



UTA.BA

FACULTAD
REGIONAL
BUENOS AIRES

TRABAJO FINAL INTEGRADOR

ESPECIALIZACIÓN EN INGENIERÍA DE
SISTEMAS DE INFORMACIÓN

Título:

“Precisión de estimaciones en desarrollo de software ágil”

Autor: Ing. Marcelo Fransoy

Tutores: Dr. Alejandro Hossian

Dr. Fernando Pincioli

Buenos Aires – septiembre/2023



Resumen

A pesar que la agilidad moderna comenzó con la publicación del manifiesto ágil en 2001, no se encuentra en la literatura aún un estudio que abarque todas las técnicas utilizadas al momento para estimar distintos aspectos de los proyectos de software ágiles, tanto académicos como industriales. Mucho menos que reporte sus niveles de precisión. El objetivo de este Mapeo Sistemático de la Literatura es disponibilizar el estado del arte en lo que respecta a las técnicas utilizadas para estimar distintos aspectos de un proyecto de Desarrollo de Software Ágil (ASD por sus siglas en inglés), junto con el grado de precisión que ofrecen. La estrategia de búsqueda en las bibliotecas seleccionadas arrojó 4315 artículos, que luego de aplicado el proceso de selección resultó en 166 artículos que tratan la estimación y precisiones en ASD. Se definen 8 preguntas de investigación y 3 preguntas de publicación. Entre los hallazgos más relevantes está la tendencia sostenida de los últimos años del uso de técnicas de estimación basadas en aprendizaje automático. Sin embargo las técnicas basadas en el juicio de expertos siguen siendo las más halladas, incluso se evidencia su uso combinado con técnicas de otros tipos. Un 72.28% de los estudios seleccionados, no revela el uso de técnicas de estimación en la industria. En cuanto al nivel de precisión hallado, el estudio concluye que existe aún un margen importante de mejora en este sentido, que no ha sido resuelto todavía y puede motivar nuevos estudios.

Palabras clave: estimación, precisión, ágil, ASD, SMS

Abstract

Although modern agility began with the publication of the Agile Manifesto in 2001, there is still no literature that covers all the techniques used to estimate different aspects of agile software projects, both academic and industrial. Much less that reports their levels of accuracy. The objective of this Systematic Literature Mapping is to make available the state of the art regarding the techniques used to estimate different aspects of an Agile Software Development (ASD) project, along with the degree of accuracy they offer. The search strategy in the selected libraries yielded 4,315 articles, which after the selection process resulted in 166 articles dealing with estimation and precision in ASD. Eight research questions and three publication questions were defined. Among the most relevant findings is the sustained trend in recent years of using estimation techniques based on machine learning. However, expert judgment-based techniques continue to be the most commonly found, even combined with other types of techniques. 72.28% of the selected studies do not show evidence of estimation techniques being used in industry. Regarding the level of accuracy found, the study concludes that there is still a significant margin for improvement in this regard, which has not yet been resolved and may motivate new studies.

Keywords: estimate, accuracy, agile, ASD, SMS



ÍNDICE

1. Introducción.....	12
1.1 Conceptos generales.....	12
1.2 Marco teórico.....	13
1.3 Alcance.....	14
2 Trabajos relacionados.....	15
3. Método de investigación.....	18
3.1 Objetivo y preguntas de investigación.....	18
3.2 Preguntas de publicación.....	19
3.3 Estrategia de búsqueda y selección de estudios.....	20
3.4 Formularios de extracción de datos.....	24
4 Resultados y discusiones.....	25
4.1 PI1: ¿Qué metodologías o métodos son mencionados?.....	25
4.2 PI2: ¿Cuáles son las técnicas de estimación utilizadas en metodologías ágiles de software?.....	27
4.3 PI3: ¿Qué variables estiman las técnicas en metodologías ágiles?.....	29
4.4 PI4: ¿Qué tipos de técnicas fueron empleadas?.....	30
4.5 PI5: ¿Qué evidencia hay acerca de la precisión de las estimaciones en metodologías ágiles?.....	32
4.6 PI6: ¿Qué diferencia en la precisión de las estimaciones reportan las metodologías ágiles frente a las metodologías tradicionales?.....	35
4.7 PI7: ¿Cuáles de las técnicas halladas fue reportado su uso en la industria?.....	36
4.8 PI8: ¿Qué resultados fueron reportados del uso de estas técnicas en la industria?.....	40
4.9 PdP1: ¿Dónde han sido publicados los estudios?.....	43
4.10 PdP2: ¿En qué año fueron publicados?.....	44
4.11 PdP3: ¿Cuáles son los países más activos?.....	45
5. Amenazas a la validez.....	46
5.1 Validez descriptiva.....	46
5.2 Validez teórica.....	46
5.3 Generalización.....	47
5.4 Validez interpretativa.....	47
5.5 Repetibilidad.....	47
6. Conclusiones.....	48
6.1 Aspectos más relevantes resultantes del estudio.....	48
6.2 Futuras líneas de investigación.....	49
REFERENCIAS.....	50
APÉNDICE A. Listado de estudios primarios seleccionados.....	53



ÍNDICE DE FIGURAS

Figura 1. Proceso de búsqueda y selección	20
Figura 2. Resultados del proceso de selección de artículos	24



ÍNDICE DE TABLAS

Tabla 1. Preguntas de investigación	19
Tabla 2. Preguntas de publicación	20
Tabla 3. Cadenas de búsqueda	21
Tabla 4. Distribución de estudios	22
Tabla 5. Criterios de exclusión	23
Tabla 6. Criterios para resolver desacuerdos	23
Tabla 7. Formulario de extracción para las PI	24
Tabla 8. Formulario de extracción para las PdP	25
Tabla 9. Enfoques y métodos	26
Tabla 10. Enfoques y métodos	26
Tabla 11. Lista de técnicas de estimación	27
Tabla 12. Qué estiman las técnicas halladas	29
Tabla 13. Tipos de técnicas	30
Tabla 14. Evidencia sobre precisión en metodologías ágiles	33
Tabla 15. Diferencias de estimaciones entre ágil y tradicional	36
Tabla 16. Técnicas con reporte de uso en la industria	36
Tabla 17. Evidencia de beneficios de las principales técnicas de estimación usadas en la industria	41
Tabla 18. Evidencia de desventajas de las principales técnicas de estimación usadas en la industria	43
Tabla 19. Número de estudios por tipo de sede	44
Tabla 20. Cantidad de artículos por país	45



ÍNDICE DE GRÁFICOS

Gráfico 1. Tendencia de los tipos de técnicas de estimación	31
Gráfico 2. Años de publicación	45



NOMENCLATURA

AB: Adaptive Boosting

ABC-PSO: Artificial Bee Colony - Particle Swarm Optimization

AHN: Artificial Hydrocarbon Network

ANFIS: Adaptive Neuro-Fuzzy Inference System

ANFM: Adaptive Neuro-Fuzzy Modeling

ANN: Artificial Neural Network

ASD: Agile Software Development

ASP: Adjusted Story Points

BBN: Bayesian Belief Network

BC: Bayes Classifier

BN: Bayesian Network

BoW: Bag-of-Words

BRE: Balanced Relative Error

CAEA: Constructive Agile Estimation Algorithmic

CBR: Categorical Boosting Regressors

CCNN: Cascade-Correlation Neural Network

CEEM: Change Effort Estimation Model

CNN: Convolutional Neural Network

COCOMO: Constructive Cost Model

COSMIC FSM: COSMIC Functional Size Measurement

DBN-ALO: Dynamic Bayesian Network - Antlion Optimization

DBN: Dynamic Bayesian Network

DN: Deep Nets

DT: Decision Tree

ECS-DBN: Evolutionary Cost - Sensitive Deep Belief Network

EE: Expert Estimation

ELM: Ensemble-based Learning Method



ENN: Elman Neuronal Network

EPA: Ensemble Prediction Average

EVM: Earned Value Management

FFBP: Feed Forward Back Propagation

FLANN-WOA: Functional Link Artificial Neural Network - Whale Optimization Algorithm

