

UNIVERSIDAD PERUANA UNIÓN
FACULTAD DE INGENIERÍA Y ARQUITECTURA
Escuela Profesional de Ingeniería de Sistemas



**Factores de calidad y buenas prácticas de desarrollo de
software ágil con Scrum en Pymes desarrolladoras de
software. Revisión sistemática de la literatura**

Trabajo de Investigación para obtener el Grado Académico
de Bachiller en Ingeniería de Sistemas

Por

Rolin Espinoza Felipe

Asesor:

M.Sc. Fredy Abel Huanca Torres

Lima, 2020

DECLARACIÓN JURADA DE AUTORIA DEL TRABAJO DE INVESTIGACIÓN

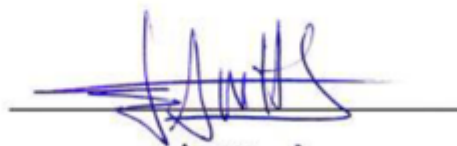
Mg. Fredy Abel Huanca Torres, de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que el presente informe de investigación titulado: *"Factores de calidad y buenas prácticas de desarrollo ágil de software con Scrum en Pymes desarrolladoras de software. Revisión sistemática de la literatura"* constituye el trabajo que presenta la estudiante **Rolin Espinoza Felipe** para aspirar al Grado de Bachiller en Ingeniería de Sistemas, cuyo trabajo de investigación ha sido realizada en la Universidad Peruana Unión bajo mi dirección.

Las opiniones y declaraciones en este informe son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente constancia en Lima, 22 de diciembre del año 2020.



Asesor

M.Sc.. Fredy Abel Huanca Torres
DNI: 01345134

ACTA DE SUSTENTACIÓN DE TRABAJO DE INVESTIGACIÓN

En Lima, Ñaña, Villa Unión, a.....los.....21.....día(s) del mes de.....diciembre.....del año 2020... siendo las.....10:00.....horas, se reunieron los miembros del jurado en la Universidad Peruana Unión campus Lima, bajo la dirección del (de la) presidente(a): Mg. Sergio Omar Valladares Castillo el (la) secretario(a): Ing. Alex Robert Villegas Cervera y los demás miembros:..... Mg. Daniel Lévano Rodríguez y el (la) asesor(a): MSc. Fredy Abel Huanca Torres con el propósito de administrar el acto académico de sustentación del trabajo de investigación titulado: "Factores de calidad y buenas prácticas de desarrollo de software ágil con Scrum en Pymes desarrolladoras de software. Revisión sistemática de la literatura"de los (las) egresados (as): a)..... Rolin Espinoza Felipe b).....

..... conducente a la obtención del grado académico de Bachiller en
 Ingeniería de Sistemas
 (Denominación del Grado Académico de Bachiller)

El Presidente inició el acto académico de sustentación invitando ...al... candidato(a)/s hacer uso del tiempo determinado para su exposición. Concluida la exposición, el Presidente invitó a los demás miembros del jurado a efectuar las preguntas, y aclaraciones pertinentes, las cuales fueron absueltas por ...el... candidato(a)/s. Luego, se produjo un receso para las deliberaciones y la emisión del dictamen del jurado.

Posteriormente, el jurado procedió a dejar constancia escrita sobre la evaluación en la presente acta, con el dictamen siguiente:

Candidato/a (a): Rolin Espinoza Felipe

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
Aprobado	19	A	Con Nominación de Excelente	Sobresaliente

Candidato/a (b):

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	

(*) Ver parte posterior

Finalmente, el Presidente del jurado invitó ...al... candidato(a)/s a ponerse de pie, para recibir la evaluación final y concluir el acto académico de sustentación procediéndose a registrar las firmas respectivas.

 Presidente
 Mg. Sergio Omar
 Valladares Castillo



 Secretario
 Ing. Alex Robert
 Villegas Cervera

 Asesor
 MSc. Fredy Abel
 Huanca Torres

 Miembro

 Miembro
 Mg. Daniel Lévano
 Rodríguez

 Candidato/a (a)
 Rolin Espinoza Felipe

 Candidato/a (b)

ÍNDICE

1.	Introducción	6
2.	Revisión de la literatura	7
2.1	Metodología de Desarrollo de Software Ágil.....	7
2.1.1	Metodologías tradicionales	7
2.1.2	Metodologías ágiles.....	7
2.1.3	Comparación entre metodologías	8
2.2.	Manifiesto ágil	8
2.2.1	Principios del manifiesto ágil.....	10
2.3	Scrum.....	11
2.4	Calidad de software	12
2.4.1	Factores de calidad de software	13
2.5	Modelos de calidad del software	13
2.5.1	Modelo de Boehm.....	13
2.5.2	Modelo de McCall	14
2.5.3	Modelo de Arthur.....	14
3.	Método de la revisión sistemática de la literatura	16
3.1	Necesidad de la revisión sistemática.....	16
3.2	Formulación de las preguntas de investigación.....	16
3.3	Definición del protocolo de investigación.....	17
3.4	Criterios de Inclusión y Exclusión.....	19
3.5	Criterios de calidad	21
4.	Resultados	23
A.	Obtener resultados de la búsqueda	23
B.	Seleccionar los estudios primarios.....	24
C.	Evaluar la calidad de los estudios.....	24
D.	Extraer los datos relevantes	24
E.	Análisis bibliométrico.....	26
F.	Sintetizar los datos extraídos	27
5	Conclusiones.....	30
6	Referencias	31

Quality factors and good practices of agile software development with Scrum in small and medium software developers. Systematic review of the literature.

Factores de calidad y buenas prácticas de desarrollo ágil de software con Scrum en Pymes desarrolladoras de software. Revisión sistemática de la literatura

Espinoza Felipe, Rolin¹

¹ Universidad Peruana Unión, Carretera Central Km 19.5 Ñaña, Chosica

² Lima, Perú

rolinespinoza@upeu.edu.pe

Resumen. La calidad de software está enmarcada en la medida que cumple con los requisitos definidos por una entidad competente, también el grado de satisfacción cliente y una conformidad con sus requerimientos totales, para ello se puede identificar ciertos factores que se debe tener en cuenta para alcanzar el estándar deseado. Además, las buenas prácticas de desarrollo de software ágil contribuyen a cumplir con mencionados requisitos de calidad e incrementar los lineamientos de desarrollo. Todo esto requiere de un cuidadoso examen de los mencionados elementos de calidad sobre las cuales finalmente se obtendrá los mejores resultados con relación al objeto de estudio definido. En este estudio se busca identificar los principales factores de calidad y buenas prácticas de calidad en el desarrollo de software ágil con Scrum principalmente en pymes desarrolladoras de software. Para encontrar los factores de calidad y las buenas prácticas en el desarrollo de software ágil se ha utilizado el método de la revisión sistemática de la literatura en las fuentes bibliográficas más populares en el entorno investigativo. De un total de 1208 artículos revisados, se seleccionaron 17 estudios que refieren a 5 principales factores y a 4 principales prácticas de calidad útiles para el presente estudio. Podemos concluir que, frente a la existencia de muchos factores de calidad de software, el componente más importante es la usabilidad y dentro de las buenas prácticas, la más utilizada es el desarrollo de las reuniones diarias en las Pymes desarrolladoras de software.

Palabras claves: Metodología ágil, desarrollo software, Scrum, metodologías ágiles de desarrollo, Pymes, CMMI, especificación de calidad, requisitos de calidad.

Abstract. Software quality is framed in the measure that it meets the requirements defined by a competent entity, also the degree of customer satisfaction and a conformity with their total requirements, for this reason it is possible to identify certain factors that must be taken into account to achieve the desired standard. In addition, the good practices of agile software development contribute to fulfill the mentioned quality requirements and increase the development guidelines. All this requires a careful examination of the mentioned elements of quality on which finally the best results will be obtained in relation to the defined object of study. This study seeks to identify the main quality factors and good quality practices in the development of agile software with Scrum mainly in SMEs software developers. To find the quality factors and good practices in agile software development, the method of systematic literature review has been used in the most popular bibliographic sources in the research environment. From a total of 1208 articles reviewed, 17 studies were selected that refer to 5 main factors and 4 main quality practices useful for the present study. We can conclude that, in front of the existence of many software quality factors, the most important component is usability and within the good practices, the most used is the development of daily meetings in software developing SMEs.

Keywords: Agile methodology, software development, Scrum, agile development methodologies, SMEs, CMMI, quality specification, quality requirements.

1. Introducción

En el proceso de desarrollo de software es probable que surjan fallos humanos. En general no es tarea sencilla realizar un excelente software, que cumpla todos los requisitos de calidad. La realidad actual es la calidad insuficiente en los productos de software construidos, principalmente por no tener en consideración las estimaciones de tiempos y los medios necesarios, junto con no recopilar de manera adecuada los requerimientos a desarrollar. En el transcurso del mundo de la ingeniería de software cabe mencionar que los métodos, instrumentos y procedimientos que forman parte de la misma van orientados a un único fin: producir software de calidad. [1]

Desde que surgió el “Manifiesto ágil” [2] en el año 2001, las pequeñas, medianas y grandes empresas de software han puesto un interés en trabajar con metodologías ágiles para lograr una mayor productividad en el negocio, debido a muchas buenas prácticas que les permiten obtener una ventaja competitiva frente al resto y asimismo, acelerar el desarrollo de sus actividades de una manera más simple y sencilla de controlar y desarrollar. Actualmente la velocidad de crecimiento en las empresas es impresionante, porque cada día son más el número de empresas que implantan metodologías ágiles dentro de su organización, en las diferentes áreas y departamentos que la conforman.

