

SNAIL a hybrid model for the management of agile web software development processes

Molina Ríos Jimmy Rolando

*Researcher, Systems Engineering Career,
Technical University of Machala, Ecuador*

Zea Ordóñez Mariuxi Paola

*Researcher, Systems Engineering Career,
Technical University of Machala, Ecuador*

García Zerda Fabricio Gustavo

*Researcher, Systems Engineering Career,
Technical University of Machala, Ecuador*

Contento Segarra María José

*Researcher, Systems Engineering Career,
Technical University of Machala, Ecuador*

Jumbo Parrales Carlos Eduardo

*Researcher, Systems Engineering Career,
Technical University of Machala, Ecuador*

ABSTRACT:

In this article we present the hybrid methodology for the development of agile software called SNAIL (Native Software of Logical Iterative Architecture) focused on the development of web applications, which works in conjunction with Software Engineering for the use of good practices within its life cycle. It is important to mention that the software development methodologies originated as an alternative of support and guidance in the process of construction of any software product, due to the complexity involved in carrying it out and in response to the problems related to its development. SNAIL arises as a result of the analysis, mapping and combination of existing methodologies and frameworks, by which a thorough investigation was conducted to obtain information on these methodologies and to determine disadvantages that can be met, in

addition to making the most of the benefits and characteristics which are inherent in each of these. This methodology is based on simplicity in the process of product construction, communication between the team of developers and code planning, its spiral model allows not only the reduction of costs due to unnecessary changes, but also the mitigation of the risk of effective way because tests are constantly carried out to verify errors and check the functionality of each part of the system.

Keywords: Agile Methodology, Hybrid Methodology, SNAIL, Software Engineering, Web applications.

I. INTRODUCTION

At present, there are different methodologies for software development, among these the traditional and agile methodologies. The traditional focus primarily on the control of the development and importance of the documentation of the system so in different large projects have justified being efficient to understand and support the implemented systems, however, demand a high level of knowledge and a long response time to adapt to changes in the course of the project. On the other hand, the agile methodologies solve the disadvantages presented by the traditional ones, being the agile more flexible with greater adaptability to the changes of the project, however, they avoid a more complete documentation and formal methods. [1][2][3]

Based on what has been said and the progress made in the industry in the development of web applications, conventional methods become complex, which is why hybrid methodologies are emerging, a new trend in the area of software engineering with the purpose of rescuing the benefits and advantages that comprise the two traditional and agile groups. [4]

Considering this, a hybrid methodology called SNAIL (Native Software of Logical Iterative Architecture) was developed; for its construction as a starting point, the feasibility of using this type of methodologies in the current context in the software development companies in Ecuador was investigated. Once this was done, it was verified through bibliographic research the use of this type of methodologies and their growing trend in the area of software engineering.

SNAIL was designed based on the combination of best practices within the RUP methodologies (Rational Unified Process, Unified Rational Process), XP (Extreme Programming, Extreme Programming), SCRUM and OOHDM (Object Oriented Hypermedia Design Model, Development Method for Hypermedia Directed by Models). In addition, it also covers the most important phases and that are crucial for a good software development.

The methodology SNAIL (Native Software of Logical Iterative Architecture) is a hybrid methodology of agile development of web applications that is based on simplicity, which generates the construction of clearer solutions, in shorter cycles and focused on what is really required by the client; communication between the work team, fostering a pleasant atmosphere and increasing productivity among the

developers; and finally, the planning of the code. An important feature of this methodology is its name that comes from the shape of the model of its phases, a spiral model; This model focuses its efforts on iterative development, where its main steps are focused on the determination of the objectives to be achieved with the web application, risk analysis, development, verification, validation and planning of the software project stages. [5]

SNAIL has the following objectives: customer satisfaction through compliance with the requirements, the empowerment of work, and the effective reduction of risk on four important variables within a project: cost, quality, time and scope.

This hybrid methodology based on the trial and error method for the creation of web applications allows the reduction of costs due to the constant evaluations performed on the deliverables at the end of each cycle. With this method SNAIL adds value to its products and also increases its quality since each deliverable will be socialized with the client who is in charge of determining the degree of acceptance with respect to it.