FNN: Fuzzy Neuronal Network

FP: Function Points

GBA: Gradient Boosting Algorithms

GBT: Gradient Boosted Trees

GLM: Generalized Linear Models

GMDH-PNN: Group Method of Data Handling Polynomial Neural Network

GMDH: Group Method of Data Handling

GRNN: General Regression Neural Network

GS: Grid Search

HKO: Kosseim and Ormandjieva

HyEEASe: Hybrid Effort Estimation in Agile Software dEvelopment

IA: Intelligence Artificial

ICR: Independent Component Regression

IRDSS: Intelligent Recommender and Decision Support System

ISC: Individual Statistical Combination

KNN: K-Nearest Neighbors

LDBNM: Learning Dynamic Bayesian Network Model

LDRNN: Long-Deep Recurrent Neural Network

LMT: Logistic Model Tree

LOC+GA: Lines Of Code + Genetic Algorithm

LOC+NN: Lines Of Code + Neuronal Network

LR: Linear Regression

LSTM: Long Short-Term Memory

MAE: Mean Absolute Error o Promedio de Error Absolute.



MAR: Mean Absolute Residual

MBE: Mean Balanced Error

MBRE: Mean Balanced Relative Error

MBREbias: Mean Balanced Relative Error bias of -0.05

MdAE: Median Absolute Error

MdBRE: Median Balanced Relative Error

MdBREbias: Median Balanced Relative Error bias of -0.05

MdMER: Median Magnitude of Error Relative

MdMRE: Median Magnitude of Relative Error

Mean RE: Mean of Relative Error

MedianRE: Median Relative Error

MER: Magnitude of Error Relative

MES: Multiagent Estimation System

ML: Machine Learning

MLNN: Multi Layer Neural Network

MLP: Multi-Layer Perceptron

MLR: Multiple Linear Regression

MMAE: Mean Magnitude of Absolute Error

MMER: Mean Magnitude of Error Relative

MMRE: Mean Magnitude of Relative Error

MR: Multiple Regression,

MRE: Magnitude of Relative Error

MSE: Mean Square Error

NB: Naves Bayes

NPC: Artificial Neural Networks with a principal component

PA: Prediction Accuracy

PCA: Principal Component Analysis

PMI-RMP: Project Management Institute – Risk Management Professional

PNN: Probabilistic Neural Network



PP: Planning Poker

PRED(X): Esta métrica mide el porcentaje de valores estimados que están dentro del X% del valor real.

PREP: Post-Requirements Estimation Procedure

PSO: Particle Swarm Optimization

PV-DM: Distributed Memory Paragraph Vector

R²: R-squared

RAE: Relative Absolute Error

RBFN-WOA: Radial Basis Function - Whale Optimization Algorithm

RBFN: Radial Basis Function Network

RF: Random Forest

RG: Random Guessing

RMSE: Root Mean Square Error

RR: Ridge Regression

RRSE: Root Relative Square Error

SA: Standard Accuracy

SD: Standard Deviation

SDBRE: Standard Deviation Balanced Relative Error

SDBREbias: Standard Deviation Balanced Relative Error bias of -0.05

SEE: Standard Error

SGB: Stochastic Gradient Boosting

SIA: Swarm Intelligence Algorithms

SLR: Simple Linear Regression

SP: Story Points

SR: Single Regression

SVM: Support Vector Machine

TF-IDF: Term Frequency-Inverse Document Frequency

TI: Tecnología de la Información

UCP: Use Case Points



WBS: Work Breakdown Structure

XGB: Extreme Gradient Boosting

XP: Extreme Programming



1. Introducción

El desarrollo de software es una disciplina que ha experimentado una transformación significativa en las últimas décadas, pasando de metodologías tradicionales a enfoques más flexibles y colaborativos conocidos como metodologías ágiles. Este cambio en la forma en que se aborda el desarrollo de software ha llevado a una reevaluación de prácticas y procesos, incluida la estimación de proyectos. La precisión en las estimaciones es un factor crítico en la gestión exitosa de proyectos de desarrollo de software, independientemente de la metodología utilizada. Sin embargo, surge la pregunta: ¿cómo se comparan las estimaciones en proyectos bajo metodologías ágiles en términos de precisión?

A continuación se abordan algunos conceptos generales en la sección 1.1, un marco teórico en la sección 1.2 y se define el alcance del presente trabajo en la sección 1.3.

1.1 Conceptos generales

La estimación en el desarrollo de software es un proceso crítico que implica predecir la cantidad de recursos, tiempo y esfuerzo necesarios para completar un proyecto de manera exitosa. La precisión en estas estimaciones es esencial para una planificación adecuada y para evitar desviaciones significativas en el desarrollo que puedan afectar el presupuesto y los plazos. Para comprender la relevancia de la precisión en las estimaciones, es esencial aclarar algunos conceptos clave:

Estimaciones en desarrollo de software

Las estimaciones en el desarrollo de software se refieren a la predicción de los recursos necesarios para llevar a cabo un proyecto de desarrollo de software. Esto incluye estimar la duración del proyecto, los costos asociados, la cantidad de personal requerido y otros recursos necesarios. Las estimaciones pueden ser de diversos tipos, como estimaciones de esfuerzo, estimaciones de tiempo y estimaciones de costo. Estas estimaciones son esenciales en la planificación y gestión de proyectos de software.

Precisión de las estimaciones

La precisión de las estimaciones se refiere a qué tan cercanas están las estimaciones a los valores reales observados una vez que se completa un proyecto. Una estimación precisa es aquella que se acerca al resultado real con un margen de error mínimo. En el contexto de proyectos de desarrollo de software, una estimación precisa significa que las estimaciones de esfuerzo, tiempo y costo son consistentes con lo que realmente se necesita para completar el proyecto.

Importancia de la precisión

La precisión en las estimaciones es crucial debido a sus implicaciones en la gestión de proyectos. Una estimación inexacta puede llevar a una serie de problemas, como retrasos en la



entrega, superación del presupuesto y la asignación incorrecta de recursos. Por otro lado, una estimación precisa permite una planificación efectiva y una toma de decisiones informada. Esto es especialmente relevante en proyectos de desarrollo de software, donde los cambios en los requisitos y las circunstancias son comunes, lo que hace que la precisión en las estimaciones sea aún más desafiante y valiosa.

Contexto de metodologías ágiles

Las metodologías ágiles, como Scrum y Kanban, se caracterizan por su enfoque en la adaptación continua, la colaboración y la respuesta rápida a cambios en los requisitos. Este contexto agrega complejidad adicional a la estimación, ya que las estimaciones deben ser lo suficientemente flexibles como para acomodar cambios constantes.

En resumen, comprender estos conceptos relevantes es esencial para explorar la precisión en las estimaciones en proyectos de desarrollo de software bajo metodologías ágiles. La precisión de las estimaciones es un factor crítico en la gestión efectiva de proyectos y tiene implicaciones significativas en la forma en que se abordan los desafíos y cambios en el desarrollo de software.

