

**Research Report**

**Methodologies for web application projects**

Author: Bertocco Thomas-Killian - #13758322

Lecturer: Dr. Insu Song

CP5046 ICT Project 1: Analysis and Design | SP53 | 2019

24 January 2020

# Abstract

Nowadays, Web applications are monopolizing the market of software products because of the many benefits they offer and the dependence of the Internet for the elaboration of all types of operations, leaving traditional desktop applications behind. This is due to their limited structure and the lack of flexibility in the face of the constant changes in technology. However, with the expansion of this market, issues linked to the development of these applications are becoming more and more common, which is why many methodologies have appeared to guide the software lifecycle and also to help mitigate and resolve existing errors at each stage of development. For this reason, a research was performed to obtain information on the methodologies and on those which, according to various authors, have essential properties in the development of the Web. The goal is therefore to compare web software development methodologies through the collection of bibliographic and analytical information.

Table of Contents

[Abstract 2](#_Toc30793031)

[Introduction 4](#_Toc30793032)

[Fundamental knowledge 5](#_Toc30793033)

[Software development 5](#_Toc30793034)

[Web applications 5](#_Toc30793035)

[Methodology for software development 5](#_Toc30793036)

[Traditional methodologies 5](#_Toc30793037)

[Agile methodologies 6](#_Toc30793038)

[Hybrid methodologies 6](#_Toc30793039)

[Methodologies for web application development 6](#_Toc30793040)

[WSDM (WEB Semantic design method) 7](#_Toc30793041)

[Phases of the WSDM methodology 7](#_Toc30793042)

[SOHDM (Scenario-based object-oriented hypermedia design methodology) 8](#_Toc30793043)

[Phases of the SOHDM methodology 9](#_Toc30793044)

[OOHDM (Object-oriented hypermedia design methodology) 10](#_Toc30793045)

[Phases of the OOHDM methodology 10](#_Toc30793046)

[WAE (WEB application extension) 11](#_Toc30793047)

[Phases of the WAE methodology 11](#_Toc30793048)

[IWEB (WEB engineering) 12](#_Toc30793049)

[Phases of the IWEB methodology 12](#_Toc30793050)

[Research methodology 13](#_Toc30793051)

[Results 14](#_Toc30793052)

[Discussion 16](#_Toc30793053)

[Conclusion 17](#_Toc30793054)

[References 18](#_Toc30793055)

# Introduction

Software development methodologies arise as an alternative and framework from the complexity involved in making software and as a response to the problems that occurred at each stage of development because it does not have adequate control which generated problems and poor software development that did not meet customer requirements.

Currently, the development of Web applications has emerged exponentially due to the impact of the Internet in the world as a means of disseminating information and other services. The complexity of the development of Web applications has increased with the technological advances in the field of programming, and various problems have also arisen as the processes of handling large volumes of information, changes in software specifications, lack of communication, and deficiency in security, so it is essential that each company working in the field of development work a methodology.

With the implementation of the methodologies within the life cycle of Web applications, not only the process of creation and development of software has been improved, but also the level of risk has been diminished, since each methodology offers a guide composed by stages and effective processes that allow obtaining quality results.

Considering the above, this bibliographic, analytical and field research presents a comparison of Web methodologies based on quality criteria and specific criteria that cover aspects such as graphic representation, design levels, life cycle, types of notation, standards, type of modeling and support tools. The methodologies that were used for the comparison are SOHDM, OOHDM, WSDM, WAE, UWE and IWEB.

Based on this comparison it can be verified that the most used methodologies cover a more significant amount of criteria and elements of almost all methodologies. These allow concentrating their efforts in Web aspects unlike the others which are focused on providing solutions to specific problems.

From the analysis of the information and comparison in the development framework of the article it was obtained as results that the OOHDM methodology is the one that fulfills almost all the characteristics for the process of development of Web applications. Besides, it was also considered by programmers and developers as the most optimal to guide the whole process.

