

Article

Approach of Agile Methodologies in the Development of Web-Based Software

Jimmy Molina Ríos ^{1,*}  and Nieves Pedreira-Souto ^{2,†} ¹ Civil Engineering Faculty, Technical University of Machala, Machala 070222, Ecuador² Computer Science Faculty, University of A Coruña, 15071 A Coruña, Spain; nieves.pedreira@udc.es

* Correspondence: jmolina@utmachala.edu.ec

† These authors contributed equally to this work.

Received: 13 September 2019; Accepted: 10 October 2019; Published: 13 October 2019



Abstract: The current inclusion of agile methodologies in web-oriented projects has been considered on a large-scale by software developers. However, the benefits and limitations go beyond the comforts that project managers delimit when choosing them. Selecting a methodology involves more than only the associated processes or some documentation. Based on the above, we could define as the main concerns the approach with which we identify the methodology, the needs of the company, the size, and qualities of the project, and especially the characteristics of agile development that they possess. However, there are several difficulties in selecting the most appropriate methodology due to the features in common; Will it be suitable for my project? What challenges will be presented in the process? Will my team understand each stage? Will I be able to deliver software that satisfies the client? Project managers create these questions, which seem manageable but have huge effects. This paper presents a systematic literature review based on the analysis of the approaches of six web development methodologies. The aim of the study is to analyze the approaches presented by relevant methodologies, identifying their common agile characteristics and managing to contrast both its benefits and limitations during a project. As a result, we could itemize five common features, which are presented within the processes; (1) flexibility, (2) constant communication of the workgroup, (3) use of UML, (4) the inclusion of the end-user and (5) some documentation.

Keywords: web-oriented software; agile approach; hypermedia approach; web development methodologies

1. Introduction

With the constant innovations within digital trends, the use of web applications has increased significantly both personally and in the business world. Providing software with complicated features and a limited delivery time has been managed with the use of agile development methodologies. Developers contemplate these methodologies at a great-scale. Undoubtedly, excellent benefits have been presented, such as agility and efficiency during the execution of the project, customer satisfaction and little documentation.

However, the simplicity and similarity of the characteristics have generated uncertainty for independent managers and developers when selecting the most appropriate methodology, above all by not having in mind the consequences of employing a process that is not capable of delivering the desired results. The most significant decision as a manager is the selection of the method to follow since it could cause complications for the development team, dilemmas with the client, and worse, end-user non-conformity with the product.

Although there are studies that identify the problems encountered during web projects, research that proposes the most important characteristics of web development methodologies is meager and not very detailed, with little consideration to the approaches that the methodologies must possess.

It is indisputable that web development includes a great diversity of approaches that developers consider when executing a web project. These approaches vary depending on the methodology to be used; considered relevant are the hypermedia approach (content-based, and task-based), the agile development approach, and finally the end user-based approach.

The proposed objective of this article is to identify the common agile characteristics of web development methodologies through the analysis of the approach of the six that have a more prominent importance in the literature. Also, to present the approach that has greater transcendence during the processes of web development, offering its benefits and limitations during the project.

The rest of this document is structured as follows—the framework used to understand the related work and the methodological process to carry out the systematic review of the literature are presented in Section 2, while Section 3 presents the results of the analysis and extraction of data executed through the SLR Procedure. Section 4 offers a discussion of the results, followed by conclusions and future work in Section 5.

2. Materials and Methods

Traditional development processes are very different to agile development processes. These include the stages for the life cycle of the product because they are not as flexible as the agile ones [1]. That is to say, in the stages of elicitation and analysis of requirements.

Agile methodologies were created to improve the software development process and the problems associated with the use of traditional methodologies, which presents rigidity and an extensive documentation process [2], as well as arising from best practice ideas from the object-oriented community.

2.1. Web-Oriented Software Development Methodologies

Web-oriented software presents a certain degree of difficulty for its design and development, due to the characteristics that must cover the duration of the life cycle. One of the main complexities is the accessibility of applications, which is due to the development of both the team and the system [3]. According to Torrecilla, Sedeño, Escalona & Mejías, the different development methodologies in web engineering “concentrate essentially on the development phases and do not cover other areas, such as project estimation; quality assurance or team management” [4]. Many web development methodologies arise from the need to implement new features within life cycle processes so they evolve and employ concepts from existing methodologies [5] that define three essential models that are exhibited in Figure 1.

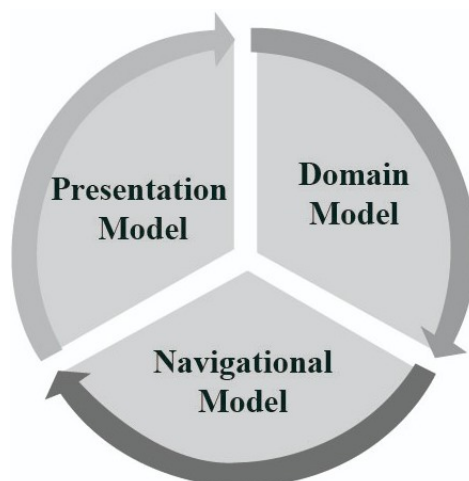


Figure 1. Web design models.

Most web development methodologies embed these models in their design and creation stages.

- **Domain Model:** Also known as a conceptual model, it describes the content of the application to be designed.
- **Navigation Model:** This model defines the navigation between the different pages that will make up the system.
- **Presentation model:** It specifies the content of the presentation of the pages towards the users, focuses on the user interface to be presented.

WebML: This methodology is used to develop web applications driven by processes and also to implement the business model [6], in addition to being used for the specification of complex requirements. Three levels are represented through unified modeling language—appearance, data transformation and content objects [1].

This is not just a methodology, it is also considered a language for the modeling and design of web-oriented applications wherein four relevant models are defined—the structural model, the hypertext model, the presentation model and the personalize model [7].

OOHDM: The life cycle used by the OOHDM (Object-Oriented Hypermedia Design Model) methodology, for the design and development of information systems, is established in three different models—the conceptual model, the presentation model and the navigation model [8]. As it is a methodology oriented to hypermedia, it bases its concepts on the design and stages that it possesses. This uses Object-Oriented data modeling to create a conceptual model [9].

W2000: The conceptual model used by the W2000 methodology is based on schemes, as well as other models oriented to the development of web applications, among them—for the design of execution, diffusion design, design of direction or navigation and finally the design of information [10]. The main focus of the methodology is in the requirements specification phase wherein it neatly makes use of UML notation to model the functional and navigational requirements for hypermedia design.

UWE: In the development of web applications, the UWE methodology stands out for the practical documentation it uses from the creation phase, using the diagrams established by UML notation, clarifying the creation of objects or elements that intervene in each process of the web application [11]. UWE's features allow the covering of the entire software life cycle of web applications while maintaining a user focus and its structure is an iterative and incremental model, which facilitates the control of constant changes in phases during project execution.

WSDM: It is considered a methodology due to its versatile characteristics in the development of applications in short periods. It is also an end-user based methodology considering the user as the main pivot of the system. However, this methodology needs a complementary methodology that carries out the project management taking control over the software life cycle [12].

NDT: According to Reference [13], NDT is a methodology that maintains a specific focus on requirements engineering, taking the objectives as guidelines for carrying out the requirements collection and validation processes. It then classifies them into information storage requirements, stakeholder requirements, functional, interaction and non-functional requirements.