1.2 Marco teórico

El marco teórico proporciona una base conceptual sólida que orienta la comprensión de los factores que influyen en la precisión de las estimaciones en proyectos de desarrollo de software bajo metodologías ágiles. En este contexto, es esencial explorar algunas teorías y conceptos clave que arrojan luz sobre este tema crítico.

Teoría de la estimación

La teoría de la estimación es un punto de partida fundamental. Esta teoría se basa en la idea de que las estimaciones son pronósticos que se basan en información disponible en un momento dado y que pueden estar sujetos a incertidumbre. La precisión de las estimaciones se ve afectada por la calidad y la cantidad de datos disponibles, así como por la capacidad del equipo para comprender y evaluar los riesgos y desafíos del proyecto. En el contexto ágil, donde los cambios son frecuentes, la teoría de la estimación debe considerar la adaptabilidad de las estimaciones a medida que evoluciona el proyecto.

Adaptabilidad

Las metodologías ágiles se centran en la adaptabilidad y la capacidad de respuesta a cambios en los requisitos del proyecto. En este sentido, la teoría de la adaptabilidad cobra relevancia. Esta teoría sugiere que las estimaciones en proyectos ágiles deben ser flexibles y ajustables a medida que se adquiere una mejor comprensión de los requisitos y las condiciones cambiantes. La adaptabilidad implica que las estimaciones iniciales pueden modificarse durante el ciclo de desarrollo del software para reflejar cambios reales en el proyecto.



Medición de la precisión

Diversas teorías y modelos se han desarrollado para evaluar la precisión de las estimaciones en proyectos de desarrollo de software. Estos modelos pueden incluir el cálculo de errores absolutos y cuadráticos, como el error absoluto medio (MAE) y el coeficiente de determinación (R^2). Los modelos de regresión son herramientas estadísticas que se utilizan para evaluar la relación entre las variables. En el contexto de la medición de la precisión, los modelos de regresión pueden ayudar a identificar patrones y tendencias en las estimaciones y sus desviaciones con respecto a los valores reales. Esto permite comprender mejor las fuentes de error y mejorar las futuras estimaciones.

Mejora continua

La mejora continua es un principio central en las metodologías ágiles. La teoría de la mejora continua sugiere que la precisión de las estimaciones puede mejorar con el tiempo a medida que el equipo adquiere experiencia y aprende de proyectos anteriores. Esto implica un ciclo de retroalimentación constante en el que las estimaciones se revisan y ajustan en función de las lecciones aprendidas.

En resumen, el marco teórico proporciona una base sólida para comprender la precisión de las estimaciones en proyectos de desarrollo de software bajo metodologías ágiles. La interacción de conceptos como la estimación, la adaptabilidad, la medición de la precisión y la mejora continua arroja luz sobre la complejidad de este tema y su importancia en la gestión efectiva de proyectos ágiles.

1.3 Alcance

Existen técnicas y herramientas para estimar el esfuerzo de un proyecto de software, tales como estimación de punto de función, estimación de punto de caso de uso, COCOMO y COCOMO II, estimación de comparación, PMI-RMP, entre otras. Se basan en la estimación de distintas variables [1] que, en el caso de las técnicas difusas, son cuantificadores lingüísticos de muchas o la mayoría de sus variables [2][3][4]. En muchos casos la opinión de los expertos es fundamental [5]. En cuanto al análisis de riesgos, la contribución de los expertos es crucial, especialmente en la definición de la jerarquía de riesgos y las probabilidades de ocurrencia.

Solo la cuarta parte de los proyectos de software tienen éxito en términos de finalización de acuerdo con lo planificado, estimado y especificado [6], a esto se lo conoce como error en la estimación. Al mismo tiempo, la naturaleza de los proyectos de TI crea una gran cantidad de riesgos [7]. Un gran porcentaje de proyectos que no alcanzan el éxito se debe a una incorrecta estimación del esfuerzo y a una mala o nula categorización del riesgo [8].

El interés en este trabajo es identificar y categorizar las técnicas y herramientas de estimación empleadas en proyectos de desarrollo de software utilizando metodologías ágiles. En este sentido, el objetivo principal es representar un estado de la aplicación de estas técnicas y herramientas haciendo hincapié en la evidencia existente sobre su precisión en la estimación del esfuerzo, entendida como la diferencia entre el esfuerzo estimado y el esfuerzo real.



A través de la Ingeniería de Software Basada en Evidencia (ISBE), se espera transformar la necesidad de información en una pregunta que pueda ser respondida, estableciendo la evidencia para responder esa pregunta y, por lo tanto, evaluando críticamente la evidencia para determinar su validez [9]. Kitchenham et al. afirman que ISBE tiene la intención de "proporcionar los medios por los cuales la mejor evidencia actual de la investigación se puede integrar con la experiencia práctica y los valores humanos en el proceso de toma de decisiones con respecto al desarrollo y mantenimiento de software" [9].

El presente trabajo constituye un Mapeo Sistemático de la Literatura (MSL) siguiendo las pautas de Petersen [10], usado para organizar los hallazgos en el campo de investigación, desde enero de 2001 hasta el 15 de agosto de 2023.

A pesar de que las prácticas ágiles ya venían siendo usadas en la industria del software durante el siglo XX en metodologías como Crystal Clear y DSDM, en el año 2001 fue que los profesionales que venían trabajando en la agilidad fueron convocados para escribir el manifiesto ágil [43], estableciendo así las bases que dieron lugar a la agilidad moderna. A su vez, al tratarse de un MSL exploratorio, el objetivo que se persigue es hallar la mayor cantidad de artículos relacionados existentes.

El resto del artículo está organizado de la siguiente manera: los trabajos relacionados se describen en la sección 2, el método de investigación se describe en la sección 3, las respuestas y conclusiones de las preguntas planteadas en la sección 4, la estrategia para mitigar las amenazas a la validez se presenta en la sección 5 y, finalmente, se ofrecen conclusiones en la sección 6.

2 Trabajos relacionados

Se encuentra una gran cantidad de trabajos relacionados en donde se hacen estudios sobre técnicas de estimación en distintas metodologías de desarrollo de software, algunos incluso escriben acerca de sus precisiones. Sin embargo, ninguno responde a las preguntas planteadas en este MSL. A pesar que tienen limitaciones en cuanto al alcance o a sus hallazgos, los trabajos que están más estrechamente relacionados a este estudio son los descritos a continuación.

Kumar et al. [18] realizan una Revisión Sistemática de la Literatura (RSL) sólo sobre las técnicas de aprendizaje automático utilizadas para la estimación de esfuerzo en proyectos ágiles. Después de comparar los valores MMRE de diferentes técnicas, han observado que DBN-ALO proporciona una mejor precisión en la estimación del esfuerzo en comparación con otras técnicas.

Bilgaiyan et al. [19] llevan a cabo una RSL exclusivamente sobre las llamadas técnicas de soft computing para estimación de costos y esfuerzo en proyectos de software ágiles, con el objetivo de brindar una descripción detallada y analítica, indicando el grado de precisión en cada caso.



Bingamawa y Kamalrudin [20] realizan una RSL sobre la estimación de costos de software en general, sin centrarse específicamente en agilidad, dividiendo las herramientas, técnicas y métodos hallados en algorítmicos, no algorítmicos e híbridos. No proporcionan mediciones sobre sus precisiones.