# Fundamental knowledge

## Software development

Software is a "set of elements in dynamic interaction organized to achieve an objective" (C. d. P. Hereder, 2004) and in addition to the hardware they allow the realization of diverse tasks in an agile way. For this reason, software development has increased notably in different fields of application, which is why before creating a new application, it is necessary to take into account several aspects, characteristics and functions that the new product will contain (R. Nori Martin, 2017) (D. Rami Cardoss, 2016). According to Pressman "it is concluded that a concentrated effort must be made to understand the problem before developing a software application" (R. S. Pressman, 2010).

## Web applications

"A Web-based application is a client/server application, where both the client (the browser, explorer or viewer) and the server (the Web server) and the protocol by which they communicate (HTTP) are standardized and do not have to be created by the application programmer" (L. More, 2002).

Nowadays, Web applications are increasingly popular and their use has monopolized the scientific, cultural, academic and business fields, among others, and this is due to the multiple advantages that the user has over desktop programs. Among others, the advantages we can mention are: multiplatform operating system, executed by any computer device that has an Internet connection, does not require the installation of programs only a browser, backups are stored in the servers, the information that is generated can be shared simultaneously by several people, the space occupied by the data is in charge of the server and is easy to use (A. Rami Martin, 2014).

## Methodology for software development

The construction of software is an activity that must be planned and systematized through strategies that guarantee its success. Any inconvenience will likely arise in the course or at the end of the development such as: changing requirements, planning or budgets that are not realistic, lack of personnel and dissatisfied customers that lead to failure. (P. Bernad, 2014) "Software development is not an easy task. As a result of this problem, an alternative has long since emerged: the Methodology" (E. Delgade, 2008).

A software development methodology is a process or set of procedures, techniques and documentation that allows developers to guide and execute the project to create new quality applications that meet customer expectations. According to Pressman, the methodology is defined as "a framework used to structure, plan and control the process of developing computer systems" (I. Sommerville, 2005).

## Traditional methodologies

The traditional methodologies appeared in the 1960s, due to a manual software development with the need to optimize the processes and objectives proposed in the development projects, "they are especially focused on the control of the process, establishing rigorously the activities involved, the artifacts that must be produced, and the tools and notations that will be used" (I. Leiva and M. Villalobos, 2015).

These methodologies divide the process or development of the system into different stages, which allows the remuneration of functions to the members of the development team and, in this way, establishes the reuse of components. They also encourage the documentation of the developed applications, thus ensuring easy understanding and maintenance.

## Agile methodologies

According to A. Navar (A. Navar, 2013), agile methodologies are flexible, that is, they can be easily modified in case the development team or the project requires it. These methodologies allow the project to be subdivided into small fractions and thus be developed autonomously in a short period estimated between two to six weeks. They are adaptable to changing requirements by the customer, delivering prototypes continually so that a better product is guaranteed. It encourages teamwork by considering the customer as part of it.

## Hybrid methodologies

From the many methodologies that exist, whether they are agile or traditional, hybrid methodologies arise, as a combination of the two previous ones, but in this case rescuing the priorities that the methodologies mentioned stand out to create a secure and flexible method that adapts to all types of projects for the development of software.

The hybrid methodologies "aim to take advantage of the existing methodologies, so that they are a combination of the best practices described in each of them". (P. Letelier, 2006). This type of methodology can also be developed by bringing together each of the best characteristics that exist within the same type of methodology.

## Methodologies for web application development

Methodologies are composed of several phases that change depending on the methodologies used. If the method has more complex steps, you will mostly have a lower number of errors concerning data quality and consistency.

The methodologies for web applications contain phases for software development that can increase or decrease depending on the method used, according to Nieves del Valle.

Most methods coincide in the following stages (S. D. Oran Jimen, 2017):

* Conceptual Design: this section covers issues related to the specification of the problem domain, through its definition and the relationships it contracts.
* Navigational Design: it is focused on access and how the data are visible.
* Design of the presentation or interface design: it focuses on the way the information is going to be shown to the users, it is worth mentioning that in this section the client is mostly involved in defining the requirements and the users defend how they want to interact with the system.
* Implementation: the construction of the software from the parts generated in the previous steps.

Next, several Web development methodologies will be analyzed together with the stages of each one of them.