It is also important to highlight that in SNAIL the client is the one who establishes the price of the project to be carried out based on the ability of the work team to establish the functionality of the web application that it can deliver over time.

With the implementation of this type of hybrid methodology within the life cycle of web applications in development projects, it has been possible to improve not only the construction process of this type of applications, but also the level of risk has been reduced, since This methodology offers a guide composed of effective phases and processes that allow each developed web application to be functional and align with the initial project expectations.

However, within the framework of development of this research is to explain in a clear and detailed each of the phases that make up this methodology such as: requirements, planning, design, programming, testing, closure and finally inbound marketing (optional in the methodology) in which a strategy for attracting customers is made.

The objective of this research is to analyze the hybrid methodologies of software development, through the application of data collection techniques and evaluation criteria, identifying which is the most appropriate in the software development process in the companies of Machala city.

II. LITERATURE REVIEW

II.I WEB APPLICATIONS

The communication process over the years has been changed due to various applications that are now presented and undoubtedly also to the use of the internet that has allowed these variants.

The internet has been the main engine on which these changes in communication revolve, which is why the use of the internet today is essential for everything. Another factor that is aimed at the use of the internet is the development of web applications.

"Web applications are nothing more than Web 2.0 office tools that are simply managed with an internet connection, and in these cases, it is possible to use the computer only as a form of remote application processes" [6]

A web application is also known as a client / server application that is composed of a set of tools oriented to the user so that he can access a server through a web browser that will be connected to the Internet or an intranet to do use of a process or service. [7]

In web applications, three levels are generally distinguished: the upper level that interacts with the user, the intermediate level who processes the data that has been received and the lower level that is responsible for providing the data.

According [8] [9] web applications are currently very important and popular due to the independence of the operating system that the user has installed, as part of this success is also based on the concept of interactivity that this type of applications have with the user.

Among other advantages that can be mentioned in web applications are: it is not necessary to perform updates by the client, centralization of information, it does not require a specific operating system, and you can work where required as long as you have a computer and connection to the network. On the other hand, among the disadvantages that can be mentioned for this type of applications are the following: network connection, complex development, response time directly linked to the provider's time that the client owns. [10].

The font size for heading is 12 bold face and subsections with 12 points and not bold. Do not underline any of the headings, or add dashes, colons, etc.

II.II METHODOLOGY FOR THE DEVELOPMENT OF SOFTWARE

A general methodology is a framework that can be used as a guide for the activities you wish to carry out. Therefore, a methodology for the development of software is a form of work that must be followed for the construction of all software products, where the tasks or phases that must be fulfilled are specified the deliverables that are going to be generated and the relationships between the different parts of the system. [11]

On the other hand, it defines methodologies as a process or set of procedures, techniques and documentation that allows guiding developers and contributing to the creation of quality applications that meet the needs of customers. [12] [13]

This framework allows the software development process to be executed in an organized manner through the determination and definition of guidelines to follow and restrictions that must be met. Currently there is a wide variety of methodologies that are framed within three major groups are: traditional, agile and hybrid, and are also intended for various types of applications are desktop, web or mobile which work with the only purpose is to guide, plan and control the entire software development process. [14]

II.III TRADITIONAL METHODOLOGIES

The traditional methodologies or also known as linear cycles arise in the 60s with the need to improve the software construction processes and bring the projects to the desired goal. This type of methodologies is based on a planned, documented work process and is mainly adapted to projects where the requirements are clearly defined. [15] [3]

In these sequential methodologies, it clearly describes a series of steps or phases that must be taken when creating an information system. These phases generally follow a specific order with a revision at the end of each phase. When the review is complete and meets the necessary requirements, the next stage can be started. [6]

The key advantage offered by this cascade model is that it allows an easy transition from distributed to distributed teams, as it provides a clear structure to organize and control activities throughout the software development cycle. Each phase, beginning with the definition of requirements until the implementation of the deliverable for the client, has clear inputs and outputs that are documented and considered final when a phase is performed. On the other hand, the inherent disadvantages in this type of methodologies are the cost of implementation, the extensive documentation, little communication with the client, and lack of flexibility due to the rigid processes that are handled. [16]