El trabajo de Vyas et al. [21] ofrece una revisión de las técnicas de estimación utilizadas también en ambas metodologías, tanto ágil como tradicional, categorizándolas en técnicas de estimación básicas, las técnicas que son técnicamente aplicables para la estimación de proyectos desarrollados utilizando metodología ágil y técnicas de estimación basadas en aprendizaje automático. Tampoco evidencia información respecto a sus niveles de precisión.

En esta otra ocasión, Bilgaiyan et al. [22] vuelven a presentar un estudio sistemático desde 2006 hasta 2015 sobre la estimación en el desarrollo ágil de software, pero restringida a costos. Haciendo una descripción de las métricas utilizadas para medir la precisión de las estimaciones de cada técnica hallada junto a sus valores reportados en sus hallazgos.

Usman et al. [23] llevan a cabo una RSL con el objetivo de proporcionar una visión detallada del estado del arte en el área de la estimación exclusivamente del esfuerzo en el ASD, analizando trabajos desde 2001 hasta 2013. Destacan que las técnicas de estimación subjetivas (EE, PP, etc.) son las utilizadas con mayor frecuencia, y que MMRE y MRE son las métricas más utilizadas para medir la precisión.

Por su parte, Dantas et al. [24] y Fernández-Diego et al. [25] realizaron una actualización de la RSL de Usman et al. [23] donde en el primero actualizaron el rango temporal desde 2014 hasta 2017, observando un aumento de evidencia de soluciones basadas en métodos de inteligencia artificial y aprendizaje automático, y en el segundo, con un rango desde 2014 hasta 2020, hallan que se mantiene la tendencia hacia el estudio de técnicas basadas en el uso intensivo de datos, con un número cada vez mayor de artículos mostrando valores de precisión aceptables, aunque muchos continuaron informando resultados inadecuados.

Sembhoo y Gobin-Rahimbux [26] presentan una revisión de trabajos limitándose sólo a aquellas que discuten el uso de modelos de aprendizaje profundo para la estimación del esfuerzo para Scrum. Descubrieron que deep-se se ha desarrollado específicamente para la estimación del esfuerzo. Además, se analizan otras técnicas de aprendizaje profundo que se han experimentado. Se identificaron una serie de métricas de rendimiento y también se comparó el rendimiento de los distintos modelos.

Arora y Chopra [27] llevaron a cabo una RSL para descubrir el nivel de precisión entre técnicas exclusivamente de aprendizaje automático, aprendizaje no automático y técnicas de estimación tradicionales, llegando a la conclusión de que las técnicas de aprendizaje automático superaron ampliamente al resto considerando la métrica MMRE.

Mahmood et al. [28] realizan una revisión sistemática de estudios desde 2000 hasta 2019 asociados con las mejores prácticas de casos de uso y técnicas de estimación del esfuerzo de desarrollo de software basadas sólo en el juicio de expertos. Hallaron que la técnica más usada en la industria del software para la estimación de esfuerzos es el juicio de expertos y que MRE, MMRE y PRED(25) son las métricas más utilizadas.



Canedo et al. [29] publicaron una RSL con el objetivo de identificar las métricas o métodos más utilizados en el desarrollo de software ágil y las métricas de tamaño más utilizadas con respecto a estimaciones de esfuerzo, plazos y costos en una planificación de proyecto de software ágil. Los resultados sugieren que PP es la técnica más popular para equipos ágiles en la fase de planificación, puntos de historia y puntos función son las métricas más utilizadas en proyectos ágiles para estimar tamaño, tiempo, esfuerzo, productividad y costo. No reflejan hallazgos sobre precisiones en las técnicas de estimación relevadas.

Durán et al. [30] llevan a cabo una RSL que identifica los atributos que influyen en la estimación de la complejidad en las historias de usuario, sin acercarse con detalles sobre sus precisiones.

A. Kaur y K. Kaur [31] exponen una RSL centrada en la estimación de esfuerzo sólo en la etapa de pruebas de desarrollo móvil. Mencionan las técnicas utilizadas y su porcentaje de precisión en términos de PA.

Alsaadi y Saeedi [32] realizan una RSL que busca presentar una descripción detallada de las técnicas de estimación en ASD pero limitándose a aquellas basadas en datos, mostrando precisiones de cada técnica relevada. Concluyen sin embargo que hay poca evidencia de trabajos en esta área.

Azzeh et al. [33] a través de una RSL muestran el estado del arte sobre técnicas de estimación de puntos de casos de uso solamente, analizando los artículos desde diferentes puntos de vista como precisión de la estimación, contexto de estimación favorable e impacto de técnicas combinadas en la precisión.

Mahmood et al. [34] en esta ocasión realizan una RSL para hallar la precisión de las estimaciones de software de técnicas de aprendizaje automático puntualmente. Es interesante el estudio comparativo que realiza de las técnicas en términos de su precisión con las métricas MRE, MMRE y PRED(25). Concluyen que la técnica con mejor precisión es ELM.

Sharma y Singh [35] llevan a cabo otra RSL para hallar las técnicas de estimación de software limitándose a las basadas en aprendizaje automático desde 2000 hasta 2017. No evidencian métricas sobre sus precisiones.

Otra RSL es el publicado por Salamea et al. [36] para evidenciar el estado del arte sobre la estimación y priorización de requisitos de calidad en el desarrollo de software, aunque sin registrar información relacionada con su precisión.

Carvalho et al. [37] hacen una RSL para recabar hallazgos sobre el uso de métricas en las estimaciones de esfuerzo y costos en software en el sector público de Brasil. La métrica más encontrada en los estudios fue el punto de historia. Además, concluyen que las métricas relacionadas con la complejidad (puntos de historia y velocidad) demostraron ser más adecuadas para recompensar el esfuerzo aplicado en la construcción de funcionalidades del software.

Altaleb y Gravell [38] presentan los resultados de una RSL sobre modelos de estimación de esfuerzo y tamaño en el desarrollo de aplicaciones móviles, haciendo foco en las utilizadas en ASD.



También encontramos en la literatura varios MSL sobre estimaciones y sus precisiones, como es el caso del elaborado por Carbonera et al. [39], donde los autores presentan una descripción detallada de los enfoques, brechas, desafíos y tendencias de las técnicas de estimación en el desarrollo de software, en el período que va desde enero del 2000 hasta diciembre del 2016. Entre sus resultados detectan que más del 70% de los estudios seleccionados adoptaron múltiples enfoques de estimación del esfuerzo. No reportan datos relacionados al nivel de predicción de cada técnica.

Fávero et al. [40] publican otro MSL donde presentan el estado del arte sobre técnicas de estimación del esfuerzo del software basadas en analogías, indicando desafíos y oportunidades de investigación. El mapeo se realizó para el período desde 2007 hasta 2017. Concluyen que el modelo de estimación por analogía ha recibido más atención y la presentan como una técnica prometedora y factible en relación con las demás. No se reportan hallazgos en cuanto al nivel de precisión en las estimaciones.

Najm et al. [41] llevan a cabo un MSL para evidenciar el estado del arte de las técnicas de estimación de software basadas en DT desde 1985 hasta 2017. Concluyen que la mayoría de los artículos tratan de mejorar los modelos DT existentes, mientras que pocos estudios han propuesto modelos novedosos con esta técnica para mejorar la confiabilidad de las estimaciones de software.

Por último, Klimczyk y Madeyski [42] realizan un MSL con el objetivo de identificar problemas de estimación y sus soluciones debido a deuda técnica previamente introducida en proyectos de software. En relación a la precisión, concluyen que su inexactitud es debido en parte a la medición técnica incorrecta de la deuda.