## WSDM (WEB Semantic design method)

It is a methodology purely for Web applications, today applications must be developed in a short period following its semantic structure of content and functionality. This is why it is considered appropriate for Web applications. However, it is not recommended for project management, for which a new methodology must be used to facilitate the software life cycle (A. Mishra, 2014).

### Phases of the WSDM methodology

The authors T. Appelmans, Y. J. Marca Aular, D. F. Flores Chicaiza agree that the WSDM methodology is a proposal for the development of web applications that focuses on the user, and considers that the user is the main actor in the system, is responsible for defining the requirements, and the users involved in the application.

In WSDM the users are classified, and based on this, design parameters are defined. This methodology focuses more on static Web sites that provide information than dynamic sites or applications. Also, it has not been so well received, because as explained above it only focuses on users and not so much on data, another important aspect in the development of Web applications.

WSD consists of three layers: context (establishment of requirements), navigation (system-user interaction), and information (relevant user data). Regarding the modeling phases for the implementation of an application, the WSDM establishes 4 phases or activities that are specified in Figure 1.

Conceptual design

Implementation design

User modeling

Implementation

**Illustration 1**: Phases of the WSDM methodology

A brief description of the phases of this methodology will follow.

* User Modeling: serves to identify potential users of the application and the information they would require from this site.
* Conceptual Design: the conceptual modeling is developed, the information is organized, the users are classified, the objects are modeled, entity-relationship diagrams are created and the navigation design is created. Each navigation design in the Web site will be different for each user profile and therefore will have its perspective. The deliverables of this phase are the conceptual model, and navigational design.
* Implementation design: Design is created based on the user's requirements, this prototype interface of the Web site should have a pleasant appearance, be efficient and safe, also here are specified the design restrictions, according to what was established in the conceptual design.
* Implementation: the selection of the development environment, construction of the architecture, codification and verification of the complete functionality of the Web application is carried out.

## SOHDM (Scenario-based object-oriented hypermedia design methodology)

It is an object-oriented methodology in hypermedia that develops designs in scenarios or panoramas. Besides, it allows for capturing the needs of the system by proposing the use of scenery. SOHDM starts from a diagram where the external entities capable of communicating with the system are identified. It is a methodology very similar to the OOHDM methodology differentiated by the use of scenarios (UNEMI, 2017).

SOHDM proposes the use of scenarios for each different event, in order to know what the needs of the system are (M. J. Escalona, 2002). Each scenario symbolizes the process of interaction that exists between the user and the system, in this process the objects involved, the flow of activities, and the operations performed are detailed. From each scenario the conceptual model can be obtained, which is reflected in a class diagram.

### Phases of the SOHDM methodology

Regarding software development management processes or life cycle, A. Nieves del Valle, A. Mishra describe 6 phases which can be seen in illustration 2.

Object-Oriented Modeling

View design

Domain analysis

Implementation

Interface design

Navigational design

**Illustration 2**: Phases of the SOHDM methodology

* Domain analysis: establishes the limits of the application to be developed, and represents them through a flow chart. Also, the use of SACs (Scenario activity charts) is made, which are nothing more than scenarios where the requirements of the application are determined.
* Object model: in this stage the SACs are used to perform the object modeling. The users are the main objects of the system, each user is described in the application development document, which includes attributes, associations and cardinality.
* Design of the views: the views are represented using navigation units, each view groups information from the application classes.
* Navigation design: It identifies the navigation of the objects within the Web application and the way they interact through links, menus, nodes, queries, among others.
* Implementation: The application interface, business logic, and database schema are generated.
* Construction: development of the final application, which meets all the needs and requirements that were initially established by users.

## OOHDM (Object-oriented hypermedia design methodology)

According to R. Soto Giorg (R. Soto Giorg, 2011). OOHDM is an object-oriented methodology that proposes a five-phase development process where UML graphic notations are combined with others typical of the methodology.

When the Internet was not accessible to all people, OOHDM was used for hypermedia applications, but thanks to the rise of the Internet, this methodology has been adapted to the development of hypermedia applications oriented to the Web, such as virtual libraries, educational sites, search engines, entertainment, etc.