2.2. Approach to Development Methodologies

2.2.1. Development Approach based on Hypermedia

Commonly, hypermedia applications are categorized into two approaches or architectures, the first concerning the content of the application, and the second focusing on the tasks or procedures they perform. According to Suh & Lee [14], among the development methodologies with a focus on tasks are SOHDM, EORM, WHDM; while those based on content are RMM, OOHDM, VHDM.

- **Content-driven Approach**

The content-based approach concentrates on structuring the hypermedia information needed to meet user needs. The project manager must establish the content employed in each node or page. In this way, it is possible to cover the objectives proposed in the planning of the project.

- **Task-driven Approach**

Better interaction with the user is created by separating the concerns related to the project according to the task. Also, this generates a better structure over the processes. The use of a task-based approach gives a prominent advantage during the gathering of system functional requirements.

This taxonomy provides an overview of the tasks that users must perform to interact with web applications hierarchically [15].

2.2.2. Agile Approach to Development Methodologies

The development of applications based on a framework of agile methodologies is implicitly subject to the use of values, principles and agile practices, which positively favor the construction of applications by diminishing aspects such as effort and time. On the other hand, the agile approach gives the faculty to be flexible to the changes generated throughout the life cycle. According to Herrera and Valencia [16], the agile manifesto employs four values during software development:

- Individuals and interactions above processes and tools.
- Software running above documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

Unlike traditional methodologies, agile ones focus on the users involved in the system and give priority to the software functionality rather than the rigorous documentation proposed by others. Additionally, it encourages the practice of support for change, leaving aside the inflexibility to modifications that the application may undergo throughout the life cycle.

2.3. Systematic Literature Review

In spite of the large number of agile methodologies for developing web-based applications, the knowledge expressed in research on them is limited. All of them are based on an approach that is lacking. Consequently, the selection and use of a property must be related to the feature specified by the system.

Despite the information provided in each methodology, it is necessary to generate an SLR, covering the approaches that can present the web development methodologies and the agile characteristics.

The approaches of agile methodologies are diverse, so it is necessary to specify those that will be made relevant to the study. The objective of the SLR is to analyze the approaches based on three characteristics—agile approach (presenting the agile principles that each web methodology possess); hypermedia approach (which classifies in task-driven and content-driven); and finally the end-user oriented approach (being relevant due to the importance of the user in agile methodologies and measuring by usability indicators using the ISO/IEC 9241).

2.3.1. Research Questions

Two research questions have been defined to guide the search of the literature in databases and to allow us to address the main scope of the document.

RQ1: What are the relevant guidelines within agile approaches, task-based and content-based approaches, and end-user-oriented approaches?

RQ1.1: Which characteristics are considered by agile web development methodologies?

RQ1.2: What features do hypermedia-based approaches (task-based approach, and content-based approach) contemplate for web application development?

RQ1.3: What are the indicators that link to end-user-oriented approaches within web development methodologies?

RQ2: What agile methodologies approaches do web-oriented software development methodologies commonly possess?

2.3.2. Review Protocol

The methodological process of searching within this research was based on the search models applied in the investigation of Javaria et al. [17] and Anureet & Kulwant [18] to encompass the purpose of the study.

The process contemplated in the literature review is made up of three phases, which were emphasized in the research in Reference [19], selecting some as priorities and mixing them with other research. The process is composed of three stages:

1. The first stage consists of searching and collecting relevant literature from the specified digital libraries. The search strings defined in Section 2.3.2.1 are used for this search.
2. The second phase consists of the selection of primary studies. This is achieved by the inclusion and exclusion criteria defined in Section 2.3.2.2.
3. The third is the assessment and gathering of data from the selected primary studies. The evaluation is based on the quality metrics assigned in Section 2.3.6.

The review protocol is performed by the authors, who assume the role of data extractors, and reviewers. A pilot test, to validate the proposed methodology to address the research questions, is employed.

2.3.2.1. Conducting the Research

The search strategies were considered according to the search process carried out in the Marques et al. [20] research, which establishes the determination of search strings and digital libraries in which to make queries strategies.

Based on the research questions posed in Section 2.3.1, keywords were identified. Table 1 shows the keywords or search strings used.

Table 1. Search String.

RQ	Key Terms
RQ1	Agile approach, task-driven approach, content-driven approach, web-centred approach to end user, web applications.
RQ2	Web development methodology, OOHDM, WebML, SOHDM, NDT, UWE, WSDM, W2000

The second search strategy consists of reviewing the literature and selecting articles within digital libraries with a high impact. The digital libraries considered within the study are presented in Table 2.

Table 2. Digital Libraries.

N	Library
1	Science Direct
2	Springer Link
3	IEEE Xplore Digital Library
4	ACM Digital Library
5	Google Scholar

The results of the search for primary studies indicate that the libraries with the highest number of selected articles are Science Direct and Springer Link, while IEEE, ACM and “Academic Search Complete” possess fewer. On the other hand, we have considered some relevant studies within libraries such as Google Scholar, redealyc and dialnet, among others.

2.3.2.2. Conditions for the Selection of Primary Studies

The primary study selection process consists of establishing inclusion and exclusion criteria in order to find studies that answer the research questions posed. Besides, metadata (such as abstract, title, keywords, research questions, and results) were considered with direct interest to the topic raised. As part of the search for related information, criteria were established to accept or exclude the research found. In this way, it is possible to obtain research focused mainly on the proposed topic. Below are the inclusion criteria considered.

- The research is focused on the use of web development methodologies.
- Research is considered as a research article, review article, or conference article.
- The research is within the domain of selected libraries.
- The range of publications of research is between 2000 and 2019.

It is important to emphasize that exclusion criteria were applied, which allowed the denial of documents with little relation to the investigation:

- The research has nothing to do with the development of web applications.
- The abstract does not provide sufficient evidence to indicate that the research is linked to web applications.
- The year of publication is not within the established range.
- The language of the publication is different from English or Spanish.

2.3.2.3. Quality Metrics of Primary Studies

For the selection of quality primary studies, metrics are established that allow the subsequent evaluation of the papers and their classification according to criteria such as Selected, Doubtful and Invalid. Each of the categories has a set of indicators related to quality metrics in terms of the metadata gathered.

Studies with “Selected” classification: This includes studies that meet the criteria for inclusion and high reliability in metadata. The metadata considered are the abstract, title, author(s), keywords, and research questions.

“Doubtful” studies: This classification considers studies that have little reliability in terms of the metadata provided or the results from the research are not relevant.

Studies with “Invalid” classification: Include papers that provide metadata without any relation to the subject matter. These are not considered for SLR.

2.3.3. Validation of Review Protocol

The process carried out for the validation of the review protocol included the execution of pilot tests based on the data extraction form. The procedure carried out consists of 3 activities, which tended to be recursive.

1. Arbitrarily select studies for RQ2 (4) and RQ1 (6).
2. Extract data from the articles, dividing the work between the authors and the external collaborator.
3. The indicators of the extraction form are validated. If the form must be modified, return to step 1 again. Otherwise, the validation process ends.

2.3.4. Identification of Relevant Research

For the process of selecting primary studies and identifying relevant research, the snowball technique applied in Wohlin's research [21] was used.

Because the search strings in Table 3 may somehow arbitrarily exclude relevant literature. The following activities are carried out for this purpose.