3. Método de investigación

El presente trabajo constituye un Mapeo Sistemático de la Literatura (MSL) siguiendo las pautas de Petersen [10], usado para organizar los hallazgos en el campo de investigación, desde enero de 2001 hasta el 15 de agosto de 2023.

Primeramente en la sección 3.1, se especificará el objetivo del trabajo junto a las preguntas de investigación definidas para lograr el mismo. Posteriormente en la sección 3.2 se detallan las preguntas de publicación del mapeo, en la sección 3.3 se describe la estrategia de búsqueda llevada a cabo, para finalizar en la sección 3.4 con la presentación de los formularios de extracción de datos confeccionados para plasmar los hallazgos.

3.1 Objetivo y preguntas de investigación

El objetivo es elaborar un estado del arte sobre las técnicas de estimación utilizadas tanto en el ámbito académico como en la industria, en metodologías ágiles de desarrollo de software, comparando además sus precisiones.



Para alcanzar este objetivo, se elaboran las preguntas de investigación (PI) descritas en la [Tabla 1](#), siguiendo las pautas definidas por Petersen.

Tabla 1. Preguntas de investigación

Referencia	Pregunta de Investigación	Descripción
PI1	¿Qué metodologías o métodos son mencionados?	Una lista de metodologías usadas, como Agile, Tradicional, Scrum, Kanban, etc.
PI2	¿Cuáles son las técnicas de estimación utilizadas en metodologías ágiles de software?	Una lista de técnicas de estimación usadas en metodologías ágiles como Scrum, Kanban, Scrumban, XP, etc.
PI3	¿Qué variables estiman las técnicas en metodologías ágiles?	Una lista de los aspectos que efectivamente son estimados, por ejemplo tamaño, esfuerzo, costo, tiempo, etc.
PI4	¿Qué tipos de técnicas fueron empleadas?	Una lista del tipo de técnicas utilizadas, por ejemplo de juicio, de cálculo o conteo, o por su uso en etapas tempranas o tardías del ciclo de vida del desarrollo de software
PI5	¿Qué evidencia hay acerca de la precisión de las estimaciones en metodologías ágiles?	Identificación de la evidencia de la precisión de las estimaciones en metodologías ágiles
PI6	¿Qué diferencia en la precisión de las estimaciones reportan las metodologías ágiles frente a las metodologías tradicionales?	Lista de las diferencias que son reportadas entre los dos tipos de metodologías
PI7	¿Cuáles de las técnicas halladas fue reportado su uso en la industria?	Lista de las técnicas de estimación cuyo uso fue reportado en la industria
PI8	¿Qué resultados fueron reportados del uso de estas técnicas en la industria?	Resultados disponibles del uso de estas técnicas en la industria

3.2 Preguntas de publicación

Se especifican también un conjunto de preguntas de publicación (PdP) para caracterizar la bibliografía y el espacio demográfico ([Tabla 2](#)).



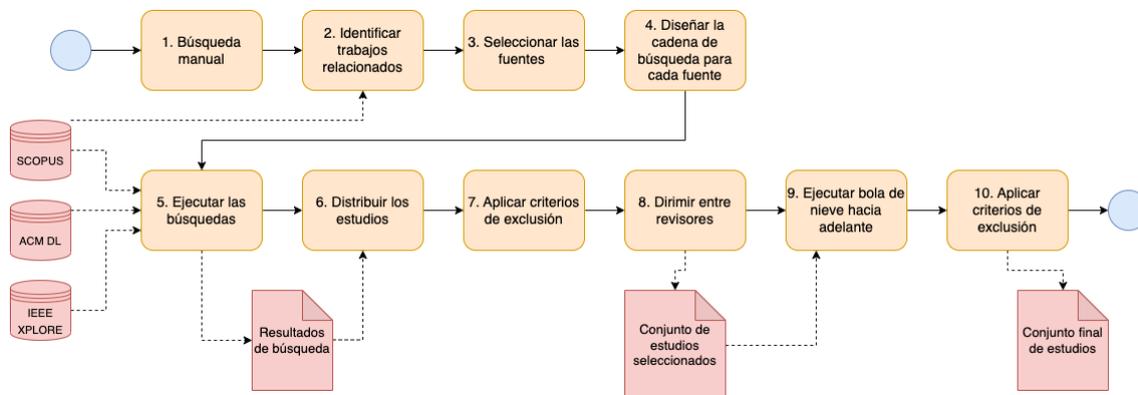
Tabla 2. Preguntas de publicación

Referencia	Pregunta de Publicación	Motivación/Descripción
PdP1	¿Dónde los estudios han sido publicados?	Conocer la distribución de los estudios por tipo de sede: conferencias, journals o workshops.
PdP2	¿En qué año fueron publicados?	Cantidad de publicaciones por año
PdP3	¿Cuáles son los países más activos?	Lista de países en donde fueron publicados

3.3 Estrategia de búsqueda y selección de estudios

La estrategia de búsqueda seleccionada incluye tres enfoques para buscar los estudios primarios. El primero es una búsqueda manual en Google Scholar de revisiones sistemáticas de literatura sobre estimaciones de software en metodologías ágiles. El segundo es una búsqueda automática realizada a través de las fuentes en línea de estudios científicos (bibliotecas digitales y bases de datos). Finalmente, se completa el conjunto de estudios utilizando la técnica de bola de nieve hacia adelante [11]. La [Figura 1](#) muestra esta estrategia.

Figura 1. Proceso de búsqueda y selección



1 Búsqueda manual

Se realiza una búsqueda manual en Google Scholar para obtener los trabajos similares al nuestro, según lo recomendado por Petersen et al. [12], ya que esto también ayuda a ajustar el enfoque del estudio.



2 Identificar trabajos relacionados

Se identifican los trabajos existentes sobre estimación de esfuerzo en metodologías ágiles. Dado que se lleva a cabo un MSL, un estudio secundario, solo otros trabajos secundarios publicados anteriormente se consideran trabajos relacionados.

3 Seleccionar las fuentes

Las bases de datos electrónicas de artículos científicos seleccionados para este estudio son Scopus, IEEE Xplore y la biblioteca digital ACM, ya que se citan repetidamente en los informes y directrices de MSL [13] [14] [15] [16].

4 Diseñar la cadena de búsqueda para cada fuente

Las cadenas de búsqueda que se utilizan para las tres bibliotecas mencionadas se indican en la [Tabla 3](#).

Tabla 3. Cadenas de búsqueda

Biblioteca	Cadena de búsqueda
Scopus	(TITLE-ABS-KEY (agil* OR ágil OR scrum OR kanban OR scrumban OR xo OR "extreme programming") AND TITLE-ABS-KEY (estim* OR predic*)) AND PUB YEAR > 1999 AND (LIMIT-TO(DOCTYPE,"ar")) AND (LIMIT-TO(SUBJAREA,"COMP "))
IEEE Xplore	("Index Terms":estim* OR "Index Terms":predic*) AND ("Index Terms":agil* OR "Index Terms":ágil OR "Index Terms":scrum OR "Index Terms":kanban OR "Index Terms":scrumban OR "Index Terms":xp OR "Index Terms":"extreme programming")
ACM DL	(Title:((agil* OR ágil OR scrum OR kanban OR scrumban OR xp OR "extreme programming")) AND Title:((estim* OR predic*))) OR (Keyword:((agil* OR ágil OR scrum OR kanban OR scrumban OR xp OR "extreme programming")) AND Keyword:((estim* OR predic*))) OR (Abstract:((agil* OR ágil OR scrum OR kanban OR scrumban OR xp OR "extreme programming")) AND Abstract:((estim* OR predic*)))



5 Ejecutar las búsquedas

Se ejecutan las búsquedas y recopilan los resultados. Estos resultados incluyen duplicados que fue necesario descartar aplicando las siguientes reglas:

- a. Estudios extendidos: conservar el último.
- b. Estudios duplicados: según la fuente, se sigue el siguiente orden: Scopus (ya que ofrece información más detallada), seguido de IEEE Xplore y, por último, ACM DL (porque no recupera los resúmenes) [17].

