipline. Software engineering researchers are looking for theories able to summarize phenomena observed in the field and/or analysed through experiments, in terms of basic concept. Theories help to communicate and share ideas and knowledge, besides giving both researchers and practitioners the possibility to implement new concepts in different contexts, such as the industrial one¹.

On the other hand, experiments conducted in empirical software engineering have the purpose to validate a theory. They have also the purpose to compare different technologies, methodologies and theories to measure their effects on software development, in a scientific way. Despite the proven usefulness of experimentations, “*very few ideas in Software Engineering are matched with empirical data*”, as highlighted by Juristo and Moreno [18]. For example, important theories such as functional programming, object-oriented programming or formal methods have never been empirically demonstrated [18]. Experiments should always be performed, because they provide evidence for – or against – a particular approach or technique, giving software engineers real benefits of existing theories. According to Juristo and Moreno, the lack of experimentation in software engineering may be due to several reasons; some of them are reported as follows:

- a belief that the traditional scientific methods are not applicable;
- a belief that experimental validation is expensive;

¹For a systematic review of the use of theories in software engineering experiments see, for example, ref. [17]

- a lack of experimental design and analysis books for software engineering;
- a belief that empirical studies conducted to validate the ideas of others researchers are not publishable;
- a lack of understanding between software developers and software engineers, due to the belief that experimental validation could slow down the work;
- quick changes in technologies;
- very large number of variables involved in software development;
- difficulty in identifying the better approach among the tested techniques;
- human factor: the same experiment can yield different results, depending on people involved;

In software engineering, it is possible to identify two complementary approaches to the empirical research, i.e. quantitative and qualitative. Qualitative research can be used to formulate hypotheses and set up different variables involved and, then quantitative research will be used to establish numerical relationships among these variables. Juristo and Moreno suggest a three-steps approach that should be done to test a new idea in software engineering (see Fig. 2.1):

1. **Laboratory experiments:** researchers verify their assumptions under controlled conditions, in order to publish a new theory, experiments and related benefits of their theory. Original experiments will then be replicated by other researchers, in order to verify the same theory and to publish new results.
2. **Quasi-experiments:** the original theory proposed by researchers is implemented by innovative developers in experimental projects, in order to verify real benefits and/or identify possible issues. Developers will then publish their results.
3. **Surveys:** the original theory is implemented in real projects by routine developers, who take associated risks. Some of the results obtained will be published by routine developers, in order to spread the innovations.

Once this approach is followed, the community is more willing to accept a new theory. First of all because it is proved it works in laboratory, subsequently because the results of the various experiments guarantee that it also works in different contexts.

The human factor is one important aspect, that has not to be overlooked when proposing and validating a new idea. In software engineering it is not possible to apply deterministic theories; it is necessary to consider, instead, the social context and the relationships among the people involved.

A software process is too complex to be represented with mechanistic or theoretical models; for this reason, an empirical model turns out to be more appropriate. The equations used in estimation models are an example of empirical model, where parameters value are obtained analysing a series of projects:

$$Effort = a \text{ size}^b \quad (2.1)$$

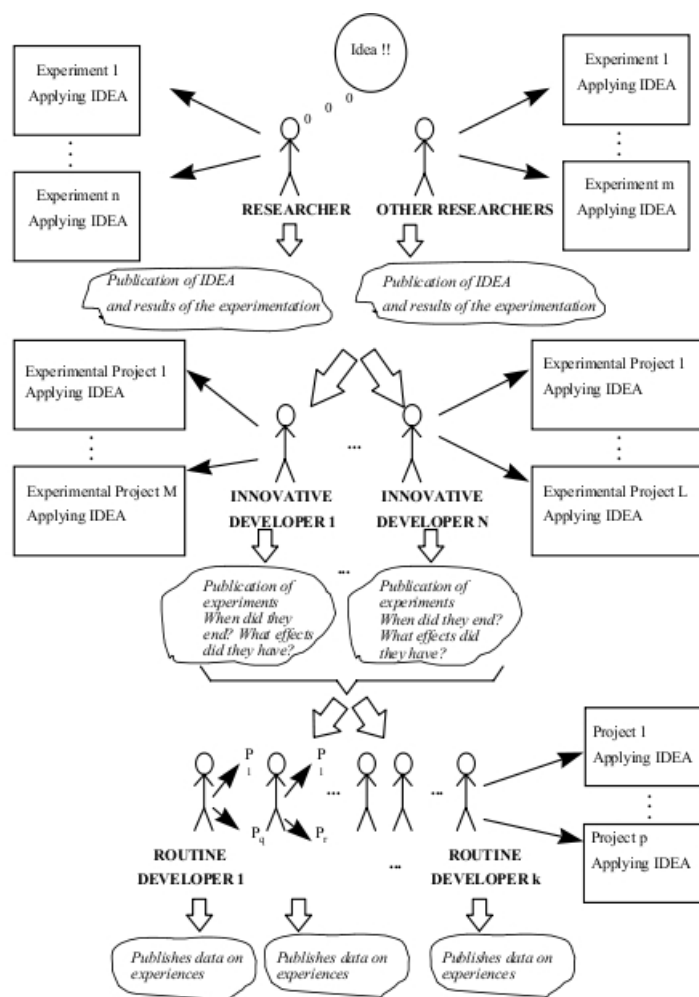


Figure 2.1: Three-steps approach according to Juristo and Moreno [18].

Hannay J. E. et al. [17] state that software estimation models may be viewed as prediction theories; i.e. theories that predict without providing explanations. They also provide a detailed description of components of theories and related experiments, proposing the scheme of Fig. 2.2.

As is can be seen by the figure, they divide the domain of experiment methodology in two levels: conceptual and operational. The conceptual level includes concepts and theories, whereas the operational level includes observations, measurements, and experiments. In software engineering experiments, variables are the following:

- **conceptual level:** actors (such as experts, project teams, software developers, etc.), software process activity and software development technology (such as design using UML, validation using functional testing, etc.), software system (such as safety critical systems, object-oriented artifact, etc.), relevant scope (such as software industry), causes (familiarity of design patterns, perspective-based reading, etc.) and effects (such as the concepts of software quality, developer performance or reliability);
- **operational level:** tasks, materials, treatments, outcomes, and experimental settings.

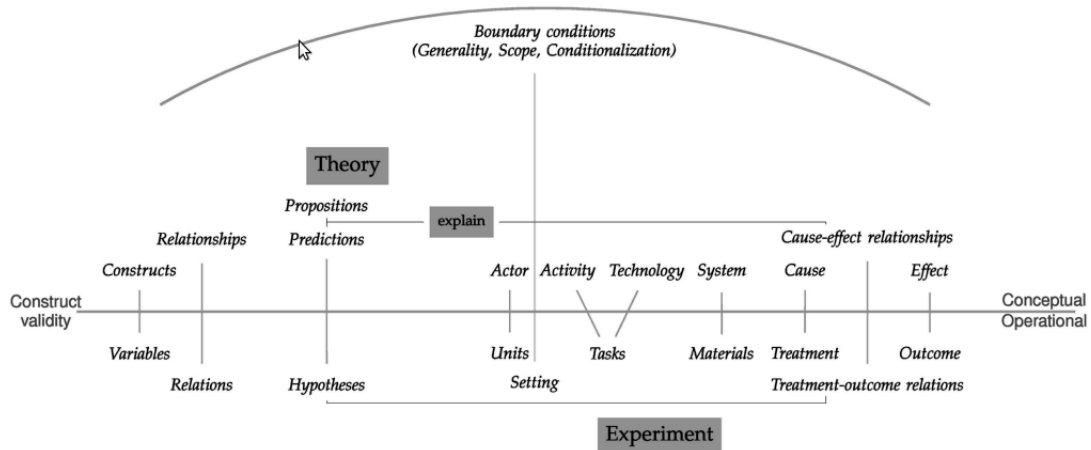


Figure 2.2: Components of theories and experiments scheme, according to Hannay J. E. et al.[17].

Experiments conducted in empirical software engineering have the purpose to investigate the relationships between cause and effect, whereas theory have the purpose to seek to explain why and how the cause-effect relationship occurs, in a precise scope. In experiments, causes are described by independent variables, while effects are described by the dependent variables. Other kind of variables, named confounding factors, may also exist, that influence the result of an experiment if added to independent variables, without the knowledge of the researcher (see Fig. 2.3).

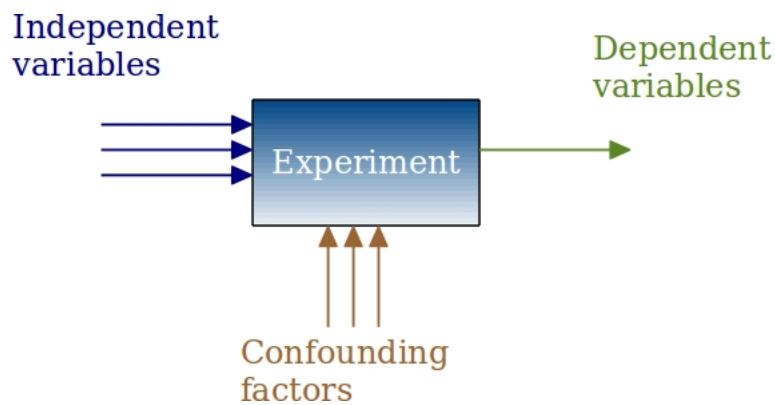


Figure 2.3: Variables in experiments

Juristo and Moreno suggest three level of investigation to identify variables and relationships among them:

1. **Survey inquiries:** the goal of this level is to identify variables that affect the development process (skills, experience, age, nationality of developers, etc.).

2. **Empirical inquiries:** the goal of this level is to extract an empirical model from observations, trying to explain how the variables affect each other, varying the values of variables among different experiments.
3. **Mechanistic inquiries:** the goal of this level is to develop a theoretical model able to explain why the variables influence the results empirically observed.

In the following sections will be shown the main software estimation models and some experiments conducted in empirical software engineering.

2.2 Psychology of prediction process

All effort estimation methodologies implicitly enclose a certain degree of uncertainty due to human judgement. This is more evident in expert-based methodologies and less in the model-based, but when you think about values to be assigned to the parameters set by different methods, their point of weakness is clearer. There are some empirical studies, that I found particular and innovative, which analyse the influence of psychological factors of the development team about the effectiveness of forecasting of used effort estimate methods.

In particular, there is a phenomenon known as anchoring and adjustment, which arises when someone is called to choose under conditions of uncertainty. According to the authors of the study, the phenomenon is especially evident when it is necessary to make quantitative judgements, just as in case of software effort estimates *“If judgement of the matter is difficult, we appear to grasp an anchor, that is, a tentative and possibly unrelated answer to the problem; and adjust such answer up or down according to our intuition or experience to reach the final result”* [19]. The experiment, consisting in estimating the time required to develop a software application, involved computer science graduate students and software developers. All participants, divided into three groups, were provided with documents on the functional requirements of the application to be estimated. The first group was given a 2 months possible value of effort (low-anchor), the second group a 20 months value of effort (high-anchor), while the remaining group was not given any value (no-anchor). The experiment result clearly showed that the effort estimation is always influenced by the anchor value both low and high, regardless of the used estimation method by the different people involved [19].

Other authors investigated about the phenomenon, showing, through an empirical study, that customer expectations can have an influence on effort estimates carried out by experts, acting as anchor-effect. For example, it may happen that, in situations of high uncertainty, as in the early stages of a project, the expert makes too optimistic prediction of effort. As the project carries on, since the client considers valid the initial estimates, the expert will have doubts in making a realistic estimate, perhaps with higher values, in order not to disappoint the expectations of the customer itself [20]. Although the results I have shown are based on empirical studies, they may partly explain why the industrial software projects are often underestimated [21].

Finally, it has been demonstrated the existence of a related, but opposite, event to those described above, however always due to a psychological phenomenon; basically, the value of estimated effort is what influences the course of the project. This phenomenon has been observed in an experimental study, in which computer science students have been involved. The experiment showed that, in case of estimates made in the early stages of the project

(when information available is still few) these can affect estimates made later, in the presence of more details and information. Furthermore, it was observed that these can also influence the course of the project in terms of quality, in case of too optimistic estimates [22].

2.3 Expert-based vs formal model

According to some empirical studies, the preferred approach for estimating the effort of software projects is the expert judgement. This could depend on the fact that formal methods are usually more complex and less flexible than the expert-based ones [21]. Some project managers choose, instead, to combine them together, which moreover leads to improve forecast accuracy, as discussed in Section 2.4 [22, 23].

It is fair someone wonders what is the best approach to follow, so it is interesting to know the views of two experts on this topic: Magne Jørgensen and Barry Boehm, which is shown below.

According to Magne Jørgensen, supporter of the expert-based approach, a formal model is unable to capture specific aspects of projects, such as the working way of the team involved in the development. He states that the expert-based approach is to be preferred in situations where there is very specific information not covered by formal models. Jørgensen also states that project managers often officially use formal methods, but in practice they turn to expert judgement. This could be due to the fact that the use of a formal method implies greater use of time, both to collect a greater amount of data and to learn how to use and calibrate it properly. A further consideration of Jørgensen regards the objectivity of formal models: it is widely assumed that they are more objective than the experts, and that they are not subject to pressure from the client or optimistic/pessimistic judgements. Anyway we have not to overlook the fact that the model inputs are always evaluated by experts, which can then affect that objectivity [22].

On the other hand, Barry Bohem states that, although formal models are not able to produce perfect estimates and are not appropriate to every circumstance, they are able to directing the project manager to the analysis of all the elements that lead to increase or to decrease the costs (cost-drivers), providing a quantitative and objective judgement on them. Formal models are created as a result of the analysis of many real-world projects and calibrated thanks to the feedback of those who used them.

Finally, Boehm says that both approaches are useful and complementary. As he observed, the organizations that get the best results in terms of prediction are those that use just a mix of parametric models and expert judgement. These good results are achieved because these organizations preserve documents and real estimates of the projects, using these data to correct and calibrate the inputs to the models for future estimates [22].