This methodology allows the development of Web applications from the use of specialized models such as conceptual, navigation and user interface to simplify and make more efficient the design of applications.

### Phases of the OOHDM methodology

The five stages of the OOHDM methodology will be described below.

Navigational design

Conceptual design

Obtention of requirements

Abstract interface design

Implementation

* Obtaining requirements: the obtention of requirements is carefully considered, so it is crucial to know the actors and tasks that must be modeled in the use cases.
* Conceptual design: the conceptual model is represented through the modeling of class diagrams based on classes, relationships, and subsystems, focusing on the semantic domain leaving aside the actors and tasks.
* Navigational design: represents the different paths that the application can execute depending on the type of user. That is, it provides a navigational context capable of performing actions through links, links or indexes that are related within the web application depending on the user profile to show their corresponding views.
* Abstract interface design: it is executed after the navigational design, where it is necessary to specify the user interfaces that will be displayed in the web application. Within this model, two sub-tasks can be identified such as structural design and behavioral design.
* Implementation: implement the Web application independently of the platform that will be used. This phase is also known as implementation since it is from here that users begin to use and take advantage of the elaborated system, through a web browser and internet connection.

It is essential to mention that the developer and designer are in charge of the technical side of the system and its final appearance, while the client verifies that it works correctly as requested in the first phase or stage.

|  |  |  |
| --- | --- | --- |
| Steps | Products | Formalities |
| Obtention of requirements | Use cases | Document Format Templates, User Interaction Diagrams (UIDs) |
| Conceptual design | Classes, subsystems, relationships, attributes | Object-oriented models |
| Navigational design | Nodes, links, access structures, contexts, navigation, navigation transformations | Abstract data views, configuration diagrams, navigation charts for ADVs |
| Abstract interface design | Abstract interface objects responses to external events, interface transformations | Abstract data views, configuration diagrams, navigation charts for ADVs |
| Implementation | Application in operation | Those supported by the environment |

**Table 1**: Products and formalities of the OOHDM methodology

## WAE (WEB application extension)

The WAE is an extension of UML, which does not focus on the object-oriented paradigm but Web elements. WAE incorporates some concepts such as JavaScript and Form. In this methodology, it covers both the server-side (server pages) and the client-side (Active X, Java applet, etc). However, object-oriented concepts (e.g. inheritance) are not sufficiently concerned with the extension. A class notation is used in the class diagram to represent an HTML page. WAE focuses primarily on the technology of the script page, such as ASP and JSP.

While Osoro (M. J. Osoro, 2009) and Mondragon (A. Mondragon, 2014) refer that WAE presents a series of stereotypes that constitute WEB elements, which can be forms, links, web pages, among others. It should be noted that despite the WAE contributed to the modeling of traditional Web applications, it still requires stereotypes and relationships that reflect interactivity, cookies, mobile communities, social networks and other notations that are applied today for Web applications.

### Phases of the WAE methodology

According to Osoro's analysis (M. J. Osoro, 2009), the WAE methodology counts are the following phases as indicated in the illustration.

Analysis and design

Obtention of requirements

Implementation

Business modelling

**Illustration 4**: Phases of the WAE methodology

* Business modeling: it comprises the flow of activities that take place within the organization, in other words, it describes which are the departments, employees and the interaction between them.
* Requirements capture: search for necessary requirements for the development of the application while appropriately symbolizing them for developers, users and customers.
* Analysis and design: analysis of all requirements that were obtained in the previous phase with a much clearer understanding of what is intended with the system. As products of this phase, sequence, component and class diagrams are created.
* Implementation: a final phase that covers the physical architecture on which the application and its environment will be implemented.

## IWEB (WEB engineering)

IWeb demands an incremental and evolutionary software process (J. D. Texier, 2008). Pressman also points out that the model in the first versions can be a paper model or a prototype, and during the last iterations more and more complete versions of the designed system are produced.

"The IWeb is divided into many structural activities, also called task regions. Generally, there are between three and six task regions, not necessarily all of which need to be applied for each iteration". (J. D. Texier, 2008). IWeb is a methodology that focuses on the creation of high-quality web applications and systems based on scientific engineering principles. These applications make access possible from remote computers.