6 Distribuir los estudios

Los estudios recuperados son distribuidos entre revisores como se muestra en la [Tabla 4](#). Nótese que se asegura que cada trabajo sea examinado por dos revisores diferentes para reducir sesgos.

Tabla 4. Distribución de estudios

Revisor	Estudios		
	0%-33%	34%-67%	68%-100%
R1	X		
R2		X	
R3			X
R4	X	X	X

La selección individual de estudios llevada a cabo por cada revisor se incluye en un conjunto único de estudios.

7 Aplicar criterios de exclusión

Los revisores efectúan la revisión de forma independiente de los estudios que le fueron asignados y deciden si son relevantes o no, con solo leer el título, el resumen y aplicar los criterios de exclusión (CE). Este criterio se describe en la [Tabla 5](#).



Tabla 5. Criterios de exclusión

CE #	Descripción
CE1	Estudio corto (menos de 5 páginas)
CE2	No trata sobre estimaciones de software
CE3	No trata sobre estimaciones de software ágiles
CE4	Estudio no revisado por pares

8 Dirimir entre revisores

Las diferencias entre los revisores se resuelven utilizando los siguientes criterios descritos en la [Tabla 6](#) [10]:

Tabla 6. Criterios para resolver desacuerdos

		Revisor 1		
		Incluir	Incierto	Excluir
Revisor 2	Incluir	A	B	D
	Incierto	B	C	E
	Excluir	D	E	F

A & B: El estudio fue incluido.

E & F: El estudio fue excluido.

C & D: El estudio fue leído completamente y clasificado nuevamente obteniendo A, B, E o F.

9 Ejecutar bola de nieve hacia adelante

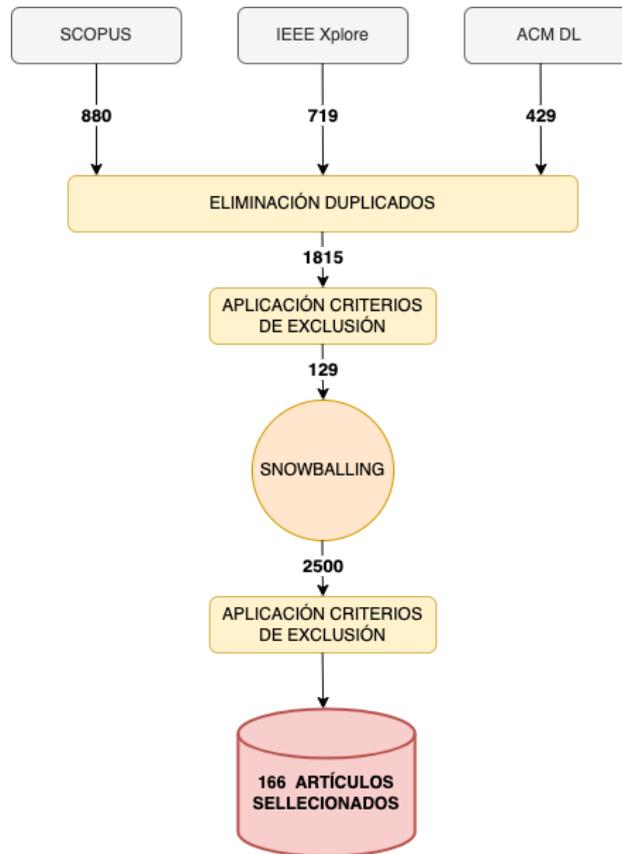
Los estudios resultantes son considerados como “artículos semilla” para ser utilizados en una técnica de bola de nieve hacia adelante, siguiendo las pautas propuestas por Wohlin [11]. La razón de ejecutar esta búsqueda complementaria tiene como objetivo enriquecer los resultados de la búsqueda automática.

10 Aplicar criterios de exclusión

Un experto aplica los criterios de exclusión a los nuevos artículos resultantes de la actividad anterior, por lo que se obtiene el conjunto definitivo de artículos, que se denominan “artículos seleccionados”.

En la [Figura 2](#), se detallan los resultados obtenidos del proceso.

Figura 2. Resultados del proceso de selección de artículos



3.4 Formularios de extracción de datos

Los datos relevantes se extraen del conjunto de estudios para responder a las ocho PI y las tres PdP. Los datos se almacenan en una hoja de cálculo con el formato que se muestra en la [Tabla 7](#) y en la [Tabla 8](#).

Tabla 7. Formulario de extracción para las PI

Estudio #ID	PI1	PI2	PI3	PI4	PI5	PI6	PI7	PI8
Estudio #1								
Estudio #2								
...
Estudio #n								
Valores aceptados	Nombres de metodologías	Nombre de técnicas	Aspectos estimados efectivamente	Nombre de técnicas	Resultados disponibles	Resultados disponibles	Nombre de técnicas	Resultados disponibles



Tabla 8. Formulario de extracción para las PdP

Estudio #ID	PdP1	PdP2	PdP3
Estudio #1			
Estudio #2			
...
Estudio #n			
Valores aceptados	Conference Journal Workshop	Año de publicación	Países

4 Resultados y discusiones

Luego de obtener los datos extraídos en los formularios de extracción mencionados, se describe mediante un análisis cada una de las preguntas de investigación y preguntas de publicación descritas. Todos los artículos seleccionados que son referenciados en cada pregunta de investigación, se encuentran referenciados en el apéndice I del presente trabajo.

Desde la sección 4.1 hasta la sección 4.11 se desarrolla el análisis de éstas preguntas.

4.1 PI1: ¿Qué metodologías o métodos son mencionados?

En esta pregunta de investigación, se pretende evidenciar las metodologías o métodos que los estudios alcanzan.

La lista de las metodologías y modelos hallados son los listados en [Tabla 9](#) y [Tabla 10](#).

El enfoque ágil es claramente el más discutido y ampliamente investigado en la literatura de desarrollo de software. SCRUM también tiene una presencia significativa en los artículos seleccionados, lo que indica su popularidad y relevancia en la industria, sumado a que se hallaron estudios en donde SCRUM se lo compara con otros enfoques ágiles.

Aunque las metodologías tradicionales tienen una presencia menor en comparación con las metodologías ágiles, aún hay interés en explorar y comprender los enfoques más tradicionales.

XP, KANBAN y TDD también tienen su espacio en la literatura, aunque con una presencia más modesta.

RUP, en comparación con otras metodologías, tiene la menor representación en los artículos seleccionados.



