

Estimating, planning and managing Agile Web development projects under a value-based perspective

C.J. Torrecilla-Salinas^a, J. Sedeño^{a,b}, M.J. Escalona^a, M. Mejías^a

^aDepartment of Computer Languages and Systems, University of Seville, Spain

^bAgencia Andaluza de Instituciones Culturales, Junta de Andalucía, Spain

Keywords:

A B S T R A C T

Management
Methodologies
Agile
Scrum
Web Engineering
e-Government

Context: The processes of estimating, planning and managing are crucial for software development projects, since the results must be related to several business strategies. The broad expansion of the Internet and the global and interconnected economy make Web development projects be often characterized by expressions like delivering as soon as possible, reducing time to market and adapting to undefined requirements. In this kind of environment, traditional methodologies based on predictive techniques sometimes do not offer very satisfactory results. The rise of Agile methodologies and practices has provided some useful tools that, combined with Web Engineering techniques, can help to establish a framework to estimate, manage and plan Web development projects.

Objective: This paper presents a proposal for estimating, planning and managing Web projects, by combining some existing Agile techniques with Web Engineering principles, presenting them as an unified framework which uses the business value to guide the delivery of features.

Method: The proposal is analyzed by means of a case study, including a real-life project, in order to obtain relevant conclusions.

Results: The results achieved after using the framework in a development project are presented, including interesting results on project planning and estimation, as well as on team productivity throughout the project.

Conclusion: It is concluded that the framework can be useful in order to better manage Web-based projects, through a continuous value-based estimation and management process.

1. Introduction

Starting a professional software development project soon raises some critical questions such as: How much will the project cost? When will it finish? How much effort must be invested in it? Will the investment be returned soon? What are the features our customers really need?

Being able to answer these questions and some others related to them is crucial for designing business strategies (e.g. financial or commercial, among others) from project results. The responses to the aforementioned questions must condition all decisions like starting one project or another, the type of product to develop and the money that must be invested in it.

It is well known that estimating and planning a development project is a compulsory and complex process [7,18,45]. To face this challenge, traditional estimation techniques focus on a predictive approach [1,12,13,37], which requires a stable and familiar environment. Essentially, these techniques begin with a strong initial requirements gathering phase to freeze user needs [53]. This approach makes these methods especially sensitive to uncertainties and changes of customer needs.

Nowadays, the rise of the Internet and the actual global and interconnected economy has increased the needs for quickly adaptation to changing customer needs. These events have emerged in parallel with the acceptance of Web Engineering as a discipline in Software Engineering [24]. Web Engineering can be defined as a set of methods, techniques and tools in Software Engineering that helps a development team build up systems on the Web. There are several characteristics that differentiate Web projects from the rest of software development projects [24,52]:

- Complex navigational structure.
- Critical interface requirements (such as unknown users or availability, among others).
- Security aspects.
- Increase on maintenance efficiency, avoiding downtimes.
- Delivery as soon as possible.
- Reduction of “time-to-market”.
- Adaptation to quick-changing requirements.

It is important to highlight that some of the aforesaid characteristics are not exclusive of Web development projects and can also appear in non-Web projects. Nevertheless, the concurrence of all of them together at the same time can be identified as a Web project specificity.

In such environments, Agile software development methodologies, with constant monitoring and measurement, and frequent intervention mainly based on the use of empirical processes [61], are turning into a solid alternative for organizations developing software to plan and estimate Web projects [4]. These methodologies offer a suitable framework for the exposed Web development characteristics [55], like quick response to changes, adaptability and reduction of development time [31,50]. In addition, as it has been mentioned, the classical approach regarding up-front requirements gathering demands a stable environment, not being the case of Web projects, where requirements change quicker. The incremental and iterative way of processing Agile methods requirements [18,27] may better fit this particular case.

In contrast, the project management classical approach states that a project succeeds when it combines achieving the goals established on variables such as cost, schedule and scope [53]. Following these criteria, the Standish Group conducts the well-known CHAOS surveys to test the projects success [72]. They define a successful project as the one that is carried out on time, on budget and includes the originally specified features. Fig. 1 shows the projects success level depending on the type of methodology used [73].

As it can be noticed, the percentage of successful projects using Agile methods is significantly higher than that of projects using traditional approaches. These results can be associated with the improvements that Agile techniques bring to project management, for example, “just-in-time” planning, iterative requirements gathering or frequent collaboration. However, it has to be added that, in terms of the above definition concerning a project success, the classical approach leaves behind crucial aspects such as quality and delivered value to customers [33]. They are main issues to address on projects, since they are related to the functionality developed and the kind of process used. Thus, using techniques that allow us to better identify and measure the value delivered

to users will improve the results of our projects. For this purpose, we suggest some techniques to take into account these variables.

Lastly, it must be kept in mind that the number of unused functionality represents expended resources that rarely return to the development organization. A survey conducted by the Standish Group [35], covering 2000 projects carried out by 1000 organizations, showed that more than half of the functionality developed on a project is hardly ever or never used. Fig. 2 shows the results of that survey.

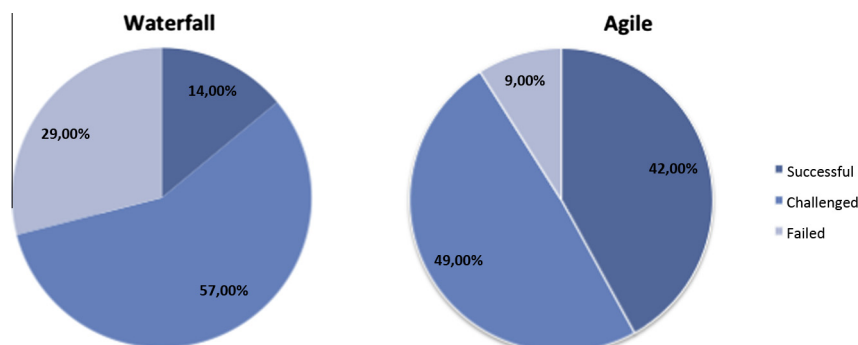
As before stated, the Agile iterative and incremental approach can better fit the special needs of Web projects in order to particularly identify what should be built and when it should be built. This approach will earlier identify these changes and will cope with them more properly to avoid designing unneeded features. It will allow a higher return on the projects investments. Based on the foregoing, this work aims to cover the following objectives:

- Proposing a framework, based on existing Agile methods, to estimate, plan and manage Web projects that, guided by the business value, will help to select what to build, estimating cost and adapting plans to a changing environment.
- Presenting the results obtained from a real experience dealing with applying the proposed framework to a project developed for a Spanish public administration office.
- Taking out the main lessons learned after applying the proposed framework, which will generalize successes and avoid failures as well as will present future lines of work.

This paper is organized into the following sections. Following this introduction, Section 2 presents the research scope, including also the research questions and methodology. Section 3 presents the related work, describing different approaches to estimate, plan and manage Web projects. Then, Section 4 provides an approach to the suggested framework to estimate, plan and manage Web projects based on Agile techniques, whereas Section 5 describes the experience of applying the proposed framework to a real project. To conclude, Section 6 states the conclusions taken out, interprets results and consequences and advances possible future lines of work.

2. Research questions, scope and method

The main question we will try to answer in order to achieve the objectives presented in the previous section is: “*Is it possible to define an Agile approach to estimate, plan and manage Web projects guided by business value?*” As it is very generic, we have tried to decompose it into the following research questions:



Source: The CHAOS Manifesto. The Standish Group. 2012

Fig. 1. Results of projects depending on the methodology used.

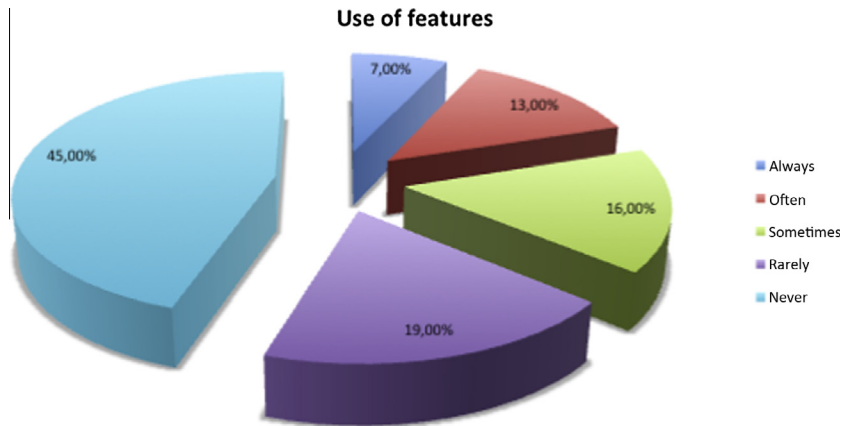


Fig. 2. Use of features and functions in a typical system.

- RQ1: What are the suitable existing techniques for estimating, planning and managing Agile Web projects?
- RQ2: Can business value help estimation, plan and management of Agile Web projects?
- RQ3: Can the identified techniques be integrated into an Agile common framework, appropriate for Web projects needs?

The answer these questions will identify the research scope, that is, the definition of a framework suitable for:

- Agile Web development projects focused on estimation, planning, and management activities.
- Managing Web development projects guided by business value, including continuous improvement throughout the project.

As described by Creswell [21], several factors must be taken into account when selecting a research approach. Creswell also states that a qualitative approach might be appropriate when the research object still needs to be understood because of little existing research. Case studies usually fall on the side of qualitative research, in which an event or activity is presented in depth. Besides, Runeson and Höst [59] analyze the suitability of using case studies in the field of Software Engineering. They put forward in their work that case studies are suitable for exploratory research, in which new ideas to study are sought.

Based on this identified research scope, the used research method includes the following steps:

- Step 1: Identify relevant related work in different contexts, focusing on the following research fields: estimation in Software Engineering, Web Engineering and Agile, Business Value Management in Agile projects and Earned Value Management in Agile projects.
- Step 2: Find out, from the identified related work, those existing techniques that better suit estimating, planning and managing Web projects and business value in an Agile way.
- Step 3: Define a coherent Agile framework suitable for Web projects needs by means of the selected techniques.
- Step 4: Validate the framework initially by means of a case study.

Fig. 3 summarizes the research method.

The case study will be defined and reported attending to Runeson and Höst's proposal, including the following steps:

- Case study definition: The case study will consist in a project to assess the proposed theoretical framework and try to answer the aforementioned research questions.

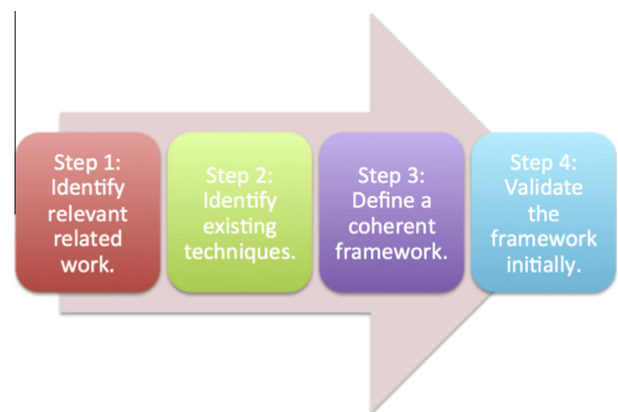


Fig. 3. Research method.

- Data collection: By means of project observations and project metrics retrieval to obtain meaningful data that will be further analyzed.
- Data analysis: It will include both quantitative and qualitative analysis. The former is based on retrieved project metrics and the latter focuses on project observations.
- Reporting: All elements described will be included in the case study report presented in Sections 5 and 6.

Fig. 4 presents the steps to design and report the case study.



Fig. 4. Case study definition.

3. Related work

As mentioned in the preceding sections, this work integrates several research areas that have to be analyzed in order to present a right related work section. In fact, our approach is similar to other approaches regarding estimation models in Web Engineering paradigm under an Agile perspective. As this general context is too young, this section includes an overview of five contexts: estimation models in Software Engineering, solutions in Web Engineering for project estimations, solutions for project estimations in Agile contexts, Business Value Management in Agile projects and Earned Value Management in Agile. Besides, a subsection describing the Scrum lifecycle has been included together with the related work, to better understand the terminology used along the paper.

3.1. Estimation models in Software Engineering

As previously introduced, estimating the effort and cost of a development project is a complex and key process that should be carried out to assess whether a project is good value for money [7]. Many models have been proposed during the last 40 years to face this problem. They can be classified in two main categories: Non algorithm-based models and algorithm-based models. This section presents a high-level overview of some of the most popular models based on these approaches, without including details that are out of the scope of this paper.

Algorithm-based models use mathematical approaches to calculate the project effort as a function of its major cost factors. Some of the most relevant algorithm-based models are listed and described below:

- **COCOMO (Constructive Cost Model):** Proposed initially by Boehm [12] in 1981, its actual version, called COCOMO II [13], was published in 2000. This model uses a basic regression formula and its results are mainly based on code-size (given in thousand lines of code). A tool called Agile COCOMO II [3] has been developed focused on this method, in order to allow the fine-tuning of the estimations according to the analogies among projects.
- **Function Points:** This model, suggested by Albrecht and Gaffney [1], provides a functionality-based measure of the program. Software Functional User Requirements are identified and the total number of function points depending on each one is categorized into one of these five types: outputs, inquiries, inputs, internal files and external interfaces. Once the function is identified and categorized into a type, it is then assessed for complexity and assigned a function point number. There are proposals associated with costs-model for Agile development projects combining story points with function points [68]. There are also case studies comparing function points estimation between Waterfall and Agile projects [69].
- **Use Case Point technique:** Proposed by Karner in 1993 [37], this model is related to Use Case modeling. As Use Case offers the functional scope of the application, the analysis can provide valuable insight on a development project size.