II.IV AGILE METHODOLOGIES

Due to the drawbacks of traditional methodologies identified as a rigid and excessive method in unnecessary documentation, agile methodologies emerge after a meeting in February 2001 in the US, in order to establish a methodology that allows developers to create software in a fast way and that responds to the changes that appear in the course of the project. [17] [18]

Agile methodologies use an iterative and team-based approach. Its main objective is to quickly deliver the application with complete and functional components. Instead of completing software development tasks in sequence, they are completed in sprints that run from about 1 to 4 weeks and where a list of deliveries is completed in each sprint. Tasks that are not completed within the sprint are reprioritized and included in future sprints. This also means that the different stages of the software development life cycle can be revised as necessary. [19] [20]

The main focus when using agile methodologies in large and distributed projects remains the same, that is, achieving customer satisfaction by providing them with valuable software and this can only be achieved by following the core values and guiding principles of the methodologies. The additional challenge of addressing the increase in complexity as a result of distribution and scaling requires careful evaluation of organizational factors and agile practices to achieve flexibility and responsiveness. [21] [22]

However, among the disadvantages that arise in this type of methodologies are little

documentation, heavy dependence on project leaders and lack of planning, delivery deadlines and project budgets. [23]

II.V HYBRID METHODOLOGIES

Hybrid or mixed so-called methodologies arise as a combination of practices and artifacts that come from different methodologies, be they traditional or agile. The hybrid methodologies base their existence on the weaknesses or deficiencies that exist in the methodologies with the purpose of creating a method that is more robust, but at the same time is flexible and that also combines the advantages that it also provides. [24] [25]

The methodologies aim to create a complete fusion that fits all types of projects. It is important to mention that the new trend that exists in software engineering is to design hybrid methodologies.

It is also important to note that this proposal is attributed to Ivar Jacobson, one of the three creators of UML (Unified Modeling Language, Unified Modeling Language, Object Management group, 2011); creator of UP (Unified Process, Unified Process), and now creator of Essup (Essential Unified Process). EssUP is a hybrid methodology that combines RUP with Scrum. [26]

II.VI METHODOLOGIES FOR WEB APPLICATIONS

According [27], software engineering is the application of a disciplined and quantifiable approach to the development, operation and maintenance of computer applications with the aim of satisfying the customer's needs with a delivery in a certain time and an efficient product. Pressman raises engineering in four layers which are: commitment to quality, process, methods and tools where it can be understood that software engineering in addition to being disciplined must be adaptable and agile.

On the other hand [28], establishes that the rapid growth of the internet and in turn the Web application has led to the development of this type of low-quality applications due to multiple problems such as the short construction period, inadequate tools, etc. Therefore, the increase in the construction of Web applications brings with it a more complex process of development. From this emerge new systematic approaches, disciplines and methodologies aimed at Web environments.

However [7], It presents specific aspects for the Web engineering environment, among which stand out: the design of business processes, case tools, code generation, collaborative web development, conceptual model of applications, integrated development environments, performance testing, personalization and application adaptation, prototyping methods, quality control and system testing, requirements engineering, applications for the semantic web, usability of applications, mobile and oblique web applications, web design methodologies, interface design, metrics for the web , risk management and projects and finally development and deployment of web

services.

As can be seen within the different aspects that are considered in Web engineering are the Web design methodologies therefore a study of some methodologies for the development of Web applications are presented below.

Particularly the methodologies according [20] to contain phases of development specifically for this type of applications which are: conceptual design, navigational design, design of the presentation or interface design and implementation

III. METHODOLOGY

The research to be carried out is of a bibliographic, analytical and field type, which will allow obtaining relevant information regarding methodologies for the development of web applications.