Tabla 9. Enfoques y métodos

ID	Metodología/Método	Cantidad	Artículos seleccionados
1	Ágil	125	S002, S003, S004, S005, S007, S008, S009, S012, S014, S015, S016, S017, S018, S019, S020, S021, S023, S024, S026, S030, S031, S033, S034, S035, S036, S038, S039, S040, S041, S042, S044, S046, S047, S048, S049, S050, S051, S052, S053, S054, S055, S056, S057, S058, S060, S062, S063, S064, S066, S067, S068, S069, S070, S071, S072, S073, S075, S076, S079, S080, S081, S082, S083, S086, S087, S088, S089, S090, S091, S092, S093, S095, S096, S097, S099, S101, S103, S105, S107, S110, S112, S114, S115, S116, S117, S120, S121, S122, S124, S125, S126, S127, S128, S129, S131, S132, S133, S134, S135, S136, S137, S138, S139, S140, S141, S142, S143, S144, S147, S148, S150, S151, S152, S153, S155, S156, S157, S159, S160, S161, S162, S163, S164, S165, S166
2	SCRUM	36	S001, S006, S010, S011, S013, S025, S028, S029, S032, S037, S043, S045, S061, S065, S074, S077, S078, S085, S094, S098, S100, S102, S104, S106, S108, S109, S111, S113, S118, S119, S130, S145, S146, S149, S154, 158
3	Tradicional	8	S003, S016, S040, S056, S062, S073, S082, S129
4	XP	5	S022, S059, S078, S084, S098
5	KANBAN	3	S027, S037, S123
6	TDD	1	S098
7	RUP	1	S001

Tabla 10. Enfoques y métodos

ID	Enfoque/Método	Cantidad	Artículos seleccionados
1	Sólo Ágil	117	S002, S004, S005, S007, S008, S009, S012, S014, S015, S017, S018, S019, S020, S021, S023, S024, S026, S030, S031, S033, S034, S035, S036, S038, S039, S041, S042, S044, S046, S047, S048, S049, S050, S051, S052, S053, S054, S055, S057, S058, S060, S063, S064, S066, S067, S068, S069, S070, S071, S072, S075, S076, S079, S080, S081, S083, S086, S087, S088, S089, S090, S091, S092, S093, S095, S096, S097, S099, S101, S103, S105, S107, S110, S112, S114, S115, S116, S117, S120, S121, S122, S124, S125, S126, S127, S128, S131, S132, S133, S134, S135, S136, S137, S138, S139, S140, S141, S142, S143, S144, S147, S148, S150, S151, S152, S153, S155, S156, S157, S159, S160, S161, S162, S163, S164, S165, S166
2	Sólo SCRUM	32	S006, S010, S011, S013, S025, S028, S029, S032, S043, S045, S061, S065, S074, S077, S085, S094, S100, S102, S104, S106, S108, S109, S111, S113, S118, S119, S130, S145, S146, S149, S154, 158
3	Ágil y Tradicional	8	S003, S016, S040, S056, S062, S073, S082, S129
4	Sólo XP	3	S022, S059, S084
5	Sólo KANBAN	2	S027, S123
6	SCRUM y KANBAN	1	S037
7	SCRUM, TDD y XP	1	S098
8	SCRUM y XP	1	S078
9	SCRUM y RUP	1	S001



4.2 PI2: ¿Cuáles son las técnicas de estimación utilizadas en metodologías ágiles de software?

El objetivo de esta pregunta de investigación es listar las técnicas utilizadas en los trabajos hallados.

La [Tabla 11](#) detalla la lista de técnicas utilizadas en los estudios seleccionados.

Tabla 11. Lista de técnicas de estimación

Tipo de técnica	Técnicas	Cantidad técnicas	Artículos seleccionados
Machine Learning	AB, ANFIS, ANFM, Agile Framework for Small Projects + Neural Networks, ABC-PSO, ABC, AHN, ANN, NPC, ASE based on Regression, Autoencoders, BoW, BoW + RF, BC, BBN, BN, CNN, CBR, DT, DN, Deep-SE, DBN-ALO, DBN, ENN, ELM, EPA, ERT, ECS-DBN, Extra Trees, XGB, FFBP, Feedforward and Feed Backward Approach-Based Estimation Model, FLANN-WOA, Fuzzy Logic, FNN, GRNN, GMDH-PNN, GBA, GMDH, Influence diagram, IRDSS, J48, KNN, HKO, LDA-based Hierarchical Clustering for Story point Estimation types & components, LDA-based Hierarchical Clustering for Story point Estimation, LDNM, LOC+GA, LOC+NN, LMT, LSTM + RF, LDRNN, MLNN, MLP, MES, NB, PSO, PNN, RBFN + WOA, RBFN, RF, Stacking, SGB, SVM, SVM + RBF, SVM + RBF + GS, SVR, SVR + Polynomial, SVR + RBFN, SIA, ZeroR, Own Method (S144, S151), TF-IDF-SVM, GPT2SP	76	S001, S002, S003, S004, S005, S006, S007, S008, S009, S010, S011, S012, S013, S014, S015, S016, S017, S018, S019, S020, S021, S022, S023, S024, S025, S026, S027, S028, S029, S030, S031, S032, S033, S034, S034, S035, S036, S037, S038, S039, S040, S041, S042, S043, S044, S045, S046, S047, S048, S049, S050, S051, S052, S053, S054, S055, S056, S057, S058, S059, S060, S061, S062, S063, S064, S065, S066, S067, S068, S069, S070, S071, S143, S144, S148, S151, S155, S157, S160, S161, S164
Estimación de Expertos	ASP, Case-based reasoning, Control Group, Crowd Estimate, EE Pura, EE + Blitz Planning, EE + Planning Game, EE using HyEEASe, Knowledge4Scrum, Manual PP [baseline], Normalized Sprint Estimation, Pair-Estimation, Pair-Estimation + Incomplete Cycle Designs, PP, PP [Averaging], PP [Consensus], PP + Checklist, Project Size Unit, Protection Poker, Risk Poker, Sprint Points, SP, SP + WBS, Technical Service Unit, Unstructured Group, UCP Test Effort Estimation Model, UCP, Wideband Delphi	28	S001, S002, S006, S008, S009, S010, S011, S013, S017, S018, S019, S020, S022, S024, S025, S026, S028, S029, S030, S031, S033, S036, S038, S039, S040, S042, S044, S044, S045, S047, S048, S050, S052, S054, S055, S056, S063, S064, S065, S067, S068, S070, S074, S075, S077, S078, S080, S081, S082, S085, S086, S088, S089, S094, S096, S097, S098, S099, S100, S103, S110, S111, S113, S118, S119, S121, S124, S125, S126, S127, S129, S130, S131, S137, S138, S141, S142, S147, S152, S153, S162, S163
Estadística	Average Productivity, Estimation by Analogy, Generalized Estimation Method, GLM, ICR, ISC, LR, Logistic Regression, Mean, Median, Monte Carlo, MR, RR, SR, Step-wise LR, Zia et al.'s regression, LOC, RG	18	S004, S007, S010, S012, S014, S017, S019, S024, S038, S049, S051, S052, S054, S056, S059, S073, S083, S093, S108, S111, S112, S136, S155, S159, S161, S164
Estimación de Expertos + Machine Learning	EE + IA, J48 + PP, LMT + PP, NB + PP, PP using HyEEASe with GBT, PP using HyEEASe, RF + PP, SP + ML, SP + Generic Algorithm, SP + LDRNN, SP + Neural Network, SP + RNN + CNN, SP Enhanced, TF-IDF + SP, UCP + MLNN, EE-SP + KNN	16	S006, S042, S053, S069, S101, S105, S107, S109, S110, S112, S114, S156