Dentro de todos estas metodologías y marcos de trabajo aplicadas, se encuentra Scrum, una de las más mencionadas y aplicadas en el contexto del mundo de software actual, pero surgen algunas interrogantes interesantes, tales como ¿En el marco de aplicación, son las pequeñas o grandes empresas las que más la adoptan? ¿Qué factores de calidad se pueden identificar dentro del marco de trabajo Scrum en las pymes desarrolladoras de software? ¿Qué buenas prácticas podemos identificar y cuáles son las más prioritarias en el marco de aplicación en las pymes desarrolladoras de software? La metodología que se utiliza para llegar al objetivo es una RSL (Revisión Sistemática de Literatura), donde se estudia y examina rigurosamente los estudios primarios que guardan una estrecha relación a los factores de calidad y buenas prácticas de calidad en el desarrollo de software con metodologías ágiles en las pymes desarrolladoras de software. Además, es útil referir que la implantación de la metodología Scrum trae múltiples ventajas a la organización tales como: Satisfacción del cliente, entrega de productos de calidad, solución rápida de problemas, transparencia en los procesos de desarrollo, etc. Sin embargo, es útil mencionar que la clave del éxito en la aplicación de métodos ágiles tiene puntos importantes a considerar tales como: Recursos humanos maduros, disciplinados, con experiencias en el área ágil y lo más importante es que el usuario final o el dueño del producto debe tener disponibilidad de tiempo para trabajar con el equipo de desarrollo.[3]

El principal objetivo de la presente revisión sistemática es hallar los principales factores de calidad y buenas prácticas de calidad en el desarrollo de software ágil con Scrum principalmente en pymes desarrolladoras de software. Este estudio está estructurado y distribuido conforme se menciona a continuación: El apartado 2 comprende la revisión de la literatura; el apartado 3 explica el método de la revisión sistemática de la literatura o RSL; y continuando con el apartado 4 en la que se describe los resultados de la revisión y la sección 5 detalla de forma breve las conclusiones, como último apartado se consideran las referencias bibliográficas.

2. Revisión de la literatura

2.1 Metodología de Desarrollo de Software Ágil

Al describir sobre las metodologías ágiles es inevitable no contemplar las metodologías de desarrollo de software tradicional. Las primeras aparecen como consecuencia de la rigurosidad y disciplina de las segundas. Principalmente para minimizar la carga burocrática que son propias de las tradicionales, en proyectos de diferente escala, como pequeñas y medianas. Por consiguiente, esta nueva corriente es adaptativa y está mucho más orientada a las personas, dejando en un segundo plano a los procesos, estos rasgos forman parte de los cuatro valores de la agilidad.

2.1.1 Metodologías tradicionales

En el mundo de las metodologías tradicionales de software es un concepto muy globalizado el tema de la planeación y la organización, ambos son considerados como puntos de inicio muy importantes. “Inician el desarrollo de un proyecto con un riguroso proceso de elicitación de requerimientos, previo a etapas de análisis y diseño. Con esto tratan de asegurar resultados con alta calidad circunscritos a un calendario. Además, inician el desarrollo de un proyecto con un riguroso proceso de elicitación de requerimientos, previo a etapas de análisis y diseño. Con esto tratan de asegurar resultados con alta calidad circunscritos a un calendario.[4]

En las metodologías tradicionales se concibe un solo proyecto, de grandes dimensiones y estructura definida; se sigue un proceso secuencial en una sola dirección y sin marcha atrás; el proceso es rígido y no cambia; los requerimientos son acordados de una vez y para todo el proyecto, demandando grandes plazos de planeación previa y poca comunicación con el cliente una vez ha terminado ésta.[5]

2.1.2 Metodologías ágiles

Las metodologías ágiles son flexibles, pueden ser modificadas para que se ajusten a la realidad de cada equipo y proyecto[6]. Como indica Boehm en la referencia[7] se inclina hacia el veloz desarrollo de aplicaciones o programas y la vida de los productos se reduce notablemente. En este ámbito inestable, que tiene como factor adjunto el cambio y el progreso rápida y continuo, la ventaja frente a la competencia se halla en aumentar la productividad y satisfacer las diferentes necesidades del cliente, y todo esto en el menor tiempo posible para otorgar un mayor valor al negocio.

Como en un punto inicial se había descrito el origen de los métodos ágiles como una reacción contra las metodologías tradicionales consideradas excesivamente pesadas, por su naturaleza muy normativa y fuerte sumisión en la planificación. Esto significa que contrasta completamente con las metodologías tradicionales, tal como está referido que las metodologías de desarrollo ágil en cambio son iterativas, se centran en el trabajo en equipo, la colaboración entre el cliente y el desarrollador, la retroalimentación de los clientes en todo el ciclo de vida del proyecto y apoyan la entrega temprana del producto.[8]

Por lo que podemos observar en la actualidad se ha ganado adhesión por las metodologías ágiles en el mundo de desarrollo de software, porque trae muchos beneficios. Esto hace que cada día la adopción de metodologías ágiles es una realidad latente.

2.1.3 Comparación entre metodologías

En la tabla I se puede apreciar algunos puntos relevantes acerca de cada metodología, muchas de ellas tienen un desglose más amplio, sin embargo, para propósitos comparativos generales se establecen y se definen de esta manera.

Metodologías tradicionales	Metodologías ágiles
Predictivos	Adaptativos
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

Figura 1. Cuadro comparativo de metodologías tradicionales y ágiles Fuente: Elab. Propia

2.2. Manifiesto ágil

Apelando a un contexto histórico, se expresa que, en el mes de febrero del año 2001, se llevó a cabo una reunión con varios académicos y expertos en temáticas del mundo del software, en Utah, uno de los estados de los Estados Unidos. La agenda de discusión fue determinar principios que aporten al desarrollo ágil de proyectos de software. Producto de este congreso fue la creación del ‘Manifiesto Ágil’ que abarca cuatro postulados y una lista de principios asociados.

Una definición elemental sobre el manifiesto ágil sería, según Qumer y Henderson-Sellers, la agilidad es un comportamiento persistente o habilidad, de entidad sensible, que presenta flexibilidad para adaptarse a cambios, esperados o inesperados, rápidamente; persigue la duración más corta en tiempo; usa instrumentos económicos, simples y de calidad en un ambiente dinámico; y utiliza los conocimientos y experiencia previos para aprender tanto del entorno interno como del externo.[7]

Por consiguiente, el desarrollo ágil no peculiariza algunos procesos o procedimientos que seguir, aunque como es sabido, es verdad que han surgido algunas prácticas relacionadas a este movimiento. El desarrollo ágil es en otras palabras una filosofía de desarrollo software que nos acompaña hacia un marco de trabajo mucho más dinámico y reducido. Según el manifiesto se mencionan cuatro postulados muy importantes en el campo, los cuales son:

Valorar al individuo y a las interacciones del equipo de desarrollo por encima del proceso y las herramientas. Este postulado enuncia que las personas son componentes primordiales en cualquier desarrollo. Tres premisas sustentan este postulado: a) los integrantes del equipo son el factor principal de éxito; b) es más importante construir el equipo de trabajo que construir el entorno; y c) es mejor crear el equipo y que éste configure el entorno en base a sus necesidades[9]

En el mundo actual, las organizaciones están compuestas en principio por personas, y para su óptimo funcionamiento y desarrollo dichas organizaciones necesitan que las personas que las integran se comuniquen de manera eficiente y eficaz. Aquí surge una palabra clave para englobar estas características vitales, “soft skills” o habilidades blandas en español, y para facilitar dicha comunicación, los métodos de trabajo ágiles tienden a implantar períodos frecuentes de inspección y adaptación, y se apoyan en una comunicación cara a cara. Incrementar la continuidad de las comunicaciones, son insuficientes para eliminar los problemas de comunicación. Habitualmente, se necesita de comportamientos clave para que los ciclos de inspección y adaptación marche bien.

Valorar el desarrollo de software que funcione por sobre una documentación exhaustiva. El postulado se basa en la premisa que los documentos no pueden sustituir ni ofrecer el valor agregado que se logra con la comunicación directa entre las personas a través de la interacción con los prototipos.[9] En un contexto más amplio, al reemplazar el término ‘software’ por ‘producto o servicio’ se encuentra una comprensión más copiosa de este valor. Un “producto o servicio” que funciona es una de las grandes diferencias que contribuye el desarrollo Ágil. La gran mayoría de los sistemas ágiles se fundamentan en la entrega al cliente de partes pequeñas de “producto o servicio” que funciona, a intervalos fijos. Se conoce como entrega «incremental». Para conseguirlo, es vital que el equipo de trabajo establezca una puntualización común de lo que entienden por «que funciona», lo que se conoce en el marco de trabajo scrum como “definición de completo” o *Definition of Done (DoD)* en inglés.