Below, some of the main non algorithm-based models are listed and described:

- **Estimation by analogy** [63]. This method consists in establishing analogies between the actual project and the costs of previous and similar projects.
- **Delphi technique** [42]. This method is centered on a panel of experts view. Several rounds of consultations using questionnaires are executed by providing participants with a consolidated

summary of the previous round results. Then, the experts review their opinion according to the previous results. There is a variant of this method, called Wideband Delphi [12], popularized by Boehm, which involves more participation on the expert's part, as estimations are discussed together in a joint meeting.

- **Expert judgment method** [36]. This method takes advantage of a group of experts' experience and understanding to figure out an estimated cost. This technique is used together with Delphi technique in order to improve and systematize the consulted experts opinion.

We have already stated that two of the main characteristics of Web projects are quick adaptation to changes and complex interfaces. On the one hand, quick adaptation to changes implies several modifications on requirements. The project estimation of algorithm-based models focuses on the estimated absolute size (number of code lines, functional requirements or Use Cases) of the project, which is obtained at the beginning after a strong requirements gathering phase. This could lead to the attempt of "freezing" requirements at the end of this phase. This restriction may not perfectly suit some Web projects, where, as previously introduced, requirements are not completely known at these early phases and users needs could change quickly during the project. On the other hand, as having complex interfaces is another characteristic of Web systems, a collaborative and iterative process between user and development team guided by mockups, wireframes and pilots will be necessary so as to design these interfaces. Due to this fact, the up-front estimation based on techniques as function points or use case points can be difficult to use. Lastly non algorithm-based models based on analogy, can better suit Web projects, since they do not consider the absolute size of the project. However, this kind of estimation usually considers effort, and not the delivered value, which is crucial to select the features that should be built.

As it is known, Agile methodologies estimation models are mainly based on non-algorithm techniques, most of them focused on Wideband Delphi methods (like the common Planning poker technique, which will be described later) [17]. We will also start our approach with this method, although it will be complemented with the relevant techniques to ensure that the most value is delivered early in the project.

To conclude, we can add that the special nature of Web development projects has not been regarded when facing the process of estimating effort and cost. On the contrary, they have been addressed as classical software development projects and they have been applied the same approaches. Nevertheless, as explained, Web projects are different from classical development projects [24,52] and have singular issues to face, both in technical aspects (e.g. complex interfaces, navigation aspects or complex maintenances, among others) and in managerial aspects (e.g. volatile requirements, short schedules, reduced time-to-market or short feedback loops), which cannot be exclusively addressed by means of classical estimation approaches.

3.2. Project estimation in Web Engineering

Since 2002, Web Engineering becomes an accepted discipline in Software Engineering [24]. It can be defined as a set of methods, techniques and tools in Software Engineering that helps development teams build up systems on the Web.

In the last years, several methodologies in Web Engineering area have been proposed. Some like UWE (UML Web Engineering) [38], IFML (Interaction Flow Modeling Language) [47] or WebML (Web Modeling Languages) [14], among the newest, offer new solutions and are widely accepted by the research community. UWE is a Model-Driven Engineering approach focusing on assisting the Web engineer in the different phases of the development

lifecycle. In addition, we can find other proposals, such as the one by Koch et al. [39], which presents a metric to measure the effort reduction resulting from applying this UWE approach. This metric is more centered on calculating the effort reduction than in forecasting the needed effort for a particular project.

Despite the large number of existing approaches, they essentially concentrate on development phases and do not cover other areas like project estimation, quality assurance or team management [28]. Nevertheless, HFPM (Hypermedia Flexible Process Modeling Strategy) [46] and NDT (Navigational Development Techniques) [29] represent two of the most relevant exceptions that can be found. Below, we will briefly describe HFPM and NDT, since defining all the existing models is out of the scope of this paper.

HFPM, proposed by Olsina in 1997, supports the project estimation phase as a task in the project lifecycle. However, it does not offer any specific technique or any technique adaptation to support this estimation, neither classic nor Agile. NDT is another exception. At the beginning, this approach did not support project estimation as a task in its lifecycle. However, with the recent evolution of the approach, it adapts the Use Case point techniques based on the Model-Driven paradigm [7]. A new tool named NDT-Counter was even included in NDT-Suite, the suite tools for NDT, to automate the application of this technique. This metric is based on an absolute magnitude (number of Use Cases), which will not perfectly cover continuous requirement changes. Therefore, this approach is hardly applicable in Agile contexts. Even though NDT supports Agile lifecycles in the development process, NDT-Suite is not appropriate for them, whereas NDT-Counter is completely designed for classical development processes.

Finally, it should be added that Mendes, in her work [45], has analyzed and compared several empirical cost estimation techniques proposed for Web development projects in the last years, most of them dealing with case studies analysis. In the paper by Mendes, eleven Web cost estimation models are analyzed, most of them, like these proposed by Mendes [44], Reifer [57] or Fewster and Mendes [30], are based on estimating absolute magnitudes (such as number of pages, number of links or COTS (Commercial Off-The-Shelf), for example), as they are derivations of algorithm-based estimation models. Some of the problems identified in the study and stressed in the conclusions are the diversity of size measures proposed in the different works and the absence of automated tools to simplify the data collection process.

Mendes's work points out the necessity and difficulty of defining a Web estimation model, due to the special characteristics of Web projects (short schedules, fluidic scope, diversity of technologies, short "time-to-market", diversity and knowledge of profiles involved in the development process and small teams).

As described before, Web development projects are characterized, among other aspects, by the use of short feedback loops, which may cause frequent changes in requirements. Besides, the special characteristics of Web interfaces demand close collaboration and iterative design, which will cause difficulties, if the project is estimated in terms of absolute magnitudes. In addition, it is worth mentioning that Mendes's work detects the trend of increased usage of Agile methods in Web development projects [4]. These two facts enhance the benefits of defining an Agile framework to support the process of estimating effort in Web projects.

Finally, it is important to highlight that none of these techniques focus on the delivery of business value, but only on a pure measure of the projects' size. This approach will not ensure that the most important features are delivered first, shortening "time-to-market", as Web development projects demand.

3.3. Agile estimation and plan

During the last years of the nineties, several methods and techniques based on an iterative and empirical approach emerged in software development projects [40]. The main goals of these practices were, on the one hand, to allow organizations to quickly adapt to clients' changing needs [50] and on the other hand, to deliver valuable results to customer as fast as possible. Some of these techniques, methods and methodologies are Scrum [70], eXtreme Programming (XP) [10], Crystal [15], Lean Software Development [51] or Feature Driven Development (FDD) [49].

The aforementioned methodologies propose a wide-ranging set of techniques to estimate and plan projects, mainly in terms of non-algorithm-based models [18,33]. Most of the Agile estimation techniques concentrate on the use of a popular one called "user stories". It was firstly introduced by eXtreme Programming [10] and then popularized by Mike Cohn [17].

A user story can be defined as a short piece of functionality that provides a customer or a user of a system with a value. A user story represents certain user needs, but not an exhaustive documentation of them. It acts as a reminder and its details are discovered during the collaboration process that runs to develop it at a certain Sprint. The details of the user story are usually recorded in a set of tests that is used to check that the story is finished. A good user story is characterized by a set of attributes, known as the acronym INVEST [10]: Independent, Negotiable, Valuable, Estimable, Small and Testable. Attending to user stories, the most relevant Agile estimation techniques are summarized below:

- **The planning game:** [64] This technique, proposed by eXtreme Programming, assumes that customers have most of the information about what has to be developed and developers have most of the information about how to implement those features. Developers estimate the cost of each feature and customers prioritize the relative importance of each feature among the rest of features. These two steps are repeated until all of the features are estimated and organized. During the process, developers and customers interact by solving doubts about priorities and estimations. Initially, this technique was used to evaluate the work in an iteration, but it can also be used to perform a complete release planning [43].
- **Planning poker:** [18,66] It was also initially used to plan the work in an iteration. Cohn [18,19] and Highsmith [33] propose this technique to perform estimation at the project level, comparing user stories instead of technical tasks. The entire development team estimates a set of features in this technique. Each member of the team has a deck of cards with a discrete subset of values (for instance 1, 2, 3, 5, and 8), representing the points to be assigned to each feature. One by one, the customer representative explains the features. Once a feature is described, the team members can ask some questions to clarify its scope. After that, each member, at the same time, shows a card from his/her deck with the estimation. If they are all the same, the feature receives the estimation; if not, members proposing the highest and the lowest estimation explain their points of view and new rounds are played until reaching a consensus. Then, the team forecasts velocity (meaning the number of points that they can deliver in an iteration, by means of historical data, a "test" iteration or an educated guess) and establishes the length of the iteration. The number of iterations is obtained by dividing the total number of points by velocity. In the same way, the duration of the project is calculated by multiplying the number of iterations by length.
- **Blitz planning:** [15] It is a variation of the Planning Game. In this technique, representatives of all stakeholders meet in the same room and brainstorm the relevant tasks to design the

desired system, by writing them on index cards. Cards are spread on a table and ordered in terms of their priorities and interdependencies. Developers estimate the effort of each task, including any external dependency on them. Then, the team identifies an implementation of the system that performs a small end-to-end feature (called Walking Skeleton), the earliest usable release (called First Delivery) and the first delivery that produces a revenue stream (called Point of Earliest Revenue). Finally, the team looks for excessive workload or a task that can either block the work or entail a high risk.

Although these techniques were initially proposed for general development projects without taking into account the special needs of Web development project, as it has been stated, the characteristics of Agile methods fit some of the Web projects requirements, such as short feedback loops, reduced time-to-market and quick changing users needs. To conclude, it has to be remarked that all of these techniques only face up effort estimation without systematically considering the business value of the delivered features. Both cases, Planning Game and Blitz planning, mention that customers will establish “priorities”, without proposing a concrete technique to do so. Particularly, Planning poker only considers effort and not business value.

3.4. Business Value Management in Agile projects

As mentioned, a critical element on Web projects, in which time-to-market is a key success factor, is building the right features in the right moment. Business value and management in Agile projects might become useful tools to better identify these features. We can find some interesting works studying the relation between Agile and Business Value Management, although as this is a relatively new field, and not many papers have been published on this topic.

Concerning delivery of value, Highsmith [33] proposes using the Agile triangle. This triangle changes the vision of the main constraints of a project: Value, Quality and Constraints (that in turn includes cost, schedule and scope). He suggests the name of “Agile triangle” to identify this set of variables. Fig. 5 shows the Agile project management triangle.

This author recommends estimating and tracking the value delivered by the project by means of the “value points technique”. We will later describe that our framework recommends combining both size and value estimation techniques to better guide the selection of features, which is a crucial element in Web projects success.

Yap’s work [78] presents an experience report describing how a company tried to find a value-based feedback mechanism involving shared responsibility between customer and team. The company applied eXtreme Programming with teams from 6 to 10

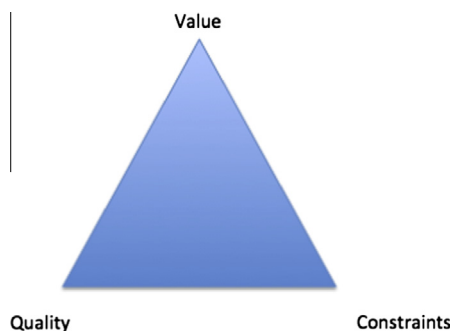


Fig. 5. Agile triangle.

developers. The report explains how the company introduced the “Value Based Investment Decisions” and “High confidence stories first” techniques, together with other practices. The former was based on retrospectives to provide feedback on what project deserved investment and the latter suggested that the stories the team thought they could finish should be first delivered in order to maximize value. The work does not propose a simple quantitative technique that should take into account both customer and developers’ points of view in order to prioritize work, although the recommended techniques can improve the business value delivered both at portfolio and project level.

Register and Golding [58] suggest using Agile techniques, mainly Agile estimating and planning proposed by Cohn [18], in order to maximize the delivered business value in build vs buy decisions. Again, this work specifically focuses on portfolio level, without proposing techniques to decide what work should be prioritized at project level.

Racheva et al. [54] present in their work the results of an exploratory cross-case study on Agile prioritization and business value delivery processes in eight software organizations. One of the research questions that this paper tries to answer is whether companies are using value-based criteria to perform value-driven decisions during Agile prioritization. One interesting conclusion obtained is that some of the companies are not only centered on the value delivered by the feature, but also on the value detracted when the feature is not delivered. In addition, it analyzes the interrelation between customer and developers during the value prioritization. Once again, this work does not suggest any quantitative technique to perform prioritization taking into account both views.

Finally, Logue and McDaid [43] study the Agile release planning process, not only to embrace the viewpoint of developers, but also the business value proposed by customers. Their work, based on the Planning Game of eXtreme Programming, puts forward the use of a business value attribute for each of the stories. In this case, customers suggest three values (optimistic, pessimistic and most likely) for the business value. The development team does the same with the story size. Once the distributions are established, a Monte Carlo simulation is carried out to distinguish how the combined size and business value of the stories are distributed. This way all participants know the likelihood of each potential release plan. This proposal, which mixes the different points of view to define a good release plan, has a drawback; it increases the overhead of the planning, and that could discourage revisiting the plan once the project starts.

To conclude, our framework offers a way of introducing business value in the process of elaborating and managing the release plan on an Agile way, without increasing significantly the overhead of the process to keep it as Agile as possible.

3.5. Earned Value Management in Agile projects

Earned Value Management [53] is a well-known tool that has proved to be useful to manage classical projects as well as to direct projects on regulated environments. However, there is not much literature linking these techniques to Agile. This section presents the related work regarding the usage of EVM techniques in Agile projects.