I conclude this brief comparison by highlighting a further “complication“ of formal models: the accuracy of the data. Data on software projects are at the base of the construction and validation of a good estimation model. As it is known, collecting data of sufficient quantity and quality is wasteful from the time point of view, and data themselves are not always easy to find. Not all organizations are willing to supply them, and the few that do it rely the task of collecting them on developers, which in turn do not always keep track of their own work in an accurate and systematic way [24]. All this is compounded by the fact that often organizations willing to collaborate are the ones who get the best results in productivity, with

evident consequences for the calibration of models [22].

2.4 Effectiveness of forecasting

In the field on forecasting studies it is known that using a combination of various forecasting methods, rather than a single one, improves their accuracy: *"combining can reduce errors arising from faulty assumptions, bias, or mistakes in data"* [23]. The combining forecasts approach consists in using the average of more independent forecasts. J.S. Armstrong, after having reviewed several empirical studies, suggests a procedure to follow for those who wish to use this approach. This procedure, among other things, consists in analysing different data and/or using at least five different forecasting methods when possible; in this way, information used in the combination will be valid and independent. He also states that *"combining forecasts is especially useful when you are uncertain about the situation, uncertain about which method is most accurate"* [23]. In addition, regarding the cases in which you want to use expert opinion, is preferable a combination of forecasts made by many experts rather than a single opinion made by the maximum expert on the matter. Finally, from the empirical studies that he examined it was found that using combining forecast approach yields usually more accurate forecasting. Under ideal conditions (high uncertainty and using multiple forecasting methods) it is possible to obtain a reduction of error forecasting equal to more than 20%.

Kocaguneli et al. also came to the same conclusion: they state that using a combination of more methods will yield more accurate estimations than using one method at a time [25].

2.5 Cost Model Overview

Research on software development effort and cost estimation has been abundant and diversified since the end of the Seventies [14, 16, 26]. The topic is still very much alive, as shown by the numerous works existing in the literature. Researchers have extensively investigated the topic, in relation to both estimation approach (regression, analogy, expert judgment, function points, simulation, etc.) and research approach (theory, survey, experiment, case study, simulation, etc.). These studies were carried out in both industrial and academic contexts. The most frequently used estimation approach is regression-based, where the CO-COMO model is the most used model [26].

As regards functional size measurement methods, they measure software size in terms of user-required functionalities. The first functional point method was created by Albrecht in 1979 [14], whereas the most modern is the COSMIC method, created by an international consortium of metrics experts [27]. For a comparison of the most widely used function points methods see, for example, ref.[27]. Finally, according to several empirical studies, expert judgment is the preferred estimation approach [5]. A recent review of empirical studies on accuracy of software development effort estimates found that, on average, accuracy of expert-based effort estimates was higher than the model-based one [28].

With regard to the validation of estimation methods, the dominant research approach is based on the use of historical data. Moreover, the context most research applies to is the industrial one [29].

Narrowing down the topic to Web applications, one of the first researchers to introduce size metrics to measure Web applications was Tim Bray [30], through statistical analysis on the basic characteristics of the main Web sites in 1995. Size metrics were proposed by Cowderoy et al.[31]. At the beginning, the models used for Web effort estimation were the same as the ones used for general software applications. One of the first researchers to introduce a method specifically devised for the Web was Reifer, through WO metric and the WEBMO model [15]. This model was later used by other researchers to perform comparisons among different estimation models, but with varying results, sometimes dissimilar from each other [6, 32, 33, 34]. Many research works on *Web effort estimation* were also carried out by Mendes and collaborators [7, 8]. Works devoted to estimate development effort in CMS projects are fewer: for example, we may quote a paper by Aggarwal et al., where the linear regression estimation model CMSEEM is proposed [9].

In general, project effort estimation models are based on cost models that consider as input a set of parameters – named cost drivers – size being the predominant one [26]. As we seen in Section 2.1, the general formula of an algorithmic effort estimation model can be expressed as:

$$Effort = a Size^b * adjustment\ factor \quad (2.2)$$

The cost drivers concerning the COCOMO II model are shown in Table 2.1, by way of example.

In ending this section, we want to underline that the existing models for classic software applications are not well suited to Web application development, as stated by many authors also in regards to CMS-based projects, so software project estimation remains a key challenge to researchers [6, 7, 8, 9, 10].

Table 2.1: Cost Drivers of COCOMO II model

Cost Drivers COCOMO II	
Product attributes	Required software reliability Size of application database Complexity of the product
Hardware attributes	Run-time performance constraints Memory constraints Volatility of the virtual machine environment Required turnabout time
Personnel attributes	Analyst capability Applications experience Software engineer capability Virtual machine experience Programming language experience
Project attributes	Application of software engineering methods Use of software tools Required development schedule

Chapter 3

A Revised Web Objects Method to Estimate Web Application Development Effort

In this Chapter the Revised Web Objects (RWO) methodology is presented. The study and the experimental validation of this methodology can be considered as the preliminary work of the study and the realization of the Web Framework Points (WFP) methodology, the most important part of my thesis.

The Revised Web Objects (RWO) methodology is a Web application effort estimation methodology based on a reinterpretation of the WO Reifer's methodology [15]. The study of the RWO methodology was inspired by a Barabino et al. work [32]. In their work, the effectiveness of Albrecht's classic Function Points (FP) [14] metric and Reifer's Web object (WO) [15] metric in estimating development effort has been compared. Following the same procedure, we apply both methods to a dataset of 24 projects of an Italian software company, in order to determine which one was more effective in estimating the effort. Since the beginning we realized that both methodologies were not easily applicable to the considered projects, because they were Web applications developed with the latest technologies. To overcome this gap, we then decided to revisit the WO methodology, in order to include new technological advancements regarding the Web application development.

Revised Web Objects is a mixed model, conciliating both the characteristics of empirical methods (i.e. the use of previous experiences in effort estimation), and algorithmic and statistical measurements. Our approach considers different weights, specifically tailored for Web applications. It starts with a preliminary categorization of the considered project, according to a web application taxonomy designed on the basis of interaction characteristics, scope of the application, dimension of the project and tools to be used to develop the solution.

The comparison among classical Function Points methods, Web Objects (WO) and RWO demonstrates the best performance of RWO in Web oriented applications.

In next section we describe our approach, in Section 3.2 we will discuss the results of the experiments performed applying our method to obtain its validation.

3.1 The proposed approach: RWO

As said before, we devised the new RWO method, that takes into account the classical parameters of WO recomputing the original indicators and, when we deem they have become obsolete due to new advances in the technology, incorporates our practical experience in effort estimation.

Of course, it is usually necessary to tune the proposed RWO method with respect to a productivity coefficient that depends on the adopted technology and, consequently, on the experience of the company performing specific projects. In this way, the proposed approach does not exclude the human factor, which is obviously unpredictable, but is based on the developers' experience and skills, and thus becomes a mixed approach.

Following the original WO indications, the elements we considered in RWO are divided in operands and operators, defined as following:

- operands: the elements themselves
- operators: the operations we can perform on the operands

Actually, in various counting examples (particularly in the White Paper describing the official counting conventions [35]), Reifer himself does not use this equation, but he just sums operands and operators, each weighted by a number that depends on the complexity of the considered item. We use the same approach for the four kinds of operands introduced by Web Objects, in the followings described with related operators and complexity weights for "Low, Medium, High" grades, reported inside the parenthesis after the name of the element, in the same order.

In the original definition, Multimedia Files (complexity Low or Medium, depending on kind of multimedia files) are dimension predictors developed to evaluate the effort required to integrate audio, video and images in applications. They are used to evaluate the effort related to the multimedia side of a web page.

In this category we can include: images, audio, video, texts. In this case, the image considered are those related to the content of a website (for example the photos or thumbnails in a photo - gallery), not the images present in the interface (icons). Audio and video are multimedia files that can be downloaded or interactively played by the users. Also in this case, audio or video files present in the interface are not considered as multimedia files. The text eligible to be considered as multimedia file is not the text present in a web page, but text files, for instance in .pdf, .doc, .odt, and other formats. Also, texts or files generated by a script (for example a form that, when compiled, generates a .pdf file as a result) are not to be considered in this category.

We redefined the original metric guidelines, in some cases already obsolete, to better fit the actual characteristics of current web applications. We upgrade the considered elements as follows:

- images:
 - generic, static format: Low
- animated images (for example, animated GIF): Low or Medium
- audio/video:
 - common A/V formats (for example MP3, AVI, Flash): Medium

streaming A/V: High

- text:

for all formats: Low

Concerning typical operators for multimedia files, we considered the following categories and weights:

- start/stop/forward for A/V files: Low or negligible
- operations on interactive elements (for example, a search on a map): Low or Medium

Web Building Blocks (complexity generally Low or Medium in some cases, depending on kind of blocks) are dimension predictors used to evaluate the effort required in the development of all the components of a page of the application in the original WO. Standard libraries (such as Windows components or graphical libraries in Java) are not considered since they are part of their own environment. Our definition considers, instead, active elements such as ActiveX, applets, agents and so on, static elements like COM, DCOM, OLE, etc., and reusable elements such as shopping carts, buttons, logos and so on. All the elements recurring on more than one page are counted just once (an example is given by the buttons performing the same operation).

We consider:

- Buttons and icons, both customized widget and static images, with the activation as the only associated operator (Low)
- Pop-up menus and tree-buttons have to be considered twice: the first time as buttons (Web Building Blocks); the second as operators (counting them as many times as the number of their functions). All these operators have a Low complexity.
- Logos, headers and footers are all static elements present in the website interface. This kind of elements are often unknown in the early stage of a project. So, their count depends on the details of the requirement document available. Concerning the complexity, we can typically consider:

Buttons, logos, icons, etc: Low

Applet, widget, etc: Medium or High

Scripts (complexity Low with different levels, depending on kind of scripts) are dimension predictors developed to evaluate the effort required to create the code needed to link data and to execute queries internal to the application; to automatically generate reports; to integrate and execute applications and dynamic content like streaming video, real-time 3D, graphical effects, guided work-flow, batch capture, etc., both for clients and for servers. It is important to clarify the difference between a script and a multimedia file: a script is the code that activates, possibly, a multimedia file.

In our model, this category also includes:

- breadcrumb: information generally present in the top of the page, allowing a quick navigation. For this element we consider a complexity Low-Medium.

- pop-ups
- Internal DB queries: queries internal to the application, with complexity depending on the adopted technology. In fact, Reifer uses the conventions defined by the Software Engineering Institute:
 - html: Low
 - query line: Medium
 - xml: High

In the projects we analyzed, we used a Low weight for DB query when a persistent framework, like Hibernate, was used. In fact, once defined the mapping of the objects in xml language, the query becomes an access to the fields of the objects, highly reducing complexity. Usually, the complexity of these elements is considered Low- Medium.

Links (complexity Low or Medium, depending on kind of links) are dimension predictors developed to evaluate the effort required to link external applications, to dynamically integrate them, or to permanently bind the application to a database.

Links are always present when the application performs queries on databases external to the application itself. Consequently, the code to access data is considered a link. In the analysed projects, the login is always considered as an external link, because the database holding the users' data is external in every case.

Concerning the complexity, Reifer counts the logical, and not the physical, lines of code. In our model, we follow the same approach used for the scripts, considering the complexity depending on the persistence technology adopted.

When evaluating the effort estimation for a web application project, the reported characteristics to be taken into account are typically not enough. In fact, web applications may have very different scopes objectives and interactivity level – from a simple collection of Web pages to a full client - server complex data processing application – and may be developed with very different technologies, characterized by very different productivities. These "environmental" and "basic genre" features must be taken into account for a realistic effort estimation. So, to incorporate this essential element influencing effort evaluation, in the early stage of the design of a web application, we also need to identify the kind of application to be developed. To this purpose, we incorporated in RWO method also a preliminary evaluation of the kind of the project. In this way, the guidelines for calculating the development effort can account for different parameters resulting from the technologies used for the development of the web application, and from the development language chosen.

Thus, the additional information we consider is the classification of each project. One of the aims of this experimentation is to confirm the general validity of the methods for different kinds of projects. Our categorization is made on the basis of three features:

- size (in terms of FP/RWO);
- level of reuse of other applications;
- productivity of the tools used.

The size is the estimation performed in terms of basic RWO measures, allowing to have a first, rough indication of the effort needed to complete the project.

The level of reuse is used to evaluate how many software component can be taken from previous projects, minimizing the development effort.

Concerning the productivity, this is a fundamental element completing the taxonomy and adopted by the company after accurate validation. Summarizing, projects are classified following the indications shown in Table 3.1.

Once a project is classified, specific weights are used to obtain the estimated effort from the computed basic RWO measures.

Table 3.1: Taxonomy of the RWO model

Acronym	Description	Features (programming language, typology, architecture)
SSP	Standard Software Project	Java, No framework, No RAD
SRP	Standard RAD Project	The skeleton of the application is developed using a RAD tool, while its detailed interface and business code are coded by hand. This category needs additional studies.
ERP	Extreme RAD Project	The application is developed using a tool that does not require particular programming skills, and no a priori knowledge, except for the ER model, constraints and validation rules. In some cases, a workflow model (static and dynamic) is needed. The RAD tool creates the database and all connected procedures. It is model-driven. An example of this kind of tools is Portofino1.
Portal	generic portal	Broadvision architecture. Generally, portals are designed for content presentation, so they have a limited or absent data processing

To evaluate our RWO approach, we performed some experiments, described in the following section.

3.2 Experimental Results

The empirical research has been performed in the context of a mid-sized Italian software company. Choosing a narrow sample for our study (projects developed by only one company) might constitute a possible threat to the generality of the results. In this experimental phase, we considered 24 projects, developed by the company, chosen among different kinds, as defined above. In this way, we were able to consider both a larger sample and a variety of cases to which apply our RWO method.