To carry out the following work, different stages have been named, which will focus on the explanation of each of the stages of the SNAIL hybrid methodology, which are:

1. Bibliographic research of works that correspond with similar studies developed in the same field (methodologies in Web applications) and with companies and programmers of the province of Gold that are within this scope.
2. Determination and processing of collected data of the different norms, standards, methods and sources found.
3. Identification of the stages or phases that make up the hybrid methodologies.
4. Description of each of the phases of the methodology.
5. Analysis of the data obtained and verification of them.
6. Elaboration of the results.

IV. RESULTS

Next, the hybrid methodology of development of web application development is presented:

SNAIL "is a hybrid methodology for the development of web applications, which is based on the simplicity, communication and planning of the developed code, its name comes from the shape of the model of its phases, since, being a spiral model, takes a shape similar to that of a snail".

Goals

SNAIL aims to fulfill three very important aspects such as: customer satisfaction, the empowerment of work and finally reduce the risk effectively on the variables of the project: cost, time, quality and scope.

Characteristics

This methodology presents six characteristics to consider for the development of web applications:

- Hybrid methodology based on tests and error for the creation of web-oriented, functional systems.
- Based on principles.
- Reduces the cost of change in all stages of the system's life cycle.
- It mixes the best practices that have been obtained to develop software and by means of its cycle in sphere way it tries to take them to its best development.
- Well defined client.
- The requirements may change.

Model

Like any methodology, SNAIL defines its model in four variables for the development of any web software project: cost, time, quality and scope; of which three of them are set arbitrarily by external actors to the developer organization, while the rest is defined by internal actors of the development team.

For this reason, SNAIL performs short development cycles or iterations, with functional deliverables at the end of each cycle. In each iteration a complete cycle of requirements, planning, design, programming, testing and closing is carried out, being an optional plus an Inbound Marketing phase, for its diffusion and publicity, but using a set of practical rules.

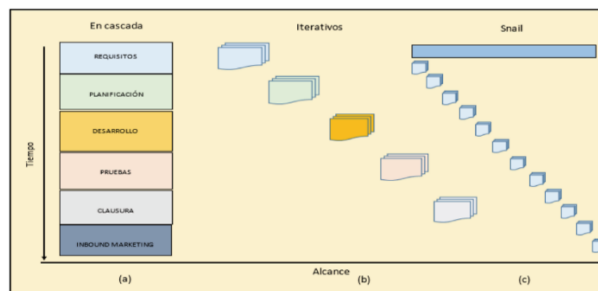


Fig 1. a) Evolution of the long development cycles. b) Shorter iterative cycles. c) Mix made by SNAIL

Phases of SNAIL

SNAIL establishes the following stages or phases described in the following table, where its respective activities that must be carried out during the software development process are defined.

Table 1. Phases and activities of the SNAIL methodology

Phases	Activities
Requirements	1. Business environment
	2. Feasibility study
	3. Identify actors or users
	4. Identify objectives or requirements
	5. Identify functional and non-functional requirements
	6. Classify functional requirements around the functionality and maintainability
	7. Validate requirements
Planning	1. Selection of functional requirements regarding the functionality and maintainability
	2. Define deliverables
	3. Estimation of costs
	4. Project speed
	5. Benchmarking scenarios
Design	1. Database design
	2. Data dictionary
	3. Conceptual design
	4. Navigational design
	5. Abstract interface design
Programming	1. Coding
	2. Standards
	3. Unit test
	4. Interconnection
	5. Integration
Test	1. Benchmarking analysis
	2. Acceptance tests
Closing	1. Present deliverables
	2. Iteral evaluation of the project
Inbound marketing	1. Advertising
	2. Branding

The SNAIL development cycle in broad strokes consists of the following steps:

1. The client establishes the business price to be implemented.
2. The programmer measures the effort necessary for its implementation.
3. The client starts asking what he wants to elaborate, according to his priorities and time constraints.
4. The programmer builds that business value.
5. Go back to step 1.

Although the life cycle of a SNAIL project is dynamic, it can be represented by the following model, identifying its phases in a simple way.

