ISSN: 2594 - 2905

INNOVATION IN SOFTWARE DEVELOPMENT TO OPTIMIZE AGREEMENT TRACKING

INNOVATION IN SOFTWARE DEVELOPMENT TO OPTIMIZE AGREEMENT TRACKING

Xochitototl Cote Itzel[1], Hernández Mora José Juan[2], Medina Barrera María Guadalupe[3], Cuatecontzi Cuahutle Elizabeth[4], Ramos Ramos Juan[5]

https://doi.org/10.61117/ipsumtec.v8i1.351

[1]Engineer in Business Management. Tecnológico Nacional de México/IT Apizaco. Graduate Studies and Research Division. Master's Degree in Computer Systems. m23370012@apizaco.tecnm.mx. Tel: 241 4172010 Ext. 146. Av C.P. 90300. https://orcid.org/0009-0008-1409-549X

[2]Doctor in Teaching Excellence. Tecnológico Nacional de México/ IT of Apizaco. Graduate Studies and Research Division. Master's Degree in Computer Systems. juan.hm@apizaco.tecnm.mx. Tel: 241 4172010 Ext. 146. Av C.P. 90300. https://orcid.org/0000-0003-2878-7290

[3]PhD in Strategic Planning and Technological Management. Tecnológico Nacional de México/ IT de Apizaco. Graduate Studies and Research Division. Master's Degree in Computer Systems. guadalupe.mb@apizaco.tecnm.mx. Tel: 241 4172010 Ext. 146. Av C.P. 90300. https://orcid.org/0000-0003-3074-0029

[4]Master in Software Engineering Management. Tecnológico Nacional de México/ IT de Apizaco. Graduate Studies and Research Division. Master's Degree in Computer Systems. elizabeth.cc@apizaco.tecnm.mx. Tel: 241 4172010 Ext. 146. Av C.P. 90300. https://orcid.org/0009-0008-0531-0354

[5]PhD in Computer Systems. Tecnológico Nacional de México/ IT de Apizaco. Graduate Studies and Research Division. Master's Degree in Computer Systems. juan.rr@apizaco.tecnm.mx. Tel: 241 4172010 Ext. 146. Av C.P. 90300. https://orcid.org/0009-0004-7440-7257

Abstract -- In software development, quality assurance and effective findings management are crucial to the success of any project. In addition to documenting the identified findings, it is essential to carry out a detailed follow-up to ensure that the necessary corrections have been implemented [1]. Therefore, in this paper, an innovative format for tracking agreements in software development is presented, which is designed to facilitate the review, validation and verification of activities.

Therefore, a format was designed for the management of agreements in software development projects, covering different stages of the life cycle of the process. Subsequently, its implementation was analyzed by means of ring charts, for the follow-up of the activities. The results of the case studies show that this implementation considerably improves the organization of the development process.

In summary, this paper shows that the proposed format is an effective tool to avoid unfinished software project cycle activities, thus improving the quality and success of software development projects.

Keywords: Communication, Software development, Software management, Validation, Verification

Abstract -- In the field of software development, quality assurance and the effective management of findings are crucial aspects for the success of any project. In addition to documenting the identified findings, it is essential to conduct detailed follow-up to ensure that necessary corrections have been implemented [1]. Therefore, this paper presents an innovative format for tracking agreements in software development, designed to facilitate the review, validation, and verification of activities.

Thus, a format is designed for managing agreements in software development projects, covering different stages of the process life cycle. Its implementation was then analyzed using ring graphs to track activities. The results from the case studies demonstrate that this implementation significantly improves the organization of the development process.

In summary, this paper shows that the proposed format is an effective tool for ensuring that activities in a software project cycle do not remain incomplete, thereby improving both the quality and success of software development projects.

Key words - Communication, Software development, Software management, Validation, Verification.