Valorar la colaboración con el cliente por sobre la negociación contractual. En el desarrollo ágil el cliente se integra y colabora con el equipo de trabajo. Se asume que el contrato en sí, no aporta valor al producto, sino que es sólo un formalismo que establece líneas de responsabilidad entre las partes.[9]

La tasa de éxito de los proyectos de desarrollo de software se ha potenciado en los últimos años como fruto de las entregas frecuentes, que permite a los clientes proporcionar comentarios sobre el software que funciona parcialmente y que se entrega a intervalos regulares. Esto permite tener una retroalimentación constante por parte del cliente y del equipo de desarrollo, y cada vez se busca estar un paso más cerca del logro de los objetivos.

El propósito de un proyecto no es entregar un producto o servicio, sino resolver un problema o necesidad, y por ello es tan fundamental la participación del cliente. Los autores del Manifiesto Ágil sabían de qué estaban hablando cuando enfatizaron que conseguir la implicación del cliente en el proceso de desarrollo de “producto o servicio” es esencial para el éxito del proyecto. Todos los sistemas de trabajo Ágiles fomentan este valor ya que incluyen un representante del cliente que trabaja mano a mano con el equipo de desarrollo.[8] Valorar la respuesta al cambio por sobre el seguimiento de un plan. La evolución rápida y continua deben ser factores inherentes al proceso de desarrollo. por sobre la capacidad de seguimiento y aseguramiento de planes preestablecidos. [9]

2.2.1 Principios del manifiesto ágil.

En marco de la definición de los principios, que están propuestas en el manifiesto ágil, se hace referencia a las características que hacen la diferencia al comparar un proceso ágil con uno tradicional da a entender las ideas más notables que presenta el marco de desarrollo ágil.

- a. Nuestra máxima prioridad es satisfacer al cliente a través de la entrega temprana y continua de software valioso.

Según Elicer Herrera para que una metodología puede ser calificada como ágil debe empezar a entregar software funcionando y útil en pocas semanas[2]. Esto permitirá que los clientes pierdan la desconfianza y la insatisfacción que tienen debido a largas esperas durante el proyecto.[10]

- b. Bienvenido a los requisitos cambiantes, incluso tarde en el desarrollo. Los procesos ágiles aprovechan el cambio para la ventaja competitiva del cliente.

Al comenzar un proyecto las organizaciones planean el análisis y la construcción del proyecto. Sin embargo, obvian los cambios que se pueden dar a lo largo de este proyecto. En ese contexto el manifiesto ágil propone una estructura adaptable al cambio y la tranquilidad del cliente al proponer sus cambios.[10]

- c. Liberar frecuentemente software funcionando, desde un par de semanas a un par de meses, con preferencia por los períodos más cortos.

El cliente siempre espera ver funcionando el programa, y es eso lo que debe entregárselo[10]. Esto le proporciona la confianza necesaria, la calma y la tranquilidad de que sus requerimientos están siendo atendidos. Por consiguiente, es necesario presentar el sistema en el menor tiempo posible y con las principales funcionalidades implementadas y funcionando.

- d. La gente de negocios y los desarrolladores deben trabajar juntos a diario durante todo el proyecto.

Si bien sabemos que el usuario no tiene conocimiento sobre lenguajes de programación, base de datos o análisis del sistema, sin embargo, el manifiesto ágil propone la participación del usuario durante la construcción del proyecto que es el encargado de validar la funcionalidad y resultados entregados por el software. Herrera Uribe menciona que, “La intervención oportuna del usuario puede resultar decisiva en el éxito de un proyecto y puede reducir el costo o el tiempo”[10].

- e. Desarrollar proyectos en torno a personas motivadas. Brindarles el entorno y el apoyo que necesitan, y confíe en ellos para hacer el trabajo.

Cuando hablamos de un entorno ágil se basa en un equipo autoorganizado y multifuncional, por lo cual la motivación es un factor importante que influye en el entorno del equipo de trabajo y mejora la productividad de ese equipo.

- f. El método más efectivo y eficiente de compartir información a, y dentro de un equipo de desarrollo, es la conversación cara a cara

El trabajo colaborativo debe estar apoyado por un sistema de comunicación entre los miembros del equipo y los clientes, que permite solucionar y validar errores y funcionalidades que garantizan el éxito del proyecto. Por lo tanto, es necesario tener una comunicación personal entre los integrantes del equipo y evitar comunicaciones terciarias porque esta comunicación puede causar problemas dentro del entorno del equipo[10]

- g. El software funcionando es la medida de progreso

Cuando se trata de establecer el estado de un proyecto, si bien existen diversas formas de medirlo, es la cantidad de requerimientos implementados y funcionando la que más claridad y confiabilidad ofrecen para establecer una medida del avance del proyecto[10]

- h. Los procesos ágiles promueven el desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios deberían ser capaces de mantener relaciones cordiales

Trabajar de forma moderada y responsable es un principio fundamental para cumplir las metas propuestas en la planeación del software. Por consecuencia los integrantes del equipo deben cumplir las tareas asignadas responsablemente y no imponer lo urgente con lo importante[10]

2.3 Scrum

Su nombre no corresponde a una sigla, sino a un concepto deportivo, propio del rugby, relacionado con la formación requerida para la recuperación rápida del juego ante una infracción menor[4]. Su primera referencia en el contexto de desarrollo data de 1986, cuando Takeuchi y Nonaka utilizan el Rugby Approach para definir un nuevo enfoque en el desarrollo de productos, dirigido a incrementar su flexibilidad y rapidez, a partir de la integración de un equipo interdisciplinario y múltiples fases que se trasladan entre sí [11].

El marco de desarrollo ágil Scrum para el desarrollo ágil de software es un marco de trabajo diseñado para lograr la colaboración eficaz de equipos en proyectos, que emplea un conjunto de reglas y artefactos y define roles que generan la estructura necesaria para su correcto funcionamiento.

Scrum utiliza un enfoque incremental que tiene como fundamento la teoría de control empírico de procesos. Esta teoría se fundamenta en transparencia, inspección y adaptación; la transparencia, que garantiza la visibilidad en el proceso de las cosas que pueden afectar el resultado; la inspección, que ayuda a detectar variaciones indeseables en el proceso; y la adaptación, que realiza los ajustes pertinentes para minimizar el impacto de las mismas[10].

Los denominados ‘Equipos Scrum’ son autónomos en la gestión, multifuncionales y laboran en iteraciones. La autogestión les posibilita seleccionar la mejor forma de realizar el trabajo, en lugar de tener que obedecer lineamientos de individuos que no forman parte del equipo y escasean de contexto. Los miembros del equipo tienen toda la ciencia y la aptitud necesaria (por ser multifuncionales) para desempeñar el trabajo. La entrega del producto se hace en iteraciones; cada iteración crea nuevas funcionalidades o actualiza las que el propietario del producto demanda.

Scrum define tres roles: el Scrum master, el dueño del producto y el equipo de desarrollo[4]. El Scrum master tiene como función asegurar que el equipo está adoptando la metodología, sus prácticas, valores y normas; es el líder del equipo pero no gestiona el desarrollo[11]. El dueño del producto es una sola persona y representa a los interesados, es el responsable de maximizar el valor del producto y el trabajo del equipo de desarrollo; tiene entre sus funciones gestionar la lista ordenada de funcionalidades requeridas o Product Backlog. El equipo de desarrollo, por su parte, tiene como responsabilidad convertir lo que el cliente quiere, el Product Backlog, en iteraciones funcionales del producto; el equipo de desarrollo no tiene jerarquías, todos sus miembros tienen el mismo nivel y cargo: desarrollador. El tamaño óptimo del equipo está entre tres y nueve personas.[11]

Scrum define un evento principal o Sprint (figura 1) que corresponde a una ventana de tiempo donde se crea una versión utilizable del producto (incremento). Cada Sprint, como en el rugby, es considerado como un proyecto independiente. Su duración máxima es de un mes. Un Sprint se compone de los siguientes elementos: reunión de planeación del Sprint, Daily Scrum, trabajo de desarrollo, revisión del Sprint y retrospectiva del Sprint.

En la reunión de Planeación del Sprint se define su plan de trabajo: qué se va a entregar y cómo se logrará. Es decir, el diseño del sistema y la estimación de cantidad de trabajo. Esta actividad dura ocho horas para un Sprint de un mes. Si el Sprint tiene una duración menor, se asigna el tiempo de manera proporcional.[2]

El Daily Scrum es un evento del equipo de desarrollo de quince minutos, que se realiza cada día con el fin de explicar lo que se ha alcanzado desde la última reunión; lo que se hará antes de la siguiente; y los obstáculos que se han presentado. Este evento se desarrolla mediante una reunión que normalmente es sostenida de pie con los participantes reunidos formando un círculo, esto, para evitar que la discusión se extienda[7].