- Search strings are applied in digital libraries.
- A validation of the articles likely to be included in the research is carried out.
- Relevant and similar literature is identified within the references of previously proposed research.
- New literature is assessed against the inclusion and exclusion criteria.
- The preceding is performed within each RQ.

Table 3. Search strings in digital Libraries

Research Questions	Search Strings
RQ 1.1	"web development" AND "agile approach" AND ("web applications" OR "software web")
RQ1	RQ 1.2 hypermedia approach AND ("task-driven approach" OR content-driven approach) AND web development AND (web applications OR software web OR hypermedia software)
	RQ 1.3 "web-centred approach" AND "end-user" AND "software development methodology" AND "web applications" AND "web software"
	"end-user" AND "usability" AND "ISO 9241-11" AND "ISO 9241-210" AND "software development methodology" AND "web applications" AND "web software"
RQ2	"web development methodology" AND "approaches" AND "features" AND "OOHDM" OR "WebML" OR "SOHDM" OR "NDT" OR "UWE" OR "WSDM" OR "W2000"

An exorbitant number of studies were obtained through digital libraries and the application of search strings. However, the snowball process was applied to each quest, getting additional investigations, which were cited within the research found.

After implementing the technique, a total of 443 studies were collected. Afterward, we proceeded to eliminate all duplicate papers, which are used in more than one investigation, resulting in the removal of 59 documents.

In the second phase, irrelevant literature was removed from the remaining set of 384 documents, resulting in 209 outstanding studies.

Finally, the inclusion and exclusion criteria were applied (mainly the type of article to which it belongs, such as review, research, or conference proceedings), thus finding a total of 188 excluded documents, resulting in 51 remaining studies with which the corresponding research was carried out.

The selection process of the literature used is illustrated in Figure 2, where it is explained from the process of searching for studies to the stages of the selection and elimination of irrelevant papers.

Unlike primary studies, the related research (Table 4) provides additional information on topics ranging from the study of web methodologies to the approach they may have.

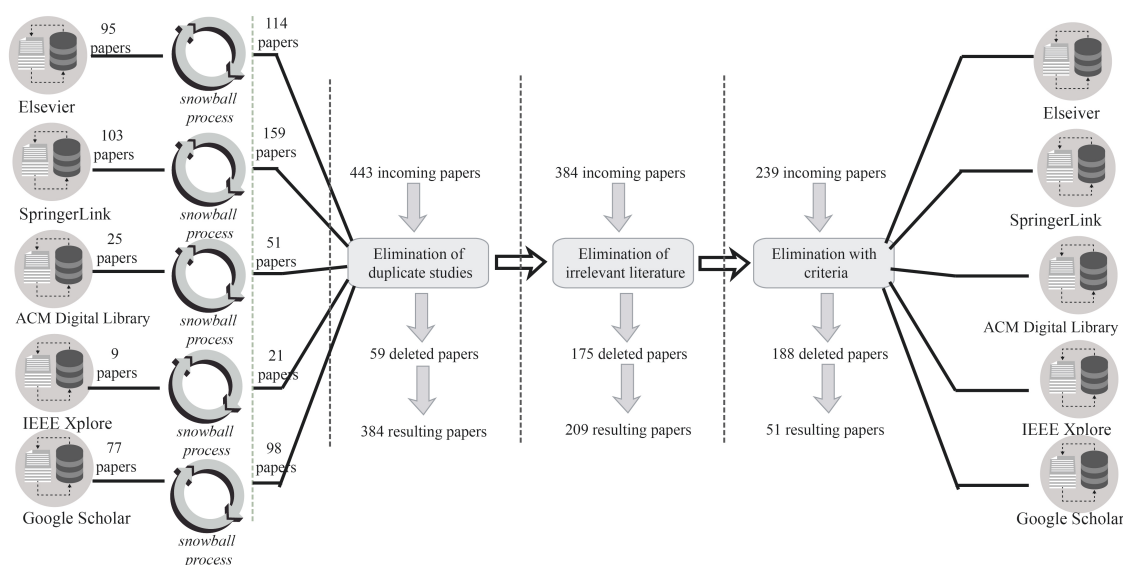


Figure 2. Selection of studies process.

Table 4. Related Research.

	Author	Title
DOC1	Madjid Sadallah, Olivier Aubert Yannick Prié	CHM: an annotation- and component-based hyper video model for the Web. [22]
DOC2	Wei Wang, Wenjie Zhang	Guest editorial: web applications and techniques. [23]
DOC3	Cristina Cachero Castro, Santiago Meliá, Marcela Genero, Geert Poels, Coral	Towards improving the navigability of Web applications: a model-driven approach. [24]
DOC4	V. V. Lipaev	Problems of the development and quality control of large software systems. [25]
DOC5	M.J Taylor, J McWilliam, H Forsyth, S Wade	Methodologies and website development: a survey of practice. [26]
DOC6	Dimitris Stavrinoudis, Michalis Xenos, Pavlos Peppas, Dimitris Christodoulakis	Early Estimation of Users’ Perception of Software Quality. [27]
DOC7	Davide Bolchini, Paolo Paolini	Goal-driven requirements analysis for hypermedia-intensive Web applications. [10]
DOC8	Mahdi Fahmideh, Gholami Mohsen Sharifi, Pooyan Jamshidi	Enhancing the OPEN Process Framework with service-oriented method. [28]
DOC9	Fernando Lyardet, Gustavo Rossi, Daniel Schwabe	Discovering and Using Design Patterns in the WWW. [29]
DOC10	T. Madhusudan, Young-Jun Son	A simulation-based approach for dynamic process management at web service platforms. [30]

2.3.5. Selection of Primary Studies

Primary studies are considered based on the relevant studies obtained by searching the databases for information so that these primary studies provide the necessary and sufficient content to cover the research questions raised within the systematic review of the literature.

The studies included in the “selected” classification are those that were considered primary studies, and therefore their information was used to answer the research questions. Meanwhile, studies classified as “dubious” and “invalid” were discarded from possible primary studies and used as related research, found for the theoretical framework and referenced articles.

For each question, a set of studies was assigned subdividing RQ2 into articles referring to web development methodologies, and RQ1 assigned to the approaches (Table 5).

Table 5. Primary studies per question

RQ	Subdivision	Selected studies
RQ1	Agile approach	[31] Agile characteristics are presented that are considered when developing hypertext applications focused on the elicitation of requirements and stages of analysis of the life cycle of the development of web applications.
		[16] Describes a short review of the agile principles and values that are put into practice by software industries and how it has contributed to the development of applications. [32] Based on proposals such as XP, they examine how the approach applied to develop an application in an unpredictable environment influences its changes and describe other benefits of the agile approach.
	Hypermedia approach	[33] In web application development he refers to how he employs the prototyping and incremental process approach and how it relates to hypermedia. [34] Describes the characteristics of the hypermedia approach and how it contributes to the educational web application development environment. [35] Presents the characteristics of task-based models, as well as the benefits and results of integrating annotations into the life cycle of development methodologies.
End-user oriented approach	[36] Describes how the end-user approach influences the use of techniques to achieve its objectives and the importance of coupling it into the life cycle of web application software. [37] Presents the features of a new methodology that emphasizes efforts to guide software designers and end users through the software development process. [38] Establishes the characteristics of model-based methodologies to improve participation and the constant requirements implied by end users.	