Alleman et al. [2] present a case study, in which they use the EVM techniques according to their client needs and combine them with some Agile practices coming from eXtreme Programming.

Cockburn, in his book presenting the Crystal methodology [15], includes some comparison between the EVM techniques and some Agile tools, like the burndown chart, but he proposes no modifications in EVM techniques to be used in Agile projects.

Solomon and Young [65] recommend a variation of EVM, named Performance Based Earned Value (PBEV) that brings to EVM important elements, such as risk management or quality aspects, compatible with Agile projects. This approach, being compatible with Agile methods, may require some extra overhead on the top of Scrum projects.

In the work by Sulaiman et al. [67] we find a very complete and detailed proposal related to the usage of EVM techniques in Scrum projects. They analyze EVM magnitudes and advise a way of calculating EVM values at release level. Our proposal is based on theirs, with some variations: first, EVM values are obtained at Sprint level to provide valuable management information more frequently in order to best suit the quick adaptation required in Web development projects; second, we do not use the Agile EVM approach to calculate the release date, but purely to monitor and control the project to keep it under its constraints; and lastly, we maintain the standard EVM terminology. Our variations to this proposal are described in subsequent sections.

3.6. Scrum

In 2001, after the appearance of several empirical software methodologies, some of the most recognized practitioners, such as Kent Beck, Alistair Cockburn, Martin Fowler, Ron Jeffries, Robert C. Martin, Ken Schwaber and Jeff Sutherland, promoted what was known as the “Agile manifesto” [8]. This manifesto included the values and principles supporting the Agile philosophy, which were shared among all methodologies that were called Agile. Some of the basic principles summarized in this manifesto were: the continuous delivery of valuable and potentially shippable software; the existence of small, self-organized teams; the use of short iterations; the inspection and adaptation of the development processes; and the ability to quickly respond to changes.

As it is known, in the last years, Agile approaches based on the “Agile manifesto” values and principles are becoming more popular in software development [11]. As Scrum is one of the most used methods [50] among them, this is the reason why we include it as a base of our proposal.

Jeff Sutherland and Ken Schwaber proposed Scrum [60], being influenced by Takeuchi and Nonaka’s work [71]. It can be defined as a framework for product development [70] and it suggests an iterative and incremental approach for project management. The development process in Scrum is divided into working cycles named Sprints, with a constant size that can differ from 2 to 4 weeks, which is repeated along the project. The main characteristic of these cycles is that they are time-boxed (they start and end with a fixed and expected date, even in the case that the planned work may not be completely finished), and they are ruled one after another, without interruptions until the end of the project.

Scrum projects start with the creation of a Product Backlog. This artifact consists of a list of all the features that can be developed during the project and it is characterized by being ordered and prioritized. The Product Backlog is a living document, which can change during the project. These features can be added, deleted, modified, re-prioritized and so on, in terms of any changes on users need.

As Scrum model states, at the beginning of each iteration, the development team runs a meeting called “*Sprint Planning meeting*”, where all team members, with the help of users and customers representatives, select the amount of work to perform during the Sprint and they commit to carry it out. They use an artifact called Sprint Backlog to facilitate this process. It has to be pointed out that the team commits to work and, at the same time, it is in charge as a group of fully developing the job by the end of the iteration, succeeding or failing as a whole. Everyday, the team checks

the progress during a short time-boxed meeting of fifteen minutes maximum, called “*Daily Scrum*”, to ensure the work is progressing satisfactorily. All impediments and blocks are collected at this meeting with the aim of addressing them.

Once the iteration ends, and in order to guarantee that the resultant product and the development process are supervised and adapted, two meetings are arranged. The first one is called “*Sprint Review meeting*”, a time-boxed meeting where the team presents the results of the iteration to the relevant stakeholders. During this meeting, several new features can be discovered and some of the previous can be modified or discarded. All of these aspects must be reflected in the Product Backlog. The second meeting is called “*Sprint Retrospective meeting*” and deals with reviewing, adjusting and improving the development process the team has executed.

4. An approach to Web Engineering estimation, plan and management with Agile methods

As put forward in the previous section, there are several approaches to software development projects estimation and plan, coming from different fields (classical project management, Agile context or algorithm-based), but none of them individually considers all the special needs Web development projects entail together with a value-based perspective. In the introduction, we advanced that there is a set of characteristics that differentiate Web projects from classical software development projects. Some of these characteristics are: need of reducing the “time-to-market”; value delivery as soon as possible; need of adaptation to continuously changing requirements; improvement of interface requirements criticality; and availability, maintenance and security. We have analyzed that Agile planning methods are appropriate to reduce “time-to-market” and adapt to changing requirements as well as Web Engineering techniques address, among other issues, complex navigation, security and user-interface criticality, among but none of them by themselves can cover all Web projects specificities.

During the preceding sections, some valuable elements have been identified in order to configure a proposal to plan, manage and estimate Agile projects guided by value. Fig. 6 offers an overview of these elements that will compose the proposed framework. They will be described in the following sections.

The framework to estimate, plan and manage Web development projects based on an Agile approach comprises the following characteristics:

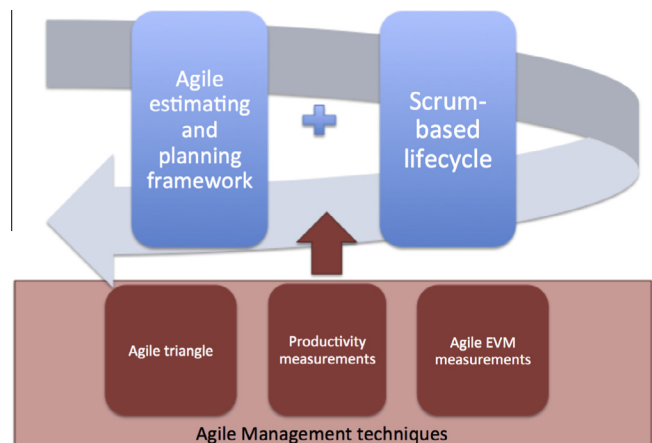


Fig. 6. Elements conforming of the proposed framework.

- The proposed framework is based on Scrum lifecycle, including ad-hoc modifications to fit the special needs of Web projects.
- The proposed framework is based on a non-algorithm estimation model, using Wideband Delphi techniques to initially estimate the project, as proposed by Agile estimation and plan methods. As stated before, these estimation techniques better meet the special characteristics of Web projects.
- The proposed framework is based on an Agile approach, making estimation and plan an iterative and incremental process during the whole project, not only a phase performed at the beginning. This approach matches well the short feedback loops that characterize Web development projects.
- The proposed framework not only considers the classical project management topics of scope, cost and schedule, but also elements like quality and business value, as proposed by the Agile triangle. As previously mentioned, this enables better identification of what should be built, a crucial element in Web development projects.
- The proposed framework uses a variant of the Earned Value Management (EVM) technique to help team members manage the project during the lifecycle and monitor constraints.
- The proposed framework also measures the team productivity with the aim of improving it along the project.

As a first conclusion, it should be highlighted that the proposed framework can be presented as a continuous Plan and Estimate–Manage–Measure–Adapt cycle. Fig. 7 outlines this cycle.

Based on the cited elements, Fig. 8 summarizes the proposed framework to plan, estimate and manage Web development projects with Agile techniques.

As mentioned before, none of the different identified techniques can, on their own, provide an acceptable framework to estimate, plan and manage Web projects in an Agile way, guided by business value. Thus, it can be pointed out that the main contribution of the framework is the combined use of all these techniques, which consequently will be able to cover all the special needs that Web projects demand.

This section describes the framework presented in Fig. 8, analyzing its different phases (planning, estimating, managing and measuring the productivity of the project) in different sections.

4.1. Project lifecycle

The project lifecycle will be divided in two phases, as shown in Fig. 8:

- Project launching.
- Project development.

The Project launching phase will cover the initial estimation and planning effort, described in Section 4.2. The Project development phase will be based on the standard Scrum lifecycle, as it is the most popular Agile approach [50], including the techniques described in Sections 4.3 and 4.4.

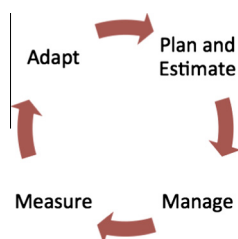


Fig. 7. The proposed cycle.

Taking as a reference the standard Scrum process, our framework will involve the following elements:

- Definition of Product and Sprint Backlogs.
- Sprint planning meeting.
- Daily Scrum.
- Sprint review meeting.
- Sprint retrospective.

As an initial element, the process will include the so-called “Sprint 0” [74] in order to establish the rules that the team will apply throughout the project. This element will be described later on.

On the top of the standard Scrum practices, and in order to keep the Product Backlog updated, 10% of the work-time will be spent in reviewing content [19]. This activity is usually named “Product Backlog grooming”. Sprint retrospectives will be organized following the principles of Agile retrospectives [25,32] including activities and innovation games, such as satisfaction histogram, radar, color dot voting, Ishikawa diagrams or 5 whys, among others. Fig. 9 shows the proposed elements of Scrum to be used.

Together with these practices, it is recommended to establish teams composed of “generalizing-specialists” [5,6], that means, people that can efficiently develop different kind of tasks.

4.2. Planning and estimating Web projects

The proposed framework is particularly based on Scrum framework, which has been previously described, including some other techniques to cover estimation, plan and management processes carried out in Web development projects. Fig. 10 presents the recommended process to estimate and plan Web projects within the framework.

The steps below must be followed to estimate and plan the project, both at Project launching and Project development phases, as an iterative process:

- Populating Product Backlog.
- Estimating Product Backlog size and value.
- Calculating ROI and organizing Product Backlog.
- Setting the iteration length and estimating team velocity.
- Developing the initial Project Plan.

Scrum does not establish what the nature of pieces of functionality, which are part of the Product Backlog, must be. Therefore, in this proposed framework these features are “user stories”, whose main characteristics have been described before. Users stories satisfactorily fit the characteristics of Web projects and they encourage collaboration between users and the development team in processes such as interface and scope definition.

A set of workshops is conducted between users and development teams at the beginning of the project to populate the Product Backlog with user stories. The initial Product Backlog is created at these workshops, including all the identified user stories. Each of them should have the following attributes:

- **Theme:** It represents a category that links a relevant number of user stories.
- **Story ID:** It offers a unique number to represent the story and helps to find and referencing it in an easy way.
- **Description:** It describes the functionality and/or value provided by the story. It should include a reference to the profile for which the story provides a value.
- **Business value:** It stores the business value the story offers. Later in this section, we will describe how this value is assigned.

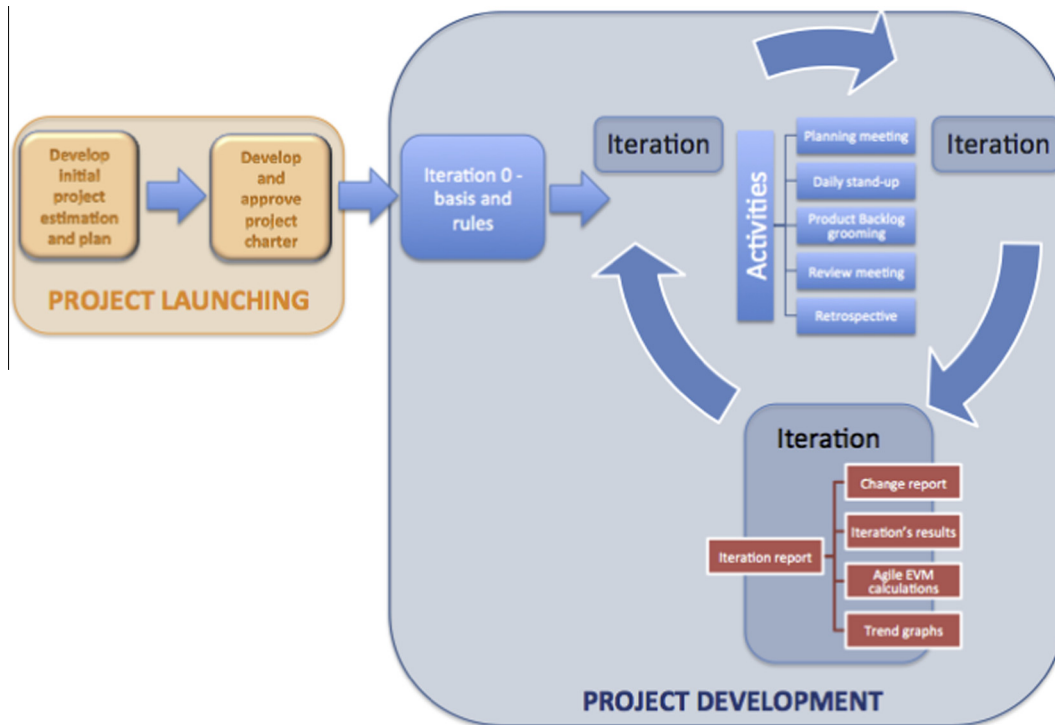


Fig. 8. The proposed framework.

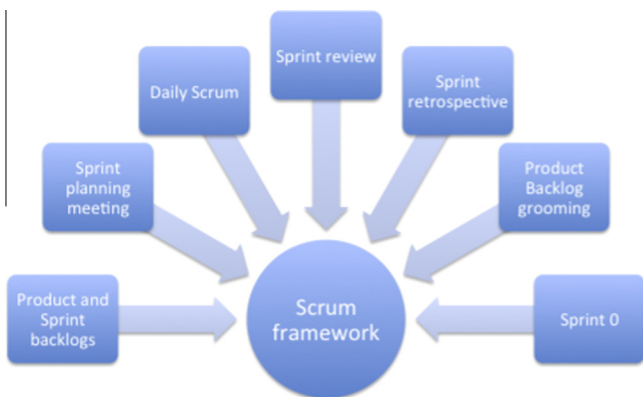


Fig. 9. Used elements in Scrum.