INTRODUCTION

Studies by Quitian Monroy JS, et al [2] show that, in software development with agile methodologies, communication is the most important practice. However, autonomous development teams operate at 50% of their capacity. In contrast, in traditional methodologies communication takes place in 76% of the cases and autonomous development teams work at 32% of their capacity, this indicates that the workflow is not constant and it is necessary to provide an understanding of the role and the specific activities to be developed by each participant in the process, which allows the optimization and automation of the process [3], as can be seen in Table 1.

Table 1. Weighting table

Ágiles		Tradicionales	
Comunicación	100%	Integración de tecnología	80%
Grupo multifuncional	95%	Integración del cliente	80%
Versatilidad	85%	Comunicación	76%
Integración de tecnología	80%	Innovación	56%
Integración del cliente	75%	Grupo multifuncional	56%
Simplicidad	60%	Mejora continua	52%
Innovación	55%	Versatilidad	48%
Grupo de desarrollo autónomo	55%	Simplicidad	44%
Mejora continua	55%	Grupo de desarrollo autónomo	32%

Source. Adapted from Quitian Monroy JS, Martínez Padua JV, Castiblanco Jiménez IA [2].

The proposed format is to innovate and record the way in which the software development team communicates, likewise, it documents the agreements that are taken in meetings related to a specific software project, with the objective that the development team reinforces its learning in relation to the projects, tasks and activities assigned as part of their work, it promotes sharing the mistakes made during the development of the processes so that everyone involved can learn from them [1].

In addition, a monitored workflow enables employees to make timely decisions, improve team communication, and drive agile work in software development processes [3]. All this is carried out under a rigorous management of activity documentation, monitoring reports, changes, operational and security risks, as well as the impacts of these risks, which enables proper monitoring and ensures the continuous availability of services and information assets [4], [5].

Problem

Given the need to document and monitor activities during the software development process, obstacles in team coordination have been identified [6]. Agile methodology, such as Kanban, offers solutions in software manufacturing, but it depends fundamentally on a well-coordinated, communicated, responsible and cohesive team [7]. Therefore,continuous monitoring of its implementation is essential.

Likewise, the Scrum methodology rigorously manages the processes to generate value to the customer, therefore, it is important to identify the tasks, to advance them if possible, but this practice weakens the quality indicators of the task performed, due to the domain of the limitation of the work in progress, in addition, bottlenecks may occur during the development stages. For example, Lean [8] proposes to perform only delivery of recommendations, which are non-specific instructions. This results in subjectivity to the interpretation of each work team.

It is therefore essential to evaluate the monitoring of projects, adapt planning, organize the work efficiently and maintain effective communication within the team to achieve the best results and maximize performance [9].

In addition, a study conducted by Torres in 2024 [10] shows that, among the most outstanding good practices in software development companies, only 2.56% verify progress with *stakeholders*.

Meanwhile, Merchán-Narváez N J, et al [11] reveals that agile methodologies face problems during implementation, such as lack of effective communication, which negatively impacts their effectiveness. Processes are highly dependent on the project manager and without proper monitoring, activities may fail. On the other hand, self-regulation can lead to undisciplined teams if they are not committed to their tasks and these are not constantly checked. In addition, customer participation in development reviews can become a challenge if the team is not fully committed.

Likewise, the lack of adequate version control, insufficient technical test management, and poor quality assurance in software development projects show an urgent need for documentation, verification, and continuous monitoring of the activities performed during the process [4].

Scope and limitations of the format

On the other hand, the assignment and identification of those responsible for the processes contributes to a better implementation of the overall workflow [12]. This assignment facilitates the automation and optimization of the activities performed and verified, ensuring that the collaborators fulfill their tasks in a timely manner.

In other words, a constant and automated workflow should, in theory, not hinder software development processes. However, there can sometimes be problems if the responsible parties do not comply with the established agreements, which can affect the quality of the product.

In addition, the lack of defined processes in companies can make overall product verification and monitoring difficult [13] and [14].