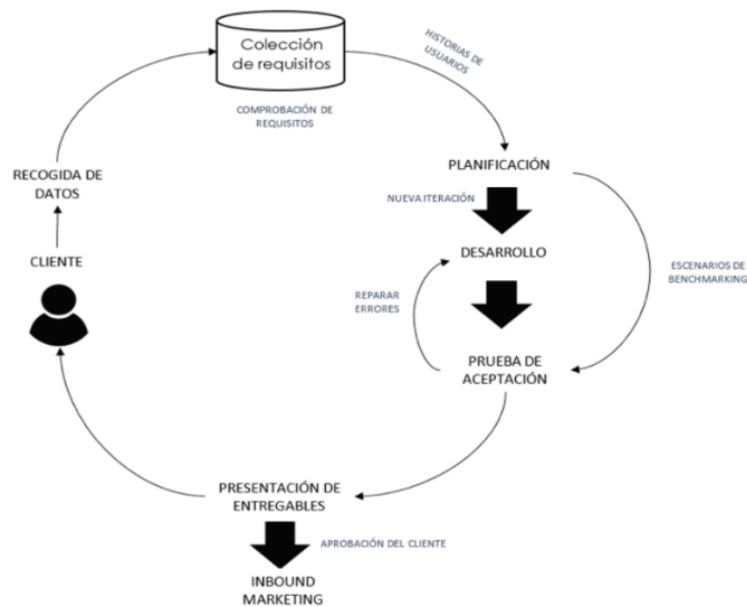


Fig 2. Phases of the SNAIL model

- Phase I: Requirements

All the requests that must be carried out during the realization of the software project are defined, the needs of the final users of the system are also analyzed to determine the objectives to be covered. It is important to point out that at this stage everything that the system needs and requires must be detailed, since these specifications will be what will continue in the following stages, without allowing to require new results in the middle of the system development process, in no way.

Business Modeling

The main objective of business modeling is to determine and specify the processes of the organization and / or business, obtaining information, activities, roles and business rules involved. Within this activity, factors such as:

- Identification of business processes and maintenance processes.

- Identify the users, departments or elements of the organization involved in the business process.
- Textual description of the business process.
- Construct an activity diagram that represents the business process.
- Specify the activities that appear in the activity diagram.
- Specify the information that flows in the activity diagram.
- Business rules.

Feasibility study

When developing a software project, the use of a feasibility study is required, where the objectives, scope and restrictions of the project are clearly established, in addition to a high-level logic model. In this sub-activity they are found:

- Technical Feasibility: Is there the necessary technology? Is it within reach?
- Economic Feasibility: Cost / Benefit Ratio. Is the cost paid?
- Operational Feasibility: What are the organizational capacities to sustain the system?

SNAIL proposes the following model or explanatory scheme for the feasibility study:

- To establish objectives
- What is the Feasibility Study for?
- Who participates in the stage?
- Activities of the stage
- Content of the feasibility study.
- Description of the main components of the Feasibility Study.

Identify actors or users and requirements

Within this sub-activity, a basic control is exercised for the administration of the requirements, in order to maintain and meet the needs in a timely manner. For the administration of requirements, the following phases are acquired:

- Collection
- Analysis
- Specification
- Check
- Tools, techniques and software

- Phase II: Planning

A framework is provided that allows the manager to make reasonable estimates of resources, deliverables, costs and time planning. These estimates are made within a limited time frame when starting a software project, and will have to be constantly

updated as the project progresses.

This phase is based mainly on the execution of one or more group planning meetings. The result of this phase is a Delivery Plan, it is also composed of the following activities:

- Selection of requirements or user stories.
- Define deliverables.
- Cost estimation.
- Speed of the project.
- Benchmarking scenarios.

- Phase III: Design

The system is restructured and organized into elements that can be developed individually, taking advantage of team development. It is convenient to distinguish between high-level or architectural design and detailed design.

For the design in the SNAIL methodology, only those user stories that the client has chosen for the proposed iteration are considered, this for two reasons:

1. It is considered that from the beginning it is unlikely to have a complete design of the system and that it is free of errors.
2. Considering the changing nature of the project, making a very complete or extensive design in the initial phases of the project to subsequently correct or change it, is considered a waste of time.