Table 5. Cont.

RQ	Subdivision	Selected studies
RQ2	WebML	[39] Establishes a perspective of how to carry out the modeling of the project that allows the comprehension and reuse to other existing models, focusing on the adaptability of the modeling of the web systems.
		[40] The approach used is characterized by its versatility to work with back-end and front-end without restricting user interaction in the graphical interface.
	OOHDM	[41] Analyzes design patterns used in web applications based on the OOHDM approach to design recycling and standardizing a web application model.
		[42] Examines the OOHDM approach to the modeling of complex web applications that are hypermedia-supported and represented through object modeling.
	UWE	[11] Characterizes the UWE methodology by its structure focused on content modeling, navigation, presentation and processes; and its relationship with UML to stage the objects that compose it.
		[12] The UWE methodology relates it to model-driven development and emphasizes the collection of functional requirements, however, it is not able to categorize them.
	WSDM	[43] As for web development, this methodology focuses mostly on users, giving way to new diagrams that prioritize the user and their interests.
		[44] Through a case study, he explains how the WSDM approach focuses on user comfort, separating application structure and workflow.
NDT	[45] Evaluates the processes of the methodology with respect to the international standard ISO/IEC 12207 that normalizes the processes of the software life cycle; and describes the approach on the design of the system.	
	[13] Describes NDT's approach as being more related to requirements engineering, however, in the implementation and testing phases it suffers from an unevenness as it is not contemplated in web development.	
W2000	[46] They elaborate a new design framework taking the approach of the W2000 methodology which is based on the agile development of prototypes, and the framework with which it works.	
	[47] Presents the methodology and approach to system functional requirements by extending UML notation, similar to HDM.	

2.3.6. Data Extraction from Primary Studies

The data extraction process was performed manually by the authors, following the structure of the data extraction form established at the planning stage.

The form allows the verification of the indicators that must be extracted from each article, referring to each classification according to the research question that they are destined to solve.

The indicators extracted in studies are:

- Agile development characteristics.
- Examples of methodologies used by each approach.
- Conditions of use.
- The advantages or strategies within the stages of the approach development process.
- End-user activities within web development.
- Common problems regarding customer satisfaction with web software.
- Usability features.

2.3.7. Data Synthesi

Once the primary studies were selected (21 papers in total) and the respective information was extracted, the thematic analysis technique embodied in the research of Sari et al. [48] and Braund and Clarke [49] is applied.

Thematic analysis was used to synthesize information corresponding to qualitative data generating patterns and themes. Within this technique, we selected an inductive approach. In other words, the data from the articles were read and extracted.

2.3.8. Threat of Validity

There were several validity threats identified during the design of this study. There are a relatively large number of web software development methodologies, but, for the analysis within this research, only 6 of them were contemplated. When expanding the work, of course, several methodologies should be included. That will allow us to identify with greater precision the characteristics they possess and the differences between each of them.

During the data collection, one researcher evaluated the studies, which can lead to subjectivism regarding the results obtained. However, to mitigate the validity at this point, we discussed some unclear studies.

Other threats to data collection are that the search strings used by research questions may exclude related terminologies, for which a snowball technique was used. Besides, the data form was reviewed by an external evaluator, which may imply a slightly different perspective from the main focus of the article.

Our research, due to the limitations and validity threats posed, is recommended to be used for intermediate studies of the proposed methodologies. Likewise, it can be used to identify the agile features that some of the web development methodologies have in common, the development approach that has greater relevance by developers, and to conclude the features that benefit the development process and which are contradictory to obtaining an effective application.

On the other hand, its use is not recommended for research that generalizes the stages and components used within the agile processes of the development methodologies. Nor should it be used to conclude with the selection of a more propitious or adequate methodology for development, because a deep comparison is not made of its limitations and benefits.

3. Results

For the presentation of the results, comparative tables are made between the selected primary studies and the main characteristics to contemplate each of the research questions.

RQ1: What are the relevant guidelines within agile approaches, task-based and content-based approaches and end-user-oriented approaches?

RQ1.1: Which characteristics are considered by agile web development methodologies?

Based on the analysis and extraction of data according to the extraction form, Table 6 is carried out, comparing the selected research and offering possible characteristics for the comparison and subsequent characterization of the development methodologies.

Table 6. Analysis of agile characteristics in development considered in primary studies

	[31]	[16]	[32]
Agile development features.	<ul style="list-style-type: none"> • People. • Resources and ontologies. • Scenarios. • Story Boarding (UI Design). • Use Case. 	<ul style="list-style-type: none"> • Simplicity. • Sustainable development. • Face to face conversation. • Self-organized teams. 	<ul style="list-style-type: none"> • Prepared for changes throughout the project. • The customer is part of the team. • Few artifacts and roles.
Agile principles.	<ul style="list-style-type: none"> • Software delivery in a short timescale. • Close communication within stakeholders. • Simplicity. • Programming documentation. 	<ul style="list-style-type: none"> • Customer satisfaction. • Acceptance of changes. • Short execution period. • Stakeholder Integration. • Constant communication. 	<ul style="list-style-type: none"> • The interactions between development team and tools. • Emphasis on functional software. • Continuous collaboration with the client. • Quick response to changes.
Conditions of environments.	<ul style="list-style-type: none"> • Software development conditions are limited by a short delivery timescale and large industries. 	<ul style="list-style-type: none"> • Motivated work environments and a good relationship between teams. 	<ul style="list-style-type: none"> • Small development team • Environment suitable for communication and collaboration. • Technologies that facilitate the change of requirements.
Problems to be solved.	<ul style="list-style-type: none"> • Speed and flexible response to change. • Functionality complexity increases. 	<ul style="list-style-type: none"> • The phases become activities that are organized in small packages for easy realization. • Direct customer participation throughout the project. 	<ul style="list-style-type: none"> • Problems posed when executing small projects, and with frequent changes.

RQ1.2: What features do hypermedia-based approaches (task-based approach, and content-based approach) contemplate for web application development?

Table 7 presents a comparison between approaches that methodologies present. Agile web-oriented methodologies are very similar due to the use of hypermedia concepts. As a result, they tend to fall into two categories, content-oriented and task-oriented methodologies.

Table 7. Hypermedia developmental characteristics analysis considered in primary studies.

	Content-Driven Methodologies	Task-Driven Approach Methodologies
Interaction with the user.	Interaction with the user is minimal.	Has greater interaction with the customer.
Conceptual model design phase.	It depends on the conceptual data model to achieve a hierarchical analysis of the content.	Analyzes and processes content data separately, according to the task at hand.
Elicitation of requirements.	The specification of requirements is complex, and brainstorming is necessary.	Requirements collection is simple, and pre-established models are used.
Methodologies.	<ul style="list-style-type: none"> - RMM - OHDM - VHDM 	<ul style="list-style-type: none"> - SOHDM - EORM - WHDM

The result presented within the systematic literature review is carried out according to the primary studies selected as adequate to contemplate the answer to the research question posed. To identify the features, the analysis of primary studies, related to hypermedia approaches, is carried out in Table 8. The selection of indicators arises from the correlation between the content of each work of the SLR. In this way, aspects were identified that allow a precise comparison of those common characteristics present within the agile approach.

Table 8. Analysis of data extracted from primary studies regarding agile approach.