Benefits of the format

-Improve the communication of the development team.
-Clarity in the objectives of each activity.
-Engaging employees in their activities.
-Integrate evidence of the review and verification of activities.
-Avoid misunderstandings between the interested parties.

Disadvantages of the implementation of the format

During the implementation of this format, we found inconveniences related to:

-The learning curve and adaptation to the new format affected productivity in the short term as in case one shown below.

-Resistance to change, where employees were not comfortable with the constant revision, making the format ineffective, as happened in case two below.

In addition, a feeling of administrative overload was created, which was the result of documentation and follow-up where it was not done before, as was the case in case three below.

General

Considering the contributions of Helen Tupper and Sarah Ellis [3], Gramuntell Bayarri V [5], IBM [9], Torres Calle [10] and Fernando Villafranca A [15], the format has been designed with the following principles:

Functionality: task management, bug tracking, reporting and analysis, checklist, feedback tracking, real-time collaboration.

-Determine a process to correlate: it is the way in which the format will be related in the workflow of the project in question. The identifiers must be unique, to avoid confusion.

-Information that identifies the project, the team and the event in question: so that all stakeholders are on the same frequency and there is no room for ambiguity.

-Workflow steps: key steps, inputs, outputs, results, metrics, personnel involved, their work areas, as well as their responsibilities within the project to be developed and verification process should be identified.

-*Feedback* and verification: the validation of stakeholder comments is analyzed to identify recurring errors and bottlenecks during the production phase in order to improve processes.

-Feedback adjustments: in order to regulate and automate the operation of the workflow within the software production team, thus achieving individual and group progress of the activities to be performed.

DEVELOPMENT

This work was carried out in an international software development company, whose name cannot be mentioned for confidentiality reasons, as well as in the Computer Center of the Master's Degree in Computer Systems of the Tecnológico Nacional de México/Apizaco Campus.

Format

Taking into account the above particularities and the contributions made by Helmut Sy Corvo [17] and Peraza-Delgado M [18], who propose the use of formats for monitoring agreements and tests, respectively. [17] and Peraza-Delgado M [18], who propose the use of formats for the follow-up of agreements and tests respectively; in the format proposal made in this project, a section for validation of the information is added, which will allow the correct follow-up of the activities so that they do not remain unfinished.

The proposed format is shown in Table 2 and consists of four sections: the first provides identification data, the second details the objectives, the third lists the agreements and the fourth focuses on the validation of the information. This last section is the most crucial to ensure the success of the activities related to the established agreements.

ISSN: 2594 - 2905

Table 2. Format for monitoring agreements in software development.

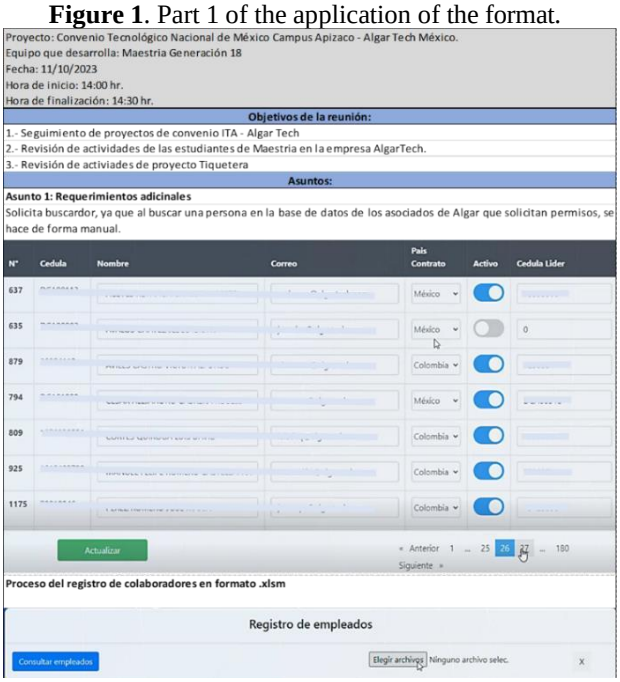
Seguimiento de Acuerdos en el Desarrollo de Software
Proyecto:
Equipo de desarrollo:
Fecha:
Hora de inicio de la reunión:
Hora de finalización de la reunión:
Ojetivos:
1.-
2.-
3.-
Acuerdos:
Acuerdo 1:
Acuerdo estalecido:
Nombre del responsable:
Constancia de notificación al responsable:
Fecha de entrega del acuerdo:
Acuerdo 2:
Acuerdo estalecido:
Nombre del responsable:
Constancia de notificación al responsable:
Fecha de entrega del acuerdo:
Acuerdo 3:
Acuerdo establecido:
Nombre del responsable:
Constancia de notificación al responsable:
Fecha de entrega del acuerdo:
Validación de información.
Relaizado por:
Firma electrónica:
Persona responsable que valida:
Firma electrónica:
Responsable que verifica:
Firma electrónica:

Source. Own creation.

Application of the format in a software development environment.

To test the effectiveness and demonstrate the benefits of the proposed format, it is implemented in the development of an application for human talent management. It is important to note that this format is compatible with both agile and traditional software development methodologies, as illustrated in Figures 1 and 2.

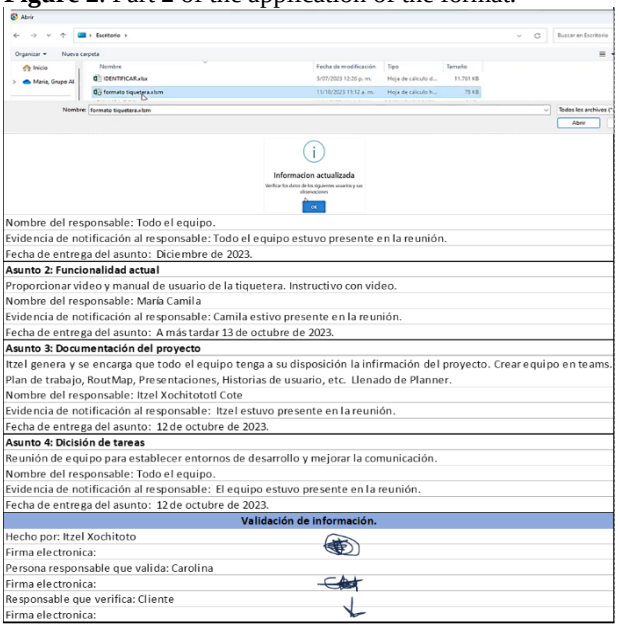
Figure 1. Part 1 of the application of the format.



Source. Own creation.

Figure 1 shows the example of filling in the information, the name of the project, the team assigned to the project, the date and time of the start and end of the meeting. Subsequently, the objectives of the meeting and the first agreement taken are shown, together with the evidence corresponding to the first issue.

Figure 2. Part 2 of the application of the format.



Source. Own creation.

Figure 2 shows the complement of the issues section, the specification of the person responsible for carrying out the activities, evidence that he/she was notified and the date on which the agreement must be delivered, at the end

is the section for validation of the information and is the most important to ensure the success of the activities and the achievement of the objectives stipulated in this document. From this follow-up, ring charts can be made to identify the progress of the activities and verify that they are completed.

DISCUSSION AND ANALYSIS OF RESULTS

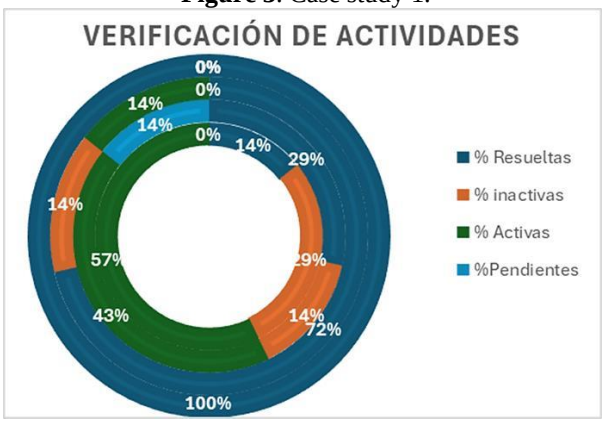
Results of the application of the format in a software development environment through case studies.