3.2.1 Dataset

The data set is built on 24 projects developed from 2003 to 2010 by the above cited company; this firm develops and maintains a fairly high number of software systems, mainly for local public administrative bodies.

The application domains are those in which the company operates: mainly Public Bodies and Health Services. Among the projects developed by the company, we chose the mentioned 24 ones, focusing our attention on the applications written using web technologies, which are now the most used by the company for developing its projects.

Each project is described by the requirement documentation, and by snapshots of the layout of their web pages. The company already performed the detailed Function Point estimate, allowing us to compare the results with the estimation done with RWO, following the rules detailed in the previous section. Before estimating, each project was first categorized according to the taxonomy described at the end of the previous section, and constituting the early step of RWO methodology.

In our experiments, the classification of each project was used to steer the subsequent phase, when weights are assigned to the required features.

The categorization of the studied projects was made on the basis of :

- the size (in terms of FP/RWO);
- the level of reuse of other applications;
- the productivity of the tools used.

The projects considered for the experiment belong to the cited groups in the same measure, with balanced dimensions and reuse levels. So, we had the same number (six) of SSP, SRP, ERP and Portal projects.

3.2.2 Effort Prediction and Evaluation Method

For each of the 24 projects we evaluated three different estimation metrics (FP, WO and RWO). Table 3.2 shows and compares the descriptive statistics related to effort estimation in person's hours. Note that the output of the three methods are the rescaled in the same way, to get an estimation of the effort, which is then compared with the actual effort declared by the company for each project.

Table 3.2: Web projects descriptive statistics (pers./hours)

Metric	Min	Max	Mean	Median	std dev
System effort in FP	60	777	312	240	236
System effort in WO	67	1342	446	355	347
System effort in RWO	42	851	282	225	220

The sizes in RWO are quite comparable to the sizes in FP. This result is encouraging, because our method, specialized for the evaluation of development effort in Web-base projects, yields results quite close to the more traditional FP method (whose use in the company has

been well established for many years), and apparently with less variability than with WO method. To evaluate the performances of the measures, we calculated the MRE ¹ (Magnitude of Relative Error) for each project. In addition, we also calculated the prediction level.

3.2.3 Results

The results obtained with the selected projects using FP, WO and RWO metrics are shown in Table 3.3. They show that RWO method perform better than, or equal to, FP on many considered projects. Consequently, we can consider RWO an overall valid alternative to FP, surely more tailored to satisfy the needs of effort prediction for a web application.

Table 3.3: MRE values

Metric	Min	Max	Mean	Median	Std dev	Pred(0.25)
FP	0	3.13	0.49	0.19	0.9	62%
WO	0.07	5.93	1.23	0.66	1.64	40%
RWO	0.01	3.05	0.45	0.19	0.85	58%

Note that non-revised WO yields poor results on the considered dataset, while our revised method yields results quite reliable for effort prediction on web application projects data. Remember that having a *Pred(0.25)* greater than 75% (more than 75% of the projects have an MRE less than, or equal to 0.25) denotes an acceptable estimation [16]. If we follow this criterion, both FP and RWO do not give acceptable estimations. However, considering current estimation models, we can affirm that RWO is an acceptable estimation method for the target projects. Apparently, RWO performs similarly to FP and both seem better than WO.

Concerning the apparent similarity between RWO and FP performance, we have to consider that several projects belonging to the considered data set do not have strong, web-specific characteristics. Moreover, there is a significant data dispersion due to the presence of projects developed using RAD technology. In fact, the RWO method appears more reliable compared to FP in the case of complete web applications, being in this case more stable and predictable. We should also consider that RWO is tailored for web applications, and could be further refined following the evolution of web technology, while such a tailoring would be much harder with FP method.

Note that the number of studied projects, even if belonging to various kinds of application, is too low to be definitive about the validity of the proposed RWO method.

3.3 Conclusions

I presented an empirical study of software development effort estimation, performed on a set of industrial projects carried on at an Italian software company. The considered data set includes 24 projects divided in 4 categories, allowing to extend and generalized results to different kinds of Web application projects. The data set is composed, in equal measure, by

¹For more details, see Section 6.2

Standard Software Projects (SSP), Standard RAD projects (SRP), Extreme RAD Project (ERP) and Portals.

The performed experiment compared the estimation power of different methods - Function Points, Web Objects, and my Revised Web Objects method.

All the estimation was done considering a productivity coefficient formulated by the company on the basis of past development experiences. I believe that entirely empirical methods are not efficient enough, because they do not give an objective measurement of the project effort, but depend on a human estimator on the basis of her own previous experience. On the other hand, mixed models take advantage of both algorithmic and empirical methods. In the real world, in fact, the early estimation of a project effort cannot be based only on one of the two aspects. For this reason, I revisited the WO method, adding other parameters designed to provide forecasts based also on human experience, and at the same time specifically formulated for the prediction of effort in developing Web applications.

In the specific context, good results were obtained both with FP and RWO methods. As previously discussed, FP method yielded good results owing to the long experience of the company developers in its use. RWO, on the other hand, was able to yield comparable - and even slightly better - results since its first use. The RWO approach accounts for specific Web application characteristics, and is suitable of further evolution, following Web application technology changes.

Besides performing a comparative analysis of these - and possibly other - effort estimation methods on a larger sample of projects, a further step in the research will consist in developing a tool, based on the proposed RWO model, allowing to perform a predictive effort estimation. The tool will allow to customize productivity parameters, so that the model could evolve following new acquired competencies, new technologies and the different Web applications considered.

Chapter 4

Adoption and use of Open-Source CMF in Italy

Thanks to the partnership, now years old, with the same software company, we could directly observe in the field the evolution of development technologies and methodologies on projects developed over a span of almost 10 years. Recently, we started new partnerships with two small software companies, so we had the opportunity to observe different areas of applicability. This fact allowed us to experiment the effort prediction methodologies in the literature and to adapt them to the changes in technologies over time, as shown in the study and experimentation of the RWO methodology, presented in Chapter 3.

Right this study highlighted the need of a methodology more strictly bounded to company context and more adaptable to different kinds of projects and technologies used by developers. For this reason, I believe it was appropriate to identify what were the last technology trends, before starting a new methodology experimentation. For lack of this kind of research in literature, I set up a survey with the main questions of interest of my research.

The main objective of the following survey was to detect what were the most used frameworks for the development of Web applications and also how the development methodology of these applications had recently evolved. This topic has become significant since when many open source frameworks, that allow to facilitate considerably the developers' task, are available.

Particularly, I wondered if and how using so powerful tools as CMF could affect the calculation of the final effort of a Web application development. To reach this objective, I formulated a questionnaire as simple as possible, in order to cut the filling time, to get as much answers as possible. The survey about the adoption and use of open source CMF was distributed to a sample of Italian software companies and Italian Internet users.

In the following sections you can see the description of the used research method, the collected data and the results of the analysis done on the data itself.

4.1 Research method and gathered data

The data have been collected through a questionnaire. The invitation to participate has been sent through e-mail to some companies that I knew and has been also done through the

publication on the main referential Italian Websites in the field. This questionnaire has been built with a Google form document, thanks to its simplicity and immediacy. It covered 10 questions pertaining to both kind and size of the respondents' belonging company and the technology used for Web applications development.

The data collection lasted about 5 months, from November 2011 - questionnaire publication date - to March 2012. The respondents final number is equal to 155 units, divided between 91 software companies and 64 freelance/individual developers relative to different Italian fields, like public administration, education, marketing, services, etc.

4.2 Data analysis and results

In this section will follow the 10 survey questions and the collected data. From the analysis of the answers, it has been possible to obtain the following information:

- kind and size of belonging company;
- typology and way of use of the possible CMF adopted.

Question n.1: Kind of belonging company.

As one can easily notice from Fig 4.1, the majority of the respondents belong to the technology field: the 27% work on software development, the 6% on Web development and the 25% on IT consulting.

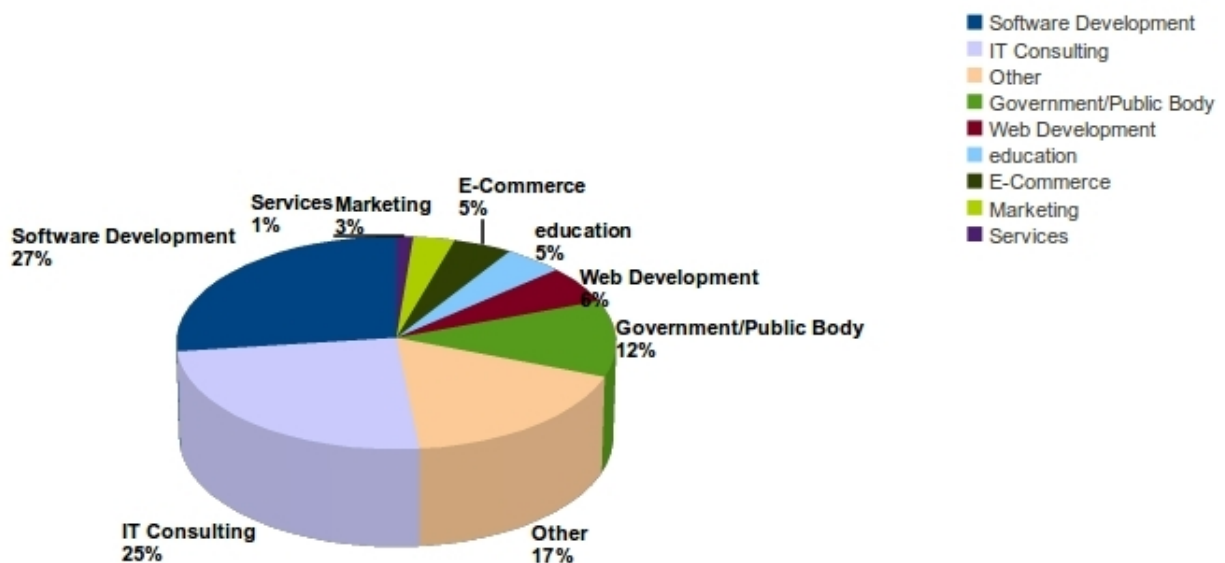


Figure 4.1: Kind of belonging company

Question n.2: Number of employees in the company.

As shown in Fig.4.2, the majority of the respondents belong to a small company, with less than 20 employees.

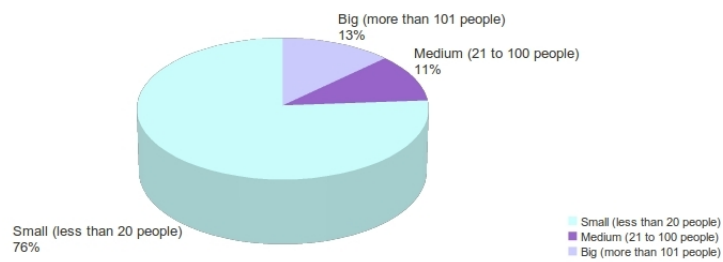


Figure 4.2: Number of employees

Question n.3: Number of developers in the company.

From the Fig.4.3 it is possible to observe that the majority of the companies whom respondents belong to, have a number of developers comprised between 1 and 5, as it was obvious to expect from the analysis of the previous answer.

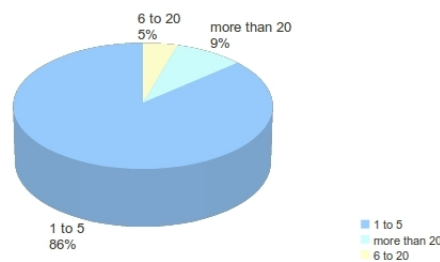


Figure 4.3: Number of developers

Question n.4: Are CMF Open Source (Joomla!, Drupal etc.) usually adopted for the Web application development in your company?

The majority of the respondents, equal to the 87%, declared to usually adopt CMF open source during the development of Web applications (Fig.4.4). The respondents that gave a negative answer to this question exited from the rest of the survey.

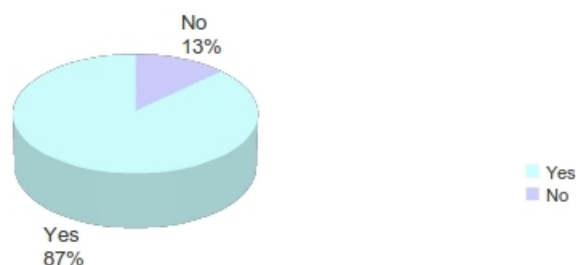


Figure 4.4: CMF usually adopted

Question n.5: What CMF are you using now? (or in your organization)

From the analysis of this question we can conclude that the most adopted CMF among the respondents is Joomla!, followed by WordPress and Drupal. The less known and less adopted CMF are Alfresco WCM and others not well specified, as shown from Fig. 4.5.

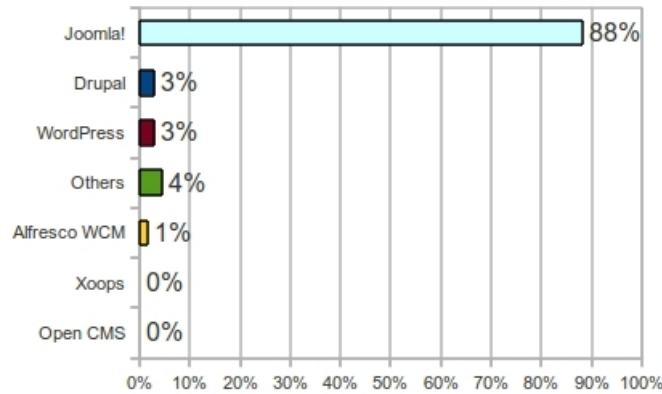


Figure 4.5: CMF used

Question n.6: What percentage of free modules/extensions/components do you use?

The majority of the respondents declared to adopt free modules and components from the CMF library, with a percentage comprised between 81% and 100% of the total. So, the majority of the respondents make extensive use of ready libraries or free downloadable ones, as inexperienced users would do, while only the 5% declared to adopt libraries with a low percentage, equal to 20% maximum, and this suggests they adopt the CMF as very expert users.