	[33]	[34]	[35]
Kind of approach.	Prototyping.	Agile.	Hypermedia.
Conditions established for its use.	Short-term projects, multimedia content management.	Organization by teams. Improvement of the final product.	Special focus on requirements and users.
Examples of methodologies.	- Scrum - EssUP (Essencial Unified Process) - W2000	- Extreme Programming (XP) - KanBan - Scrumban	- NDT - WSDM - OOHDm
Strategies in stages of the development process.	- Employ User Stories. - Task board. - Use meta-models for information, navigation, etc.	- Perform unit tests. - Whiteboard with sticky notes for process management.	- Generation of models for system analysis. - Capture of requirements through UML notation.

RQ1.3: What are the indicators that link to end-user-oriented approaches within web development methodologies?

Based on the analysis and extraction of data according to the extraction form shown in the previous sections, Table 9 is carried out, which compares the selected research and conceptualizing the characteristics and problems that the web development methodologies have, focusing mainly on their features in the end-users.

Table 9. Analysis of data extracted from primary studies regarding hypermedia approaches.

	[36]	[37]	[38]
End-user activities within web development.	- Requirements specifications. - End users formulate a narrative description of the solution. - Verify that the provisional designs output work properly.	They perform two fundamental roles: 1.Perform assigned tasks. 2.Participate as an interested party in the domain.	Puts the user in front and gives him the ability to create his own applications on the fly that satisfy a situational need.
Common Problems.	- They do not define the requirements correctly from the beginning. - They do not perform a proper test for error detection.	The end user pretends to be a consumer and designer.	The main problems are presented at the design stage, for the use of interfaces, organization and coordination of resources.
Usability features.	The study has not specify features of usability.	- The environments that employ the end user approach must meet usability requirements. - It is evaluated by workshops.	- Visibility. - Feedback. - Consistency. - Non-destructive operations. - Discoverability. - Scalability. - Reliability.

The importance of the end-user during the software development process lies mainly in the usability features, the user experience, and the interaction between the system. To this end, it considers more feasible the use of the characteristics of Regulation 9241 as the bases necessary for the evaluation of the usability of the systems concerning the end-user.

The research of the mentioned authors identifies the focus of the final user in different activities. However, it can be appreciated that they consider fundamental user participation within the Project. An end-user can act as a consumer (to test the software) and as a designer (to make the software) and, depending on their function, will carry out the necessary activities.

The indicators established for the comparison were defined based on general and basic knowledge about the inclusion of an end-user in the development process. On the other hand, the indicator related to usability is applied due to the crucial link between quality characteristics and the correct development process, thus considering the importance that the authors bring to this issue.

RQ2: What agile methodologies approaches do web-oriented software development methodologies commonly possess?

The methodologies selected for the study (WebML, OOHDM, UWE, WSDM, NDT and W2000) present similar characteristics in terms of the flexibility of the agile development and the iterative process during its life cycle. However, there are also differences, which characterize each of these methodologies. These characteristics are presented in Table 10, according to selected indicators of the different approaches studied in research question RQ1.

Each one of the identified characteristics are pronounced in the primary studies considered. However, there was some discrepancy between some authors, so it was convenient to reflect the features that had greater acceptance and were mentioned more in the research papers.

Table 10. Comparison of the methodologies within the perspective of agile development approach.

Activities	WebML	OOHDM	UWE	WSDM	NDT	W2000
Agile features	Iterative development. Correction of errors.	Small development group. Adaptation to changes.	Correction of errors. Simplicity in the code.	User Orientation. Few documentation.	Unit tests. Function. Prioritization.	Flexibility to change. User Orientation.
Application in distributed areas	Does not apply	Applies	Does not apply	Applies	Does not apply	Applies
Kind of Hypermedia Approach	Content-driven	Model-driven	Content-driven	Model-driven	Task-driven	Task-driven
End-User Intervention	Throughout the development process.	Throughout the development process.	Throughout the development process.	Throughout the development process.	Throughout the development process.	Throughout the development process.
End-User based approach	Applies	Applies	Applies	Applies	Applies	Applies

It is necessary to take into account the circumstances in which it is advisable to use different approaches and this will depend on the requirements of the project to be developed.

Table 11 presents a comparison between the different approaches to development methodologies, evaluating them according to the functional requirements of web-engineering obtained from Paydar & Kahani’s research [50], and good practices suggested in Garousi, Petersen & Ozkan’s research [51].

The rating used in Table 11 refers to four categories (High–Medium–Low–No apply) essential for the evaluation of the requirements posed for each of the development approaches. Each one is assessed according to the specific criteria presented in Table 12. The conclusion and establishment of each one of the categories for the requirements and achieving the completion of the table is done based on the primary studies analyzed within the SLR.

Table 11. Comparison between approaches of web development methodologies.

Requirements	Hypermedia-based approach		Agile approach	End-User based approach
	Content-driven	Task-driven		
Allows reuse.	High	Medium	High	No apply
Allows integration of types of software artifacts.	High	Low	High	Medium
Ensures constant maintenance.	High	Low	High	High
Allows flexibility.	High	High	High	Medium
Ensures communication with the development team.	Low	Medium	High	High
Enables end user collaboration	Low	Medium	High	High
It is accessible to constant modifications	High	High	High	High

Table 12. Comparison of the methodologies within the perspective of agile development approach.

Requirements	Categories	
Allows reuse.	High	The interface design is specified in detail. Provides documentation of each of its functionalities and activities during the life cycle.
	Medium	Certain functionalities and components can be used in other projects. Documentation is poor.
	Low	It does not provide the necessary resources to use the modules in other projects.
Allows integration of types of software artifacts.	High	The life cycle process integrates in its entirety various artefacts, such as UML, object-oriented, among others.
	Medium	Allows the integration of artefacts in certain stages of the life cycle.
	Low	Includes the use of artifact only in the main stages.
Ensures constant maintenance.	High	Provides the proper guidance for hassle-free maintenance.
	Medium	Maintenance is performed correctly but can have setbacks.
	Low	Constant maintenance of the project is complex, and time consuming.
Allows flexibility.	High	Allows requirements to be updated and aggregated without any problem.
	Medium	The implementation of requirements implies difficulties for the working group and the development process.
	Low	It is not possible to implement new requirements or modifications at any stage, only at the beginning.
Ensures communication with the development team.	High	Communication and interaction among team members occurs at each stage of development.
	Medium	There is constant communication but it does not include all the members of the working group.
	Low	Communication is mild and only occurs at certain stages.
Enables end user collaboration.	High	The end user interacts throughout the process and is a constant part of the software design and modification.
	Medium	The user only intervenes in the necessary stages (requirements and evaluation). Minor changes are discussed.
	Low	The end user only intervenes in specific stages and tasks. There is no follow-up during the development process.
It is accessible to constant modifications	High	Changes can be made at any stage of the life cycle.
	Medium	Changes can be made at specific stages of the life cycle without seriously affecting the process.
	Low	Changes can simply be made in the initial stage.

3.1. Quantitative Analysis of Studies

The majority of the selected studies present a large-scale knowledge of a topic that emphasizes the importance of our research. A quantitative analysis allows us to ensure the validity and reliability of the results proposed within this study. Thus, they allow us to identify a number of papers that support the decisions made.

The total research articles employed in the SLR analysis and in the resolution of the RQs is 21 (Table 5). The number of studies that support the results acquired in research question 2, are shown in Figure 3.