Taking into account the previous format, the graph of the progress and review of the activities stipulated in the format is shown; in three case studies.

Case Study One

Figure 3 shows the follow-up of the agreements shown in the application example (Figures 1 and 2).

Figure 3. Case study 1.



Source. Own creation.

Figure 3, represents the progress of the monitored activities, this follow-up allowed the application under development to be improved and the agreements made to be concluded after four revisions. Where, the first iteration of the graph shows that 14% of the activities are resolved, 29% of the activities are inactive, that is, they could not be started, 57% are being carried out, so they have the label of active and no pending activities are found. During the second progress review, 29% of the activities are already resolved, there are still activities that have not been initiated and represent 14%; 43% of the activities are being carried out and inconsistencies were found in 14% of the activities, so they were identified as pending. Meanwhile, in the third monitoring of the activities, 72% are already completed and approved, 14% are inactive, 14% are active and there are no pending activities. With this follow-up, the entire team was able to complete its activities, including the inconsistency, which, without verification and validation follow-up, would have remained unfinished.

It should be noted that, in this project, the inconsistency had its origin in the learning curve that was implemented in the entire development team; however, the correct follow-up and communication remedied the detail.

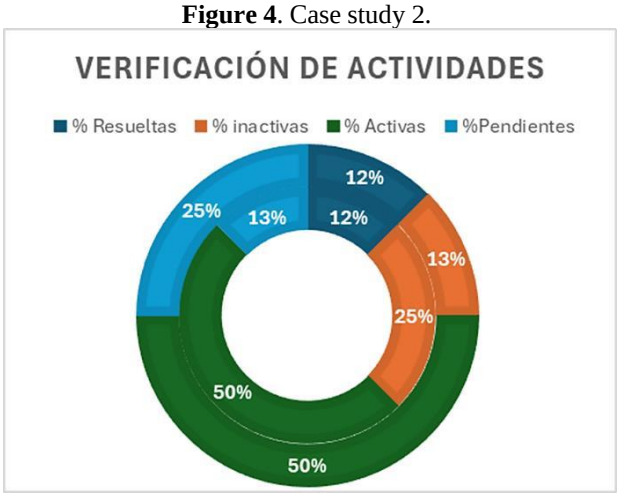
This indicates that validation and verification are important when developing quality software and with the specifications requested by the customer.

Case Study Two

In this case, there was a noticeable resistance to change from the team. Members were uncomfortable with the constant validation and verification of activities, and showed dissatisfaction with the implementation of the new processes. This resistance made the proposed format for monitoring agreements less effective and led to poor execution, negatively affecting its effectiveness.

In addition, changes in project scope were not adequately managed, leading to unfinished activities, as shown in Figure 4. This figure also illustrates how insufficient review of progress can lead to non-compliance with established agreements.

Figure 4. Case study 2.



Source. Own creation.

Taking into consideration the above arguments, Figure 4 shows that, during the first review, 12% of the activities were resolved, 25% were inactive, 50% were active and 13% of the activities were inactive during the first iteration of the format.

Subsequently, in the second iteration, no progress was made with the resolved activities and only 12% continued, which gave us an indicator that something did not work correctly; 13% of the activities remained inactive, which shows that the active ones should have increased, but did not; finally, the pending activities with inconsistencies increased to 25%, resulting in the activities of case two remaining unfinished.