- **Proposed by:** It identifies who proposes the user story.
- **Date:** It specifies the date when the user story is included in the Product Backlog.
- **Comments:** It records any additional comment that can clarify the scope of the user story such as restrictions, dependencies, limitations and special cases to take into account or examples, among others. This is a living attribute that has to be updated during the project as a result of the collaborative relation among users, customers and development teams.
- **How to test it:** It registers a description of any test that helps to assert that the story is really executed. It is used as a basis to automate tests, if appropriate.

The most popular Agile project management systems include most of these properties. For instance, VersionOne [76] offers Theme, Story ID, Description, Size, Proposed by, Date and Comments. Agile plugin for JIRA [34] recommends Theme, Story ID, Description, Business value, Size, Proposed by, Date and Comments. Finally, Redmine [56] gives the possibility of defining custom fields to apply to user stories. It has to be mentioned that none of the systems offers ROI property.

- **Size:** It stores the relative size of the story, in comparison with the rest of stories of the Product Backlog.
- **Return of investment (ROI):** It stores the relation between cost and value provided by each story.

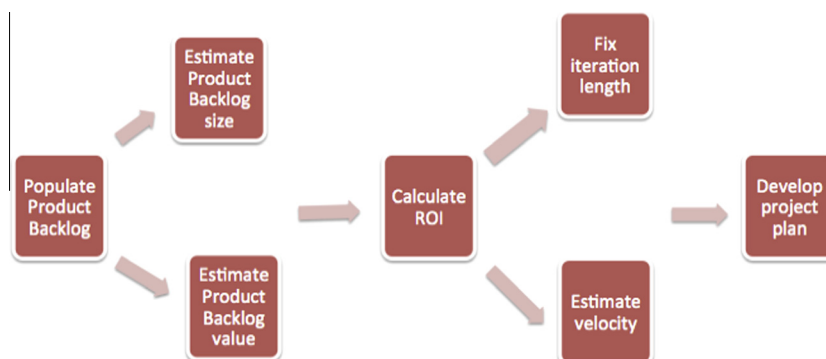


Fig. 10. Agile estimating and planning process within the framework.

During the creation of the Product Backlog, an initial estimation of the amount of needed work and its value is developed. For this purpose, every user story includes an attribute called “Size” representing the estimated amount of work required to finish the story in comparison with the rest of the stories included in the Product Backlog answering the question: “How difficult is to build a certain piece of functionality?”

Classical estimation approaches focus on calculating the total amount of effort needed to finish some task whereas Agile approach proposes evaluating the relative size of a feature in relation to other features by means of an ideal measure unit called “story point” [18]. To compare them, one of the user stories of the Product Backlog is chosen to act as a basis for comparison and the rest of the stories are estimated taking it as reference. The most common approach [18] consists in selecting a small feature to act as a basis for comparison.

It is recommended to pick a discrete universe of measurements [18] to make the estimation process easier. In this particular case, the use of a variant of Fibonacci’s sequence is proposed (0, 1, 2, 3, 5, 8, 13 and 20). This framework advises to use Planning poker as an estimation technique to perform all the estimation process, keeping in mind that only members of the development team can estimate the size of each story. Users or customers work in liaison with the development team by answering questions and clarifying any doubt regarding the scope of the stories.

In addition to the size of each story, the attribute “Value” is also estimated in order to obtain the Product Backlog size. This attribute intends to answer the question: “How much value does this feature bring to the organization?” Using this attribute is very important, as it integrates users opinion, which is crucial in Web development projects. The “Value point analysis” technique [33] is the key to address this process. It deals with assigning a different value to each story from a discrete set of values, by means of different scale (the following values are proposed: 500; 1000; 2000; 5000; 10,000 and 20,000). Only customers, users or their representatives, depending on how the workshops are organized, can assign this value. The total amount of value points that users can give to the Product Backlog is limited, aiming at visualizing the existence of limited resources in each project and ensuring the real prioritization of stories.

At the end of the workshops, each of the elements of the Product Backlog has two values; size (in story points) and value (in value points). The former represents how much the organization has to pay for building the feature (either by investing money or resources) and the latter represents how much value this feature implies. Using both, a relation between cost and benefit of every user story can be calculated, and that value is assigned to the “Return Of Investment” (ROI) attribute of the story [20]. The ROI will present a measure of the cost-benefit ratio, seemed as an easy way to calculate how much the company will pay to obtain the different features of the Product Backlog.

Using this value in the project has two main goals:

- Allow ordering the Product Backlog taking into account both business and development points of view
- Avoid introducing a significant amount of overhead in the estimation process that could impede frequent re-planning cycles.

The equation to calculate this value is the following:

$$\text{ROI} = \text{Value (in value points)} / \text{Size (in story points)}$$

As previously introduced, one of the characteristics of the Product Backlog is that it is an ordered document. In this line, ROI is proposed with the aim of organizing Product Backlog items to address, mainly, those stories that have the highest ratio between the benefits provided and the effort needed. This ordering will

provide an initial proposal for both business and technical teams that might be adjusted in accordance with the needs and constraints (for instance, a feature either represents a competitive advantage or it is a technical constraint). The ordered Product Backlog can be readjusted after the agreement reached by all stakeholders during the release planning discussions.

The team has an ordered list of the initial scope of the project, once this process is finished. Consequently, they are ready to develop an initial project plan. However, it is essential to estimate two important magnitudes to carry it out: the team velocity and the duration of the iteration. Velocity can be defined as a measure of a team rate of progress [18] and represents the amount of story points finished during the iteration. The value can be obtained by means of different methods, like executing test iteration, using historical data from previous projects or forecasting the value. The length of the iterations must be the same during the project to guarantee velocity value consistency. This length is established according to several factors [18] like users need of feedback or changes, or uncertainty of functionality or technology, among others.

Velocity represents the number of story points that can finish in one iteration, therefore, if the total amount of story points of the Product Backlog is divided by velocity, the result will be the initial number of iterations needed to finish the project:

Number of expected iterations

$$= \text{Total number of story points} / \text{Velocity}$$

Once the duration of the iteration has been calculated, the initial estimation for the project length results from multiplying this value by the number of expected iterations:

$$\text{Initial duration of project} = \text{Number of expected iterations} * \text{Length of iteration}$$

Lastly, and by means of the organized Product Backlog, the estimated team velocity and the established duration of the iterations, an initial release plan can be developed by allocating each story in an iteration in terms of the “Return of Investment” attribute value and size [49].

To conclude, it should be reminded that estimating and planning in Agile is not a phase, but a process, thus this initial plan needs to be reviewed and updated in each iteration. Everybody knows that the continuous control and modification of the development process is the key for success in empirical processes [51]. To guarantee this, the Product Backlog needs to be updated (sometimes deleting, adding or modifying stories or some others reassessing their value or size, for example), the real velocity of the team needs to be tracked, and depending on the changes, the project plan also needs to be updated. Next section further explains the proposed techniques to achieve this goal.

4.3. Managing Web projects

This sub-section presents our advised management process for Web development projects, describing at the beginning the management workflow, how the Agile triangle helps to guide the project to succeed, how to use Agile EVM techniques and how to use and present the quantitative data to better manage the project. Our management process is a value-guided quantitative process, which proposes tools to quantitatively identify the most relevant features of the customer so as to deliver them firstly. Ours is a Scrum-based iterative approach, consequently, it will meet the specific Web development requirements in three ways.

- Increasing feedback loops: The iterative approach will provide “check-points” at every Sprint. This will allow adjusting the planning, based on new priorities coming from any of the stakeholders, by means of the value or the size of each story.
- Improving adaptation to changes: The proposed framework will allow quick adaptation to changes, providing the tools to reassess and re-plan at Sprint level.
- Reducing “time-to-market”: As a consequence of the foregoing, the right features will be built in the right moment, with short development cycles.

4.3.1. Management workflow

The management process starts when the initial planning finishes and lasts until the end of the project. Fig. 11 illustrates the management workflow.

As Fig. 10 shows, after developing the initial project estimation and plan, the senior management team of the organization supporting the project must approve it. The general approach for this task deals with making a project charter. It is an executive document, as defined by the Project Management Institute [53], where the main elements of the project are presented, including the cost that carrying out the project implies as well as the main risks.

Developing a project charter is considered a valuable tool in the Agile world, even though its development entails considering the “Barely sufficient” approach proposed by Alistair Cockburn [16]. This project charter may include at least the initial project plan, its main risks and costs as well as the main involved stakeholders. This approach meets properly the characteristics of Web projects, as they are normally short projects with short “time-to-market” requirements, not having time to be spent in long documentation phases.

Once approved, the project starts running iteration-by-iteration, trying to keep a sustainable and continuous development rhythm. However, first of all, a special “Sprint 0” is executed [74] in order to establish, among others, the rules and criteria for quality management. This “Sprint 0” is a time-boxed iteration, lasting the same period of time as the rest of Sprints, where the team establishes the general basis and rules for the project: quality assurance, project data management, “definition of done” and project measurement and analysis. When it concludes, the normal iteration-guided lifecycle of an Agile project begins with considering that all quality and management data established during “Sprint 0” is gathered during ordinary Scrum meetings.

4.3.2. Usage of Agile triangle

As before stated, the traditional approach to project management usually takes into account cost, schedule and scope

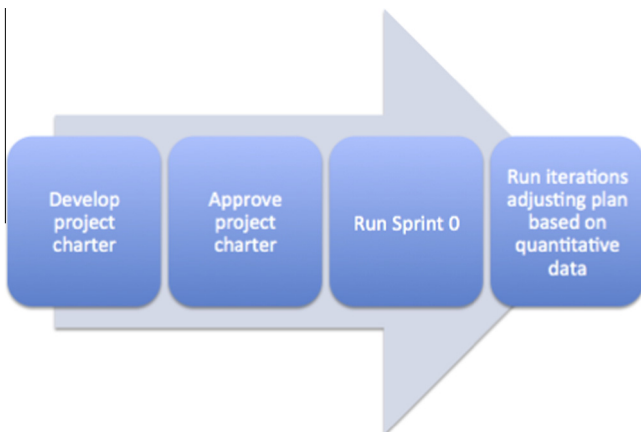


Fig. 11. Management workflow.

restrictions by means of a tool known as “Iron triangle”. Nevertheless, achieving the goals over those variables successfully does not guarantee that the most valuable features are delivered to customers and final users. This fact is even more important in Web development projects, as they are normally short projects carried out by small teams, where developing or not the right feature implies the success or failure of the project.

We previously referred that the Agile triangle recommends the usage of Value, Quality and Constraints as guides to assess the success of a project. Our framework proposes a set of techniques that helps to track the goals achieved during the project to successfully manage these variables. The necessary data is gathered during the previously cited meetings (Sprint Planning Meeting, Daily Scrum, Sprint Review and Sprint Retrospective) in order to reduce the overhead as much as possible. They are estimated at iteration level to allow the team to modify the process with the aim of improving results.

The first of the proposed restrictions is “Value”. Each user story is evaluated during the initial planning and assigned a value given in value points by means of “Value Points analysis”. Then, a “Return of Investment” of each story is calculated through this value as well as the size of each story. The present framework proposes both to structure the Product Backlog in a descending order to that of the “Return of Investment” attribute and develop those stories that have the closest relation between value given and cost needed. This approach provides teams with a quantitative measure of the value that each feature assumes for users, not only at the start of the project, but also through all Sprints. In consequence, the team can focus on the right feature in the right moment, what represents one of the key success factors in Web development projects.

Tracking the percentage of achieved business value at the end of each Sprint is suggested as an additional tool. This figure is calculated by means of the following formula:

$$\% \text{ Delivered Value} = \frac{\sum \text{Value of finished stories/}}{\sum \text{Value of all stories on Product Backlog}}$$

A chart representing its evolution through iterations is used to offer this value. Fig. 12 shows an example of this chart.

This tool tends to help the team deliver most of the project value during the initial iterations of the project. The case study presented afterwards will confirm that the calculations of these metrics do not imply significant overhead to the project. On the contrary, they constitute a useful tool to guide the development.

The second variable proposed by the Agile triangle is “Quality”. It has to be mentioned that within Agile philosophy “Quality has to be built-in along the project” [51]. When talking about quality, its two faces must be reminded: product quality and process quality,

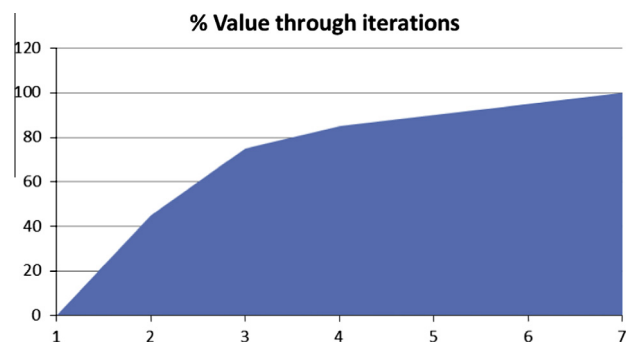


Fig. 12. Evolution of the value delivered through iterations.

being necessary to measure and manage both during the project. Agile proposes several techniques, most of them based on “Test-first” philosophy, such as Test-Driven-Development [9,10], to guarantee product quality. However, the detailed description of these techniques is out of the scope of this paper. With regard to quality, it has to be added that only the story points of stories that pass the defined test and customer acceptance count on team velocity.