La Revisión del Sprint ocurre al final del Sprint y su duración es de cuatro horas para un proyecto de un mes (o una proporción de ese tiempo si la duración es menor). En esta etapa: el dueño del proyecto revisa lo que se hizo, identifica lo que no se hizo y discute acerca del Product Backlog; el equipo de desarrollo cuenta los problemas que encontró y la manera en que fueron resueltos, y muestra el producto y su funcionamiento. Esta reunión es de gran importancia para los siguientes Sprints.[7]

2.4 Calidad de software

Una definición sencilla de la calidad de software se plantea que es a concordancia con los requisitos funcionales y de rendimiento enmarcados, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente[12]

Por tanto, se considera como calidad de software al producto elaborado con una metodología estándar y un marco de trabajo adecuado para la agilización del desarrollo en el proyecto de software. Además, se concibe que la calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario[12]

Con las referencias anteriores se puede ver que la calidad de un software tiene que ver por un lado con el rendimiento y la correcta recopilación de los requerimientos, así como del uso de una buena metodología y, por otro lado, tiene que ver con la utilidad y con la eficiencia, confiabilidad, seguridad y disponibilidad. Y por consiguiente se puede referir como síntesis que la calidad del software puede ser entendida como el grado con el cual el usuario percibe que el software satisface sus expectativas.[13]

El vocablo ‘calidad’ es un concepto variado; hay distintas dimensiones del mencionado término, diversos temas que tienen que ver con ella. Algunos son manifiestos y elementales; otros son perspicaces y complejos. Unos son usualmente permitidos; otros son simplemente entendidos. El concepto de ‘calidad’ del software se refiere, principalmente, al hecho de especificar, analizar, diseñar e implementar software que cumpla plenamente los requerimientos del usuario, sin errores y con un funcionamiento adecuado. Ello involucra a todas las etapas del ciclo de vida y, por tanto, tiene relación con los diversos métodos y herramientas que pueden utilizarse para alcanzar una buena calidad [13]

2.4.1 Factores de calidad de software

- ✓ **Corrección.**
El grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente
- ✓ **Fiabilidad**
El grado que se puede esperar de una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida
- ✓ **Eficiencia**
La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados
- Integridad**
El grado con que puede controlarse el acceso al software o a los datos a personal no autorizado
- ✓ **Facilidad de uso**
El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.
- ✓ **Facilidad de mantenimiento**
El esfuerzo requerido para localizar y reparar errores
- ✓ **Flexibilidad**
El esfuerzo requerido para modificar una aplicación en funcionamiento.
- ✓ **Facilidad de prueba**
El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos
- ✓ **Portabilidad**
El esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo
- ✓ **Reusabilidad**
Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones
- ✓ **Interoperabilidad**
El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos

2.5 Modelos de calidad del software

A medida que la industria del software está alcanzando el desarrollo a gran escala, se hallan diversos defectos en creación de software. Y el resultado ha sido la producción de sistemas que han tenido muy altos costes de rendimiento que no han consumado las verdaderas y principales necesidades de los usuarios.

Todo ello ha llevado a la comunidad científica a buscar un medio de mejorar la calidad del software y una forma adecuada para medirla. En ocasiones, hay razones legales para proporcionar un software de calidad, particularmente cuando están en juego grandes cantidades de dinero o, incluso, la vida de personas. En otras ocasiones, dar satisfacción a los usuarios del producto[13] es, en pocas palabras, un asunto de suficiencia propia e imagen de la organización.

En los siguientes apartados se presentan algunos de los modelos de calidad más significativos existentes en la actualidad, ordenados cronológicamente. Hay que destacar que el modelo que más relevancia ha tenido hasta el presente, y así se refleja en la literatura relacionada con el tema, ha sido el modelo de McCall, en el que se basan muchos de los modelos publicados.

2.5.1 Modelo de Boehm

B. W. Boehm diseñó un árbol que representaba las características de la calidad del software (Figura 2.3) cuyo primer nivel reflejaba los usos sobre los que se podía basar la evaluación del software [Boehm, 76] [Boehm, 78]. El nivel inferior proporcionaba un conjunto de características primitivas completamente independientes entre sí. Estas primitivas se combinaban en conjuntos de condiciones necesarias que conformaban las características del nivel intermedio. Las características primitivas se podían medir utilizando una serie de medidas.[13]

2.5.2 Modelo de McCall

James A. McCall estableció un modelo para obtener una medida de la calidad del software. El modelo se basa en la división jerárquica de la calidad que se muestra en la Figura 2.4. En el nivel superior, se identifican los principales aspectos de la calidad del software: factores. Por debajo de estos factores se encuentran una serie de atributos que, si se encuentran presentes en el software, proporcionan las características representadas por los factores. Estos atributos se denominan criterios. Por último, se sitúan las métricas que proporcionarán información cuantitativa sobre distintos aspectos de los criterios.[13]

El modelo de McCall tiene dos niveles de aplicación: desde el punto de vista de la gestión del proyecto y desde el punto de vista de la gestión de la calidad. El modelo está pensado inicialmente para los paradigmas tradicionales y, para su evaluación, incorpora un conjunto de medidas normalizadas, la mayoría como listas de comprobación, otras como una cuenta de un atributo concreto y otras tomadas de otros autores. Además, también se otorga un peso a cada medida, obtenido por correlación en datos empíricos obtenidos de programas de las Fuerzas Aéreas Norteamericanas. La principal ventaja de este modelo sobre el precedente es que está diseñado para ser utilizado no sólo sobre el código, sino también sobre la documentación. Y, lo que es más importante, por primera vez en un modelo de calidad, el autor propone su utilización durante las distintas etapas del ciclo de vida de construcción del software, aunque sin detallar cuáles serían los pasos de su aplicación.[13]

2.5.3 Modelo de Arthur

L. J. Arthur presentó una pequeña variación al modelo de McCall consistente en añadir tres nuevos criterios, así como en modificar las relaciones existentes entre éstos y los factores.

Auditabilidad: son los atributos que posibilitan la elaboración de auditorías sobre el software, los datos y sus efectos.

Complejidad: son aquellas características del software que reducen la claridad de un módulo o del programa.

Seguridad: son atributos del software que proporcionan control y protección, tanto al software como a los datos.

Arthur presentó una ligera modificación al árbol de calidad de McCall. Según el autor, el modelo serviría para evaluar las necesidades del sistema y transmitir las a los programadores. Por primera vez, se aporta la idea de agregar los valores por el árbol, indicando que la calidad es una función de los factores y cada factor, a su vez, es una función de los criterios, aunque no define cómo deben ser estas funciones.[13]

Su modelo de calidad está enmarcado para utilizarlo solamente sobre el código (aplicado a los paradigmas tradicionales). Divide los criterios en características medibles y mesurables, para los que plantea unos conceptos y medidas simples, muchas sin organizar. Todo esto lo hace más difícil de manejar.

3. Método de la revisión sistemática de la literatura

La metodología de investigación denominada como la Revisión Sistemática de la Literatura (RSL), ha sido construida para recoger y examinar toda la información y evidencia utilizable sobre el objeto de estudio. Dicha metodología incluye una serie de procedimientos definidos en un protocolo de revisión que especifica los pasos a seguir. La metodología RSL fue elaborada y enfocada para la toma de decisiones en las investigaciones que se producen. Esta metodología presenta una arquitectura diseñada por 6 etapas, donde cada etapa es una serie de procesos que el investigador debe realizar y analizar para tomar decisiones acertadas en cuanto a la información que tomará para su investigación.[10]

3.1 Necesidad de la revisión sistemática

La revisión sistemática que se presenta en esta investigación surge por la necesidad de identificar los factores de calidad del software y las buenas prácticas enmarcadas hacia el desarrollo ágil de software, en los diferentes estudios que se puedan seleccionar de los principales repositorios y bibliotecas virtuales sobre investigación en el área de ingeniería de software. Se aplicará la metodología mencionada anteriormente, detallando cada uno de los pasos y estableciendo los criterios necesarios para el éxito de la presente investigación. De acuerdo a la plantilla Goal, Question, Metric (GQM por sus siglas en inglés) para establecer el objetivo de la investigación [10], se pueden observar los componentes descritos en la Tabla II.

Además, este requerimiento se cimenta en la creciente necesidad por el desarrollo de software de calidad, es un tema latente en el campo del desarrollo de software, además todo esto sugiere un desafío para los desarrolladores en las decisiones que se toman en los proyectos determinados para la definición de las actividades, marco de trabajos y metodologías a emplearse.

CAMPO	VALOR
Objeto de estudio	Calidad de desarrollo de software ágil con scrum
Propósito	Reconocer y describir
Foco	Factores y métricas
Involucrados	pymes desarrolladoras de software, compañías de software, industria de software, medianas empresas de software
Factores de contexto	Ninguno

Tabla 2. Elaboración del objetivo de la investigación.

3.2 Formulación de las preguntas de investigación

Para la definición y estructuración de las preguntas de investigación se tomó como referencia la finalidad de la investigación expuesta en la sección anterior. En la Tabla III se muestran las preguntas propuestas y la motivación para cada una de ellas.