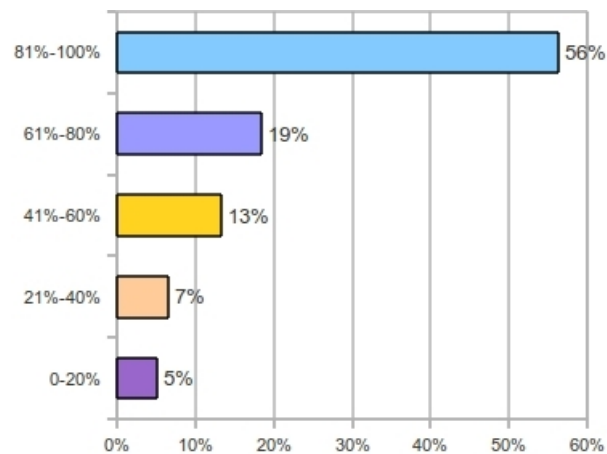


Figure 4.6: Adoption of the library percentage

Questions n.7/8 Have you ever customized any module/extension/component? What percentage do you usually customize the module/extension/component with?

In this section I analyse two connected questions. The majority of the respondents (70%) declared to edit modules from the library (Fig. 4.7); the same respondents, at the next question, declared mainly to edited these modules around 40% of the code compared to the original, as shown from Fig. 4.8. The respondents that gave a negative answer to this question exited from the rest of the survey.

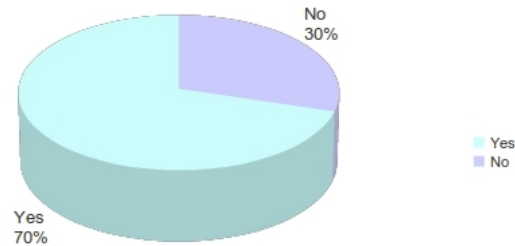


Figure 4.7: Library editing percentage

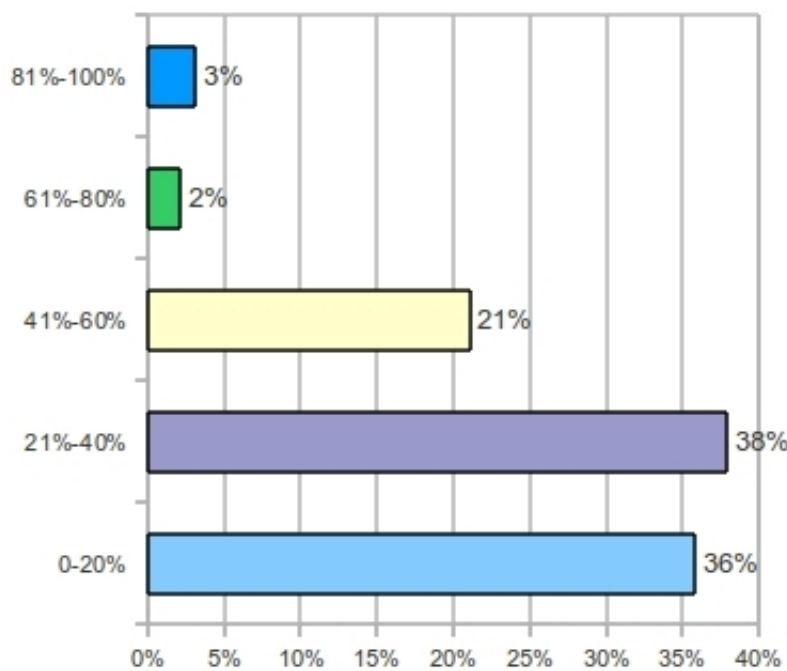


Figure 4.8: Modules editing percentage

Question n.9: How long usually does it take to customize the module/extension/component to fit your needs?

From the analysis of the question, I can assume that the respondents edited modules quickly; then the majority of them, equal to 70%, took from one hour to maximum one work day of time (Fig. 4.9).

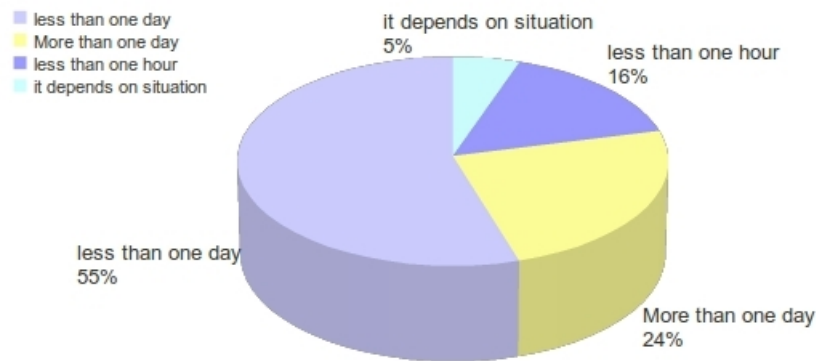


Figure 4.9: Necessary time for editing

Question n.10: Have you ever bought any ready modules/extensions/components?

From the question analysis we can conclude that there is a good percentage of users that usually buy ready modules, although there are many free libraries and despite it is quick to edit library modules, as declared in the last answer.

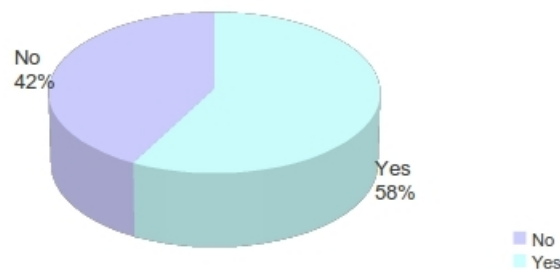


Figure 4.10: Ready modules buying percentage

4.3 Summary of the results

From the analysis of the results it is possible conclude that the majority of the respondents declared to:

- belong to a small company of the technological field, with less than 20 employees and with a number of developers comprised between 1 and 5.
- usually adopt open source CMF for the Web application development, among which Joomla! is the most frequently used.

- make extensive use of ready libraries or free downloadable ones, declaring to edit these modules around 40% of the code compared to the original.
- edit modules from the library quickly and there is a good percentage of users that usually buy ready modules.

Chapter 5

The Web Framework Points Methodology

Combining the experience gained through the study and experimentation of RWO methodology presented in Chapter 3, with the results issued from the survey presented in Chapter 4, it took shape the Web Framework Points methodology, that will be proposed in this Chapter.

As highlighted in the survey presented in Chapter 4, the latest trend¹ in Web applications development is the prevailing usage of Content Management Frameworks (CMF). For this reason, I decided to focus my research work on this direction, elaborating a methodology specifically built for effort estimation of projects where a CMF is in use.

5.1 Content Management Framework

The effort estimation methodology – outlined below – was devised starting from a thorough observation of the development cycle of Web applications developed with Content Management Frameworks available with an Open Source license, such as Joomla!, Drupal, etc.[1, 2].

A Content Management Framework (CMF) is a high-level software application that allows for the creation of a customized Web Content Management System (WCMS). A WCMS is a software tool that can be used by both technical and general staff, and that allows for the creation, editing, management and publishing of a wide asset of multimedia content, in a website. CMFs greatly help to organize and plan a WCMS, freeing the site administrator from all aspects related to Web programming (knowledge of scripting languages, server installation, database creation etc.).

Every CMF, apart from the basic functionalities for creation and management of dynamic pages, have libraries of modules and add-on components readily available to users. By using such libraries, even the most knowledgeable programmer can be free from the task of writing code parts on easy and recurring functionalities, with the advantage of focusing on specific functionalities for her or his own application.

The web developer using a CMF has many options: using just ready-made modules (and components) for the entire application, editing and customizing the available modules to

¹As reported on the survey in Chapter 4, 90% of respondents usually adopt Open Source CMF when developing Web applications.

her or his liking (a chance specific to open source CMFs), or planning and programming new, completely original, modules². In the final estimation of development effort, the use of ready-made modules and components will clearly have a different impact compared to programming new ones starting from scratch. Similarly, editing modules and components in order to customize them will have a different impact altogether.

5.2 The Web Framework Points estimation approach

At the beginning of this thesis it has been shown the anchor-effect phenomenon, able to influence effort evaluation, regardless of the used estimation method. I had then drawn attention to the debate concerning which was the most reliable and accurate, among all the existing estimation methods, in particular through a direct comparison between formal methods and expert judgement. I came to the conclusion that both approaches could not be objective and that you get the best performances using a mix between them. However, formal methods have the advantage of helping project managers to identify and quantify the main cost-drivers of a project and they are based to an high number of real cases. Finally, I have seen that by combining together multiple forecasting methods it is possible to obtain more accurate predictions.

All newly made considerations led to devise an effort estimation methodology that was as far as possible free from anchor-effect, that included specific aspects of the projects and which included three forecasting models (expert-based, analogy-based and regression-based) within a basic mathematical model, in order to improve the accuracy of prediction. Let's see in detail these features:

- **Specific aspects:** after having carefully analysed a sample of software projects, the main cost-drivers have been identified, quantifying them with respective effort and weights.
- **Three forecasting models:** the quantitative value of each cost-driver has been the result of regression-based and analogy-based analysis, that I made on the considered projects dataset. This value changes together with the overall complexity of the project, whose assessment is always carried out by a final expert-based analysis.
- **Absence of anchor-effect:** when using the model, the project manager does not know in advance effort values of each individual cost-driver, so his judgement will not be subject to anchor-effect; cost-drivers merely invite the user to identify all those aspects of "cost" that characterize a project.

²As reported on the survey in Chapter 4:

- 68% of respondents frequently uses components of the library, and the same respondents state they use the components with a frequency between 61 and 100% on the total development of an application;
- 5% of respondents uses modules from the library, and the same respondents state they use the modules with a frequency between 0 and 20%;
- 64% of respondents edits modules from the library, changing usually about 40% of the code compared to the original.

- **Formal model:** at the root of the methodology has been used a simply but effective mathematical model, along the lines of functional models existing in literature (FP, WO, etc.).

The proposed methodology is meant for Web applications developed with CMFs, regardless of the specific technology used to implement the frameworks. It is essentially based on two separate phases, that can be accomplished in parallel, and on a final merge between the data coming from these phases.

One of the phases is the *Size Estimation* which, starting from the requirements, considers the various elements that typically contribute to the size of the application, and weights them with their relative difficulty to implement. Since the work on the various elements is made in different ways – it may be writing code in a programming language, writing style sheets, writing XML to configure an interface, designing the schema of a database, editing a map, or other activities – the resulting size is not expressed in units like lines of code or the like, but is in fact a table with values estimating the size and the relative difficulty to implement the various elements that typically constitute a CMF application.

The other phase of Web Framework Points methodology is the identification of the Cost Model that is characteristic of the organization, and is often specific of a given team working in it. The Cost Model is identified just once, and is valid for all the projects that the team carries on using the same, or similar, tools. The Cost Model gives an estimate, this time in man-days, of the typical effort needed to implement the various elements identified in the methodology, at the various difficulty levels. In practice, it is a table of values that have to be multiplied by the corresponding size estimates to yield the global estimate.

The last step is the computation of the sum of the size of the elements multiplied by the corresponding costs. It is performed straightforwardly, and yields the global estimate of the effort to implement the system, in man-days. Of course, if requirements change after the Size Estimation, the estimation should be updated, and the final effort estimation recalculated. Fig. 5.1 gives a schematic overview of the Web Framework Points methodology.

In the following sections the three steps just shown will be described with more details.

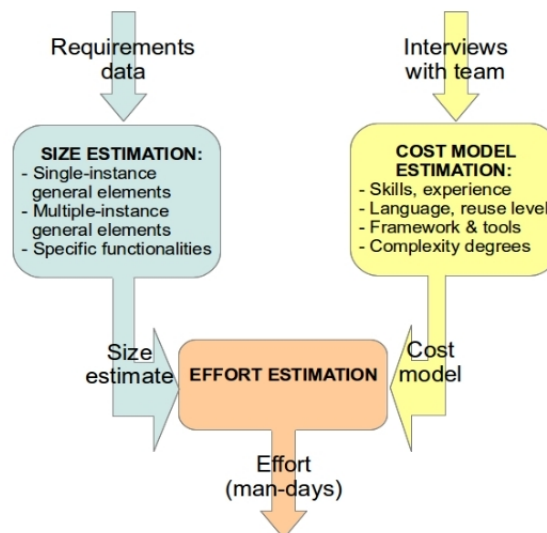


Figure 5.1: Web Framework Points methodology scheme

5.3 Size Estimation of a Web Application

Following the analysis on a sample of Web applications and of their development cycle, distinctive and recurring elements were found. They were divided into two sets: *general elements* and *specific functionalities*. Each element found is marked by a complexity degree, depending on various factors: context of application, existence or absence in the used CMF library, customization, reuse, etc. The weighted sum of each element makes up the size estimation of the Web application. In this way, size estimation is performed in terms of functionalities offered by the application to the user, as in Albrecht's classic FP metric [14], but everything is now contextualized to the present time.

5.3.1 General Elements

General elements are defined as all the preliminary analysis and planning activities, as well as the essential elements for creating the main structure of an application, like basic image elements and some information content, usually static and with low or no interaction with the user. Basic, necessary elements for interaction in an application belong to this class. Some elements are single-instance, while for others there might be a number of instances; all elements have a complexity that can be low, medium-low, medium-high or high.

Single-instance general elements

Below is a list of the 15 single-instance elements, each with its own definition. These elements can be present or not, but if they are present, their number is just one.

CONTEXT AND EXTERNAL ENVIRONMENTAL ANALYSIS

- **Context and user-base analysis:** critical issues and opportunities of the informative space where the Web application is to be run.
- **Analysis of on-line demand-and-offer:** critical summary and review of gathered materials (market analysis, interviews, focus groups, etc.).
- **Newsletter:** policies on spreading and publishing content, how frequently, to whom, etc.
- **Customizations by editorial staff:** feasibility of updates to the site from outside. Options (software-side) to edit the template, in case external staff is planned to be in charge.
- **Site findability and positioning verification:** operations related to the positioning of the site on search engines.