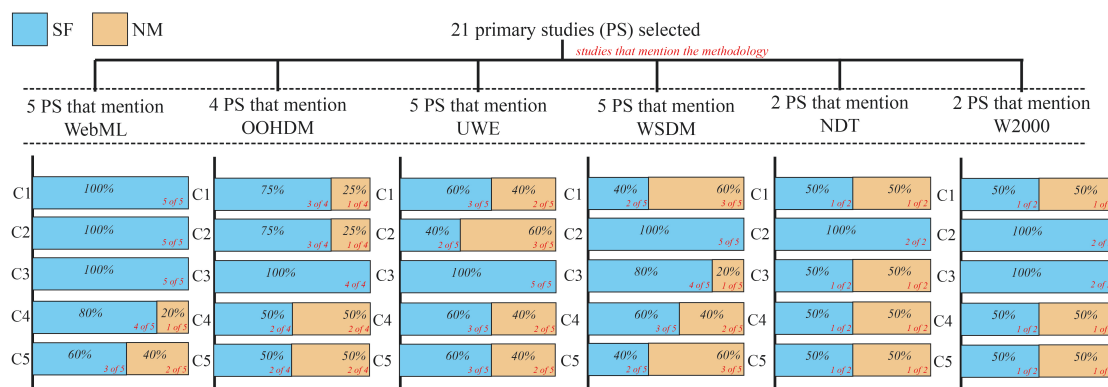


Figure 3. Quantitative analysis of primary studies by methodology.

In Figure 3, we represent quantitatively the studies that support each decision made; therefore, we identified the number of papers that names each of the methodologies to be analyzed. It is important to realize that some research mentions more than one methodology, so it was used for the analysis of both.

The first sub classification identifies 5 of 21 papers that mention WebML, as well as UWE, WSDM. On the other hand, 4 of 21 papers mention OOHDM and 2 of 21 mention NDT, likewise W2000.

Then we made a diagram to identify the percentage of studies that were in favor of each feature established and which do not mention them. For this reason, two categories of documents were proposed—SF (studies that support our decision) and NM (studies that do not mention the activity).

3.2. Common Characteristics of Agile Methodologies

One of the objectives of the study is the identification of common agile characteristics of the methodologies. To this end, the results obtained from the research questions were analyzed. The following characteristics were identified:

- Flexibility before changes of requirements at any stage of the life cycle. Without affecting the development process and planning.
- Constant communication between the development team, despite distributed environments. Including meetings, communication of functionalities and modifications made to the project.
- Inclusion of the end-user throughout the software life cycle. As a result, developers will know the right modifications at the right time.
- Necessary and accurate documentation. Without having to maintain extensive paperwork, these methodologies perform specific documentation of the functionalities of the application.
- Implementation of UML artifacts for the elaboration of documentation and design models.

Each of the characteristics presents a benefit and a limitation within the project. Some of them are explained below:

Flexibility: It is well known that this characteristic is fundamental for agile development. It allows a reduction of the risks and delays of delivery of the software; the project can admit changes during any stage and when finalizing the software fulfills the expectation of the client.

Nevertheless, this feature can affect the developer's work environment because of the need to include the user throughout the process. This point will be explained in more detail within the end-user inclusion feature.

Another change that must be considered due to this characteristic is the modification of the initial approach. It is true that, as the development process progresses, the software must be tested by the client in order not to exclude functionalities that he wants. Notwithstanding, the repeated modifications mean that the user is not decided about the functionality of the system, causing the main scope of the application to be altered, sometimes objected from the original purpose.

Ongoing communication: A good project is born from good communication. Communication is one of the features that bring more benefits because it provides a better working environment for staff. This condition also addresses the use in distributed environments, without considering the setbacks that may occur during the process.

Implementation of UML artifacts: the use of the UML language makes it easy to design models and diagrams, which are required for the specification of the documentation.

End-user inclusion: Another known feature within the agile approach is total user inclusion. The collaboration helps the project to develop according to the needs of the client without excluding any functionality. It helps to guarantee quality software and client satisfaction.

However, total client inclusion leads to internal problems within the development process. One of them, which is very little known, is the discomfort of the developer and analyst. The issue arises because the user does not know what he wants, allowing with each revision to add a new requirement or modify an existing one.

Modifications imply a redesign of the application by the analyst and a modification in terms of coding and sometimes to the database by the developer. Although the modifications are not the most appropriate for the application, they are necessary to obtain customer satisfaction.

One of the recommendations to avoid the above is to establish limitations and tasks specific to the client. Also, limit the range of requirements changes that users can make and establish the initial scope without major subsequent modifications.

Necessary documentation: Most project managers are inclined to agile methodologies due to their lack of documentation. It allows managers to reduce the project life-cycle and focus their attention on other concerns. However, little documentation may affect the project in the future. One weakness identified is that changes are not made efficiently.

If you want to add new functionality to the application, understanding the structure of the system will be complex. That is because changes and updates are superficially documented. Managers do not include all the relational and navigational models of the application carried out, generating confusion in both the developer and the analyst.

4. Discussion of the Results

As has been mentioned within the research, a development methodology may have several approaches, of which just three have been analyzed (hypermedia, agile, and end-user-oriented approach). Despite having a different perspective, each one is important at the moment of efficiently running a software project. However, through the analysis tables obtained in the results, it is possible to state that the task-oriented approach presents a greater degree of acceptance as well as the end-user-oriented one.

By implementing a task-based approach, the project tends to produce quick and efficient results. Because each team member is bound to a specific activity, so the functionalities of the application are covered. Additionally, the implementation of modules presents a hierarchy and little risk in the project. But what happens if one task has delays or failures? Or if the results that the person in charge carried

out are not the most convenient for the activity that continues? These are some of the questions a manager should consider when employing a task-driven approach.

Indeed, a task-based or activity-based approach is much more effective than a content-based one. Mainly because of the decrease in errors that usually occur during the execution of a web project. However, it is essential to consider the following characteristics when working with a task-driven distribution. (1) Distribute the activities according to the qualities of each member; (2) Consistency and hierarchy of each of the system's functionalities must be maintained, from highest to lowest; (3) Delays should be taken into account and avoid linking activities that are detrimental to planning; (4) The revisions of the results must be constant, correcting errors if necessary; (5) The functionalities of the system must satisfy the requirements of the user, and the user must verify them.

On the other hand, the end-user approach is mentioned in most development methodologies. The high degree of acceptance and favorable responses within projects makes this approach one of the most important during the analysis. It is indisputable that the inclusion of the user within the processes is a priority when speaking of agile practices. However, the pressure exerted by them can generate annoyances to the developer, mainly by limiting the development to modifications outside the initial scope of the project. Also, this approach is not very accurate in some methodologies, considering "the end-user" as a member of the team that plays two roles. The first is the designer (in charge of performing design tasks), and the second is the consumer (in charge of testing the application from the user's perspective). Assuredly, the main objective of the study contemplated the identification of the agile characteristics that the methodologies have in common, for which it was necessary to analyze the literature based on relevant factors. As a result, five features were obtained.