Retrospective meetings are the main tools to direct the process quality. During these meetings, the established development process is checked through a systematic analysis by means of innovation games and other techniques, with the aim of tracking the real root causes of the identified problems and impediments. Some metrics, dealing with discussions on these meetings, are introduced in the following section. Our framework recommends collaborative tools, like Wiki pages, in order to collect the conclusions of each retrospective meeting and track the solution along the following iteration of the project.

The last magnitude included in the Agile triangle is “Constraints”. This magnitude involves all classical project constraints, like scope, cost and schedule. Our framework suggests a variation on the Agile-based approach to EVM techniques proposed by Sulaiman et al. [67] to follow the evolution of this variable. As it is known, these techniques allow measuring the relationship among cost, scope and schedule in the course of a project [53]. EVM proposes two main indexes to track the relations among constraints. These indexes are *Cost Performance Index* (CPI) and *Schedule Performance Index* (SPI). CPI appraises the relation between estimated cost and real cost and SPI determines the relation between estimated schedule and real schedule. If the values of these indexes are close to 1, it means the project is evolving as it was conceived.

Our framework advises measuring both indexes at the end of each Sprint and analyzing them in the retrospective meeting. That would help to identify problems or obstacles early during the project execution and correcting them in the forthcoming iterations.

4.3.3. Agile Earned Value Management

The following data is calculated at the beginning of the projects, with the aim of obtaining Agile EVM values:

- **Estimated average cost of the team per hour:** It represents the average labor cost of team per hour.
- **Estimated number of hours per iteration:** It is obtained when planning the first iteration, and it varies depending on the number of team members and the hours they work in the project.
- **Estimated average cost per iteration:** It is the result of multiplying the estimated average cost per hour by the estimated number of hours per iteration.

Estimated average cost per iteration

= Estimated average cost per hour of team

* Estimated number of hours per iteration

- **Budget At Completion (BAC):** It refers to the total estimated cost of the project. It is the result of multiplying the number of expected iterations by the estimated average cost per iteration.

BAC = Estimated average cost per iteration

* **Number of expected iterations**

The following parameters are calculated at the end of the iteration, based on previous values:

- **Expected Percent Completed (EPC):** It is obtained by dividing the planned completed story points at the end of the iteration by the total amount of story points of the Product Backlog.

$$EPC = \frac{\sum \text{Planned completed story points at iteration } i}{\sum \text{Total stories on Product Backlog}}$$

- **Actual Percent Completed (APC):** It is obtained by dividing the real completed story points at the end of certain iteration by the total amount of story points of the Product Backlog.

$$APC = \frac{\sum \text{Real completed story points at iteration } i}{\sum \text{Total stories on Product Backlog}}$$

- **Planned Value (PV):** It is obtained by multiplying EPC at certain iteration by the BAC of the project.

$$PV = EPC \text{ at iteration } i * BAC$$

- **Earned Value (EV):** It is obtained by multiplying APC at certain iteration by BAC of the project.

$$EV = APC \text{ at iteration } i * BAC$$

- **Actual Cost (AC):** It is obtained by multiplying the Estimated average cost of the team per hour by the real number of working hours at the end of certain iteration.

$$AC = \text{Estimated average cost of the team per hour} * \text{Real number of working hours at iteration } i$$

- **Cost Performance Index (CPI):** It is obtained by dividing EV by AC at the end of certain iteration.

$$CPI = EV/AC$$

- **Schedule Performance Index (SPI):** It is obtained by dividing EV by PV at the end of certain iteration.

$$SPI = EV/PV$$

At the end of each Sprint, the number of finished story points and working hours dedicated to the project is tracked and the values of the exposed magnitude are calculated. Once analyzed, corrective actions are performed in order to adapt the project to the new situation.

4.3.4. Management reports

One of the most relevant things to handle during a project concerns managing expectations. This framework recommends tools like burn-down charts, burn-up charts, velocity charts or change reports, among others [18, 62], to address the challenge and inform the relevant stakeholders on the project's progress during the iteration.

A special report is generated to inform the relevant stakeholders at the end of the Sprint. This report might include the following information:

- **Change report:** It refers to stories that have changed during the iteration (e.g. finished, non-finished, discarded, re-estimated or included in the Product Backlog).
- **Iteration result:** It deals with story points and value points finished, percentage of value delivery or real number of working hours dedicated to the project.
- **Agile EVM calculations:** It is used to show the status of the project in relation to the designed plans.
- **Iteration burn-down chart:** It shows how the work progresses during the iteration.
- **Project burn-down chart:** It represents the evolution of the real completed work of the project regarding the expected completed work.

- **Project burn-up chart:** It offers the evolution of the real completed work in the project in relation to the remaining work in the Product Backlog.
- **Evolution of delivered value through iterations:** It represents how much value is being delivered to customers and users.

This report can be generated as a printable document or can be simply posted in a collaborative tool (like VersionOne [76]) or in the team room for consultation. It plays the role of “information radiator” [15].

Most of the reports are available in tools like VersionOne, with the exception of those regarding Agile EVM calculations and the evolution of delivered value through iterations. These can be easily generated with Excel spreadsheets.

4.4. Measuring productivity in Web projects

This framework proposes tracking some productivity metrics to motivate teams to improve their performance during the project development. As Web development teams are normally small groups of people, it is important to take most of them to deliver value to customers.

One of the goals of Agile projects is to maintain a lightweight process, which enables adapting to changes that provide customers and users with competitive advantages. These metrics, suggested by Downey and Sutherland [26], can be obtained without increasing overhead expenses of the process during the meetings, since the finished story points and dedicated working hours are only needed to calculate them. Metrics are evaluated per Sprint and analyzed during the Sprint retrospective. Our framework includes the following metrics from Downey and Sutherland’s set of metrics:

- **Team velocity (in hours):** It is defined as the sum of the finished story points in certain iteration multiplied by the average number of hours per story point. It represents the average amount of functionality delivered by the team per iteration.

Team velocity(hours)

$$= \sum (\text{Finished story points in iteration}) \\ * \text{Average number of hours per story point in project}$$

- **Work capacity:** It is defined as the sum of working hours spent during certain Sprint, whether the user story in which development the hours were spent in was finished or not. It represents the average cost spent by the team per iteration.

Work capacity = \sum Working hours in iteration

- **Focus factor:** It is defined as the team velocity measured in hours divided by the work capacity. It represents the relation between the dedicated working hours and velocity in hours. Besides, it shows whether the team is over or under the forecasted capacities.

Focus factor = Team velocity(in hours)/Work capacity

- **Percentage of accepted work:** It is defined as the result of dividing the working hours dedicated to finished user stories during certain iteration by the total dedicated working hours of the iteration. It represents the percentage of working hours dedicated to deliver features to the user in certain iteration.

% of accepted work

$$= \frac{\sum (\text{Dedicated working hours to finished stories in iteration})}{\sum (\text{Total dedicated working hours in iteration})}$$

- **Target Value Increase (TVI+):** It is defined as the finished story points of the actual iteration divided by the average story points of all finished iterations. It represents the team productivity increase in relation to the average performance throughout the project.

$$\text{TVI+} = \frac{\sum (\text{Finished story points in iteration})}{\text{Average story points per iteration}}$$

Tracking these metrics will uncover the number of working hours along the iteration; the number of features that will be delivered; the measure of the team’s forecasted capabilities; the percentage of the team’s effective work that produces features to deliver and the team productivity improvement. These values will allow the team to self-manage. As it has been stated, this framework proposes to calculate them per Sprint and analyze them during the iteration retrospective. Guiding discussions will enhance finding out the root of each impediment or problem and modifying the process to increase results.

To summarize, it has to be mentioned that our main contribution is the combined use of the described productivity metrics together with the usage of the rest of Agile techniques, mainly Agile EVM. The aim is to propose a direct streamlined lightweight systematic process to improve the team productivity through Sprints.

5. A practical example

This section describes a practical sample of the proposed framework, being our first empirical experience regarding the application of this framework. A single experience can or cannot validate it, although it can help to provide the first insights for our further research. Some of the results described in this section have been already presented in our previous paper [75]. That was focused on assessing the suitability of Agile practices in Public Administration and its adaptability to different kinds of projects (Web projects and infrastructure projects), giving only high-level details of the project.

In this section, the full project and its results are described in depth. The example, called *eBOJA* project, was developed within the Ministry of Culture and Sports of the Regional Government of Andalusia (Junta de Andalucía), in Spain. It is important to mention that all economic data included in this section are pure estimations used for management purposes, not representing any real expenditures.

5.1. The background of the project

Junta de Andalucía is the name of the regional government of the Spanish region of Andalusia, the body that has developed this project. The Information and Communication Technologies (ICT) Department of the Ministry of Culture and Sports has led the project in liaison with other Departments belonging to the regional government.

The *eBOJA* project is mainly an e-Government project. The main legal framework to develop this kind of projects within the Spanish Public Administration is the Law 11/2007 [41]. According to the principles it establishes, Junta de Andalucía has developed an e-Government framework called W@NDA [77]. W@NDA project’s axes focus on process reengineering and administrative procedure simplification, with the aim of reducing costs and time and improving citizens’ satisfaction. Additionally, the Official Journal of Junta de Andalucía (BOJA) is the name given to the official journal of the regional government [22,23], being issued on paper along the last thirty years. Some years ago, Junta de Andalucía

started a project to develop the necessary infrastructure to transform this journal into an electronic one, keeping all legal warranties. As a result of this initiative, the BOJA is issued electronically, with total legal validity, since May 10th 2012 [48].

Once the infrastructure is available, all institutional bodies of Junta de Andalucía will adapt their systems and processes to use it. The ICT Department of the Ministry of Culture and Sports started *eBOJA* project, to cope with that goal. Two main systems have been used within the project: a Web application for internal communications and a Web application to electronically sign official documents, both of them developed under the aforementioned *W@NDA* project. *eBOJA* project included, among others tasks, the following: design of administrative procedures, deployment of applications within the Ministry's infrastructure, development of software APIs to protect the internal systems, interconnection of the applications with the general infrastructure or all the aspects related to change management.

It is worth highlighting that this is a key system, because all regulations approved by the Ministry shall be published through this tool. In contrast, the large number of involved stakeholders has given the project an extra difficulty: managing a broad set of stakeholders, sometimes with opposed interests, as well as managing a big change in an uncertain environment, has posed a great challenge.

5.2. The technical context of the project

As it was previously introduced, *eBOJA* project is part of the e-Government strategy of the Ministry of Culture and Sports, aligned with the general e-Government strategy of Junta de Andalucía. Fig. 13 shows the main components that interoperate within the project.

Below, we will briefly describe the elements conforming this figure:

- A workflow engine for administrative procedures, called TREW@. Based on it, a Web tool to operate the procedures, called eCO, which is a Web application centered on JEE architecture that uses Spring, Hibernate and JSF, runs as a front-end of TREW@ (a.1). It also includes a graphic design tool to develop administrative procedures, based on AWT, where these procedures are designed through an XML-based language called XPDL (eXtended procedure Description language) (a.0). TREW@ contains an Oracle database schema and Java API to programmatically have access to the defined procedures.
- A Web application to electronically sign documents, called PORTAFIRMA. This Web application based on JEE architecture, using Spring, Hibernate and JSF. PORTAFIRMA compiles all the functionality required to electronically sign any administrative

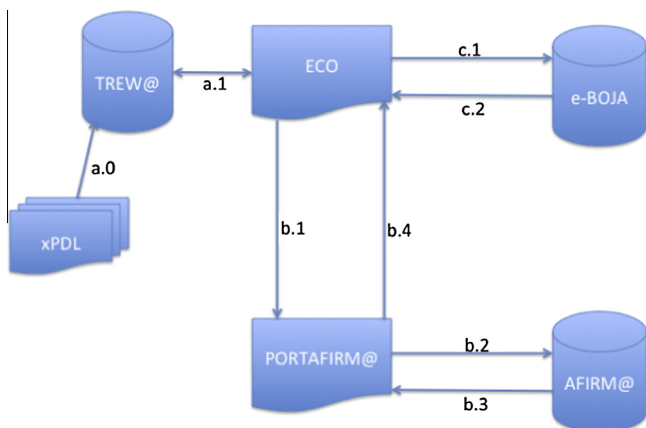


Fig. 13. e-Government solution architecture for eBOJA.

document. It acts as the corporate multi-PKI platform front-end for authentication and electronic signature of Junta de Andalucía, called @FIRMA [77] (b.2, b.3). This platform is used by all Spanish public administrations, providing them with fast and effective authentication and electronic signature services. PORTAFIRMA communicates with TREW@ workflow engine via DSS-complaint Web Services (b.1, b.4).

- A Web application to electronically submit documents to the Official Journal of Junta de Andalucía, called ELECTRONIC REMISSION. This application receives the documents sent by eCO through a Web Services layer (c.1, c.2). This platform has been developed and it is currently maintained by the infrastructure of the Ministry of Presidency. It is carried out through JEE architecture, using Spring and Hibernate.

5.3. The organizational environment of the project

This section summarizes some of the environmental characteristics of the project, pointing to the different elements influencing it. Fig. 14 outlines them as follows.

The most influencing element on the project is the composition of the team. In the case of *eBOJA*, the project comprised a team of four members. Such a team was recognized by the following characteristics:

- To be a real multi-disciplinary team, with different level of experience and knowledge.
- To be used to work as a team and to self-organization.
- To be integrated by experts in their own fields to guarantee the knowledge of the basic tasks that they would be able to develop.

It must be highlighted that, with the exception of one member, no other team member had real experience in Agile methodologies. Before starting the project, the team members received a training session regarding the grounds of Agile and during the project they were coached by the one having Agile expertise.

Furthermore, *eBOJA* faced one main constraint, the schedule. That meant a strict deadline fixed by law established by the Regional Government. For that reason, delivering the project on time was one of the main goals of the team.