Adicionalmente en la Tabla IV se presentan las preguntas de bibliometría propuestas con la finalidad de obtener visibilidad sobre la evolución y tendencia de los estudios en el tiempo.[14]

3.3 Definición del protocolo de investigación

El reglamento de la revisión consiste en la determinación formal sobre la secuencia de instrucción en el proceso de la revisión sistemática. Por consiguiente, con la finalidad de dar a conocer determinadas cuestiones del estudio, se muestra los criterios que son usados para la realización de la búsqueda de la literatura relevante para llevar a cabo la investigación a desarrollar en este estudio.

PREGUNTAS DE INVESTIGACIÓN:

1. ¿Qué factores de calidad de software podemos identificar en la metodología ágil Scrum?

3. ¿En qué porcentaje se encuentra implementado Scrum tanto para las pymes y las grandes compañías?

3. ¿Qué buenas prácticas son las más relevantes para alcanzar un software de calidad en el desarrollo ágil con Scrum?

Figura 3. Preguntas de investigación. Fuente: Elab. Propia

PREGUNTAS BIBLIOMÉTRICAS:

1. ¿Cuál es la cantidad de publicaciones por tipo de artículo sobre los factores de calidad de software con Scrum?

2. ¿Cómo ha evolucionado en el tiempo la frecuencia de las publicaciones sobre los factores de calidad de software con Scrum?

3. ¿Cuáles son las publicaciones en las que se han encontrado estudios relacionados sobre los factores de calidad de software con Scrum?

Figura 2. Preguntas bibliométricas

Definición de las cadenas de búsqueda

Cadena de búsqueda. La estrategia elegida para la elaboración de la cadena de búsqueda fue la estrategia PICO a lo largo de un proceso iterativo en el cual se realizaron los ajustes convenientes para la selección de resultados. [10]

Población:

- ✓ Término principal 1: factores de calidad de software
- ✓ Términos alternos: propuestas, estudios, procedimientos, técnicas
- ✓ Justificante: se selecciona el término por ser el objeto de estudio de la revisión a ejecutar y se obtienen los términos alternos que representan las variantes o cercanos al término principal.
- ✓ Término principal 2: buenas prácticas de calidad
- ✓ Términos alternos: actividades ágiles, practicas, criterios, recomendaciones
- ✓ Justificante: se selecciona el término por ser el tipo de análisis a ejecutar y se obtienen dichos términos alternos por ser aquellos los diferentes tipos de comparaciones existentes.

Intervención:

- ✓ Entidad: Uso de metodologías de desarrollo ágil.
- ✓ Término principal 1: metodología de desarrollo ágil.
- ✓ Términos alternos: Marco de trabajo scrum, framework scrum, prácticas ágiles, método scrum.
- ✓ Justificante: se selecciona el término por ser el elemento sobre el cual se realizará el análisis y se tiene como resultado referidos términos alternos por ser estos los tipos de objetos.

Comparación:

- ✓ no aplica ya que en la RSL no se hace contraste alguno con algún patrón de referencia.

Resultado:

- ✓ Entidad: Propuestas y experiencias de comparación de herramientas de software
- ✓ Término principal: experiencias
- ✓ Términos alternos: propuestas, evaluación
- ✓ Justificante: se seleccionan dichos términos puesto que es lo que se busca obtener como resultado de la investigación.

Tipo de diseño.

En la presente revisión no está contemplado esta temática, principalmente porque se desea examinar y analizar las experiencias de usuario y con ello establecer algunos criterios de análisis sobre las métricas y factores de calidad en el desarrollo de software con el marco de trabajo scrum.

Idioma.

El idioma para la definición de la cadena de búsqueda ha sido el inglés porque en la mayoría de bibliotecas virtuales el idioma en la que están almacenados los artículos es el referido inicialmente

Haciendo caso de las recomendaciones de la estrategia PICO, se tiene que adicionar los operadores booleanos de búsqueda, entre los demás componentes ya definidos anteriormente: Población AND Intervención AND Resultado. Para mayor detalle, se muestra la tabla V, allí observa los términos en el idioma inglés y, además, los conectores lógicos que se está usando en la cadena de búsqueda de la presente revisión.

CONCEPTO	TÉRMINOS	TERMS
Población	factores, componentes, atributos, características y requisitos	("factors" or "components" or attributes or characteristics or requirements)
Intervención	Desarrollo de software ágil, Marco de trabajo scrum, framework scrum, prácticas ágiles, método scrum	("Agile software development" o "Scrum framework" o "Agile practices" o "Scrum method")
Comparación	no aplica	no aplica
Resultado	Propuestas y aplicaciones de las métricas de calidad de software, experiencias, estudios	("Proposals" AND "applications") AND (software quality metrics) AND(experiences OR studies)
Contexto	No aplica	No aplica

Figura 4. Términos en inglés y conectores lógicos usados en la búsqueda. Fuente: Elab. Propia

3.4 Criterios de Inclusión y Exclusión.

Después de ejecutar la cadena de búsqueda en las diferentes y con base a la estructura de reglas elaborados por Kitchenham, se tienen que evaluar los resultados con el fin de establecer los estudios primarios que responden de una manera clara las preguntas de investigación. En la tabla VI y VII se muestran los criterios más resaltantes que se ha tomado para evaluar los documentos de estudio.

Código	Descripción del criterio de inclusión
C.I.1	Se consideran todos aquellos artículos provenientes de librerías digitales determinadas.
C.I.2	Los artículos deben provenir del área de Ingeniería de Software concretamente.
C.I.3	Se aceptarán artículos que contengan estudios o análisis de factores de calidad de software bajo metodologías de desarrollo ágil.
C.I.4	Se considerarán todos los artículos que se encuentren dentro del rango de temporalidad definido. (2015-2020)
C.I.5	Se aceptarán artículos provenientes de revistas científicas
C.I.6	Se aceptarán artículos de revisión
C.I.7	Se aceptarán solamente los documentos cuyo tema de publicación sea de 'calidad de software'
C.I.8	Se aceptarán artículos de investigación
C.I.9	Se aceptarán los documentos de tipo 'Journals'

Tabla 1. Criterios de Inclusión

Código	Descripción del criterio de exclusión
C.E.1	Serán excluidos los artículos duplicados sobre factores de calidad de software
C.E.2	Serán rechazados los artículos que no se encuentren en idioma inglés.
C.E.3	Serán rechazados los artículos de contenido similar, quedándose solo los que
C.E.4	Serán excluidos los artículos cuyo título no tenga relación con el objeto de estudio

Tabla 2. Criterios de Exclusión

Temporalidad. Es menester tomar en cuenta los estudios realizados en los últimos 10 años porque se requiere analizar artefactos y/o metodologías de software que permanezcan vigentes. Además, se considera que si bien el entorno sobre el que se ejecuta esta investigación es el de calidad de software en el marco de trabajo scrum, y se sabe que estamos ante un contexto muy cambiante en el transcurso del tiempo relacionado a las tecnologías; a pesar de todo esto, se nota la existencia de instrumentos y metodologías que están vigentes por medio de una serie de reajustes, considerándose cualquier análisis hecho durante mencionado trayecto recorrido relevante e importante para la investigación en particular.[14]

Fuentes de datos. Las fuentes bibliográficas presentes en la internet e incluidas, debido a que fueron tomados en cuenta por la relevancia en materia de conocimiento válido para la extracción de artículos son los siguientes:

- ✓ ACM Digital Library (<https://dl.acm.org/>)
- ✓ Science Direct (<http://www.sciencedirect.com>)
- ✓ Google Scholar (<https://scholar.google.com/>)
- ✓ IEEEExplore (<http://www.ieee.org/web/publications/xplore/>)

Procedimientos para la selección de estudios. El sistema estructurado elegido para la selección de los documentos en la RSL que finalmente son los artículos, es el siguiente:

✓ **Paso 1:**

El primer paso fue iniciado con la búsqueda de la cadena PICO, en las librerías digitales incluidas previamente, y para ello con la aplicación de los criterios de inclusión y exclusión de acuerdo a la tabla VIII. Las referencias encontradas en todos los documentos o artículos de la búsqueda fueron asegurados para un siguiente refinamiento.

✓ **Paso 2:**

En el segundo paso, se observaron el tipo de artículo y el lugar de procedencia, incluyendo solo aquellas que provinieran de una revista científica. Además, analizando todas las publicaciones, se procedió solamente a aceptar dentro del marco de artículos a los que son de clasificación como ‘artículos de investigación’ (‘research article’).

✓ **Paso 3:**

En el tercer paso, se revisaron los artículos y solo se aceptaron aquellos que estén dentro de la categoría de ‘calidad de software’ o ‘software quality’ asimismo, se consideró los demás criterios especificados en la tabla VIII, tales como aceptar o incluir aquellos documentos de tipo ‘journals’ con lo cual se redujo considerablemente el tamaño de resultados en la plataforma digital.

✓ **Paso 4:**

En este paso se siguió con un examen minucioso del contenido de los títulos y los resúmenes de los artículos seleccionados posterior al paso 3, además de ello, se procedió a excluir a todos aquellos elementos que no estén en idioma inglés. Posterior a ello se procedió a dejar de lado aquellos que sean de contenido duplicado o similar, y finalmente se aplicó el criterio de eliminar o no contar con los que no forman parte con el objeto de estudio. Todos estos aspectos están enmarcados en la tabla VIII.