- Flexibility before changes of requirements in any stage of the life cycle. Without affecting the development process and planning.
- Constant communication between the development team, despite distributed environments. Including meetings, communication of functionalities, and modifications made to the project.
- Inclusion of the end-user throughout the software life cycle. As a result, developers will know the right modifications at the right time.
- Necessary and accurate documentation. Without having to maintain extensive paperwork, these methodologies perform specific documentation of the functionalities of the application.
- Implement UML artifacts for the elaboration of documentation and design models.

Each feature presents a range of benefits during the development process. However, it is astonishing that most of the studies do not identify or consider the limitations that these features present. Flexibility, ongoing communication, and end-user inclusion could affect the project and, on a minor-scale, the developer. Allowing modifications in each review, which sometimes are drastic, implies the developer has to redesign the interface and source code, continually putting pressure on his work. Also, change requirements in some stages involves a variance in the initial scope that the project possesses. Nevertheless, some strategies permit limiting the inclusion of the end-user without diminishing flexibility.

On the other hand, the importance of documentation has been replaced by minimal paperwork (only what is necessary). Allowing less complexity for the manager, but more likelihood of future risks in terms of revisions and updates of the project. Based on the above, the use of UML diagrams has facilitated the execution of both conceptual and navigational models, but they are not evaluated and redesigned with each change management performed.

5. Conclusions and Future Work

In this study, the analysis of six web development methodologies was presented, considering three approaches—the agile approach, the hypermedia approach and, finally, the end-user-oriented approach, each of which is contemplated relevant to the development process according to the features they present and the environment in which they are used.

Given the results, it can be concluded that web development methodologies do not embrace a specific approach. So, each one of them has its characteristics for which they implement the most pertinent approach. It is important to assess the environment of the work, the scope of the project, the approach and the characteristics of the development methodology to select the most suitable methodology.

However, it was identified that the evaluated web development methodologies, in their entirety, include a user-end and hypermedia-based approach. According to its characteristics, the task-based approach presents greater beneficence within the life cycle, because the separation in functionalities offers greater comfort to the work group and efficiency at the moment of modifications.

One of the objectives of the study is the identification of common agile characteristics. Based on the results obtained during the review, we determined five common features—(1) Flexibility before changes of requirement in any stage; (2) Constant communication between the members of the teamwork; (3) Inclusion of the final user during all the cycle of life; (4) Necessary and precise Documentation, without needing to maintain an extensive paperwork; (5) Use of UML devices for the elaboration of the documentation and modeling of design.

The five characteristics mentioned must be used correctly during the process. The manager must establish the limitations, functions and interventions that the end-user will have within the project, avoiding the modifications desired by the client deviating the project from its original scope, and the members (developers, testers, among others) are negatively affected.

One of the areas on which we are working is the proposal of strategies that allow us to evaluate a methodology for its selection. Establishing benefits and disadvantages that the decision has within a web project, as well as the problems that could arise. In this way, future difficulties within the life cycle are avoided because when a company selects a methodology it must know several indicators that validate its efficiency in the project. An example is the use of methodologies that are not accessible to change requirements at any stage and their effect on the progress of the project.

Author Contributions: Investigation, J.M.R. and N.P.-S.; Methodology, J.M.R. and N.P.-S.; Writing—original draft, J.M.R. and N.P.-S.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SLR: Systematic Literature Review

References

1. Heeager, L.T.; Nielsen, P.A. A conceptual model of agile software development in a safety-critical context: A systematic literature review. *Inf. Softw. Technol.* **2018**, *103*, 22–39. [[CrossRef](#)]
2. Eom, H.E.; Lee, S.W. Human-centered software development methodology in mobile computing environment: agent-supported agile approach. *EURASIP J. Wirel. Commun. Netw.* **2013**, *2013*, 111. [[CrossRef](#)]
3. Martín, A.; Rossi, G.; Cechich, A.; Gordillo, S. Engineering Accessible Web Applications. Anide Aspect-Oriented Approach. *World Wide Web* **2010**, *13*, 419–440. [[CrossRef](#)]
4. Torrecilla-Salinas, C.J.; Sedeño, J.; Escalona, M.J.; Mejías, M. Estimating, planning and managing Agile Web development projects under a value-based perspective. *Inf. Softw. Technol.* **2015**, *61*, 124–144. [[CrossRef](#)]
5. Garrigós, I.; Gomez, J.; Houben, G.J. Specification of personalization in web application design. *Inf. Softw. Technol.* **2010**, *52*, 991–1010. [[CrossRef](#)]
6. Hermida, J.M.; Meliá, S.; Montoyo, A.; Gómez, J. Applying model-driven engineering to the development of Rich Internet Applications for Business Intelligence. *Inf. Syst. Front.* **2013**, *15*, 411–431. [[CrossRef](#)]