### Phases of the IWEB methodology

The following are the seven phases of the IWeb methodology.

Navigation design

Architectural design

Production

Content design

Engineering

Generation of pages

Analysis

Planification

Formulation

Testing and customer evaluation

Interface design

**Illustration 5**: Phases of the IWEB methodology

* Formulation: In this first phase, the objectives, goals, the scope of the application and its first delivery are identified. Fundamental aspects to consider in this phase are to know the motivation for the development of the application, to verify if it is necessary or not, and to identify who is going to use it.
* Planning: estimate the general cost of the project to be carried out, as well as contingency plans due to possible risks, the scope and describe the quality and management of the application in terms of changes.
* Analysis: establish the design and technical requirements, also analyzing the content of the same, its iteration, functionality, and configuration.
* Engineering: achieve the integration of architectural, navigation and interface design. This is why it is divided into content design, production, architectural design, navigation design, and interface design
* Page generation: the designs of the previous stage are integrated through tools such as programming and labeling languages that serve as a basis for the construction of the web application.
* Testing: the business logic applied in the system is checked, and data inputs and outputs are verified in order to discover functionality, behavior or performance errors.
* Customer evaluation: it allows correcting errors thanks to the iterations carried out in order to polish the application in comparison to previous iterations.

# Research methodology

The research to be carried out is bibliographical and analytical. It will provide relevant information on web application development methodologies. In order to carry out the following work, different stages have been named, which will be focused on the comparison of methodologies in Web applications, which are:

1. Bibliographic research of works that correspond with similar studies developed in the same field.
2. Determination and processing of collected data.
3. Establishment of a comparative analysis of the different methodologies in the development of Web applications.
4. Analysis of the data obtained and verification of the same.
5. Elaboration of the results and the conclusion on methodologies in Web applications, selection of the best alternative.

# Results

Below is a series of comparisons of the methodologies each with different criteria covering aspects of requirements, development, design, and quality. Table 2 shows a comparison of the requirements of each methodology studied.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirements | Methodologies | | | | | |
| WSDM | SOHDM | OOHDM | UWE | WAE | IWEB |
| Data | X | x | x | x | x | x |
| User’s interface |  | X | x | x | x | x |
| Navigation |  |  | x | x | x | X |
| Personalization | X |  | X |  |  |  |
| Transactional |  | X |  | x |  |  |
| Non-functional | x | x | x | x | x | X |

**Table 2**: Comparison of requirements in the web environment covered by the methodologies

Analyzing the results and taking into account that the methodologies are ordered in chronological order. It can be seen that in the beginning the methodologies were only focused on the data and the interface provided to the user, while in the most current ones the importance of dealing with customization, navigation, transactional, and non-functional requirements is highlighted. It is also important to mention that the OOHDM and UWE methodologies are the ones that contemplate most of the requirements in the Web environment.

Table 4 shows a comparative analysis between the different methodologies, their techniques, notation, and their support tools. Abbreviations were established in table 3 in order to make the names of the methodologies more understandable (B. Mercerat, 2001).

|  |
| --- |
| abbreviations or acronyms |
| Entity – Relationship (E-R) |
| Object Oriented (OO) |
| Relationship Management Data Model (RM DM) |
| Graphical User Interface (GUI) |
| Abstract Data View (ADV) |
| Object Modeling Technique (OMT) |
| Unified Modeling Language (UML) |

**Table 3**: Table 4’s abbreviations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methodology | Modeling technique | Graphical representation | Notation | Support tool |
| OOHDM | OO | Class diagram  Navigation diagram, class + context  ADV configuration diagram and ADV diagram | OMT/UML  ADV | OOHDM-Web |
| SOHDM | OO | Activity scenarios diagram  Class structure diagram  OO views  Navigational connection diagram  Pages diagram | Proprio |  |
| WSDM | E-R / OO | E-R class diagram  Navigation layers | E-R / OMT  Proprio |  |
| WAE | OO | UML diagram | UML | Rational Rose |
| UWE | OO | UML diagram | UML | Magic Draw |
| IWEB | E-R / OO | Navigational diagram  Architecture diagram | UML | XML, HTML |