PROCEDIMIENTO	CRITERIO DE SELECCIÓN
Paso 1	C.I.1, C.I.4, C.I.6
Paso 2	C.I.5, C.I.8, C.E.2
Paso 3	C.I.7, C.I.9
Paso 4	C.E.4, C.I.3, C.E.3, C.E.2

Tabla 3. Procedimiento y utilización de los criterios.

3.5 Criterios de calidad

Esquema de evaluación de la calidad de estudios. Siguiendo con los lineamientos planteados en la guía de Kitchenham[15], continuando con el sistema de pasos se define el compendio de análisis de calidad, con lo cual se finaliza el estudio de los artículos elegidos.

Para verificar el cumplimiento de los criterios de ambos tipos, se estableció toda una serie de listas de consideraciones. Cada consideración tiene presente un puntaje en base en la escala de Rouhani[11], pasando a detallar sobre lo que trata son los siguientes puntos: Sí cumple (S) = 1, Cumple parcialmente (P) = 0.5 y No cumple (N) = 0. Al final todos los resultados son presentados tal como lo detalla el esquema en la tabla IX.

Estrategia para la recopilación de datos. Se desarrolló la estrategia para la selección de datos por medio del bosquejo de un formulario, con la finalidad de extraer toda la información relevante y vital para responder las interrogantes de investigación planteadas, el cual se describe en la Tabla X. Se tomaron en cuenta las recomendaciones dadas por Kitchenham y Brereton [16] para la construcción del mismo.

Estrategia para la síntesis de datos. El conjunto de pasos para el resumen de la información extraída para esta RSL, se explica de la siguiente manera, es la consumación de la síntesis de la literatura fundamentado en el marco sugerido por Popay [11], el mencionado método consta de los siguientes pasos:

1. Se va a ejecutar un resumen preliminar en base al objetivo de conseguir un detalle inicial de patrones durante el recorrido de cada uno de los estudios elegidos con relación a lo detallado en la Tabla XI.
2. Explorar los vínculos entre las conclusiones de los hallazgos de los estudios y los puntos más relevantes de la población de estudio, funcionamiento y contexto de los referidos.
3. Explorar cuán compacto y qué nivel de calidad tiene el estudio para sacar conclusiones bien fundamentados y sustentados sobre los resultados y el estándar del resumen.

N°	Criterio de evaluación de calidad
1	<p>¿El método seleccionado para llevar a cabo el estudio ha sido documentado apropiadamente? S: El método seleccionado ha sido documentado apropiadamente. P: El método seleccionado ha sido documentado parcialmente. N: No se ha documentado el método seleccionado.</p>
2	<p>¿El estudio aborda las amenazas a la validez? S: El estudio aborda las amenazas totalmente P: El estudio aborda las amenazas parcialmente N: No se detallan amenazas</p>
3	<p>¿Se han documentado las limitaciones del estudio de manera clara? S: Las limitaciones se han documentado claramente. P: Las limitaciones se han documentado parcialmente. N: No se ha documentado las limitaciones.</p>
4	<p>¿Los aportes del estudio para las comunidades científica, académica o para la industria han sido descritos? S: Los aportes del estudio han sido mencionados claramente. P: Los aportes del estudio han sido mencionados parcialmente. N: No se han mencionado aportes.</p>
5	<p>¿Los resultados han contribuido a responder las preguntas de investigación planteadas? S: Los resultados han contribuido a responder todas las preguntas de investigación. P: Los resultados han contribuido a responder algunas preguntas de investigación. N: Los resultados no han contribuido a responder las preguntas de investigación.</p>

Tabla 4. Criterios de Evaluación de Calidad Fuente: Elab. Propia

Criterio	Detalle	Relevancia
Título del estudio		
Autor		
Año de publicación		
Resumen		
Tipo de estudio		
Método ágil usado		
Entorno del estudio		
Fecha de la extracción		
Base de datos encontrada		
Resultados obtenidos		

Tabla 5. Formulario de extracción de datos.

	Beneficios	Limitaciones
Descripción textual		
Tabulación		
Agrupación		
Análisis temático		

Tabla 6. Síntesis Preliminar. Fuente: Elab. Propia.

Validar el protocolo de investigación. El protocolo que se ha usado para el desarrollo de la RSL (Systematic Review of the Literatura) fue verificado por el ingeniero Omar Loayza Jara.

4. Resultados

De acuerdo a los pasos definidos en la sección III, el primer paso para la selección de estudios consiste en la ejecución de la cadena de búsqueda en las librerías digitales seleccionadas. En la Tabla X se muestran los resultados y las cadenas de búsquedas empleadas.

A. Obtener resultados de la búsqueda

- ✓ En la mayoría de casos fue necesario ajustar la cadena de búsqueda de acuerdo a la sintaxis de cada librería o a la cantidad de resultados obtenidos ya que en algunos casos era excesiva. Se realizaron los ajustes de acuerdo al siguiente detalle:
- ✓ La fuente de datos de IEEE Xplore tenía presente la capacidad límite de tratar un máximo de cinco caracteres variables definida en la cadena.
- ✓ La base de datos de Google Scholar mostraba muchos resultados de poca relevancia por lo que se suprimió parte de los caracteres variables de la cadena para maximizar la calidad de la búsqueda.

Con el fin de poseer un gran control sobre el vocablo que formarían parte de la cadena de búsqueda se optó por usar la opción de “Advanced Search” o búsqueda avanzada en algunas de las fuentes virtuales. Una vez puesta en marcha la búsqueda se exportaron los resultados obtenidos que son detallados a continuación:

- ✓ Science Direct: la librería tiene la opción de exportar cada documento en formato pdf, y es menester mencionar que la totalidad de los artículos seleccionados al final tienen formato pdf.
- ✓ IEEE Xplore: la librería tiene la opción de exportar cada documento en formato pdf, y es menester mencionar que la totalidad de los artículos seleccionados al final tienen formato pdf.

Cadena	Base de datos	Total
("factor software quality" OR "agile software development") AND (enterprise OR companie OR team OR firm OR SME) AND ("scrum"))	ScienseDirect	404
("factor software quality" AND "software quality" OR "agile software development") AND (enterprise OR companie OR team OR firm OR SME) AND ("scrum"))	IEEE Xplore	109
("factor software quality" AND "software quality" OR "agile software development") AND (enterprise OR companie OR team OR firm OR SME) AND ("scrum"))	ACM	649
"factor quality" OR "metric quality" AND ("scrum" OR "agile software development") AND "software quality"	Google Schoolar	46

Tabla 7. Cadenas de búsqueda y fuentes bibliográficas registradas

B. Seleccionar los estudios primarios

Los artículos que encontramos en todas las bibliotecas virtuales han sido exportados y confirmados en formato pdf, además, el cual fue tomado como punto de referencia para hacer la selección de estudios primarios continuando con los lineamientos planteados en la sección que antecede.

A continuación, en la Tabla XIII se pueden ver los resultados de la recopilación de estudios realizados en las diferentes fuentes:

Base de datos	Artículos descubiertos	Paso 1	Paso 2	Paso 3	Paso 4
ScienseDirect	404	267	15	9	3
IEEE Xplore	109	55	10	5	3
ACM	649	363	321	16	8
Google Scholar	46	20	10	5	3
Total	1208	705	356	35	17

Tabla 8. Resultados del método de selección de estudios. Fuente: *Elab. Propia*. Tabla XIII. Resultados del método de selección de estudios.

C. Evaluar la calidad de los estudios

Teniendo el total de 32 estudios resultantes se usó la lista de criterios de filtro que ya se han definido en el apartado III. Los resultados de la examinación se pueden ver en la Tabla XIV. A partir de la tabla se infiere que solo el 20% de los estudios obtuvieron una puntuación menor al 50% del puntaje total, entonces podemos considerar que es un buen indicador de la calidad de los estudios elegidos para la RSL.

D. Extraer los datos relevantes

En acuerdo con lo que se ha detallado en la guía de Kitchenham y Charters [15], los prontuarios para la selección de datos tienen que cumplir con el requisito de ser diseñados con el fin de recolectar toda la información requerida para solucionar las preguntas de investigación del presente estudio.

Sabiendo esto se construyó un formulario descrito en la sección III para la selección de la información vital de todos los artículos seleccionados que contribuyen en la solución de las preguntas de investigación planteadas. Todos los artículos que fueron seleccionados fueron leídos y paralelamente se procedió con el llenado de sus formularios correspondientes, para lo cual se tenía que traducir el artículo al idioma español.

ID	C1	C2	C3	C4	C5	Total
1	1	0.5	1	0.5	1	4
2	1	1	1	1	1	5
3	0	1	1	0.5	0.5	2.5
4	1	1	1	1	1	5
5	1	1	1	0.5	1	4.5
6	1	0.5	1	0.5	0.5	3.5
7	0	1	1	0.5	1	3.5
8	1	1	1	1	0.5	4.5
9	1	0.5	1	0.5	1	4
10	1	0.5	1	0.5	1	4
11	1	0.5	1	1	0.5	4
12	0.5	1	1	0.5	1	4
13	1	0.5	1	1	1	4.5
14	1	0.5	1	0.5	1	4
15	1	0	1	1	0.5	3.5
16	1	0.5	1	1	1	4.5
17	1	0.5	1	1	1	4.5

Tabla 9. Evaluación de la calidad de estudios. Fuente: Elab. Propia

En la Tabla XIII se puede ver un ejemplo de la información relevante fruto de la búsqueda criteriosa de todos los artículos elegidos. Los demás detalles ya fueron explicados anteriormente con relación a esta información.