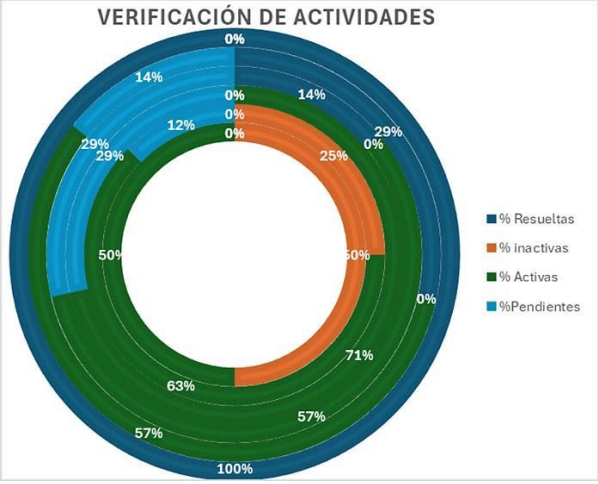
With all this, it is shown that the review format also serves as an essential tool to validate software quality, since it provides a record of progress in relation to the established agreements. However, resistance to the new

format prevented its effective use, thus emphasizing the need for greater acceptance and adaptation to the new monitoring practices.

Case study three

To demonstrate the compatibility of the format with different methodologies, Figure 3 shows the follow-up of activities during the design of a web application for school tracking. In this case, both the initial agreements and the tasks assigned to each team member were considered.

Figure 5. Case study 3



Source. Own creation.

Figure 5 shows that, at the beginning of the activities, there were 50% active and 50% inactive, in the second iteration 25% inactive, 63% active, 12% pending and no resolved activities were reported; for the third iteration, there were no more inactive activities, 71% active and the pending activities increased to 29%, related to inconsistencies in the activities carried out, likewise, no resolved activities were reported; for the fourth iteration, 14% of the activities were completed, 57% were active, the pending activities with inconsistencies did not decrease, but neither did they increase and no resolved activities were reported; For the fifth iteration, 29% of the activities were already resolved, there were no more inactive activities, 57% of the activities were active and the pending activities decreased to 14%; it can be observed that the constant revision resulted in the sixth iteration in 100% of the activities being concluded and accepted, despite the fact that the team felt overworked due to the constant revisions and validations.

This shows that, in order to achieve the established objectives, revisions and validations must be carried out, demonstrating that the application of the proposed format is a constant work and ensures the quality of the product delivered.

CONCLUSIONS

The implementation of the proposed format allows clear identification of roles and responsibilities, which prevents duplication of activities. This facilitates planning, communication and quality assurance, as the format provides a clear view of tasks and a defined path for follow-up. In addition, lack of communication and inconsistent practices can lead to disagreements. Therefore, constant monitoring is crucial to reduce the effects of excessive improvisation and lack of control [19]. A process has been developed to manage maintainability from the early stages of the software life cycle by specifying the inputs and outputs for each activity assigned to particular roles [20].

This process includes roles for verification and validation of maintainability. And it is here where the proposed format takes the fundamental role as a support and follow-up tool.

On the other hand, there were inconsistencies during implementation; these inconsistencies were addressed with careful planning and clear communication, thus minimizing the negative impact and ensuring a successful transition to the implementation of the proposed format.

In summary, the proposed format not only effectively organizes activities and roles, but also strengthens project planning and control. The implementation of a detailed approach for the verification and validation of maintainability consequently ensures a comprehensive and effective management of the software lifecycle.

As future research, we propose the use of the format in software development management under a hybrid software development methodology called "AgileSegura" based on DevSecOps and SCRUM.

Future lines of work could focus on:

-Optimizing the implementation of agile methodologies with a structured format and how to integrate this format within the most popular agile frameworks, without imposing a rigid structure that may affect the adaptability of the team.

-Improving maintainability verification in the software lifecycle, to identify how these practices can evolve throughout the project lifecycle. New quality metrics specific to maintainability could be explored, as well as automation tools that integrate continuous verification throughout the development process, ensuring that software is easy to maintain, modify and expand over time.

-The impact of constant monitoring and control on project success may seem counterproductive in an agile environment, and specific methods and tools could be investigated to integrate monitoring into agile teams

without generating administrative overhead, ensuring that the project stays focused, on budget and on schedule.