Additionally, *eBOJA* was developed thanks to the collaborative work performed among several stakeholders:

- The ICT Department System Operation teams, who were in charge of some of the necessary tasks to deliver users stories on time.
- The Directors Board of the Ministry, who was responsible for supporting the team during release and change management processes.

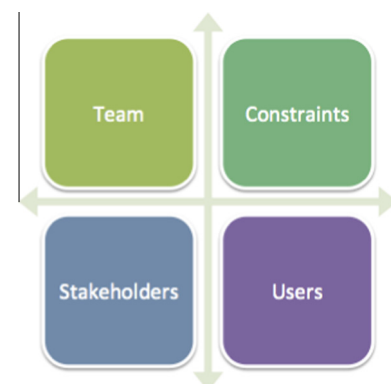


Fig. 14. Elements having impact on the project.

- Several other Ministries, who were in charge of delivering on time some of the components used in the project.

The large number of stakeholders (for instance the different Ministries involved) gave this project an extra difficulty, apart from the technical aspects.

Lastly the project influences both internal and external users:

- From internal user perspective, *eBOJA* represents a great change in the Ministry's internal users behavior since, before being implemented, all tasks dealing with publishing laws and regulations were manually issued on paper. The deployment of the system implies a real change of paradigm, which has to be carefully handled.
- From external user perspective, mainly citizens and companies, the project is relevant since all laws and regulations approved by the Ministry of Culture and Sports shall be published through electronic means. As examples, the system manages both, all procurement processes and all financial aids granted to the cultural industry.

5.4. Results

The project started in December 2011 with the project-launching phase, including the development of a project charter and two workshops performed in the first weeks of January, to allow the team defining the required features in the shape of user stories. The initial Product Backlog included 14 user stories, scoring a total of 56 story points. Four more user stories were added in the course of the project, one of the original was deleted and some of the initial estimations were updated. These facts entailed that the Product Backlog size ranged from 74 to 56. Using Agile planning and estimating framework, the next step was establishing the duration of the iteration and velocity. The iteration length was fixed in three weeks and, because of the absence of historical data, team velocity was forecasted taking into account the available working hours of each team member during the iteration. [Table 1](#) shows the initial forecasted velocity.

The team developed an initial project plan and estimation cost after establishing the length of the iteration and team velocity. [Table 2](#) summarizes the main aspects of the initial plan, including a level of uncertainty suggested by the actual phase of the project [18].

Sprint-0 started in January 30th 2012, after defining the initial plan. The team was able to lay the foundation for managing the project with this first iteration. Once finished, the normal Sprint-based lifecycle started. [Table 3](#) represents the main figures of the project.

As it can be observed in [Table 3](#), the project included 5 iterations (4 of them lasting three weeks and one four weeks), being finished in June 23rd 2012. It has to be mentioned that the Ministry of Culture and Sports was the first body of Junta de Andalucía in adapting the systems to the new official journal. Regarding the last Sprint, it lasted an extra week, in order to include all remaining stories, avoiding a short final Sprint of one week.

As mentioned, size, composition and priorities varied during the project and ROI was a key element to decide over Product Backlog changes. The new features were included during Product Backlog Grooming sessions, where business representatives provided them

with value estimation. The team estimated their size and calculated their ROI. If the ROI of one of the story was higher than a previous one, then the Product Backlog would be reordered and the Release Plan changed. [Table 4](#) presents the results of the project in relation to the values included in the initial plan.

Regarding Agile management tools used, [Fig. 15](#) presents the burn-up chart of the project, with the evolution of the remaining amount of work in comparison with the finished amount of work, both measured in story points.

This tool shows how much work was finished attending to the remaining amount of work. It reflects the changes on the size of the project caused by the continuous reassessment of priorities, using value and ROI as main tools. [Table 5](#) shows the evolution per iteration in order to measure the delivered value.

Based on these data, [Fig. 16](#) lets us know the accumulated business value delivered per iterations.

As it was already mentioned, the main goal of this tool is helping the team deliver most of the value at the beginning of the project. It can be noticed that this goal was only partially achieved due to several reasons, such as the necessity of delivering first some technical stories not providing clear value to the customer or the initial fluctuation in velocity (as the value delivered was linked to the finished stories and during the first three iterations the value changed).

Moreover, [Table 6](#) shows EVM calculations for each of the project's iterations as an Agile management tool in order to control the project constraints.

[Table 6](#) also reveals how the different indicators used to control project constraint were evolving through Sprints. It is worth pointing out that these indicators were used as a main tool to learn through the project in the Sprint retrospective meetings. They proved to be very useful, and as shown, they later helped to stabilize the project in the last Sprints. [Fig. 17](#) demonstrates the evolution of the planned value against the earned value.

It can be observed that the team was overcommitted during the first three iterations. Specially, in Sprint 3, the team was not able to deliver what was committed, because of some uncertainties on the requirements definition and lack of knowledge on the team part. From iteration 4 on, the team improved their estimations. As mentioned, this stabilization can be caused by the different project indicators described in the proposed framework, as they were used to re-estimate the Product Backlog during the different Sprint meetings. [Fig. 18](#) represents the evolution of the earned value against the actual cost.

It shows that the team tended to overestimate the necessary amount of work. This element is important, as confirms that the team improved its work, but keeping a conservative approach during the last Sprints of the project. The team discussed about this fact during the Sprint retrospective meetings based on the data, but it decided to keep a conservative approach during the whole project, probably due to either a natural tendency to self-protection or, as it was its first Agile experience, some lack of knowledge about the context or the framework. [Fig. 19](#) shows the evolution of CPI and SPI indexes through the project's iterations.

Lastly, [Table 7](#) offers the results of the productivity metrics obtained during the project estimated at iteration level, which allowed the team to assess and adapt its behavior throughout the project.

[Table 8](#) offers the average data of each metric for the entire project.

One relevant thing that can be highlighted is how the team improved its performance, as it can be seen in the average value of TVI+, which is above 100%, even with oscillations in team velocity during Sprint 3. Another figure that can be emphasized is the percentage of accepted work, which shows how the team was able to focus on valuable work in the project.

Table 1
Initial forecasted velocity.

Available hours	Estimated initial velocity	Hours per story point
150	11	13.64

Table 2
Summary of the initial project plan.

	Value	Uncertainty ($\pm 25\%$)	Initial estimation
Total No of story points	56	14	56 \pm 14
Velocity (forecast)	11	3	11 \pm 3
No of expected iterations	5	1	5 \pm 1
Length of iteration	3 weeks	N/A	N/A
Duration of project	15 weeks	3 weeks	15 \pm 3 weeks
Hours per iteration	150	38	150 \pm 38
Hours per Project	764	191	764 \pm 191
Cost per iteration	€ 3694.34	€ 923.59	€ 3694.24 \pm € 923.59
Total Project cost	€ 18807.55	€ 4701.89	€ 18807.55 \pm € 4701.89

Table 3
Main figures of the project.

Sprint	Start date	End date	Working days	Est. velocity	Real velocity	Finish. story points	Product Backlog story points.	Remain. story points	Estimated hours	Real hours
1	30/01	20/02	15	11	6	6	56	50	150	117
2	21/02	14/03	15	14	14	20	74	54	126	79
3	15/03	09/04	15	12	2	22	74	52	150	88
4	16/04	07/05	15	10	15	37	71	34	147	118
5	08/05	23/06	20	19	19	56	56	0	101	157
									674	559

Table 4
Project results vs initial plan values.

	Results	Planned
Average velocity	11.20	11 \pm 3
Total working days	80	N/A
Total working hours	559	764 \pm 191
Total story points	56	56 \pm 14
Hours per story point	9.98	13.64

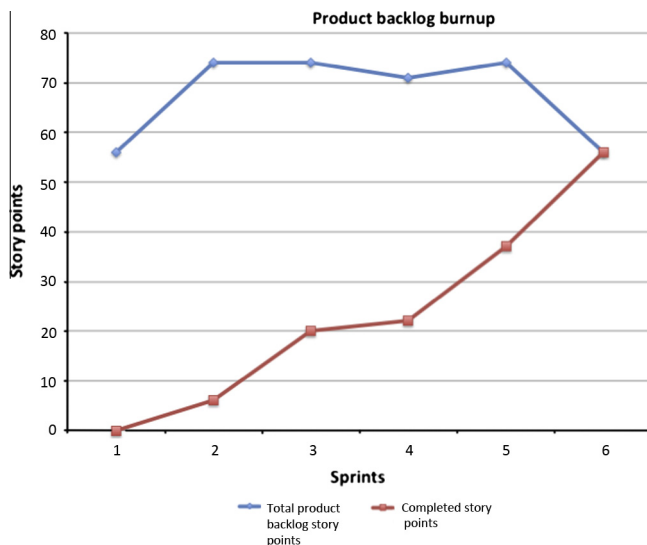


Fig. 15. Burn-up chart of the project.

Table 5
Delivered value by Sprint.

Sprint	Value of finished stories (estimated)	Value of finished stories (real)	% Delivered value (%)	% Delivered value (accumulated)
1	13,000	11,000	16.18	16.18
2	24,000	24,000	35.29	51.47
3	6500	1000	1.47	52.94
4	11,000	11,500	16.91	69.85
5	20,500	20,500	30.15	100.00

Another important aspect that can be extracted from the presented values is that they show an overview of the team performance along the project. This fact clarifies whether some negative values are only isolated cases or are related to underlying problems. Fig. 20 represents the relation between team velocity and work capacity.

It is realized that during the project, the relation between these two magnitudes was primarily above 100%. Normally, a value higher than 80% reveals a tendency to overestimation [26]. This fact is aligned with the results obtained from EVM calculations. Fig. 21 confirms the evolution of the percentage of accepted work.

This value is between 80% and 95%, which means the team mainly focused on finishing work concerning delivered features at the end of each Sprint. Fig. 22 outlines the evolution of TVI + during the project.

The team improved its capacity to deliver a value to the customer in each Sprint, with the exception of Sprint 3.

To sum up, the correct use of this value can help any team know its situation, and within the Sprint retrospectives, adapt and improve estimations and performance.

5.5. Discussion on the results obtained

In this section we will analyze the results presented in the preceding section, in order to help us come to general conclusions. As mentioned, both quantitative (through project metrics) and qualitative (through project observation) data have been gathered, and this section will address both of them separately.

5.5.1. Quantitative data analysis

This section will analyze the project metrics presented in Section 5.3 with the aim to obtain meaningful information.

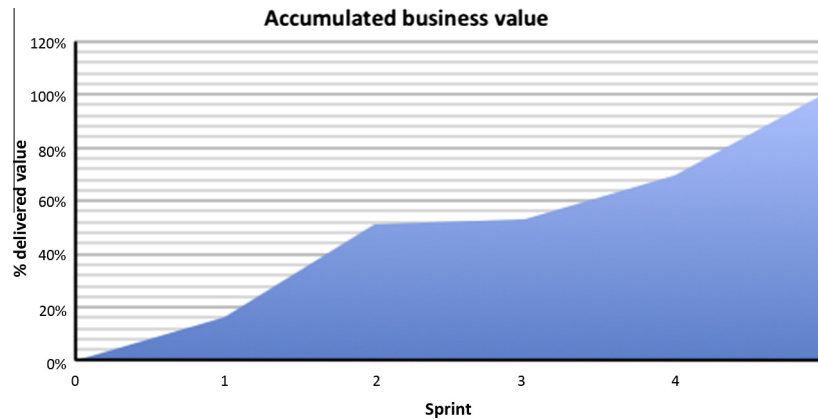


Fig. 16. Delivered value per Sprint.

Table 6
EVM calculations.

Sprint	Plan. compl. story points	EPC (%)	APC (%)	Estimated work hours	Real no. of working hours	PV	EV	AC	SPI	CPI
1	11	19.64	10.71	150	117	€ 3694.34	€ 2015.09	€ 2880.54	0.55	0.70
2	14	35.71	35.71	126	79	€ 6716.98	€ 6716.98	€ 4825.52	1.00	1.39
3	12	57.14	39.29	150	88	€ 10747.17	€ 7388.68	€ 6992.08	0.69	1.06
4	10	57.14	66.07	147	118	€ 10747.17	€ 12426.42	€ 9897.24	1.16	1.26
5	19	100.00	100.00	101	157	€ 18807.55	€ 18807.55	€ 13762.58	1.00	1.37

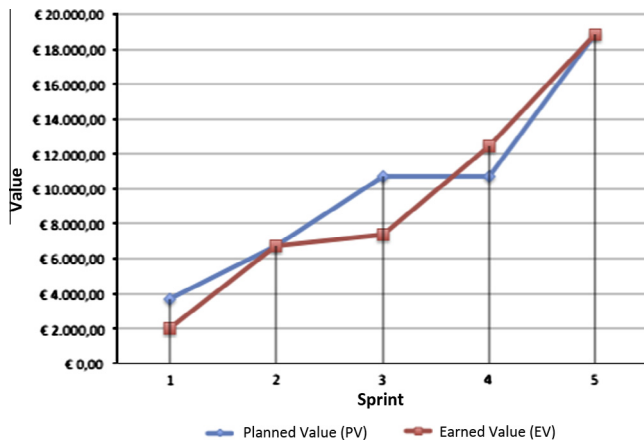


Fig. 17. Planned and earned value per iteration.