In the design phase we find the following activities:

Database design

The process of database design is composed of subprocesses that start from the collection of database requirements. This lies in the development of three database designs which are: conceptual, logical and physical the same that are made through the use of specific techniques and methods.

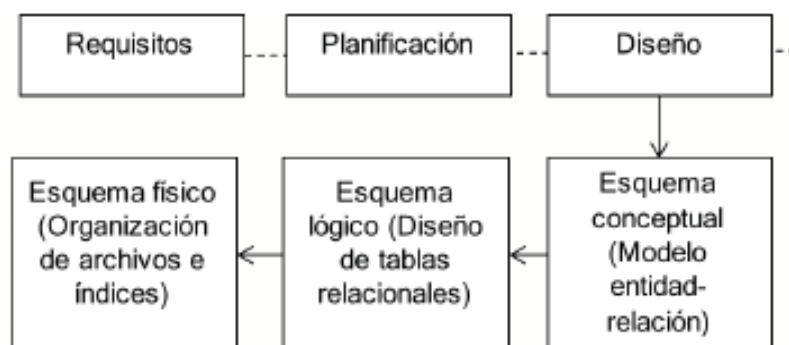


Fig 3. Database development process

Data Dictionary

It is defined as a structured list where all data that are part of the system are described, with a set of clear and rigorous definitions, so that the analyst and the user understand the inputs, outputs, storage elements and intermediate calculations.

Conceptual design

This phase is taken from the OOADM methodology, [29] where a conceptual scheme is drawn up or designed, represented by the objects of the domain, the relationships and existing collaborations established between them. In the SNAIL methodology, the conceptual scheme is constructed by classes, relationships and subsystems. Classes are described as in traditional object-oriented models. Even so, the attributes have the advantage of being able to be of multiple types to represent different perspectives of the same entities of the real world.

Navigational design

A navigational model is constructed as a view on a conceptual design, allowing the construction of different models according to the different user profiles. Each navigational model provides a subjective view of the conceptual design.

Abstract interface design

Once the navigational model is defined, the interface aspects must be specified. This means that the way in which navigational objects can appear must be defined, how will interface objects activate navigation and the rest of the application's functionality? What interface transformations are pertinent? and When is it necessary to do them?

- Phase IV: Programming

The source code is transcribed, making use of prototypes, as well as tests and trials to avoid having errors and having them, to be able to correct them in time. Regardless of the programming language that is handled and its version; libraries and components are created that will be reused within the same project to make the programming process much faster.

Within this phase are the following activities:

- Coding.
- Standards.
- Unit test.
- Interconnection.
- Integration.

- Phase V: Tests

The parts that have already been programmed come together to structure the system, in addition it is checked that it works correctly and that it meets the requirements before being delivered to the client. The tests can be executed at any point of the system development process; it is here where the test levels allow to understand with greater vision the different points or stages where certain types of test can be executed.

Next, the classification of the types of tests of a software project is mentioned:

- Unit or component tests.
- Integration testing.
- System tests.
- Detection and correction of errors.
- Test methodology.

- Phase VI: Closing

It is the last of the phases, if the Inbound Marketing (optional) is considered. This phase comprises the management process of the same, and applies both to the project in general and to each of the phases of its life cycle. It is usually the shortest phase of a project, even so, it should not be considered the least important compared to the others. Because in this phase the formal closure of the project is established, it reviews the successes and failures with a view to improving the next one.

Closing a project involves a set of processes, such as startup, planning, execution or control. According to the PMBOK [30] the closure of projects includes two processes: 1) Close the project or the project phase, 2) Close the acquisitions.

Submit deliverables

Establish the deliverables as they were related in the project and quality plans and their final status (completed, abandoned, postponed, closed).

Comprehensive evaluation of the project

Iteration is a set of time-adjusted tasks that focus narrowly on producing an executable or presentable. The Evaluation iteration involves a certain amount of revision (of existing work products), as well as a change in the revision.

In short, it is necessary for a certain number of revisions to finalize a product which will be delivered and do it with quality, this is achieved through the creation of intermediate products and the evaluation of the suitability of the product architecture. Initially and regularly, the quality of the final product increases.