ACKNOWLEDGMENT

The authors would like to thank the Tecnológico Nacional de México Campus Apizaco, Graduate Studies and Research Division for the loan of the laboratory and equipment to carry out this project, and the International Software Development Company for opening its doors and allowing the development of the research.

BIBLIOGRAPHY

[1] Dirección General de Cómputo y de Tecnologías de Información y Comunicación. (2022) "Prácticas para el desarrollo de software"[Online] Dirección General de Cómputo y de Tecnologías de Información y Comunicación. Universidad Autónoma de México UNAM [Accessed August 05, 2024] Available at: https://www.red-tic.unam.mx/desarrollo-de-software

[2] Quitian Monroy JS, Martinez Padua JV, Castiblanco Jimenez IA.(2022) "Characterization and comparison of agile and traditional product development methodologies."[Online] Cien.Ing..
[Internet].];32(2):9-26. [Accessed August 19, 2024Available from: https://doi.org/10.18359/rcin.5168

[3] Helen Tupper andbSarah Ellis, (2023) " How to help your team learn in the Flow of work", [Online] Harvard Business Review February. [Accessed August 05, 2024] Available at: https://hbr.org/2023/02/how-to-help-your-team-learn-in-the-flow-of-work

[4] Pinzón López L M, García Rodríguez J C, (2024) "Restructuring of the software change control process for a higher education institution" [Online] Degree work. Universidad de Bosque [Accessed August 05, 2024] Available at: https://hdl.handle.net/20.500.12495/12693

[5] Gramuntell Bayarri, Vicente, (2023) "Analysis and comparative of project management tools" [Online] Degree work. Universidad Politécnica de Valencia, , [Accessed August 05, 2024] Available at: http://hdl.handle.net/10251/197137

[6] ALMAGUER TORRES R M, PEREZ CAMPANA M and AGUILERA GARCIA L O, (2020) "Procedure for integrated and process-based management of local development projects." Rev retos [online]., vol.14, n.1 [Accessed 2024-08-22], pp. 89-115 . Available at: <http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2306-91552020000100089&lng=es&nrm=iso>. Epub 05-Jan-2020. ISSN 2306-9155.

[7] Castillo Anzures M., Guaña Moya E.J.,(2024) "Kanban: An agile methodology for efficient workflow management in software development, a systematic review."
[Online]Global Engineering Journal. Vol.3 No.1, 2024 17-28. [Accessed August 05, 2024] Available at : https://doi.org/10.62943/rig.v3n1.2024.68

[8] Gaete J, Villarroe R, Figueroa I, Cornide-Reyes H, Muñoz R, (2021)" Agile application approach with Serum, Lean and Kanban" [Online] Ingeniare. Rev. chil. ing. vol.29 no.1 Arica mar. 2021 [Accessed August 05, 2024] Available at : http://dx.doi.org/10.4067/S0718-33052021000100141

[9] IBM.(2020)What is a workflow? [Internet] IBM [Accessed August 05, 2024] Available at: https://www.ibm.com/es-es/topics/workflow

[10] Torres Calle, Cristian Esteban, Carrión Narváez, Bryam Sebastián, (2024) "Empirical practices in software development project management from adaptability principle analysis" [Online] Paper.
undergraduate degree. Universidad Politécnica Salesiana, Ecuador. [Accessed August 05, 2024] Available at: https://dspace.ups.edu.ec/handle/123456789/28055

[11] Merchán-Narváez N J, Palma-Peralta E E, Poma-Japón D X, (2024) "Comparison of agile methodologies for software development" [Online] Refereed multidisciplinary journal of scientific research Vol. 8 No. 1 (2024): January-March frequency issue [Accessed 05 August 2024] Available at: https://doi.org/10.56048/MQR20225.8.1.2024.5052-5074

[12] VERA R A, GOMEZ O, MENDOZA J and PECH J, (2020) "Influence of Role Theory on Management activities: A controlled experiment with Software Engineering students". RISTI [online]. 2020, n.39 [Accessed 2024-08-22], pp.131-147. Disponível em: <http://scielo.pt/scielo.php?script=sci_arttext&pid=S1646-98952020000400010&lng=pt&nrm=iso>. ISSN 1646-9895. https://doi.org/10.17013/risti.39.131-147.