SITE STRUCTURE

- **Content architecture:** content management planning: document types and management types (e.g.: listing texts by expiring date or by type/topic, by priority/deadlines, user type, etc.).

- **Management and re-aggregation of tags and keys:** categorization and classification of content and information on the Website.
- **System infrastructure:** arrangements for the required infrastructure at system level.
- **General search engine on site:** a basic (standard) search engine or a customized one, present in the application.
- **Preparation of bare mockup, requirements and navigation:** decision as to how navigation should be done, what is to be highlighted, content management solutions.
- **Content management system:** creation of components for content management.

GRAPHIC AND MAPS

- **Production of logo and corporate image:** thorough study of design and meanings.
- **Graphic layout production:** layout elaborated by graphic artists, starting from bare mockup (title, footer, static elements in interface).
- **Creation of ad hoc texts, pictures and/or videos:** development of original multimedia content for the Web, on request by the customer, on specific topics.
- **Map (or background):** management of necessary backgrounds for creation of geo-referenced information into the application.

Multiple-instance general elements

Below is a list of the 4 multiple-instance elements, each with its own definition.

- **Community and social management:** managing the presence of the Web application on the main social networks, as static (simply sharing contents) or dynamic (an intelligent and more complex management style). One instance per social network.
- **Templates and navigation system:** planning of main templates (home page, content pages, search pages, etc.), menu and cross-section views (view by user, view by life events, etc.). One instance per template.
- **User role management:** Front-end user registration and customization of access type to the site depending on user type. One instance per user type.
- **Multilingualism:** simple translation of the site and re-planning of some parts depending on language. One instance per each language.

5.3.2 Specific Functionalities

This category includes all elements needed for interaction between application and user, concerning the specific features of the application. These are functionalities expressly created, thus with a high customization level and database interaction (authentication, profiling, data input forms, etc).

As done previously, functionalities are evaluated by number of instances, as well as by complexity level, which can be low, medium-low, medium-high or high. For instance, in the case of the number of tables that have to be created in the DB, we will consider separately the number of low complexity tables, of medium-low complexity tables, and so on, multiplying each number by a weight depending on their complexity and summing up the four factors.

Below is a list of elements describing the 11 multiple-instance specific functionalities, each with its own definition. These elements can be present or not; if they are present, their number can be more than one.

QUERY AND REPORTING

- **DB and internal Query creation:** number of tables in the DB.
- **Report system design:** number of reports.
- **External Query:** number of queries to external DBs.

CARTOGRAPHIC AND MULTIMEDIA

- **Cartographic data base:** use and management of pre-existing data bases needed to include geo-referenced information into the application (e.g. data bases on hospitals/hotels/companies etc.). Ad-hoc cartographic data bases belong to the "DB and external query creation" category.
- **Creation and inclusion of customized maps:** creation and inclusion of maps with placeholder icons, lines, selection tools, videos or pictures, through the use of Google Maps JavaScript API, or similar APIs - number of different maps.
- **Clickable maps:** number of pictures/graphs with hypertextual links to other sites or other sections of the same site.
- **File types managed by the application:** number of different file types the application needs to manage.

EXTERNAL ACCESSIBILITY

- **Management of reserved areas:** definition of access levels (management of content approval workflow: e.g. none, reading, writing, adding/deleting documents, adding new pages, etc.) and functionalities of each reserved area (page or site section) - number of different areas.
- **External system access:** number of accesses to different external applications.
- **Services available outside of the application:** number of Web services the system provides and/or uses.
- **Data input models:** number of modules specific to the application.

5.4 Complexity Degree

Determining the complexity degree of each element is one of the most critical steps in the methodology, because it is left to the project manager's own experience and knowledge of her or his team of developers. The degrees that can be associated to each element are four: low, medium-low, medium-high or high complexity. We decided to use a 4-degree ordinal scale to avoid giving the user of the method the chance to choose a "fully balanced" judgment - that is not to perform a choice. In all cases, the user must choose between "low" and "high", albeit in different levels.

The complexity degree to be assigned to analysis and planning is strongly related to the context and size of the application; thus, it must be assessed on an empirical basis. As far as development of CMF modules or plugins is concerned, we can generally consider:

- **Low complexity** when the element is present in the CMF library or when pre-existing elements are used without substantial changes;
- **Medium-low complexity** when the element is present in the CMF library but a customization is needed, or when pre-existing elements are used with non-substantial changes;
- **Medium-high complexity** when the element is not present in the CMF library and therefore there is a need for it to be implemented, or when the customization of an element in the library is substantial;
- **High complexity** when the element is not present in the CMF library and its implementation is complex or when the customization of an element in the library is very high.

5.5 Cost Model

The cost model used in the Web Framework Points methodology is an empirical model, that includes the following parameters, named *cost-drivers*:

- similar projects developed by the team;
- team members skills;
- software reuse;
- development experience of the team.

As one would expect, the values of the cost drivers differ among teams, so the cost model is not fixed and predetermined, but we need to calibrate it according to the characteristics of the development team. For privacy reasons, we hide the name of the companies we worked with for empirical evaluation of our method. We will simply call them company/team A, B and C. Table 5.1 shows a brief description of the three companies involved in our study. For calibrating the cost model, we interviewed the project manager of each team, asking to state a quantitative judgment that included an overall evaluation of the above mentioned cost drivers, for each element mentioned in sections 5.3.1 and 5.3.2. This was made before

Table 5.1: Software Companies Main Features

Company/Team	Size	n. employees	Foundation year
A	micro/small	15	2007
B	medium	399	1988
C	small	20	2004

the beginning of the estimation phase. Note that, if it is known that some of the elements will never appear in the projects carried on by the company, the corresponding rows of the table can be overlooked.

This procedure has been done only once before the estimation phase, and the values obtained are valid for all company projects ³.

As a result, we obtained the development effort estimate expressed in man-days for each element, and for each degree of complexity. These estimates are specific of each team, and represent the respective cost model. As an example, Table 5.2 shows the cost model pertaining to Team A.

5.6 Calculation of the Estimation

After considering every element, each one of which is weighted with its own complexity, the effort estimation of the Web application results from the simple sum of all elements:

$$Effort_{estimation} = \sum_{j=1}^M EG_j c_j + \sum_{k=1}^N FS_k c_k \quad (5.1)$$

Where:

EG_j is the j -th general element, of c_j complexity, and M is the total number of general elements;

FS_k is the k -th specific functionality, of c_k complexity, and N is the total number of specific functionalities.

³Project manager could omit few elements of methodology WFP that were not covered in the projects examined

Table 5.2: Cost Model of the Team A

Elements	Complexity degree			
	Low	Medium-Low	Medium-High	High
Context and user-base analysis	0.5	1	2	5
Analysis of on-line demand-and-offer	0.5	1	3	5
Newsletter	0.5	2	4	6
Customizations by editorial staff	0.5	1	5	8
Site findability and positioning verification	0.5	1	2	3
Content architecture	0.5	1	2	3
Management and re-aggregation of tags and keys	0.5	1	1.5	2
System infrastructure	1	2	3	5
General search engine on site	0.5	1	2	5
Preparation of bare mockup, requirements and navigation	0.5	1	2	4
Content management system	0.5	1	1.5	2
Production of logo and corporate image	1	1.5	2	3
Graphic layout production	1	1.5	2	3
Creation of ad hoc texts, pictures and/or videos	0.5	1	2	3
Map (or background)	0.5	1	1.5	2
Community and social management	0.5	1	1.5	2
Templates and navigation system	1	3	5	7
User role management	0.5	1	2	4
Multilingualism	1	2	5	8
DB and internal Query creation	0	0.25	0.6	1
Report system design	0.5	1	1.5	2
External Query	0.5	1	1.5	2
Cartographic data base	0.5	1	1.5	2
Creation and inclusion of customized maps	0.5	1	3	5
Clickable maps	0.2	0.5	0.75	1
File types managed by the application	0.5	1	1.5	2
Management of reserved areas	0.5	1	1.5	2
External system access	1	3	6	10
Services available outside of the application	0.5	1	1.5	2
Data input models	0.25	0.5	0.75	1

Chapter 6

Experimental Results

As stated previously, the WFP methodology has been the subject of experimentation on real projects developed by Italian software companies; in this chapter the results of this experimentation will be presented.

The methodology outlined here can be considered to be generally valid, since the elements presented in sections 5.3.1 and 5.3.2 are common to many Web applications. On the other hand, the calibration of the method through the choice of the complexity degree to assign to each element is strongly dependent on the team developing the application. Therefore, the experimental findings shown below are to be considered of limited external validity, although they represent an interesting validation case of a methodology on real data.

Testing the validity of the Web Framework Points effort estimation model through comparison with other methods usually used in literature, such as FP, COCOMO, WO etc., has not been possible, because these methods measure different elements from those we considered, and are, therefore, hardly comparable with it.

6.1 Dataset

Software companies involved in testing the methodology WFP, faithfully reflect the general Italian production sector, made up principally of SMEs. In Italy there are over 6000 companies of software and ICT services and most of them are small: the 70% of turnover is less than 2 billion € a year, 39% of companies have less than 5 employees, 77% do not reach 20 employees and less than 1% have more than 500 employees [36].

We considered a dataset made by 19 projects, developed with Content Management Frameworks between 2009 and 2012, by the three software companies A, B and C. These companies provided us all relevant data about the considered projects, and namely their requirements, original estimations and overall development effort computed after the project completion. Note that our estimation results refer to projects that were already completed. We had no chance to estimate projects at their beginning, using only their requirements, and then to compare the results with the real effort after the actual completion of the project.

The studied projects were generally aimed to develop applications for public administrations, health services and other customers; they were all finished successfully.

The first activity we performed was to build the complete cost model for the three consid-

Table 6.1: Software Companies estimation methods and technologies used

Project	Estimation method	Technology
1/Team A	Analogy & expert judgment	jAPS
2/Team A	Analogy & expert judgment	jAPS
3/Team A	Analogy & expert judgment	Joomla!
4/Team B	expert judgment	Joomla!
5/Team B	expert judgment	Joomla!
6/Team B	expert judgment	Joomla!
7/Team B	expert judgment	Joomla!
8/Team B	expert judgment	Joomla!
9/Team B	expert judgment	Joomla!
10/Team B	expert judgment	Joomla!
11/Team B	expert judgment	Joomla!
12/Team B	expert judgment	Joomla!
13/Team B	expert judgment	Joomla!
14/Team B	expert judgment	Joomla!
15/Team B	expert judgment	Joomla!
16/Team C	Analogy	Joomla!
17/Team C	Analogy	Joomla!
18/Team C	Analogy	Joomla!
19/Team C	Analogy	Joomla!

ered companies A, B and C, as described in section 5.2 . Then, for each project, the elements discussed in sections 5.3.1 and 5.3.2 were extracted from their original requirement documents. For each element present in each examined project, we multiplied their relative size for the estimated complexity factor, following the respective cost model table (accounting also for the number of instances with different complexity level). We then summed up the value of all these elements, as reported in eq. 5.1 of Section 5.6, obtaining the provisional *effort estimate*.

We then compared the final actual effort of the completed projects with the provisional *effort estimates*. We considered both the provisional effort estimate provided by each company and the provisional effort estimate calculated with the Web Framework Points methodology. The analysis showed differences in estimation data, as expected. In Section 6.3 this analysis is shown. Table 6.1 shows the effort estimation methods and technologies used by each company.

6.2 Effort Prediction and Evaluation Method

We evaluated the effectiveness of the methodology in predicting the effort of the analyzed applications through the calculation of the MRE (Magnitude of Relative Error) factor for each project, a measure commonly used in the Web estimation literature for prediction accuracy.

$$MRE = \frac{Effort_{ACTUAL} - Effort_{ESTIMED}}{Effort_{ACTUAL}} \quad (6.1)$$

Similarly to what reported in [33], we completed the error evaluation by calculating the prediction level $Pred$, defined as the proportion of the observations within a given level of accuracy:

$$Pred(l) = \frac{k}{N} \quad (6.2)$$

By performing N total observations, if k is the number of observations with an MRE less than or equal to l , $Pred(l)$ is the percentage of projects with a MRE less than or equal to l . For instance, if we have 10 projects, with just 8 out of them having an MRE less than 0.2, then $Pred(0.2) = 0.8$. Conte et al. [16] suggest an acceptable threshold value for the mean MRE to be less than or equal to 0.25, and for $Pred(0.25)$ to be greater than or equal to 0.75. In other words, more than 75% of the projects should have an MRE less than 0.25.

6.3 Results

Table 6.2 summarizes the results for the examined dataset. As shown there, the method gives good results, with very low MRE values. As confirmed by the more detailed MRE analysis shown in Table 6.3, the Web Framework Points methodology has a value of $Pred(0.25)$ equal to 79%, so it fully satisfies Conte's criterion. This means, for our methodology, a good estimation power regarding the effort needed to build the application.

Tables 6.4 and 6.5 show the values of provisional effort estimation of projects calculated using the corresponding company's methodology. In this case, $Pred(0.25)$ is equal to 47%, so it does not satisfy Conte's criterion.

As can be seen from Fig. 6.1 and 6.2, the WFP methodology trend is to underestimate the effort (63%), while the company's methodologies trend is to overestimate this value (58%).