- Phase VII: Inbound Marketing

In this phase (optional) a strategy is made, based on attracting customers with useful, relevant content and adding value in each of the stages of the buyer's journey. With

the Inbound marketing phase, potential customers find your system through different channels such as blogs, search engines and social networks.

V. DISCUSSION

At the time of developing a software many times the question of the development methodology to be used is born, there are a number of traditional and agile methodologies; of which many of the times do not meet the desired expectations by the developer team, however, due to the need to encompass the virtues and positive aspects of many methodologies, it has become a tendency to build hybrid methodologies that meet these needs.

For this reason, came the creation of SNAIL (Native Software of Logical Iterative Architecture), a hybrid methodology for the management of processes of agile development of Web software, taking good practices of methodologies and frameworks as RUP (Rational Unified Process), XP (Extreme Programming), SCRUM and OOADM (Object Oriented Hypermedia Design Model).

VI. CONCLUSION

Through the information presented, it is concluded that:

- Development methodologies play a very important role in the construction of a software product, being a guide for developers in order to achieve quality features such as functionality, security, usability, among others.
- The birth of the SNAIL hybrid methodology provides developers with an alternative to achieve a high level of quality in their web software products, since it supplements the disadvantages of others by taking advantage of their best practices, obtaining great benefits during the software's life cycle.
- Due to the high demand for quality web applications today, companies and independent developers seek to build a product that meets the needs of the client, which is why development methodologies have gained strength in recent years, SNAIL being an excellent agile development alternative

REFERENCES

- [1] J. Rolando, M. Ríos, M. Paola, and Z. Ordóñez, "Comparación de metodologías en aplicaciones web," *3c Tecnol.*, vol. 7, no. 1, pp. 1–19, 2018.
- [2] D. Benslimane, S. Dustdar, and A. Sheth, "Services mashups: The new generation of web applications," *IEEE Internet Comput.*, vol. 12, no. 5, pp. 13–15, 2008.
- [3] R. Lujan, "Modelos de desarrollo de software y su relación con la Innovación. En el sector SSI de Entre Ríos," vol. 5, no. 2, pp. 52–53.

- [4] J. R. Molina Ríos, M. P. Zea Ordóñez, M. J. Contenido Segarra, and F. G. García Zerda, “Metodologías de desarrollo en aplicaciones web,” *3c Tecnol. ISSN-e 2254-4143*, Vol. 6, No. 3, 2017, págs. 54-71, vol. 6, no. 3, pp. 54–71, 2017.
- [5] J. R. Molina Ríos, M. P. Zea Ordóñez, N. M. Loja Mora, F. F. Redrován Castillo, M. R. Valarezo Pardo, and J. A. Honores Tapia, *SNAIL, UNA METODOLOGÍA HÍBRIDA PARA EL DESARROLLO DE APLICACIONES WEB*, Primera. Machala: 3ciencias, 2018.
- [6] J. Molina, M. Zea, J. Honores, and A. Gómez, “Analysis Methodologies Web Application Development,” *India Publ.*, vol. 11, no. 16, pp. 9070–9078, 2016.
- [7] N. Faci, Z. Maamar, V. Burégio, E. Ugljanin, and D. Benslimane, “Web 2.0 applications in the workplace: How to ensure their proper use?,” *Comput. Ind.*, vol. 88, pp. 1–11, 2017.
- [8] J. Molina, N. Loja, M. Zea, and E. Loaiza, “Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python,” *Rev. Latinoam. Ing. Softw.*, vol. 4, no. 4, pp. 201–207, 2016.
- [9] A. L. Cardador Cabello, *Implantación de aplicaciones web en entornos internet, intranet y extranet*, Primera. Málaga: IC Editorial, 2015.
- [10] J. Faircloth, *Web applications and services*. 2017.
- [11] M. Manimaran, A. Shanmugam, P. Parimalam, N. Murali, and S. A. V. Satya Murty, “Software development methodology for computer based I&C systems of prototype fast breeder reactor,” *Nucl. Eng. Des.*, vol. 292, pp. 46–56, 2015.
- [12] K. Sousa, J. Vanderdonckt, B. Henderson-Sellers, and C. Gonzalez-Perez, “Evaluating a graphical notation for modelling software development methodologies,” *J. Vis. Lang. Comput.*, vol. 23, no. 4, pp. 195–212, 2012.
- [13] L. A. M. Zaina and A. Álvaro, *A design methodology for user-centered innovation in the software development area*, vol. 110. Elsevier Ltd., 2015.
- [14] G. Pantaleo and R. Ludmilla, *Ingeniería de Software*. Buenos Aires: Alfaomega, 2015.
- [15] A. Navarro Cadavid, J. D. Fernández Martínez, and J. Morales Vélez, “Revisión de metodologías ágiles para el desarrollo de software,” *Prospectiva*, vol. 11, no. 2, pp. 30–39, 2013.
- [16] G. Papadopoulos, “Moving from Traditional to Agile Software Development Methodologies Also on Large, Distributed Projects.,” *Procedia - Soc. Behav. Sci.*, vol. 175, pp. 455–463, 2015.
- [17] P. Rodríguez, M. Mäntylä, M. Oivo, L. E. Lwakatare, P. Seppänen, and P. Kuvaja, “Advances in Using Agile and Lean Processes for Software