**Table 4**: Comparison of methodologies in web application development

Table 5 shows a design comparison based on the three typical levels of web development: conceptual, structural and visible.

|  |  |  |  |
| --- | --- | --- | --- |
| Methodology | Concept level | Structure level | Visible level |
| OOHDM | Classes  Perspective  Relation - OO | Link | ADV  In context |
| SOHDM | Scenarios:  -Event  -Activity  Activity flow | Navigational link  Views – OO:  -Base  -Association | UI component:  -Choose  -Search input text  -Button |
| WSDM | Object  Perspective  Relation | Link  Component  -Navigation  -Information  -External  Navigation path | -None |
| WAE | Relation-OO | Link  Targeted link  Redirect  Build  Send | Frame set  Form |
| UWE | Meta-model  OO  Web scenario | Logical and navigational model  -Association  -Temporal interaction | Views represented by UML diagrams |
| IWEB | Agile process | -Navigation | Web Apps |

**Table 5**: Comparison of design concepts of web development methodologies

Based on the comparison previously made and based on the research carried out by Silva (B. Mercerat, 2001), it can be said that the most widely used methodology in web application development is SOHDM, taking into account that this method offers more secure processes focused on quality metrics aspects that verify that applications have higher reliability, consistency, and security.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Criterias | Methodologies | | | | | |
| OOHDM | SOHDM | WSDM | WAE | UWE | IWEB |
| Functionality | Adequacy  Interoperability | Adequacy  Interoperability | Adequacy  Interoperability | Adequacy  Interoperability  Security | Adequacy  Interoperability  Security | Adequacy  Interoperability  Security |
| Fiability | Recoverability | Recoverability | Low | Low | Recoverability | Low |
| Usability | Learning  Understanding  Processing | Processing | Processing | Low | Learning  Understanding  Processing | Low |
| Efficiency | Medium | Medium | Medium | Low | Low | Low |
| Portability | adaptability | adaptability | adaptability | Adaptability  Coexistence | adaptability |  |
| Maintainability | Ease of analysis, change and testing | Medium | Medium | Low | ease of analysis and change | Medium |

**Table 6**: Comparison of quality criteria for web development methodologies

Table 6 shows a comparison of the quality criterion based on ISO/IEC 9126 with the different methodologies it has studied.

As it can be seen in the table, most of the characteristics according to the quality criteria established by ISO/IEC 9126 (R. L. Gran, 2015) are covered by OOHDM methodology, since it presents a better adaptability to changes, it is easy to learn, and it allows the quick correction of errors and problems presented throughout the life cycle.

# Discussion

Web software development methodologies play an essential role in the development and implementation of this type of application. Besides, there are different Web software development methodologies that can be applied to the construction of software. There is not a better methodology than another one, but each one is adapted to some particular characteristics of the product to be obtained, the best-known ones according to the research carried out are SOHDM, OOHDM, WSDM, WAE, UWE and IWEB (J. P. Ucan-Pech, 2014) (P. Baldeon and C. Hinojos, 2012).

The OOHDM methodology is the most widely used for the process of developing Web applications due to its easy adaptability to any project. It is also important to mention that there is a tiny margin of difference between companies and developers that apply methodology as those who do not apply. We could emphasize the use of methodologies as an indispensable framework for the creation of software products in subsequent studies.

# Conclusion

The developers must develop software using methodologies that allow them to guide their work and to adapt the final product to aspects such as functionality, security, consistency, and reliability, which allow the system to follow quality standards and to comply with the objective for which it was created.

Regardless of the methodology that the professional chooses, the various tasks set out in the previous sections must be carried out to ensure an efficient product whose functionalities are following what was previously established by the client.

The comparison carried out allowed us to determine that the OOHDM methodology offers an optimal framework for the development of Web applications since it facilitates the work within the development team and speeds up the processes by optimizing their stages, in addition to contemplating more stages in the development life cycle and precise object modeling.