Table 6.2: Effort Estimate on 19 Datasets Pertaining to Real Projects

Project n.	Provisional effort estimate through WFP methodology [man-days]	Final actual effort [man-days]	Final - provisional effort [man-days]	Qualitative judgment	MRE
1/Team A	24.5	24.75	0.25	underestimated	0.01
2/Team A	53.25	58.75	5.5	underestimated	0.09
3/Team A	56.2	64.5	8.3	underestimated	0.13
4/Team B	201	196	-5	overestimated	0.03
5/Team B	263	248	-15	overestimated	0.06
6/Team B	799	1047	248	underestimated	0.24
7/Team B	39	40	1	underestimated	0.03
8/Team B	185	218	33	underestimated	0.15
9/Team B	241	300	59	underestimated	0.20
10/Team B	19	13	-6	overestimated	0.46
11/Team B	28	35	7	underestimated	0.20
12/Team B	349	403	54	underestimated	0.13
13/Team B	411	326	-85	overestimated	0.26
14/Team B	274	142	-132	overestimated	0.93
15/Team B	1925	1988	63	underestimated	0.03
16/Team C	13.38	10	-3.38	overestimated	0.34
17/Team C	21	25	4	underestimated	0.16
18/Team C	28.5	30	1.5	underestimated	0.05
19/Team C	10.38	10	-0.38	overestimated	0.04

Table 6.3: MRE Statistics of WFP methodology

Min	Max	Mean	Median	Std dev	Pred(0.25)	%Pred(0.25)
0.01	0.93	0.19	0.13	0.21	15	79 %

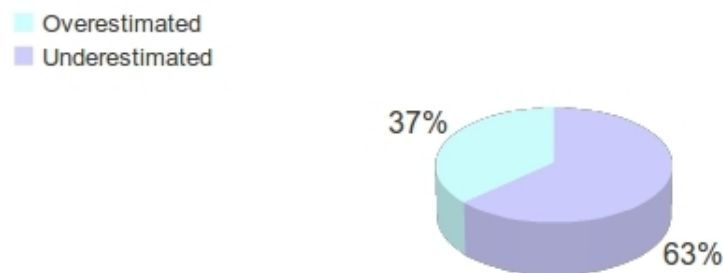


Figure 6.1: WFP qualitative judgment

Table 6.4: Companies Effort Estimate on 19 Datasets Pertaining to Real Projects

Project n.	Provisional effort estimate through company methodology [man-days]	Final actual effort [man-days]	Final - provisional effort [man-days]	Qualitative judgment	MRE
1/Team A	40	24.75	-15.25	underestimated	0.62
2/Team A	66.8	58.75	-8.05	underestimated	0.14
3/Team A	110	64.5	-45.5	underestimated	0.71
4/Team B	168	196	28	overestimated	0.14
5/Team B	360	248	-112	overestimated	0.45
6/Team B	930	1047	117	underestimated	0.11
7/Team B	42	40	-2	underestimated	0.05
8/Team B	261	218	-43	underestimated	0.20
9/Team B	260	300	40	underestimated	0.13
10/Team B	24.17	13	-11.17	overestimated	0.86
11/Team B	30	35	5	underestimated	0.14
12/Team B	125	403	278	underestimated	0.69
13/Team B	125	326	201	overestimated	0.62
14/Team B	125	142	17	overestimated	0.12
15/Team B	1200	1988	788	underestimated	0.40
16/Team C	15	10	-5	overestimated	0.50
17/Team C	30	25	-5	underestimated	0.20
18/Team C	40	30	-10	underestimated	0.33
19/Team C	15	10	-5	overestimated	0.50

Table 6.5: MRE Statistics of Companies Effort Estimates.

Min	Max	Mean	Median	Std dev	Pred(0.25)	%Pred(0.25)
0.05	0.86	0.36	0.33	0.24	9	47 %

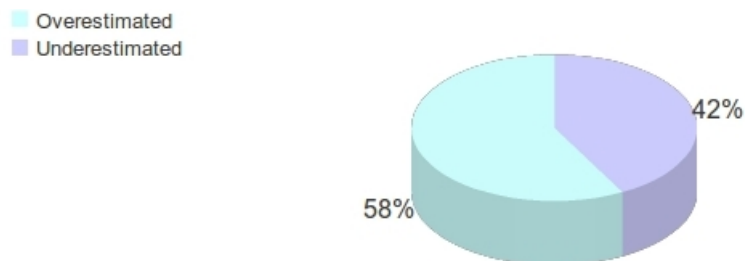


Figure 6.2: Companies qualitative judgment

Chapter 7

Web Framework Points Tool

In this section it will be shown the main features of the interactive application ¹ implementing the Web Framework Points methodology, object of the present thesis. I used this application both to collect projects data from the companies and to estimate the effort. From the experimentation described in the Chapter 5, I obtained three Cost Models, one for each company involved (Cost Model 1, 2 and 4). The fourth Cost Model (Cost Model 3) has been obtained from an average of values between the Cost Model 2 and 4. Cost Models are summarized in the following:

- **Cost Model 1:** freelance developer/ micro company, from the Team A;
- **Cost Model 2:** small company, from the Team C;
- **Cost Model 3:** medium company;
- **Cost Model 4:** big company, from the Team B.

These four cost models have been incorporated in the application, and have been used to estimate the effort of the project belonging to the three categories of companies involved in the experimentation. Obviously, for a correct use of the tool, the projects in question (to be estimated) have to be Web applications developed through CME.

7.1 Technology used

The design of the WFP software system components has been made following the Model View Controller (MVC) architectural pattern. According to this pattern, the representation of information is separated from the user's interaction:

- A **Model** is an object representing data (e.g. a database table).
- A **View** is an object managing the visualization of the state of the Model (data and information).

¹Available on the <http://tulipano.dibe.unige.it/stimaEffort-webapp/home>

- A **Controller** is an object interpreting the inputs from the user, and offers facilities to change the state of the model.

Fig. 7.1 shows the relationship among the three objects.

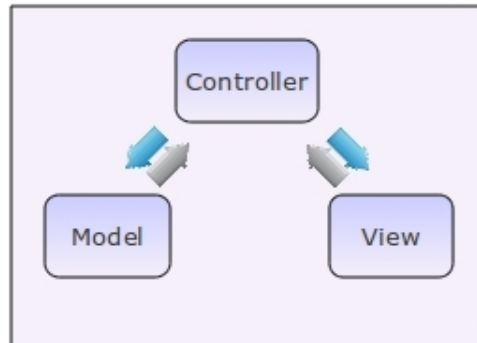


Figure 7.1: MVC Pattern

The application development has been made using the object-oriented programming language Java, while for the management of the entire project have been used the open source frameworks Spring, Hibernate and Maven. Finally, the open-source object-relational database PostgreSQL has also been used (Tab 7.1).

Table 7.1: Technology used in the WFP application

Design Pattern	MVC
Language	Java
Framework	Spring, Hibernate, Maven
DataBase	PostgreSQL

7.2 Functioning

The main steps of the exchange of requests among the different components of the WFP system and the final user are summarized in the following:

1. The user requests a specific URL.
2. The server executes the logic of the controller corresponding to the required URL.
3. The controller responds with the corresponding view.
4. The user interacts with the view that sends the various requests to the controller.
5. The controller communicates to the component model for saving/retrieving information, according to the input data and the actions undertaken by the user.

7.3 Architecture

In Fig. 7.2, the architecture of the entire Web Framework Points application is shown.

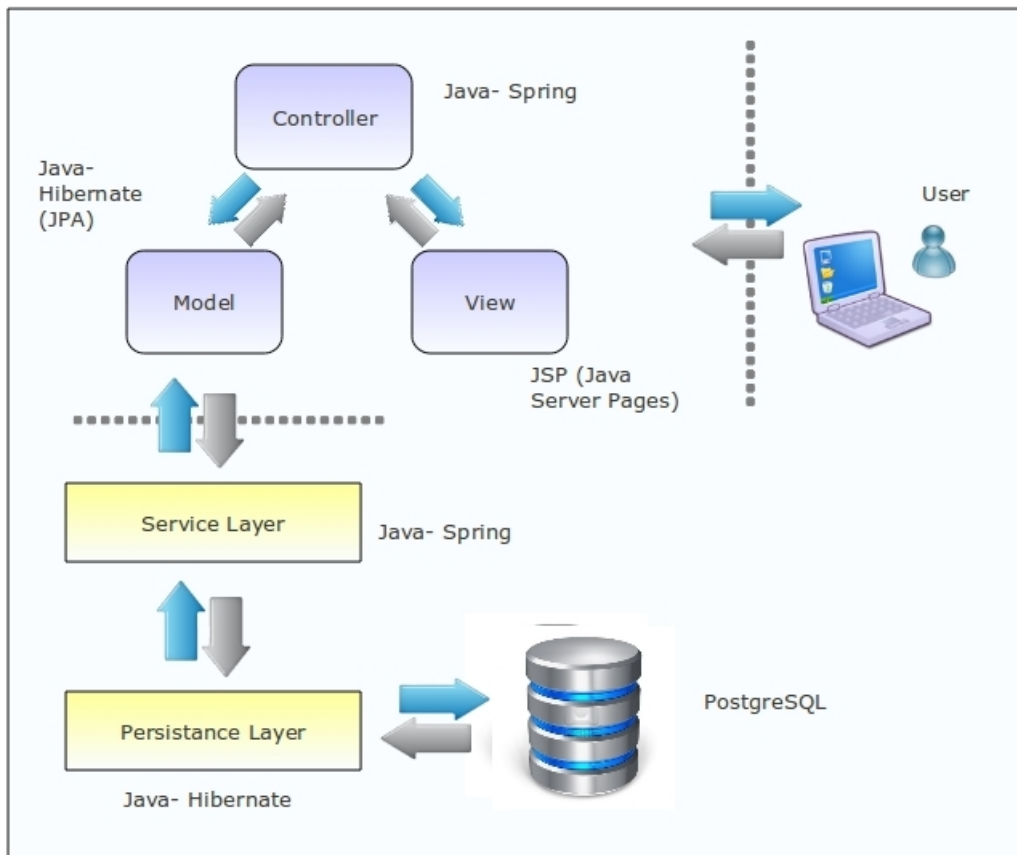


Figure 7.2: WFP architecture

7.4 Home Page

In the application main screen there is the **REGISTRATION** form, where the user can enter its own data and those relative to the own company, choosing among: freelance developer, micro, small, medium or big company, specifying the number of people that form the developing team of the Web applications. After the registration, it is possible to access to the application by entering the username and the password in the **LOGIN** form, as shown in Fig.7.3.

The screenshot displays the home page of the Web Framework Points tool. At the top, there is a navigation bar with links for 'Projects', 'Add Project', 'Copy Project', and 'Contact us', along with flags for the United Kingdom and Italy. The main header features the 'Web Framework Points' logo and a brief description: 'Tool for estimating the effort of Web applications developed with a Content Management Framework (CMF)'. To the right of the header is a 'LOGIN' form with fields for 'UserName' and 'Password', and a 'Login' button.

The central part of the page is dominated by the 'REGISTRATION' form. It includes the following fields:

- Email Address *
- Name and last name
- Company Name
- Turnover (k€) with a value of 0.0
- Number of developers with a value of 0
- Password *
- City
- Country
- Number of employees * with a dropdown menu set to 'Freelance Developer'
- Sector * with a dropdown menu set to 'Software Development'

 Below the form, there is a checkbox for 'I read and I accept the Privacy Policy . All collected data will be used for research purposes only.' and a note '* Mandatory fields'. A 'Register' button is positioned at the bottom of the registration section.

On the right side, there is a 'USEFUL LINKS' section with a 'VIEW MORE' button. It lists four links:

- User's Guide**: WFP User's Guide.
- Agile Group**: Software Engineering Group of the University of Cagliari.
- DITEN**: Department of Naval, Electric, Electronic and Telecommunications Engineering of the University of Genova.
- BO.ME.CC.**: smart BOat, software MEmrics and Cloud Computing.

The footer contains the 'Agile Group' logo, a navigation menu with links for 'Home', 'About Us', 'Contact Us', 'Privacy Policy', and 'Terms Of Use', the 'DITEN' logo, and a 'Back To Top' link.

Figure 7.3: Registration Form

7.5 Add Project form

Once logged in, the user can enter the relative data about his/her own projects, in the **Add Project** section. As it can be seen from Fig.7.4, it is necessary to add the name of the project, the features pertaining to the development (operating system, methodology, IDE, language and framework used) and the number of people of the team involved to the project.

It is also necessary to specify if the project is concluded or not, because it is possible to use the WFP tool both for having an effort estimation and for verifying to what extent the estimation done differs from the real value.

If the project is already concluded, it is necessary to indicate the number of man-days that has been used to carry out the entire project, from the idea to its complete realization.

If the project is at an early stage, or has not been concluded yet, it will be necessary to indicate the number of man-days that the user thinks will be required for the entire implementation.

The screenshot displays the 'Add Project' form within the 'Web Framework Points' application. The form is structured as follows:

- Project Name:** A text input field.
- Number of people in the team:** A dropdown menu with '1-2' selected.
- Number of days estimated / actual:** A text input field with '0.0' entered.
- Operating System used to develop the project:** A dropdown menu with 'Linux' selected.
- Language:** A dropdown menu with 'Java' selected.
- Methodology:** A dropdown menu with 'Scrum' selected.
- Framework:** A dropdown menu with 'spring' selected.
- IDE:** A dropdown menu with 'eclipse' selected.
- Project Concluded:** Radio buttons for 'Yes' and 'No', with 'No' selected.

A 'Forward' button is located at the bottom right of the form. To the right of the form is a 'USEFUL LINKS' sidebar with a 'VIEW MORE' button. The sidebar contains the following links:


- User's Guide:** WFP User's Guide.
- Agile Group:** Software Engineering Group of the University of Cagliari.
- DITEN:** Department of Naval, Electric, Electronic and Telecommunications Engineering of the University of Genova.
- BO.ME.CC.:** smart BOat, software MEmics and Cloud Computing.