Criterio	Detalle	Relevancia
Título del estudio	Significance of Agile Software Development and SQA Powered by Automation	-
Autor	Gonen, Bilal Sawant, Dipali	PB-1
Año de publicación	2020	PB-2
Resumen	Today's software industry is fast moving and has daily changing demands. Many organizations are struggling to cope s.	PB-2
Tipo de estudio	Conference paper	PB-3
Método ágil usado	scrum	PI-1
Entorno del estudio	Agile software development	PI-2
Fecha de la extracción	2020	PI-3
Base de datos encontrada	IEEE explore	PI-3
Resultados obtenidos	1	-

Tabla 10. Ejemplo de extracción de datos de un estudio primario

E. Análisis bibliométrico

Llegados a este punto se procede a puntualizar el análisis minucioso de los artículos escogidos para la presente RSL en perfecta armonía con los factores requeridos para este punto del examen.

1. Pregunta de bibliometría 1 (PB-1)

¿Cuál es la cantidad de publicaciones por tipo de artículo sobre los factores de calidad de software con Scrum?

En la Fig. 1 se observa la cantidad de estudios clasificados por tipo de artículo. Además, se puede interpretar que los artículos de investigación ('Research article') representan el 47% del total de conocimientos registrados para la presente RSL; posteriormente se tiene a los artículos científicos y artículos de conferencia con 18% y 17% respectivamente; finalmente se puede concluir que los artículos de investigación es la mayor fuente para identificar y detallar los factores y buenas prácticas de calidad de software con el uso de metodologías ágiles.



Figura 5. Porcentaje de publicaciones por tipo de estudio

2. Pregunta de bibliometría 2 (PB-2)

¿Cómo ha evolucionado en el tiempo la frecuencia de las publicaciones sobre los factores de calidad de software con Scrum?

En la Fig. 2 se observa que la frecuencia de publicaciones entre los años 2018 y 2019 tuvo un aumento considerable, a comparación de los años anteriores. De un total de 17 estudios entre los años 2015 al 2018 se han publicado un 71% de los estudios seleccionados, y por otra parte un 29% de los estudios se han publicado entre los años 2019 y 2020.



Figura 6. Cantidad de publicaciones conforme el paso del tiempo

3. Pregunta de bibliometría 3 (PB-3)

¿Cuáles son las publicaciones en las que se han encontrado estudios relacionados sobre los factores de calidad de software con Scrum?

En la Fig. 3 se puede observar los estudios clasificados por su dominio de publicación y vemos que existe una mayor recurrencia de participación en el caso de las revistas de ciencias de la computación (Journal of Computing Sciences) con un 35 % del total; seguidamente se puede observar un 24 % de participación por parte de las conferencias de desarrollo de software ágil (Conference Agile software development) y los últimos tienen 6% de frecuencia.

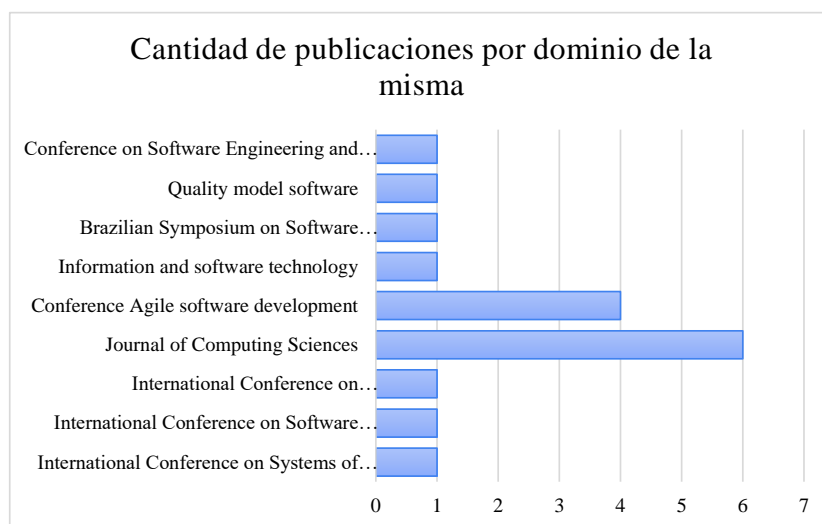


Figura 7. Cantidad de publicaciones por dominio de la misma. Fuente: Elab. Propia.

F. Sintetizar los datos extraídos

A continuación, se muestra como producto de la realización de la RSL y conforme el procedimiento establecido en el apartado III del actual estudio, por todo lo referido anteriormente se eligieron los 32 artículos en los que se

evidencian una lista de los factores y buenas prácticas en el marco de trabajo ágil Scrum. La lista de los factores de calidad y buenas prácticas halladas, se encuentra descrita en la tabla XVI y en ella se indica o se puntualiza cada uno de los puntos relevantes para el vigente estudio.

Id artículo	factores de calidad	Tamaño de empresa	Buenas prácticas
A001	Flexibilidad	PYMES	Reuniones diarias
A002	Eficiencia	PYMES	Reuniones de revisión
A003	Interoperabilidad	Grande	Reuniones diarias
A004	Flexibilidad	PYMES	Backlog del producto
A005	Utilizabilidad	PYMES	Reuniones diarias
A006	Seguridad	Grande	Reuniones diarias
A007	Seguridad	PYMES	Reuniones diarias
A008	Utilizabilidad	PYMES	Reuniones diarias
A009	Utilizabilidad	Grande	Reuniones diarias
A010	Eficiencia	PYMES	Reuniones de revisión
A011	Utilizabilidad	PYMES	Reuniones diarias
A012	Utilizabilidad	Grande	Reuniones diarias
A013	Utilizabilidad	PYMES	Reuniones diarias
A014	Utilizabilidad	PYMES	Reuniones diarias
A015	Seguridad	Grande	Gráficos de Burndown
A016	Flexibilidad	PYMES	Reuniones de revisión
A017	Utilizabilidad	PYMES	Reuniones de revisión

Figura 8. Síntesis de datos extraídos. Fuente: Elab. Propia.

1. Pregunta de investigación 1 (PI-1)

¿Qué factores de calidad de software podemos identificar en el marco de trabajo ágil Scrum?

A través de la extracción de datos de los artículos se logró hallar diferentes factores que son considerados prioritarios en los diferentes entornos de aplicación y además se pudo constatar que hay algunas que son fundamentales en cualquier Pyme que aplique Scrum.

En la Fig. 4 podemos ver que en un 47% de las veces hay una preferencia por el enfoque de la Utilizabilidad, seguido de un 18% por parte de la seguridad, y además están presentes los factores de calidad de Eficiencia, Flexibilidad e Interoperabilidad. La utilizabilidad se relaciona con el grado de satisfacción en el uso del producto, y hay tres métricas principales para su medición eficacia, eficiencia y satisfacción.

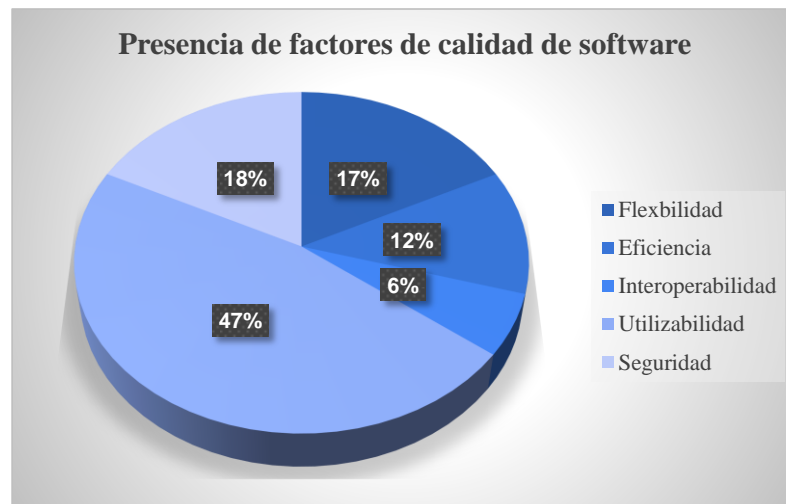


Figura 9. Porcentaje de factores de calidad de software más relevantes.

2. Pregunta de investigación 2 (PI-2)

¿En qué porcentaje se encuentra implementado Scrum tanto para las pymes y las grandes compañías?

En la fig. 5 se puede evidenciar que el 71% de las veces es implementada en Pymes el marco de desarrollo ágil Scrum y un 29% corresponde a las grandes empresas.