# References

|  |
| --- |
| C. d. P. Hereder, IT and communications in the company, illustrated, Ed., Belgrade:  ESIC, 2004. |
| R. Nori Martin, The Software Development Process, Second ed., Smashwords Edition, 2017. |
| D. Rami Cardoss, Software Development: Requirements, Estimates, and Analysis, Second ed., Createspace Independent Pub, 2016. |
| R. S. Pressman, Software Engineering. A practical approach, Seventh ed., London: Mc Graw-Hill, 2010. |
| L. More, Web application programming: history, basics, and web clients, First ed., San Francisco: University Club, 2002. |
| A. Rami Martin, Web applications, Second ed., Zagreb: Paraninfo Editions, S.A., 2014. |
| P. Bernad, «Risk management in software development projects», *Faculty of Engineering University of Antioquia,* vol. I, nº 70, pp. 223-243, 2014. |
| E. Delgade, «Software development methodologies. What is the path?», *Architecture and Engineering Magazine,* vol. II, nº 3, pp. 1-7, 2008. |
| I. Sommerville, Software Engineering, Seventh ed., Toronto: Pearson Education, 2005. |
| I. Leiva and M. Villalobos, «Hybrid agile method for developing software on mobile devices», Chilean *engineering,* vol. XIII, nº 3, pp. 473-488, 2015. |
| A. Navar, «Review of agile methodologies for software development», *Prospective,* vol. XI, nº 2, pp. 30-39, 2013. |
| P. Letelier, «Agile methods for software development: eXtreme Programming (XP)», *Administrative Science and Technique,* vol. VI, nº 26, pp. 1-6, 2006. |
| S. D. Oran Jimen, «Hybrid methodologies for software development» *Digital University Review,* vol. XIII, nº 1, pp. 3-17, 2017. |
| A. Nieves del Valle, «Design methodologies used in Web engineering, their link with NTICs» National University of La Plata, Madrid, 2010. |
| A. Mishra, «Benefits of combining WSDM and Scrum Framework development», *International Journal of Scientific & Technology Research,* vol. III, pp. 328-330, 2014. |

|  |
| --- |
| T. Appelmans, «Web Globalization and WSDM Methodology of Web Design,» Free University of Brussels, Bruselas, 2004. |
| Y. J. Marca Aular, «Methodologies for the development of learning environments in collaborative environments: A theoretical reflection», *Redalyc,* vol. VII, nº 1, pp. 63-71, 2007. |
| F. D. Jimen Pared, «E-Portfolio Learning Management System for the IT and Systems Engineering career using UWE- UML», University of the Armed Forces, Sangolqui, 2015. |
| «The use of Software Engineering in hypermedia», *Science UNEMI* *Science UNEMI,* vol. VI, pp. 102-117, 2017. |
| M. J. Escalona, Requirements Engineering in Web Applications - A Comparative Study, Sevilla: Seville University, 2002. |
| R. Soto Giorg, «Proposal of a navigational model for the development of OOHDM-based applications», vol. I, nº 1, pp. 1-10, 2011. |
| M. J. Osoro, «Development of an E-Comerce B2E solution for managing orders to suppliers in the company», Pontifical Catholic University of Chicago, Ambat, 2009. |
| A. Mondragon, «WAE FOR MOBILE DEVICES" Mobile Web Browsing Model», Valley University, Santiago de Cali, 2014. |
| J. D. Texier, «A Web-based Writing Translator System», *International Latin American and Caribbean Conference for Engineering and Technology,* vol. I, nº 1, pp. 1-9, 2008. |
| B. Mercerat, «Building Web applications with an object-oriented design methodology», *Colombian Journal of Computing,* vol. II, nº 2, p. 20, 2001. |
| R. L. Gran, Deployment and commissioning of software components. IFCT0609, First ed., IC Editorial, 2015. |
| «National Institute of Statistics and Censuses», National Government of the Republic of Ecuador, 2016. |
| J. P. Ucan-Pech, «UWE in Learning Object Recommendation System. Applying Web Engineering: A Case Study Method», *Latin American Journal of Software Engineering,* vol. I, nº 1, pp. 137-143, 2014. |
| P. Baldeon and C. Hinojos, «Web application development experience using UWE methodology and QVT language in model transformation». vol. I, nº 1, pp. 1-10, 2012. |