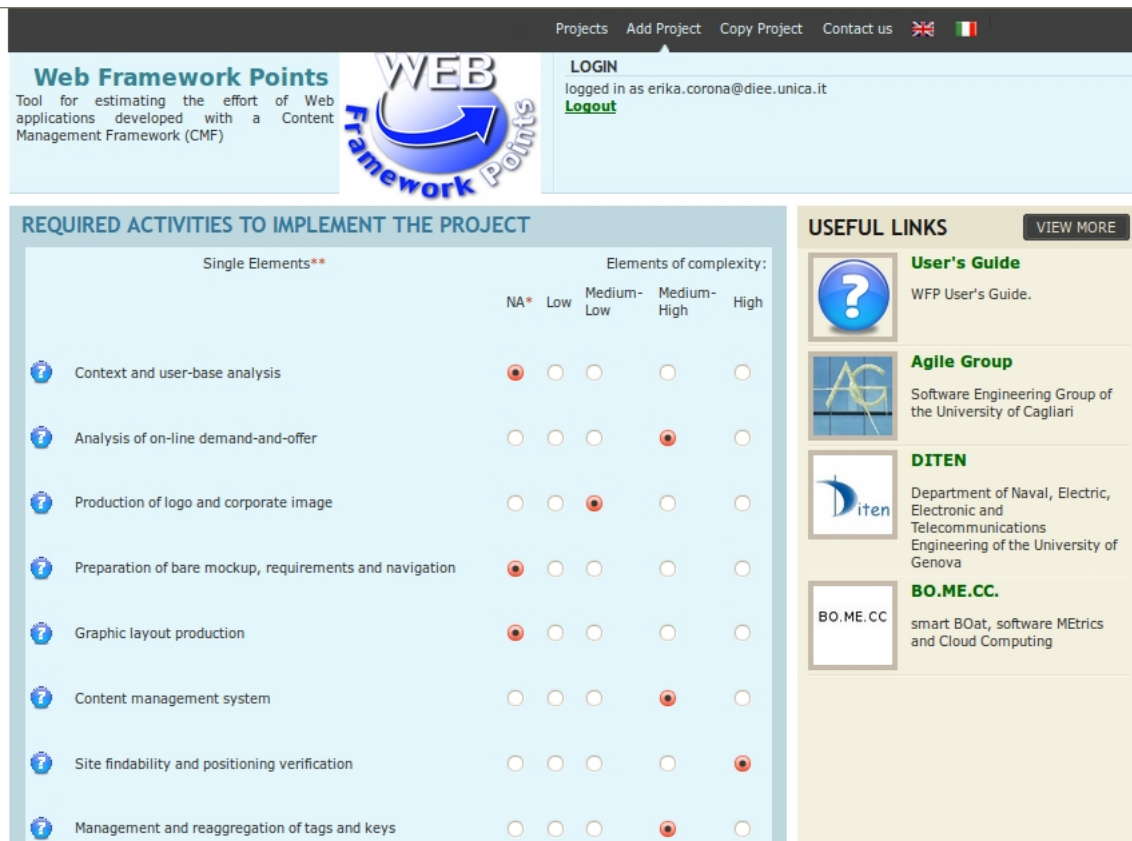
The footer of the page includes 'Agile Group', navigation links ([Home](#), [About Us](#), [Contact Us](#), [Privacy Policy](#), [Terms Of Use](#)), 'DITEN', and a 'Back To Top' link.

Figure 7.4: Add Project Form

7.6 Activities required to implement the project

By clicking on the **Forward** button, the user can access to the **Activities required to implement the project** section, divided between **Single elements** and **Multiple elements**. They represent the main steps of the development cycle and the necessary elements to create the application. Each element is characterized by a complexity degree that can be low, medium-low, medium-high or high.

- **Single elements:** the list of the single elements is set up by closed answers. They will be counted one time for each application to be developed. The project manager will simply have to select the degree of complexity that each item is supposed to have in the application. By clicking on the icon  (to the left of each item) the user can read an explanation regarding each item of the list. If in the application to be developed there is a not predicted item, it will be sufficient to select NA (not applicable), as it can be seen in Fig.7.5.



The screenshot displays the 'Web Framework Points' tool interface. The main section is titled 'REQUIRED ACTIVITIES TO IMPLEMENT THE PROJECT' and contains a table of 'Single Elements**'. The table has columns for 'NA*', 'Low', 'Medium-Low', 'Medium-High', and 'High'. Each row represents an activity, and the selected complexity level is indicated by a red dot in the corresponding radio button.


Single Elements**	NA*	Low	Medium-Low	Medium-High	High
Context and user-base analysis	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Analysis of on-line demand-and-offer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Production of logo and corporate image	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Preparation of bare mockup, requirements and navigation	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Graphic layout production	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Content management system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Site findability and positioning verification	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Management and reaggregation of tags and keys	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

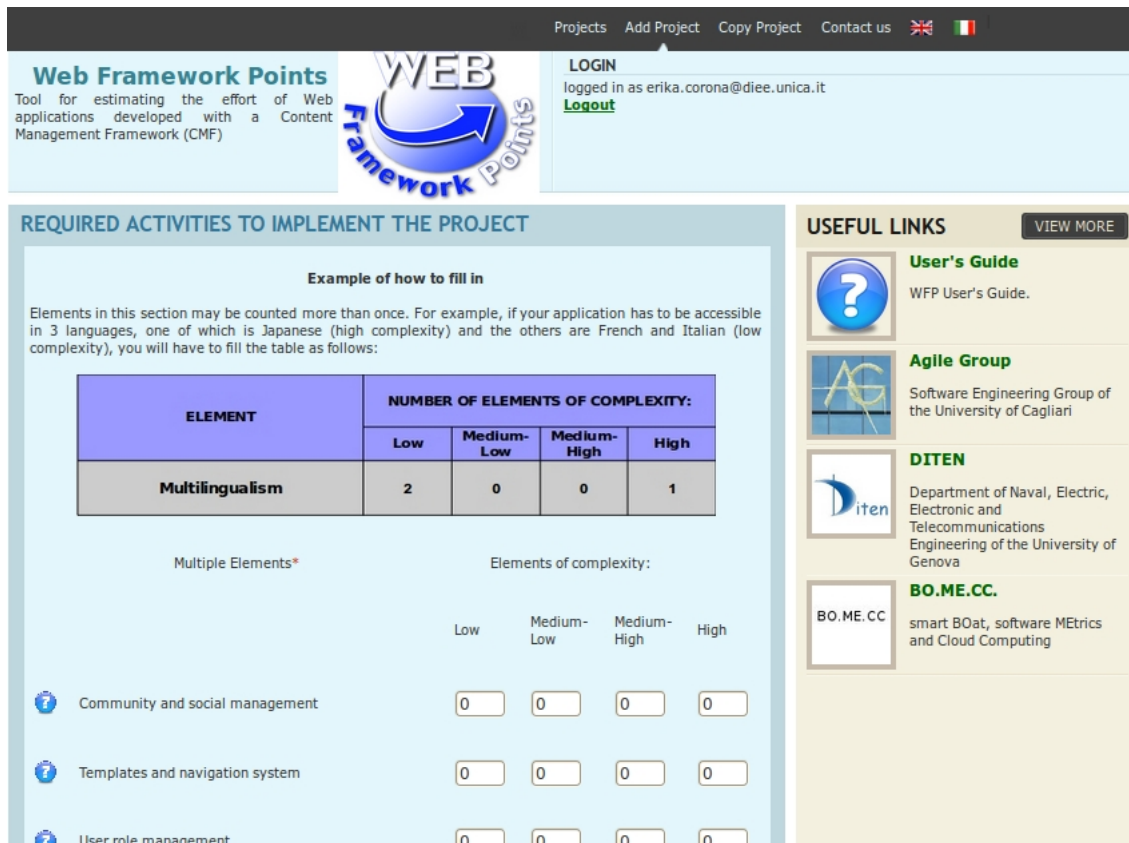
The sidebar on the right contains 'USEFUL LINKS' with a 'VIEW MORE' button. The links listed are:

- User's Guide**: WFP User's Guide.
- Agile Group**: Software Engineering Group of the University of Cagliari.
- DITEN**: Department of Naval, Electric, Electronic and Telecommunications Engineering of the University of Genova.
- BO.ME.CC.**: smart BOat, software METrics and Cloud Computing.

Figure 7.5: Single Elements



After completing the list of these single elements, by clicking on the **Forward** button the user can access to the next list:

- **Multiple Elements:** the elements on this section may be counted more than once. The project manager will have to evaluate their complexity and to enter the elements number. For example, if the application to be developed has to be accessible in 3 languages, one of which is Japanese (high complexity) and the others are French and Italian (low complexity), the table has to be completed as shown in Fig.7.6. Also in this case, by clicking on the icon  (to the left of each item) the user can read an explanation regarding each item of the list.



Web Framework Points
Tool for estimating the effort of Web applications developed with a Content Management Framework (CMF)

WEB Framework Points

Projects Add Project Copy Project Contact us  

LOGIN
logged in as erika.corona@diee.unica.it
[Logout](#)




REQUIRED ACTIVITIES TO IMPLEMENT THE PROJECT

Example of how to fill in

Elements in this section may be counted more than once. For example, if your application has to be accessible in 3 languages, one of which is Japanese (high complexity) and the others are French and Italian (low complexity), you will have to fill the table as follows:

ELEMENT	NUMBER OF ELEMENTS OF COMPLEXITY:			
	Low	Medium-Low	Medium-High	High
Multilingualism	2	0	0	1

Multiple Elements* Elements of complexity:

	Low	Medium-Low	Medium-High	High
 Community and social management	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
 Templates and navigation system	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
 User role management	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

USEFUL LINKS

[VIEW MORE](#)





-  **User's Guide**
WFP User's Guide.
-  **Agile Group**
Software Engineering Group of the University of Cagliari
-  **DITEN**
Department of Naval, Electric, Electronic and Telecommunications Engineering of the University of Genova
-  **BO.ME.CC.**
smart BOat, software MEmrics and Cloud Computing

Figure 7.6: Multiple Elements

7.7 Project DB

After completing the list, the user can click the **Save project** button and the project will be stored in the personal DB, for a further consultation and/or future changes (Fig.7.7). It will be possible than to know the effort estimation by clicking on **Effort Estimation**, edit or delete the project and to obtain a report of the results. It will be also possible to duplicate a project with the same feature as the one already inserted, by clicking on the **Copy Project** section (in the main banner), as it is shown in Fig.7.8.

The screenshot displays the 'Web Framework Points' tool interface. At the top, there is a navigation bar with links for 'Projects', 'Add Project', 'Copy Project', and 'Contact us', along with flags for the United Kingdom and Italy. Below this, the main header area includes the tool's logo and a description: 'Web Framework Points Tool for estimating the effort of Web applications developed with a Content Management Framework (CMF)'. A user is logged in as 'erika.corona@diee.unica.it' with a 'Logout' link.

The main content area is divided into two sections. On the left, under the heading 'MY PROJECTS', there is a list of three projects: 'Progetto Uno', 'Progetto Due', and 'Alpha'. Each project entry includes links for 'Edit', 'Delete', and 'Report Effort estimation'. On the right, under the heading 'USEFUL LINKS', there are four links: 'User's Guide' (WFP User's Guide), 'Agile Group' (Software Engineering Group of the University of Cagliari), 'DITEN' (Department of Naval, Electric, Electronic and Telecommunications Engineering of the University of Genova), and 'BO.ME.CC.' (smart BOat, software MEtrics and Cloud Computing).

At the bottom of the page, there is a footer area with the 'Agile Group' logo and a navigation menu: 'Home | About Us | Contact Us | Privacy Policy | Terms Of Use'. The 'DITEN' logo is also present in the footer. A 'Back To Top' link is located in the bottom right corner.

Figure 7.7: Project DB

Projects Add Project Copy Project Contact us

Web Framework Points
Tool for estimating the effort of Web applications developed with a Content Management Framework (CMF)

WEB Framework Points

LOGIN
logged in as erika.corona@diee.unica.it
[Logout](#)

COPY PROJECT

Enter the name of the project to be copied Enter the name of the new project

USEFUL LINKS [VIEW MORE](#)

- User's Guide**
WFP User's Guide.
- Agile Group**
Software Engineering Group of the University of Cagliari
- DITEN**
Department of Naval, Electric, Electronic and Telecommunications Engineering of the University of Genova
- BO.ME.CC.**
smart BOat, software MEtrics and Cloud Computing

Agile Group [Home](#) | [About Us](#) | [Contact Us](#) | [Privacy Policy](#) | [Terms Of Use](#) **DITEN**

[Back To Top](#)

Figure 7.8: Copy Form

The screenshot displays the Web Framework Points tool interface. At the top, there is a navigation bar with links for 'Projects', 'Add Project', 'Copy Project', and 'Contact us', along with flags for the United Kingdom and Italy. The main header area includes the tool's name, a description, a logo, and a user login status.

Web Framework Points
 Tool for estimating the effort of Web applications developed with a Content Management Framework (CMF)

LOGIN
 logged in as [erika.corona@ticom.it](#)
[Logout](#)

PROJECT REPORT

Company Features

Project Name	Alpha
Number of people in the team	1
Methodology	Scrum
Framework	Joomla!
IDE	eclipse
Project Concluded	No

Single Elements

Context and user-base analysis	none
Analysis of on-line demand-and-offer	medium-low
Production of logo and corporate image	medium-high
Preparation of bare mockup, requirements and navigation	none
Graphic layout production	none
Content management system	medium-high
Site findability and positioning verification	none
Management and reaggregation of tags and keys	medium-high
Content architecture	none
General search engine on site	none
System infrastructure	none

USEFUL LINKS [VIEW MORE](#)

- User's Guide**
WFP User's Guide.
- Agile Group**
Software Engineering Group of the University of Cagliari
- DITEN**
Department of Naval, Electric, Electronic and Telecommunications Engineering of the University of Genova
- BO.ME.CC.**
smart BOat, software MEtrics and Cloud Computing

Figure 7.9: A project report

Chapter 8

Validity Analysis

Essentially, in a software engineering context, there are four types of validity that contribute to judge the overall validity of a research, i.e. internal, construct, external and conclusion validity [37]. This Chapter explains these concepts and identifies possible threats to validity regarding the obtained experimental results of Web Framework Points methodology.