7. Di Lucca, G.A.; Fasolino, A.R. Testing Web-based applications: The state of the art and future trends. *Inf. Softw. Technol.* **2006**, *48*, 1172–1186. [[CrossRef](#)]
8. Haustein, S.; Pleumann, J. A model-driven runtime environment for Web applications. *Soft. Syst. Model.* **2005**, *4*, 443–458. [[CrossRef](#)]
9. Atzeni, P.; Merialdo, P.; Mecca, G. Data-Intensive Web Sites: Design and Maintenance. *World Wide Web* **2001**, *4*, 21–47. [[CrossRef](#)]
10. Bolchini, D.; Paolini, P. Goal-driven requirements analysis for hypermedia-intensive Web applications. *Requir. Eng.* **2004**, *9*, 85–103. [[CrossRef](#)]
11. Nieves-Guerrero, C.; Ucán-Pech, J.; Menéndez-Domínguez, V. UWE en sistema de recomendación de objetos de aprendizaje aplicando ingeniería web: un método en caso de estudio. *Rev. Latinoamericana de Ingeniería de Softw.* **2014**, *2*, 137–143. [[CrossRef](#)]
12. Ríos, J.R.M.; Ordóñez, M.P.Z.; Segarra, M.J.C.; Zerda, F.G.G. Comparación de metodologías en aplicaciones web. *3c Tecnología: Glosas de Innov. Aplicadas a la Pyme* **2018**, *7*, 1–19. [[CrossRef](#)]
13. Escalona, M.J.; Torres, J.; Mejías, M. NDT: Navigational Development Techniques. Proceedings of IV Jornadas de Trabajo Dolmen, Alicante, Spain, 12–14 November 2003.
14. Suh, W.; Lee, H. A methodology for building content-oriented hypermedia systems. *J. Syst. Softw.* **2001**, *56*, 115–131. [[CrossRef](#)]
15. Valderas, P.; Pelechano, V. Introducing requirements traceability support in model-driven development of web applications. *Inf. Softw. Technol.* **2009**, *51*, 749–768. [[CrossRef](#)]
16. Uribe, E.H.; Ayala, L.E.V. Del manifiesto ágil sus valores y principios. *Scientia Et Technica* **2007**, *XIII*, 381–386.
17. Imtiaz, J.; Sherin, S.; Khan, M.U.; Iqbal, M.Z. A systematic literature review of test breakage prevention and repair techniques. *Inf. Softw. Technol.* **2019**, *113*, 1–19. [[CrossRef](#)]
18. Kaur, A.; Kaur, K. Investigation on test effort estimation of mobile applications: Systematic literature review and survey. *Inf. Softw. Technol.* **2019**, *110*, 56–77. [[CrossRef](#)]
19. Brereton, P.; Kitchenham, B.A.; Budgen, D.; Turner, M.; Khalil, M. Lessons from applying the systematic literature review process within the software engineering domain. *J. Syst. Softw.* **2007**, *80*, 571–583. [[CrossRef](#)]
20. Marques, M.; Simmonds, J.; Rossel, P.O.; Bastarrica, M.C. Software product line evolution: A systematic literature review. *Inf. Softw. Technol.* **2019**, *105*, 190–208. [[CrossRef](#)]
21. Wohlin, C. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, London, UK, 13–14 May 2014; pp. 38:1–38:10.
22. Sadallah, M.; Aubert, O.; Prié, Y. CHM: An annotation- and component-based hypervideo model for the Web. *Multimed. Tools Appl.* **2014**, *70*, 869–903. [[CrossRef](#)]
23. Wang, W.; Zhang, W. Guest editorial: web applications and techniques. *World Wide Web* **2015**, *18*, 1391–1392. [[CrossRef](#)]
24. Cachero Castro, C.; Meliá, S.; Genero, M.; Poels, G.; Calero, C. Towards improving the navigability of Web applications: A model-driven approach. *Eur. J. Inf. Syst.* **2007**, *16*, 420–447. [[CrossRef](#)]
25. Lipaev, V.V. Problems of the development and quality control of large software systems. *Program. Comput. Softw.* **2005**, *31*, 47–49. [[CrossRef](#)]
26. Taylor, M.J.; McWilliam, J.; Forsyth, H.; Wade, S. Methodologies and website development: A survey of practice. *Inf. Softw. Technol.* **2002**, *44*, 381–391. [[CrossRef](#)]
27. Stavrinoudis, D.; Xenos, M.; Peppas, P.; Christodoulakis, D. Early Estimation of Users' Perception of Software Quality. *Softw. Qual. J.* **2005**, *13*, 155–175. [[CrossRef](#)]
28. Gholami, M.F.; Sharifi, M.; Jamshidi, P. Enhancing the OPEN Process Framework with service-oriented method fragments. *Softw. Syst. Model.* **2014**, *13*, 361–390. [[CrossRef](#)]
29. Lyardet, F.; Rossi, G.; Schwabe, D. Discovering and Using Design Patterns in the WWW. *Multimed. Tools Appl.* **1999**, *8*, 293–308. [[CrossRef](#)]
30. Madhusudan, T.; Son, Y.J. A simulation-based approach for dynamic process management at web service platforms. *Comput. Ind. Eng.* **2005**, *49*, 287–317. [[CrossRef](#)]
31. Wills, G.B.; Abbas, N.; Chandrasekharan, R.; Crowder, R.M.; Gilbert, L.; Howard, Y.M.; Millard, D.E.; Wong, S.C.; Walters, R.J. An Agile Hypertext Design Methodology. Proceedings of the 18th Conference on Hypertext and Hypermedia, Manchester, UK, 10–12 September 2007; pp. 181–184.

32. Canós, J.H.; Letelier, M.C.P.P. Metodologías ágiles en el Desarrollo de Software. Available online: <http://roa.ult.edu.cu/handle/123456789/476>, accessed on 13 October 2019.
33. Álvarez, D.J.S.; Villamizar, I.A.G.; Villalba, R.L. ISO SPICE en sistemas hipermedia educativa. *Rev. UIS Ing.* **2003**, *2*, 63–72.
34. Hernandez, A.; Martínez, M.D. La investigación evaluativa: Enfoque estratégico para una educación a distancia en entornos virtuales de calidad. *Aula de Encuentro.* **2014**, *16*, 108–129.
35. Feldmann, M.; Hubsch, G.; Springer, T.; Schill, A. Improving Task-driven software development approaches for creating service-based interactive applications by using annotated web services. Proceedings of the Fifth International Conference on Next Generation Web Services Practices, Prague, Czech, 9–11 September 2009; pp. 94–97.
36. Lizcano, D.; Alonso, F.; Soriano, J.; Lopez, G. A Web-centred Approach to End-user Software Engineering. *ACM Trans. Softw. Eng. Methodol.* **2013**, *22*, 36:1–36:29. [[CrossRef](#)]
37. Costabile, M.F.; Fogli, D.; Mussio, P.; Piccinno, A. A meta-design approach to end-user development. Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05), Dallas, TX, USA, 20–24 September 2005; pp. 308–310
38. Rivero, J.M.; Luna, E.R.; Grigera, J.; Rossi, G. Improving user involvement through a model-driven requirements approach. Proceedings of 2013 3rd International Workshop on Model-Driven Requirements Engineering (MoDRE), Rio de Janeiro, Brazil, 15 July 2013; pp. 20–29
39. Moreno, N.; Fraternali, P.; Vallecillo, A. WebML modelling in UML. *IET Softw.* **2007**, *1*, 67–80. [[CrossRef](#)]
40. Brambilla, M.; Fraternali, P. Large-scale Model-Driven Engineering of web user interaction: The WebML and WebRatio experience. *Sci. Comput. Program.* **2014**, *89*, 71–87. [[CrossRef](#)]
41. Rossi, G.; Schwabe, D. Object-Oriented Design Structures in Web Application Models. *Ann. Softw. Eng.* **2002**, *13*, 97–110. [[CrossRef](#)]
42. Schmid, H.A. Model Driven Architecture with OOHDM. Proceedings of Workshops in connection with the 4th International Conference on Web Engineering (ICWE 2004), Munich, Germany, 28–30 July 2004.
43. Aular, Y.J.M.; Pereira, R.T. Metodologías para el desarrollo de ambientes de aprendizaje en entornos colaborativos: Una reflexión teórica. *Multiciencias* **2007**, *7*, 63–71.
44. De Troyer, O.; Casteleyn, S. Modeling complex processes for Web applications using WSDM. Proceedings of the Third International Workshop on Web-Oriented Software Technologies, Oviedo, Spain, 14–18 July 2003.
45. Amengual, E.; Mas, A.; Escalona, M.J.; Ramos, I.; Ruiz, M. Correspondencia de procesos de la metodología NDT con el estándar ISO 12207. *REICIS. Rev. Española de Inno. Calidad e Ingeniería del Soft.* **2009**, *5*, 6–19.
46. Baresi, L.; Franca, M.M. W2000 as a MOF Metamodel. Proceedings of the 2002 World Multiconference on Systemics, Cybernetics and Informatics, Orlando, FL, USA, 14–18 July 2002.
47. Paiano, R.; Pandurino, A. From Design to Development: a W2000-Based Framework; Issues and Guidelines. *Id. Gr. Publ.* **2003**, pp. 500–503.
48. Sari, A.; Tosun, A.; Alptekin, G.I. A systematic literature review on crowdsourcing in software engineering. *J. Syst. Soft.* **2019**, *153*, 200–219. [[CrossRef](#)]
49. Braun, V.; Clarke, V. Using thematic analysis in psychology. *Qual. Res. Psychol.* **2006**, *3*, 77–101. [[CrossRef](#)]
50. Paydar, S.; Kahani, M. A semantic web enabled approach to reuse functional requirements models in web engineering. *Autom. Soft. Eng.* **2015**, *22*, 241–288. [[CrossRef](#)]
51. Garousi, V.; Petersen, K.; Ozkan, B. Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review. *Inf. Softw. Technol.* **2016**, *79*, 106–127. [[CrossRef](#)]