- Development,” *Adv. Comput.*, 2018.
- [18] M. Younas, D. N. A. Jawawi, I. Ghani, T. Fries, and R. Kazmi, “Agile development in the cloud computing environment: A systematic review,” *Inf. Softw. Technol.*, no. June, pp. 0–1, 2018.
- [19] P. Letelier, M. Canós, E. Sánchez, and M. Penadés, “Metodologías Ágiles en el Desarrollo de Software,” *Val. Val. España*, pp. 1–8, 2003.
- [20] R. N. A. Del Valle and R. Bertone, “Metodologías De Diseño Usadas En Ingeniería Web, Su Vinculación Con Las Ntics,” p. 58, 2009.
- [21] J. Britto, “Comparación de metodologías ágiles y procesos de desarrollo de software mediante un instrumento basado en CMMI,” *Sci. Tech.*, vol. 21, no. 2, pp. 150–155, 2016.
- [22] Ö. Özcan-Top and O. Demirors, “Application of a software agility assessment model – AgilityMod in the field,” *Comput. Stand. Interfaces*, 2018.
- [23] B. M. Montero, H. V. Cevallos, and J. D. Cuesta, “Metodologíaa ágiles frente a las tradicionales en el proceso de desarrollo de software,” *Espirales Rev. Multidiscip. Investig.*, vol. 2, no. 17, 2018.
- [24] M. Jiménez, S. Hernández, and D. Orantes, “Metodologías híbridas para desarrollo de Software: una opción factible para México,” *Rev. Digit. Univ.*, vol. 1, no. 1067–60710, pp. 1–17, 2012.
- [25] R. D. A. Araújo, S. Soares, and A. L. I. Oliveira, “Hybrid morphological methodology for software development cost estimation,” *Expert Syst. Appl.*, vol. 39, no. 6, pp. 6129–6139, 2012.
- [26] I. Leiva Mundaca and M. Villalobos Abarca, “Método ágil híbrido para desarrollar software en dispositivos móviles,” *Ingeniare. Rev. Chil. Ing.*, vol. 23, no. 3, pp. 473–488, 2015.
- [27] P. J. V Roldán Martínez, David Valderas Aranda, *Aplicaciones Web. Un enfoque práctico*. RA-MA S.A, 2010.
- [28] A. Ramos Martín and M. J. Ramos Martín, *Aplicaciones Web, Ediciones. Ediciones Paraninfo, S.A.*, 2014.
- [29] B. Mercerat and D. A. Silva, “Construyendo aplicaciones web con una metodología de diseño orientada a objetos,” *Rev. Colomb. Comput.*, vol. 2, no. 2, pp. 1–21, 2001.
- [30] “Gestión de las Adquisiciones del Proyecto - La guía PMBOK.”