8.1 Internal validity

A research has internal validity if there is a causal relationship between independent and dependent variables and if there is only one explanation for the research results. Some factors that could threaten the internal validity are:

- **confounding variables:** variables that can modify the relationship among experimental variables, such as external events, testing effects, selection effects etc.;
- **subject effects:** behavioural modification of subjects involved in the research due to the knowledge of being observed or studied (i.e. Hawthorne effect, acquiescence bias, etc.);
- **experimenter bias:** it occurs when the experimenter influences the participant's behaviour (personal characteristics, variations in tone of voice, body posture, etc.) or when external factors are accidentally introduced by the experimenter into the observed phenomenon (i.e. errors in measurements).

In my study, the factors that could threaten the internal validity are as follows:

- **testing effects:** the participants of the research (project managers) could have used the WFP methodology in different ways in the projects. This could be due to misinterpretation of elements definition or due to the learning process;
- **selection effects:** selection of participating companies has not been done by involving all categories of interest in a homogeneous way. It had not been possible to include the same number of freelance developers, micro, small, medium and big companies participating to the research. The experimentation was done on a voluntary and free basis;

- **experimental mortality:** some subjects participated to the experimentation only with one or two projects, thus choosing not to pursue the experimentation (dropout);
- **Hawthorne effect:** in qualitative assessment of different elements, some subjects could have specified complexity values lower/higher than those actually experienced, depending upon the attention being received by the researcher.

8.2 External validity

To ensure external validity of a study, the generalizability is essential. It refers to whether the results can be generalized to populations/situations/times/environments different from those in the experiment. Generally, there are three types of possible generalization:

- **population generalization:** generalization of the results from the considered sample to a larger population;
- **environmental generalization:** regards the confidence you can have in generalizing your results across situations not included in your study;
- **time generalization:** the extent to which the results of the conducted research apply at different time.

Results of my research are closely tied to the moment in which the research is carried out, because technologies change continuously. Therefore, there are no prerequisites to consider the time generalization factor. The threat to the external validity may be due to the following factor:

- **inadequacy of the sample:** the number of companies involved in the experimentation is not high. Also, the number of projects on the considered dataset is low, although the sample of the companies is representative of the typology of a medium Italian company.

8.3 Construct validity

Construct validity concerns the correspondence between research and theory that underlies the research itself. From this point of view, a research is valid if one may exclude alternative explanations of the results obtained in relation to the theory of reference, and if one is measuring what it is expected to measure. Factors which may threaten this kind of validity are the following:

- **inadequate analysis of conceptual constructs,** due to the lack of analysis of the subject under study;
- **inadequate operationalization of theoretical constructs:** in order to study a concept properly, it is necessary to transform the concept into concrete operations;
- **ambiguity of independent variables:** when the independent variables suggested by the researcher have not been evaluated carefully and they do not match or represent the problem domain adequately.

In my research, the threats to construct validity may be due to the fact that some secondary elements, that are necessary for the construction of Web applications (see Chapter 5), may have been overlooked and therefore not included in the list.

8.4 Conclusion validity

A research has conclusion validity when the observed results in experiments are due to the cause-effect relationship with the considered variables apart from due to other factors, such as randomness. Threats to conclusion validity could be due to:

- **fishing:** incorrect hypothesis concerning correlation among considered variables;
- **small sample size:** results can be derived from a statistically insignificant sample.

Regarding my research, the threat may be due to the fact that the sample size of the companies and the projects is statistically small.

8.5 How to reduce threats to validity identified

Essentially, threats to internal, external and conclusion validity could be reduced by including larger and homogeneous samples in the experimentation. This could be done by considering an equal number of freelance developers, micro, small, medium and large companies, contributing a larger number of projects to the research.

Regarding the threats to construct validity, it would require a further investigation among other project managers, in order to find possible elements (cost-drivers) neglected in the previous experiments (although the subjects involved in research considered the elements of the methodology adequately representative).

Chapter 9

Conclusions

In this thesis, a new methodology of effort estimation for Web applications developed using a Content Management Framework (CMF) has been presented. The reason for proposing a new methodology for effort estimation was the realization of the inadequacy of the Revised Web Objects (RWO) method – which I had previously developed – in estimating the development effort of the latest Web applications. Thanks to the partnership, now years old, with the same software company, I could directly observe on the field the evolution of development technologies and methodologies on projects developed over a span of almost 10 years.

Recently, I also started new partnerships with two small software companies, so I had the opportunity to observe even more application areas and technologies. New key elements for analysis and planning were identified; they allowed for the definition of every important step in the development of a Web application through a CMF. After calibrating the method, I tested it on a dataset made by 19 projects, developed between 2009 and 2012 by the three companies. The data on all the previously described elements in the projects were provided during the requirement gathering (estimation level), as well as after the development was finished (final level). My findings show that the application of the method gives very low MRE values: the Web Framework Points methodology has a value of $Pred(0.25)$ equal to 79%, so it satisfies the acceptance criterion by Conte. This means, for my methodology, a good estimation value.

This work represents a confirmation of the good result we obtained in our 2012 work [13], and the base to conduct further experimental validation, with the purpose to make the Web Framework Points methodology a useful tool for software companies.

Bibliography

- [1] Joomla! website, www.joomla.org. [cited at p. 1, 29]
- [2] Drupal website, <http://drupal.org>. [cited at p. 1, 29]
- [3] WordPress website, wordpress.org. [cited at p. 1]
- [4] K. El Emam, A. G. Koru, "A replicated survey of IT software project failures", *Software, IEEE*, vol. 25(5), pp. 84-90, Sept.-Oct. 2008. [cited at p. 1]
- [5] K. Moløkken, M. Jørgensen, "A review of software surveys on software effort estimation", in *Proceedings of the International Symposium on Empirical Software Engineering, ISESE'03*, pp. 223-230, Rome, Italy, 2003. [cited at p. 1, 11]
- [6] T.C. Hooi, Y. Yusoff, Z. Hassan, "Comparative study on applicability of WEBMO in Web application cost estimation within Klang Valley in Malaysia", *Computer and Information Technology Workshops, CIT Workshops 2008, IEEE 8th International Conference on*, pp.116-121, July 2008. [cited at p. 1, 12]
- [7] E. Mendes, N. Mosley, S. Counsell, "The application of case-based reasoning to early Web project cost estimation", in *Proceedings of the 26th Annual International Computer Software and Applications Conference, COMPSAC 2002*, pp. 393-398, Oxford, England, August 2002. [cited at p. 1, 12]
- [8] E. Mendes, N. Mosley, S. Counsell, "Early Web size measures and effort prediction for Web cost estimation", in *Proceedings of the 9th International Symposium on Software Metrics, METRICS'03*, pp. 18-39, Sydney, Australia, September 2003. [cited at p. 1, 12]
- [9] N. Aggarwal, N. Prakash, S. Sofat, "Web hypermedia content management system effort estimation model", *ACM SIGSOFT, Software Engineering Notes*, vol. 32 (2), March 2009. [cited at p. 1, 12]
- [10] L. Mangia, R. Paiano, "MMWA: a software sizing model for Web applications", in *Proceedings of the 4th International Conference on Web Information Systems Engineering, WISE 2003*, pp.53-61, Rome, Italy, December 2003. [cited at p. 1, 12]
- [11] R. Folgieri, G. Barabino, G. Concas, E. Corona, R. DeLorenzi, M. Marchesi, A. Segni, "A revised Web Objects method to estimate Web application development effort", in *Proceedings of the 2nd International Workshop on Emerging Trends in Software Metrics, WETSOM 2011*, pp. 59-64, Honolulu, Hawaii, USA, May 2011, ACM. [cited at p. 1]
- [12] E. Corona, M. Marchesi, G. Barabino, D. Grechi, L. Piccinno, "Size Estimation of Web Applications through Web CMF Objects", in *Proceedings of the 3rd International Workshop on Emerging Trends in Software Metrics, WETSOM 2012*, Zurich, Switzerland, June 2012. [cited at p. 1, 2]

- [13] Corona, E., Concas, G., Marchesi, M., Barabino, G., Grechi, D., "Effort Estimation of Web Applications through Web CMF Objects", 2012 Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement MENSURA, 2012. [cited at p. 1, 2, 59]
- [14] A.J. Albrecht, "Measuring application development productivity", Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium, pp. 83-92, Monterey, California, October 1979, IBM Corporation. [cited at p. 1, 11, 13, 32]
- [15] D. Reifer, "Web development: estimating quick-to-market software", Software, IEEE, vol. 17 (6), pp. 57-64, Nov-Dec 2000. [cited at p. 1, 12, 13]
- [16] S. D. Conte, H. E. Dunsmore, V. Y. Shen, "Software engineering metrics and models", Benjamin-Cummings Publishing Co, 1986. [cited at p. 1, 2, 11, 19, 41]
- [17] Hannay, J. E., Sjöberg, D. I., Dyba, T., "A systematic review of theory use in software engineering experiments", Software Engineering, IEEE Transactions on, 33(2), 87-107, 2007. [cited at p. v, 5, 7, 8]
- [18] Juristo, N., Moreno, A. M., "Basics of software engineering experimentation", Springer Publishing Company, Incorporated, 2010. [cited at p. v, 5, 7]
- [19] J. Aranda and S. M. Easterbrook, "Anchoring and Adjustment in Software Estimation", European Software Engineering Conference/ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE'05), Lisbon, Portugal. [cited at p. 9]
- [20] M. Jørgensen and D.I.K. Sjøberg, "The Impact of Customer Expectation on Software Development Effort Estimates", International Journal of Project Management, 2004, 22: p. 317-325. [cited at p. 9]
- [21] M. Jørgensen and D.I.K. Sjøberg, "Impact of Effort Estimates on Software Project Work". Information and Software Technology, 2001. 43(15): p. 939-948. [cited at p. 9, 10]
- [22] M. Jørgensen and Boehm, B. "Software Development Effort Estimation: Formal Models or Expert Judgment?", IEEE Software, March-April 2009, pp. 14-19. [cited at p. 10, 11]
- [23] J. Scott Armstrong, "Combining Forecasts", The Wharton School, University of Pennsylvania, 2001. [cited at p. 10, 11]
- [24] M. Shepperd and M. Cartwright, "Predicting with Sparse Data", IEEE Transactions on Software Engineering, vol. 27, no. 11, pp. 28-998, 2001. [cited at p. 10]
- [25] E. Kocaguneli, T. Menzies, and J. Keung, "On the value of ensemble effort estimation" IEEE TSE, p. 14p. (in press), 2012. [cited at p. 11]
- [26] B.W. Boehm, "Software engineering economics", Software Engineering, IEEE Transactions on, vol. SE-10(1), pp. 4-21, January 1981. [cited at p. 11, 12]
- [27] COSMIC organization Website, www.cosmicon.com. [cited at p. 11]
- [28] M. Jørgensen, "Forecasting of software development work effort: Evidence on expert judgment and formal models", International Journal of Forecasting, vol. 23 (3), pp. 449-462, 2007. [cited at p. 11]
- [29] M. Jørgensen, M.J. Shepperd, "A systematic review of software development cost estimation studies", Software Engineering, IEEE Transactions on, vol. 33(1), pp. 33-53, 2007. [cited at p. 11]

- [30] Bray, T, "Measuring the Web", Proceedings of the 5th International World Wide Web Conference, Paris, May 1996. [cited at p. 12]
- [31] A.J.C. Cowderoy, A.J.M. Donaldson, J.O. Jenkins, "A metrics framework for multimedia creation", Proceedings of the 5th Software Metrics Symposium, Metrics 1998, pp. 144-147, November 1998. [cited at p. 12]
- [32] G. Barabino, G. Porruvecchio, G. Concas, M. Marchesi, R. De Lorenzi, M. Giaccardi, "An empirical comparison of function points and Web Objects", in Proceedings of the International Conference on Computational Intelligence and Software Engineering, CiSE 2009, pp.1-4, Wuhan, China, December 2009. [cited at p. 12, 13]
- [33] M. Ruhe, R. Jeffery, I. Wiczorek, "Using Web Objects for estimating software development effort for Web applications", Proceedings of the 9th IEEE International Software Metrics Symposium, METRICS '03, pp.30-37, September 2003. [cited at p. 12, 41]
- [34] F. Ferrucci, C. Gravino, S. Di Martino, "A case study using Web Objects and COSMIC for effort estimation of Web applications", Proceedings of the 34th Euromicro Conference, Software Engineering and Advanced Applications, SEAA '08, pp. 441-448, September 2008. [cited at p. 12]
- [35] Reifer D.J., "Web Objects Counting Conventions", Reifer Consultants, Mar (2001). [cited at p. 14]
- [36] Assintel, "Modelli organizzativi nelle aziende di software e servizi di informatica", Assintel, 2000. [cited at p. 39]
- [37] Conradi, R., Wang, A. I., "Empirical methods and studies in software engineering: experiences from ESERNET", (Vol.2765). Springer., 2003. [cited at p. 55]

List of Publications Related to the Thesis

Conference papers

- R. Folgieri, G. Barabino, G. Concas, E. Corona, R. DeLorenzi, M. Marchesi, A. Segni, *A revised Web Objects method to estimate Web application development effort* in Proceedings of the 2nd International Workshop on Emerging Trends in Software Metrics, WETSOM 2011, Honolulu, Hawaii, USA, May 2011.
- E. Corona, M. Marchesi, G. Barabino, D. Grechi, L. Piccinno, *Size Estimation of Web Applications through Web CMF Objects* in Proceedings of the 3rd International Workshop on Emerging Trends in Software Metrics, WETSOM 2012, Zurich, Switzerland, June 2012.
- E. Corona, G. Concas, M. Marchesi, G. Barabino, D. Grechi, *Effort Estimation of Web Applications through Web CMF Objects* in 2012 Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement, MENSURA 2012.

Journal papers

- E. Corona, G. Concas, M. Marchesi, G. Barabino, D. Grechi, *Web Framework Points: an Effort Estimation Methodology for Web Application Development* (based on this thesis).
in Progress