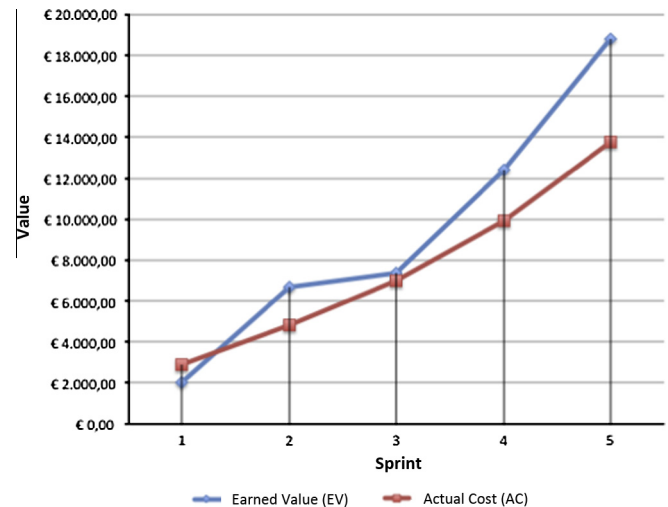


Fig. 18. Earned value and actual cost per iteration.

As an initial thought, Fig. 17 states that the framework allows the team to improve estimation regarding scope through Sprints, becoming a valuable learning tool. This framework also provides frequent feedback loops that are very suitable for Web development projects requirements. As the data presented in Table 5 displays, modifications on the Product Backlog (stories added, deleted and adjusted) are guided by the value that users and customers assign. This enables reassessing and reordering the Product Backlog and Release Plan throughout the project, and being able to deliver not only the planned system, but also the desired system by the end of the project.

The framework executes project plan and estimation as a continuous task along the project, not exclusively in the initial phase, as it can be observed in Tables 3 and 6. This approach helps teams to deal with Web projects with changing requirements that sometimes are unknown at the beginning. This fact, together with the possibility of having an initial project plan, could allow organizations to make medium and long-term decisions in relation to their project portfolios. The process of continuous planning is shown in

Table 2 and the subsequent table, which illustrate the initial plan and how it evolves throughout the project.

In addition, project estimation and planning do not depend on late product size (such as lines of code or number of pages), which leads towards "requirements-freeze", but on relative estimations, including periodic milestones, which reflect and adapt estimations to the new circumstances, as Tables 2 and 3 show. For this purpose, value estimations and ROI calculations proved to be a very useful asset, as they headed the process of re-estimation and the decision of what must be developed and when.

Nevertheless, the "Size" attribute of user stories jointly evaluates risk, complexity and uncertainty elements of each attribute. This fact simplifies the estimation process as well as makes estimations oscillate during the project, as Table 3 represents. For instance, the estimation of the "Size" attribute of certain story usually decreases when risks are reduced or uncertainties are resolved.

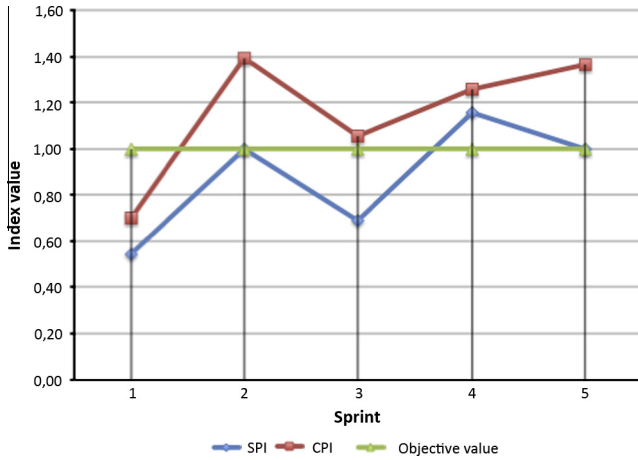


Fig. 19. Evolution of CPI and SPI per iteration.

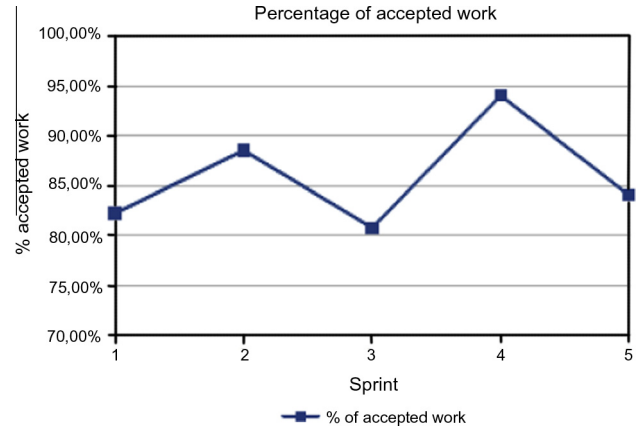


Fig. 21. Percentage of accepted work.

Table 7
Productivity metrics by Sprint.

Sprint	Real velocity	Average Hour/Story point	Team velocity (in hours)	Work capacity	Focus factor (%)	Accepted work	% Of accepted work	TVI+ (%)
1	6	19.5	117	117	100	96.25	82.26	100
2	14	9.79	137.03	78.75	174	69.75	88.57	140
3	2	12.86	25.73	87.25	29.49	70.5	80.80	27.27
4	15	10.83	162.47	117.75	137.98	110.75	94.06	162.16
5	19	9.96	189.15	156.75	120.67	131.75	84.05	169.64

Table 8
Productivity metrics for the project.

Number of Sprints	5
Team velocity (in story points)	11.2
Hours/story point	9.96
Team velocity (in hours)	126.27
Team capacity	111.5
Focus factor	111.25%
% Of accepted work	85.95%
TVI+	119.82%

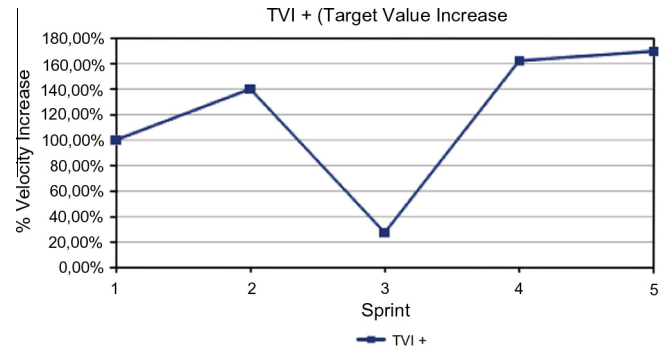


Fig. 22. Target value increase per Sprint.

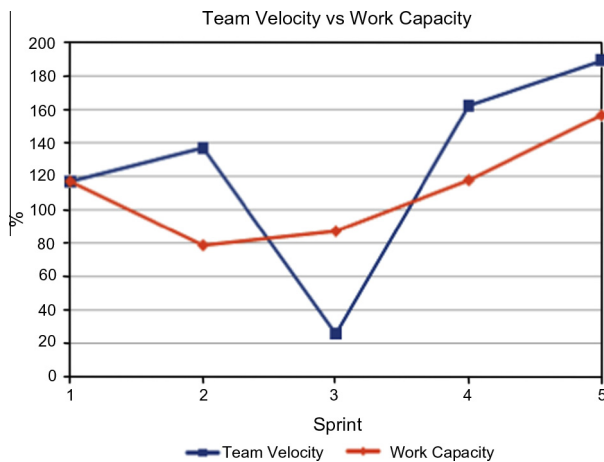


Fig. 20. Team velocity vs work capacity.

There is also an element to take into account: some technical non user-valuable stories need to be developed first, in order to provide technical architecture. The presented data confirm that this fact partially impeded the goal of delivering the higher value first, as Table 5 outlines, and should require some adaptations to the framework.

Tools such as charts representing the evolution of the delivered value like Fig. 16, based on the “Value” attribute of user stories given by customers, let the team know whether it is delivering most valuable stories first and then, if it focuses on costumers needs. Tracking this value through iterations enables quick adaptation and correction of any non-desired effects. However, as referred, the goal consisting in clearly delivering up-front value has not totally been met due to technical constraints and the initial fluctuation of velocity. A mechanism to solve this problem and spread the business value of the business stories to the technical ones could be put in place, and it could constitute the object of a future proposal.

5.5.2. Qualitative data analysis

This section will expose and discuss the different qualitative project observations. As an initial one, and regarding Plan and Estimate phases, we can point out that the framework involves all members of the team in planning the project. Everybody discusses and achieves a common estimation of the feature by means of Planning poker, which was particularly chosen for being one of

the estimation techniques that can better meet the characteristics of Web development, as explained in Section 4.2. It was proven to be an efficient and collaborative tool to estimate the work both at project and Sprint levels. Additionally, introducing value estimation in each user story makes users and customers be engaged with the estimation process and also enables identifying the real needs.

Besides, the proposed framework allows the team to learn and improve how to estimate during the project as well as visualize its own capacity of delivering finished features to the customer. Using productivity metrics and EVM calculations to guide Sprint retrospectives, as shown in the project data, helped to achieve this goal. As the management phase concerns, Agile teams can introduce aspects like provided value and quality, that are also crucial for the project to succeed thanks to the proposed change in the classical “Iron triangle”. The double check on quality (product and process) ensures that quality is built-in during the project. Sprint review meetings, where the team showed the developed features fulfilling the testing criteria, additionally became key tools.

Teams can control project constraints, like schedule or cost, through EVM calculations based on the Agile approach. These calculations, obtained at a Sprint level, can be appreciated as useful tools during iteration retrospectives, so as to obtain indications on potential problems in development processes. The “Status report” constitutes a useful tool to provide all project stakeholders with information such as changes on the Product Backlog, features and value delivered or constraints status, for example. This can enhance the feeling of openness and teams and stakeholders’ degree of commitment throughout the project. Using this tool was very important in a project like this, with a large number of different stakeholders, as it helped to handle the different expectations. In Measure and Adapt phases, EVM calculations and productivity metrics, together with innovation games, can offer a very practical tool to identify potential problems and obstacles and locate their roots causes. The correct use of this data can implement the capacity of the teams to lead projects to the correct direction.

It is important to mention that having a member of the team fully experienced in the use of Agile, acting as a coach of the rest of his/her colleagues, appeared to be very valuable, too. This colleague facilitated the transition from a classical to an Agile approach, even if elements like self-protection on time estimations did not completely disappear.

Finally, it has to be added that the size of the project (a small project of 4 persons during 6 months) can also influence the results. On the one hand, the reduced size favored the capability of quick adaptation as well as communication among the team. The short duration helped the team focus on the main goals (to deliver on time, which, as mentioned, was the main restriction of the project), but maybe it also constrained the learning process (as shown in the continuous overestimation tendency of the team). As a main conclusion, it can be stated that the framework proved to correctly behave in this type of projects.

6. Conclusions and future work

This section states the research conclusions, stressing the strengths and weaknesses of the proposed Agile framework, including relevant information resulting from applying it to a real project. These conclusions can provide valuable insights to guide the future work, although further research is needed.

As previously stated in Section 2, the general research question is structured into 3 research questions. Based on the results, some arguments can be provided to answer them, as shown below:

- *RQ1: What are the suitable existing techniques for estimating, planning and managing Agile Web projects?* To answer this question, it must be commented that we have assessed and a set of

suitable practices (as Planning Poker, Value points estimation, ROI calculation or Agile EVM, among others) suitable to estimate, plan and manage Agile Web projects in Sections 3 and 4.

- *RQ2: Can business value help estimation, plan and management of Agile Web projects?* With regard to this research question, Section 4 explains a way to identify business value linked to desired features, named ROI, and how to use it as guidance for re-estimation and adaptation in Agile Web projects.
- *RQ3: Can the identified techniques be integrated into an Agile common framework, appropriate for Web projects needs?* To answer this research question it must be clarified that some of these identified practices have been adapted and modified in order to integrate them into a coherent framework, presented in Section 4, with the goals of better suiting the characteristics of Web projects and keeping their agility.

As a conclusion, and with the aim of answering our main question about the feasibility of an Agile approach to plan, estimate and manage Web projects guided by value, we have identified and integrated an existing set of Agile practices in a framework and they have been tested in a real-world project. As analyzed, the proposed framework focuses on a continuous Plan and Estimate–Manage–Measure–Adapt cycle, using different techniques in each phase to help teams handle Web projects. It is worth pointing out that the proposed framework seems to fit well the special characteristics of Web projects described in Section 1.

The preceding sections have introduced an Agile approach to Web development project estimation, plan and management, as well as the result of an empirical example dealing with the practical application of the framework. The proposed framework tries to address the main characteristics of estimation and management in Web development projects, by offering a balance between agility and ability to change and medium and long-term capability to plan and control project constraints. The framework also includes some techniques and metrics that provide teams with objective data, for their processes of continuously improving their performance.

Although the proposed framework has been designed with the aim of covering all specific characteristics of Web projects, as mentioned before, some of them are shared with non-Web system. Even though the scope of our research is Web environments, the suitability of the framework to more general projects could be assessed as part of future research.

The results of the practical experience, based on the framework, are very encouraging and evidence that this kind of approach will be very appropriate to Web development projects. However, further research is needed to clarify how the business value is assigned to user stories and their interdependencies.

Extending the proposed framework, including other Agile practices and methods, by creating one that would be successfully assessed against a maturity model like CMMI would be very interesting. This extended framework will supply organizations with the possibility of combining quick responses, changing capacities and lightweight processes with continuous improvement, systematic use of data and advanced management practices, as well as validating these capabilities with a well-known and well-established model like CMMI.