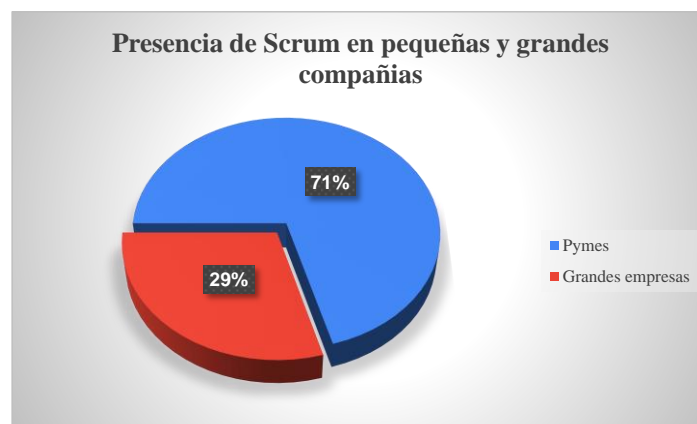


Figura 10. Presencia de Scrum en las entidades

3. Pregunta de investigación 3 (PI-3)

¿Qué buenas prácticas son las más relevantes para alcanzar un software de calidad en el desarrollo ágil con Scrum?

En la presente recolección de datos se ha podido constatar la gran relevancia que tienen algunas buenas prácticas de calidad en las pymes. En la fig. 6 se puede evidenciar que el 65% de los estudios considera como punto vital las reuniones diarias, un 23% corresponde a las reuniones de revisión, un 6% a los gráficos de Burndown y otro 6% al Backlog del producto. De todo ello se puede inferir que la práctica de calidad más usada en las pymes son las reuniones diarias.

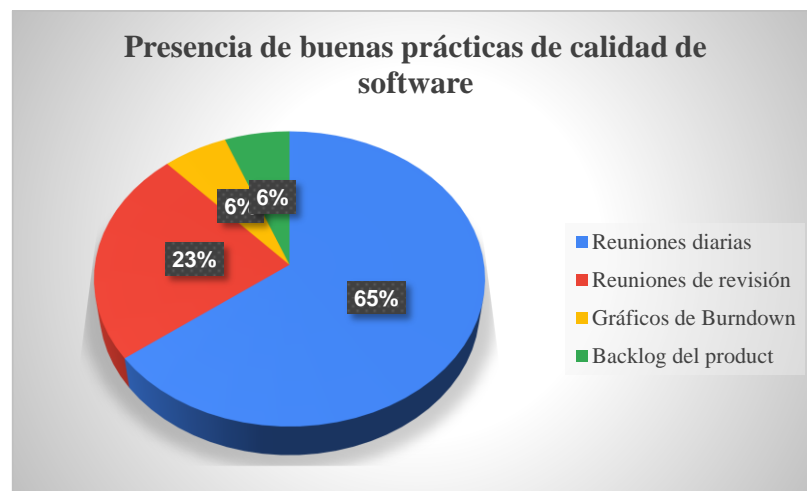


Figura 11. Presencia de buenas prácticas de calidad de software.

5 Conclusiones

Para el presente artículo se utilizó las herramientas y la estrategia PICO de la RSL para investigar de manera rigurosa los estudios primarios que finalmente resultaron ser muy compactos, todos relacionados a las empresas (Pymes y grandes), en identificar y detallar los factores y buenas prácticas de calidad de software con el uso de metodologías ágiles. Se tomaron en cuenta para el análisis 17 estudios primarios, cuyo respectivo examen nos permite llegar a las siguientes conclusiones:

- ✓ Entre los factores de calidad identificados en el presente estudio, la utilizabilidad es el más prioritario en materia de aplicación en las empresas (pequeñas, medianas y grandes) que trabajan con la metodología de desarrollo ágil de software. Este factor es el grado de esfuerzo necesario para aprender, operar y utilizar el software por parte del cliente, se compone de tres métricas de evaluación: eficacia, eficiencia y satisfacción. En los proyectos ágiles de software, uno de los componentes principales es el cliente, por ello, se designa un responsable que está en constante interacción con el cliente para verificar, medir y comprobar el grado de utilizabilidad que tiene cada entregable de producto.
- ✓ Scrum es el marco de desarrollo ágil más utilizada en las empresas de menor tamaño o comúnmente conocido como Pymes y en menor medida en las grandes empresas.
- ✓ La práctica de calidad más usada por las entidades que aplican el marco de desarrollo ágil Scrum, son las reuniones diarias, como punto base para tal afirmación se considera 65% de participación.

Además, se ha realizado un análisis bibliométrico y se detalla dentro del mismo la clasificación de los artículos por año de publicación, donde se puede observar un incremento en el número de estudios importantes publicados 2018 y 2019, estos datos prueban que existe un interés continuo en la incurrencia sobre estos temas. Entre otros resultados se tiene que concluir que, en el marco de tipo de estudio utilizado para el desarrollo de la presente RSL, los artículos de investigación es la mayor fuente que son útiles para el referido estudio y además nos guían en el objetivo de identificar los factores y buenas prácticas de calidad de software con el uso de metodologías ágiles.

6 Referencias

- [1] E. Marcos, “Investigación en Ingeniería del Software vs. Desarrollo Software,” *Nueva Publicación de Prueba*, 2010, [Online]. Available: <http://educacion.usach.cl/ojs/index.php/ojsprueba1/article/view/6>.
- [2] A. C. M. DAMIAN FARROW, JOSEPH BAKER, “ANÁLISIS DE LAS METODOLOGÍAS ÁGILES Y SU INCIDENCIA EN LA CREACIÓN DEL PORTAFOLIO DE SERVICIO PARA LA UNIDAD DE EXTENSIÓN UNIVERSITARIA DE LA UNIVERSIDAD TÉCNICA DEL NORTE DE LA CIUDAD DE IBARRA,” *Nhk 技研*, vol. 151, pp. 10–17, 2015, doi: 10.1145/3132847.3132886.
- [3] C. P. and F. P. W. Ortega, “Mapeo sistemático sobre la evaluación de la agilidad en organizaciones de desarrollo de software,” *ITECKNE Innovación e Investig. en Ing.*, vol. 16, pp. 64–76, 2019.
- [4] A. N. Cadavid, “Revisión de metodologías ágiles para el desarrollo de software,” *Prospectiva*, vol. 11, no. 2, p. 30, 2013, doi: 10.15665/rp.v11i2.36.
- [5] H. Khurana, “Agile : The Necessitate Of Contemporary Software Developers,” vol. 3, no. 2, pp. 1031–1039, 2011.
- [6] O. Paul, “Pruebas De Calidad Aplicadas Al Sitio Web Allison,” *Inst. Tecnológico México*, p. 142, 2017.
- [7] P. Rodríguez, “Estudio de la aplicación de metodologías ágiles para la evolución de productos software,” *Metodol. Agil. para la Evol. Prod. Softw.*, vol. 1, p. 146, 2008, [Online]. Available: 38279714_Estudio_de_la_Aplicacin_de_Metodologas_giles_para_la_Evolucin_de_Productos_Software.
- [8] K. Mendes, C. Karina, and C. Pablo, “Inserción del mantenimiento en los procesos ágiles,” *XVIII Congr. Argentino Ciencias la Comput.*, pp. 910–919, 2001, [Online]. Available: http://sedici.unlp.edu.ar/bitstream/handle/10915/32422/Documento_completo.pdf?sequence=1.
- [9] K. Mendes Calo, E. C. Estevez, and P. R. Fillotrani, “Evaluación de metodologías ágiles para desarrollo de software,” *XII Work. Investig. en Ciencias la Comput.*, pp. 455–459, 2010, [Online]. Available: <http://hdl.handle.net/10915/19546>.
- [10] F. D. E. I. Y. Arquitectura, “Factores que afectan la productividad del equipo Scrum: Una revisión sistemática de la literatura.,” pp. 1–128, 2017.
- [11] G. Hernández, Á. Martínez, R. Jiménez, and F. Jiménez, “Métricas de productividad para equipo de trabajo de desarrollo ágil de software: una revisión sistemática,” *TecnoLógicas*, vol. 22, pp. 63–81, 2019, doi: 10.22430/22565337.1510.
- [12] E. J. B. VILLANES, “Método Para La Evaluación De Calidad De Software,” *Univ. San Martin Porres*, 2015.
- [13] L. J. Fuertes Castro, “MODELO DE CALIDAD PARA EL SOFTWARE ORIENTADO A OBJETOS presentada en la UNIVERSIDAD POLITÉCNICA DE MADRID para la obtención del GRADO DE DOCTOR EN INFORMÁTICA,” 2002, [Online]. Available: http://oa.upm.es/34988/1/TD_Fuertes_JOSE_LUIS.pdf.
- [14] Y. Romero, K. Melendez, and A. Dávila, “Modelos para el análisis comparativo de herramientas de software: Una revisión sistemática de la literatura.”

- [15] M. Muñoz and M. Peralta, “Situación actual sobre la implementación del perfil básico ISO/IEC 29110 en EMP: una revisión sistemática de la literatura,” *RISTI - Rev. Ibérica Sist. e Tecnol. Informação*, no. 36, pp. 1–14, 2020, doi: 10.17013/risti.36.1-14.
- [16] O. Gonzales and J. Abanto, “Análisis comparativo de patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad,” *patrones diseño y Apl. móviles.*, p. 77, 2016, [Online]. Available: <https://repositorio.upeu.edu.pe/handle/UPEU/533>.