[13] Flores-CernaF, Sanhueza-Salazar VM, Valdés-González HM, Reyes-Bozo L, (2022) "Agile methodologies: an analysis of organizational challenges for their implementation" [Online] Vol. 43 No. 1 (2022): January-April 2022 [Accessed August 05, 2024] Available at: https://doi.org/10.14483/23448350.18332

[14] Campos Wright W, Machado Paredes A,(2023) "Artificial Neural Networks for estimating effort in software projects. A comparative study", [Online] International scientific convention, UCIENCIA/23, [Accessed August 05, 2024] Available from: https://repositorio.uci.cu/jspui/handle/123456789/10713

[15] Fernando Villafranca A, Mexicano Santoyo A, Carmona Frausto J C, Dina Valenzuela B, Fernanez Bonilla O D, (2022) "Modelo para la Gestión de Desarrollo de Software en una Empresa Mediana a Través de Formatos Digitales Basados en Moprosoft" [Online] Papers from the International Research Congress Academia Journals Oaxaca 2022 [Accessed 05 August 2024] Available at: https://cdvictoria.tecnm.mx/oferta/maestria_sistemas/Repositorio/2022%20Articulo%20arbitrado%20Modelado.pdf

[16] Núñez-Sánchez Y, González-TorresA, (2020) "Gestión de terceros en el desarrollo de software: propuesta de una metodología (Third-party management in software development: proposal of a methodology) " [Online] Enfoque UTE Revista Vol. 11 No. 2 (2020)

[Accessed August 05, 2024] Available at: https://doi.org/10.29019/enfoque.v11n2.621
[17] Helmut Sy Corvo.(2021) "Minuta Lifeder", [Online] Humanities, Management and Economics [Accessed August 05, 2024] Available from: https://www.lifeder.com/minuta/
[18] Peraza-Delgado M, (2022) "Method for managing a project through digital media in the testing stage: PROMADE case" [Online] Revista Nacional de Administración ISSN 1659-4932, Rev. Nac. Adm. vol.13 n.2, 2022 [Accessed August 05, 2024] Available at: http://dx.doi.org/10.22458/rna.v13i2.3947
[19] Pardo-Calvache CJ, Salazar-Escobar GA, Vargas-Arias GF, Masso-Daza JE, (2020) "Metrics based on a risk-guided approach to assess communication, cooperation and coordination in global software development teams" [Online] Rev. Fac. ing. vol.29 no.54 Tunja Jan./Mar. 2020 Epub Dec 30, 2020 [Accessed Aug 05, 2024] Available at: https://doi.org/10.19053/01211129.v29.n54.2020.11759
[20] Tamayo Espinosa L, Silega Martínez N, (2021) "Maintainability management from early stages in software development" [Onine] Rev cuba cienc informat [online]. 2021, vol.15, n.1 [Accessed 2024-08-19], pp. 52-69 . Epub 31-Mar-2021. ISSN 2227-1899. Available at: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992021000100052&lng=es&nrm=iso

ROLE OF CONTRIUTION TABLE

Role	Author(s)
Conceptualization	Itzel Xochitototl Cote
Data Curation	José Juan Hernández Mora
Methodology	María Guadalupe Medina Barrera
Project management	Elizabeth Cuatecontzi Cuahutle
Resources	Juan Ramos Ramos
Supervision	Juan Ramos Ramos
Validation	María Guadalupe Medina Barrera
Drafting (Original Draft)	Itzel Xochitototl Cote
Drafting Proofreading and editing.	José Juan Hernández Mora