Acknowledgements

This research has been supported by the MeGUS Project (TIN2013-46928-C3-3-R) of the Ministry of Economy and Competitiveness, Spain. We would like to thank the Regional Ministry of Culture and Sports of Junta de Andalucía for allowing us to issue this data

References

- [1] A.J. Albrecht, J.E. Gaffney, Software function, source lines of codes, and development effort prediction: a software science validation, *IEEE TSE SE-9* (1983) 639–648.
- [2] G. Alleman, et al., Making agile development work in a government contracting environment – using earned value to measure velocity, in: *AGILE 2003, Proceedings of Agile Conference*, 2003, Salt Lake City, Utah, 25–27 June 2003.
- [3] Agile COCOMO II. <<http://sunset.usc.edu/cse/pub/research/AgileCOCOMO/AgileCOCOMOII/Main.html>> (accessed 1.9.14).
- [4] S.W. Ambler, *Lessons in Agility from Internet-Based Development*, IEEE Software, 2002. pp. 66–73. (March–April).
- [5] S. Ambler, Generalizing Specialists: Improving Your IT Career Skills. <<http://www.agilemodeling.com/essays/generalizingSpecialists.htm>>, 2012 (accessed 1.9.14).
- [6] J. Appelo, *Management 3.0. Leading Agile Developers, Developing Agile Leaders*, Addison-Wesley, Boston, 2009.
- [7] J. Armario, J.J. Gutiérrez, M. Alba, J.A. García-García, J. Vitorio, M.J. Escalona, Project estimation with NDT, in: *ICSOFT 2012 Proceedings of the 7th International Conference on Software Paradigm Trends*, Rome, Italy, July 24–27 2012, pp. 120–126.
- [8] K. Beck, et al., Manifesto for Agile Software Development. <<http://www.agilemanifesto.org>>, 2001 (accessed 1.9.14).
- [9] K. Beck, *Test-Driven Development by Example*, Addison-Wesley, Boston, 2002.
- [10] K. Beck, C. Andres, *Extreme Programming Explained: Embrace Change*, second ed., Addison-Wesley, Boston, 2004.
- [11] A. Begel, N. Nagappan, Usage and perceptions of agile software development in an industrial context: an exploratory study, in: *ESEM '07, Proceedings of the 1st International Symposium on Empirical Software Engineering and Measurement*, Madrid, Spain, September 21–27 2007, IEEE.
- [12] B.W. Boehm, *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [13] B.W. Boehm, C. Abts, A. Winsor Brown, S. Chulani, B.K. Clark, E. Horowitz, R. Madachy, D.J. Reifer, B. Steece, *Software Cost Estimation with COCOMO II*, Prentice-Hall, Englewood Cliffs, NJ, 2000.
- [14] S. Ceri, P. Fraternali, A. Bongio, Web modelling language (WebML): a modelling language for designing web sites, *Comput. Networks* 33 (1–6) (2000) 137–157.
- [15] A. Cockburn, *Crystal Clear: A Human-Powered Methodology for Small Teams*, Addison-Wesley, Boston, 2004.
- [16] A. Cockburn, Balancing Lightness with Sufficiency. <<http://alastair.cockburn.us/Balancing+lightness+with+sufficiency>>, 2010 (accessed 1.9.14).
- [17] M. Cohn, *User Stories Applied: For Agile Software Development*, Addison-Wesley, Boston, 2004.
- [18] M. Cohn, *Agile Estimating and Planning*, Addison-Wesley, NJ, 2005.
- [19] M. Cohn, *Succeeding with Agile Using Scrum*, Addison-Wesley, Boston, 2009.
- [20] R. Corral, Gestión ágil de proyectos: experiencias prácticas y resultados obtenidos. <http://www.ermua.es/pags/noticias/archivos/2011/20110331/presentacion_rodrigo_corral.pdf> (accessed 1.9.14).
- [21] J.W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Method Approaches*, second ed., SAGE, 2003.
- [22] Decreto 1/1979, de 30 de julio, sobre creación y publicación del Boletín Oficial de la Junta de Andalucía. <<http://www.juntadeandalucia.es/boja/1979/1/3>>, 1979 (accessed 1.9.14).
- [23] Decreto 205/1983, de 5 de octubre, por el que se aprueba el Reglamento del Boletín Oficial de la Junta de Andalucía. <<http://www.juntadeandalucia.es/boja/1983/82/1>>, 1983 (accessed 1.9.14).
- [24] Y. Deshpande, S. Marugesan, A. Ginige, S. Hanse, D. Schawabe, M. Gaedke, B. White, *Web engineering*, *J. Web Eng.* 1 (1) (2002) 3–17.
- [25] E. Derby, D. Larsen, *Agile Retrospectives. Making Good Teams Great*, The Pragmatic Bookshelf, Dallas, 2006.
- [26] S. Downey, J. Sutherland, Scrum metrics for hyperproductive teams: how they fly like fighter aircraft, in: *HICSS 2012, Proceedings of the 45th Hawaii International Conference on System Science*, 2012, Maui, Hawaii, USA, January 4–7 2012.
- [27] T. Dybå, T. Dingsoyr, Empirical studies of agile software development: a systematic review, *InfSoftwTechnol* 50 (9–10) (2008) 833–859.
- [28] M.J. Escalona, J. Torres, M. Mejías, J.J. Gutiérrez, D. Villadiego, The treatment of navigation in web engineering, *Adv. Eng. Softw.* 38 (4) (2007) 267–282.
- [29] M.J. Escalona, G. Aragón, NDT: a model-driven approach for web requirements, *IEEE Trans. Softw. Eng.* 34 (3) (2008) 370–390.
- [30] R. Fewster, E. Mendes, Measurement, Prediction and Risk Analysis for Web Applications, in: *Proceedings of Metrics '01*, London, 2001.
- [31] H. Glazer, J. Dalton, D. Anderson, M. Konrad, S. Dhrum, CMMI or agile: why not embrace both!, in: *CMU/SEI-2008-TN-003*, Pittsburgh, 2008.
- [32] D. Gray et al., *Gamestorming. A Playbook for Innovators, Rulebreakers and Changemakers*, O'Reilly, CA, 2010.
- [33] J. Highsmith, *Agile Project Management: Creating Innovative Products*, second ed., Addison-Wesley, NJ, 2009.
- [34] JIRA Website. <<https://www.atlassian.com/es/software/jira>> (accessed 1.9.14).
- [35] J. Johnson (Ed.), XP2002, Report to the Third International Conference on Extreme Programming and Agile Processes in Software Engineering, Alghero, Italy May 26–29, 2002.
- [36] M. Jorgensen, *Practical Guidelines for Expert Judgment-Based Software Effort Estimation*, OsloUniv., Norway, 2005 (Software).
- [37] Gustav Karner, Resource Estimation for Objectory Projects, *Objective Systems SF AB*, 1993.
- [38] N. Koch, A. Knapp, G. Zhang, H. Baumeister, *UML-Based Web Engineering: An Approach Based on Standards*, Web Engineering: Modelling and Implementing Web Applications, Springer, 2008.
- [39] N. Koch, A. Knapp, S. Kozuruba, Assessment of effort reduction due to model-to-model transformations in the web domain, in: *ICWE 2012, Proceedings of the 12th International Conference on Web Engineering*, Berlin, Germany 23–27 July 2012.
- [40] C. Larman, V.R. Basili, *Iterative and Incremental Development: A Brief History*, *IEEE Comput.* 36 (6) (June 2003) 47–56.
- [41] Law 11/2007, 22th June, on Citizen's Electronic Access to Public Services. <<http://www.boe.es/boe/dias/2007/06/23/pdfs/A27150-27166.pdf>>, 2007 (accessed 1.9.14).
- [42] K.K. Lilja, K. Laakso, J. Palomki, Using the delphi method, in: *PICMET11, Proceedings of the 2011 Portland International Center for Management of Engineering and Technology Conference: Technology Management in the Energy Smart World*, Portland, Oregon, USA, July 31–August 4, 2011, IEEE.
- [43] K. Logue, K. McDaid, Agile release planning: dealing with uncertainty in development time and business value, in: *ECBS 2008, Proceedings of the 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems*, 2008, Belfast, Northern Ireland, March 31 2008–April 4 2008.
- [44] E. Mendes, S. Counsell, N. Mosley, Measurement and effort prediction of web applications, in: *Proceedings of the 2nd ICSE Workshop on Web Engineering*, Limerick, Ireland, June 2000.
- [45] E. Mendes, N. Mosley, *Web Cost Estimation: An Introduction*, Web Engineering: Principles and Techniques, IGI Global, 2005.
- [46] L. Olsina, Building a Web-based information system applying the hypermedia flexible process modelling strategy, In: *Workshop on Hypermedia Development Processes, Methods and Models (Hypertext 98)*, Pittsburgh, USA, 1998.
- [47] OMG – Object Management Group, IFML: The Interaction Flow Modeling Language. <http://www.ifml.org/?page_id=99> (accessed 1.9.14).
- [48] Orden de 23 de abril de 2012, por la que se regula la inserción de documentos en el Boletín Oficial de la Junta de Andalucía. <<http://www.juntadeandalucia.es/boja/2012/86/1.html>>, 2012 (accessed 1.9.14).
- [49] S.R. Palmer, *A Practical Guide to Feature-Driven Development*, Prentice-Hall, NJ, 2002.
- [50] M. Pikkarainen et al., *The Impact of Agile Practices on Communication in Software Development*, *Empirical Software Engineering*, Springer, 2008 (May 2008).
- [51] M. Poppendieck, T. Poppendieck, *Lean Software Development. An Agile Toolkit*, Addison-Wesley, Boston, 2003.
- [52] R.S. Pressman, *What a Tangled Web We Weave*, IEEE Software, 2000 (January–February 2000).
- [53] Project Management Institute, *Project Management Book Of Knowledge (PMBOK) Guide*, fourth ed., Project Management Institute, 2008.
- [54] Z. Racheva, M. Daneva, K. Sikkel, A. Herrmann, R. Wieringa, Do we know enough about requirements prioritization in agile projects: insights from a case study, in: *Proceedings of the 18th IEEE International Requirements Engineering Conference*, Sydney, Australia, September 27 2010–October 1 2010), RE 2010.
- [55] H. Ran, et al., Agile web development with web framework, in: *WiCOM '08, Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing*, Dalian, China, October 12–17 2008, IEEE, 2008.
- [56] Redmine Website. <<http://www.redmine.org/>> (accessed 1.9.14).
- [57] D.J. Reifer, *Web Development: Estimating Quick-to-Market Software*, IEEE Software, 2000 (November–December 2000).
- [58] M. Register, T. Golding, Using agile for buy vs build decisions, in: *AGILE 2008, Proceedings of the Agile Conference*, 2008, Toronto, Canada, 4–8 August 2008.
- [59] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, *Empirical Softw. Eng.* 14 (2009) 131–164.
- [60] K. Schwaber, Scrum development process, in: *OOPSLA '95, Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages and Applications*, Austin, TX, USA 15–19 October 1995, ACM.
- [61] K. Schwaber, *Controlled Chaos: Living on the Edge*, American Programmer, 1996. April.
- [62] K. Schwaber, *Agile Project Management with Scrum*, Microsoft Press, Redmond, 2004.
- [63] M. Shepperd, C. Schofield, B. Kitchenham, *Effort Estimation Using Analogy*, Dept. of Comput., Bournemouth Univ. Software Engineering, 1996.
- [64] J. Shore, S. Waden, *The Art of Agile Development*, O'Really Media, CA, 2008.
- [65] P. Solomon, R. Young, *Performance-Based Earned Value*, Wiley-IEEE Computer Society, 2006.
- [66] G. Smith, A. Sidky, *Becoming Agile in an Imperfect World*, Manning Publications, CT, 2009.
- [67] T. Sulaiman, B. Barton, T. Blackburn, AgileEVM – Earned Value Management in Scrum Projects, in: *AGILE 2006, Proceedings of the Agile Conference*, 2006, Minneapolis, Minnesota, 23–28 July 2006.
- [68] K. Sungjoo, C. Okjoo, B. Jongmoon, Model-based dynamic cost estimation and tracking method for agile software development, in: *IEEE/ACIS ICIS 2010, Proceedings of the 9th Computer and Information Science Conference*, Kaminoyama, Japan 18–20 August 2010.
- [69] J. Sutherland, A. Viktorov, J. Blount, N. Puntikov, Distributed scrum: agile project management with outsourced development teams, in: *HICSS 2007*,

- Proceedings of the 40th Annual Hawaii International Conference on System Sciences, Waikoloa, Big Island, Hawaii. 3–6 January 2007.
- [70] J. Sutherland, K. Schwaber, The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. <<http://www.scrum.org/Scrum-Guides>>, 2011 (accessed 1.4.14).
- [71] H. Takeuchi, I. Nonaka, The new product development game, *Harvard Bus. Rev.* 64 (1) (1986) 137–146 (January–February 1986).
- [72] The Standish Group International, Inc., Extreme Chaos, The Standish Group International Inc. <<http://www.standishgroup.com>>, 2001 (accessed 1.9.14).
- [73] The Standish Group International Inc, The CHAOS Manifesto 2012. <<http://www.versionone.com/assets/img/files/CHAOSManifesto2012.pdf>>, 2012 (accessed 1.9.14).
- [74] C.J. Torrecilla-Salinas, et al., A scrum-based approach to CMMI maturity level 2 in Web Development environments, in: iiWAS '12, Proceedings of the International Conference on Information Integration and Web-based Applications & Services 2012, Bali, Indonesia December 3–5 2012, ACM.
- [75] C.J. Torrecilla-Salinas, et al., Agile in public administration: oxymoron or reality? An experience report, in: CAiSE '13, Proceedings of the Industrial Track of the Conference on Advanced Information Systems Engineering 2013, Valencia, Spain, June 21, 2013.
- [76] VersionOne Website. <www.versionone.com>. (accessed 1.9.14).
- [77] W@NDA Project, Workflow in Andalusia Public Administration. <<https://ws024.juntadeandalucia.es/ae/descargar/2793>>, 2012 (accessed 1.9.14).
- [78] M. Yap, Value based extreme programming, in: AGILE 2006, Proceedings of the Agile Conference, 2006, Minneapolis, Minnesota, 23–28 July 2006.