Tipo de técnica	Técnicas	Cantidad técnicas	Artículos seleccionados
Estimación de Expertos + Algorítmica	COCOMO, COCOMO II, CAEA, FP, FP Analysis for Software Development Phase, Hybridized CEEM, Simplified FP, SP + FP, UCP + COCOMO, PREP, Own method S145-S149-S150-S154-S165	15	S019, S030, S050, S052, S054, S056, S071, S072, S082, S097, S117, S118, S122, S125, S134, S139, S145, S149, S150, S154, S158, S165
Algorítmica	CEEM, COSMIC FSM, EVM, HyEEASE, Own Algorithmic Method, Software Maintenance Effort, Estimation Model, System Design Instability, Event Points, Own method S146, PCA-based	10	S015, S016, S018, S021, S034, S041, S046, S060, S062, S076, S078, S079, S082, S087, S095, S096, S104, S106, S133, S146, S159, S166
Algorítmica + Estadística	COSMIC FSM + SLR, COSMIC FSM + MLR	2	S128
Estimación de Expertos + Algorítmica + Machine Learning	COCOMO II & Fuzzy Logic	1	S023

Un 56% de las técnicas encontradas, utilizan aprendizaje automático ya sea de forma independiente como combinado con otras. Se identifican 76 técnicas de aprendizaje automático diferentes, que van desde algoritmos clásicos hasta enfoques más avanzados como redes neuronales y algoritmos de ensamblaje (stacking, boosting). Estas técnicas se utilizan para la estimación de diferentes aspectos en el desarrollo de software. Los artículos seleccionados abarcan un amplio rango de enfoques, lo que refleja el creciente interés en la aplicación de aprendizaje automático en la estimación de proyectos de software. Las técnicas halladas con más frecuencia son DT, GRNN, GBA, KNN, MLP, NB, RF, SGB, SVM y SVR. A su vez, se evidencia un aumento del estudio e interés de la técnica deep-se (técnica de aprendizaje automático de última generación existente) en los estudios de los últimos años.

Hay un 16.86% de técnicas basadas en estimación de expertos, que incluyen métodos como pair-estimation, protection poker y wideband delphi. Estas técnicas dependen del juicio y conocimiento de expertos en el campo para realizar estimaciones. La variabilidad en las técnicas muestra que hay una diversidad de métodos para abordar la estimación basada en la experiencia.

Las técnicas estadísticas para la estimación en el desarrollo de software incluyen desde métodos simples como media y mediana, hasta enfoques más avanzados como regresión logística y métodos generalizados de estimación. Las técnicas estadísticas son fundamentales para analizar datos históricos y realizar predicciones basadas en patrones pasados.

En resumen, esta tabla demuestra la diversidad y evolución en las técnicas de estimación utilizadas en el desarrollo de software. Desde el uso de algoritmos clásicos hasta la aplicación de enfoques de aprendizaje automático, se observa un intento de combinar y aprovechar diferentes paradigmas para lograr estimaciones más precisas y confiables.



4.3 PI3: ¿Qué variables estiman las técnicas en metodologías ágiles?

Con esta pregunta se busca evidenciar qué variables estiman las técnicas utilizadas en los estudios seleccionados. La [Tabla 12](#) muestra este detalle.

Tabla 12. Qué estiman las técnicas halladas

ID	Variables	Cantidad	Artículos seleccionados
1	Esfuerzo	104	S002, S009, S011, S012, S013, S014, S015, S016, S021, S022, S026, S029, S030, S035, S038, S045, S048, S053, S054, S056, S057, S058, S059, S060, S062, S063, S067, S068, S069, S070, S071, S072, S074, S078, S079, S081, S082, S083, S086, S088, S091, S092, S093, S094, S096, S097, S098, S100, S101, S102, S105, S106, S107, S108, S110, S111, S112, S113, S114, S115, S116, S117, S118, S119, S120, S121, S122, S123, S124, S125, S126, S127, S128, S129, S130, S131, S132, S134, S135, S136, S137, S138, S139, S141, S144, S145, S147, S148, S151, S152, S153, S154, S155, S156, S157, S158, S159, S160, S161, S162, S163, S164, S165, S166
2	Costo	52	S001, S003, S005, S006, S007, S008, S009, S011, S015, S017, S018, S019, S023, S024, S025, S028, S031, S032, S033, S037, S039, S040, S042, S043, S044, S046, S049, S050, S051, S052, S055, S060, S061, S064, S066, S067, S073, S077, S085, S086, S091, S095, S109, S133, S134, S140, S145, S149, S150, S162, S165, S166
3	Tamaño	18	S004, S019, S020, S034, S041, S047, S049, S050, S052, S076, S080, S087, S089, S096, S099, S101, S103, S104
4	Tiempo	18	S009, S011, S013, S015, S020, S027, S028, S060, S062, S075, S117, S132, S133, S134, S140, S142, S143, S146
5	Riesgo	3	S036, S048, S119
6	Velocidad	2	S009, S084
7	Mantenibilidad	1	S090
8	Productividad	1	S065
9	Cobertura de pruebas	1	S010

El 62.65% de las técnicas encontradas en la literatura, estiman el esfuerzo de un proyecto de software, sin embargo se encuentra que frecuentemente además del esfuerzo, se suelen estimar otros componentes como el costo financiero o el tamaño del proyecto.

Se evidencia que varios artículos que estudian la estimación de costos, también estiman el tiempo. Esto demuestra la fuerte relación entre estos dos aspectos de un proyecto. Las precisiones halladas en estos artículos siempre demuestran ser ligeramente diferentes.



Aspectos como la velocidad y la productividad, a pesar que se encontraron artículos que los estiman específicamente, pueden ser deducidos a partir de otros como el tamaño y el tiempo de un proyecto.

Hay áreas menos estudiadas como la mantenibilidad y la cobertura de pruebas, que podrían representar oportunidades de investigación futura en este campo.

4.4 PI4: ¿Qué tipos de técnicas fueron empleadas?

Se pretende dejar plasmadas con esta pregunta de investigación, qué tipos de técnicas son consideradas en los estudios seleccionados.

La lista de los tipos de técnicas utilizadas son las listadas en la [Tabla 13](#).

Tabla 13. Tipos de técnicas

ID	Tipo de técnica	Cantidad	Artículos seleccionados
1	Estimación de Expertos	81	S001, S002, S006, S008, S009, S010, S011, S013, S017, S018, S019, S020, S022, S024, S025, S026, S028, S029, S030, S031, S033, S036, S038, S039, S040, S042, S044, S045, S047, S048, S050, S052, S054, S055, S056, S063, S064, S065, S067, S068, S070, S074, S075, S077, S078, S080, S081, S082, S085, S086, S088, S089, S094, S096, S097, S098, S099, S100, S103, S110, S111, S113, S118, S119, S121, S124, S125, S126, S127, S129, S130, S131, S137, S138, S141, S142, S147, S152, S153, S155, S163
2	Machine Learning	50	S003, S004, S005, S007, S012, S024, S025, S027, S028, S032, S034, S035, S037, S042, S043, S051, S057, S058, S059, S061, S066, S073, S077, S084, S090, S091, S092, S093, S094, S100, S102, S107, S108, S112, S115, S116, S120, S123, S132, S135, S140, S143, S144, S148, S151, S155, S157, S160, S161, S164
3	Estadística	26	S004, S007, S010, S012, S014, S017, S019, S024, S038, S049, S051, S052, S054, S056, S059, S073, S083, S093, S108, S111, S112, S136, S148, S159, S161, S164
4	Algorítmica	24	S015, S016, S018, S021, S034, S041, S046, S060, S062, S076, S078, S079, S082, S087, S094, S095, S096, S104, S106, S133, S146, S159, S162, S166
5	Estimación de Expertos + Algorítmica	22	S019, S030, S050, S052, S054, S056, S071, S072, S082, S097, S117, S118, S122, S125, S134, S139, S145, S149, S150, S154, S158, S165
6	Estimación de Expertos + Machine Learning	12	S006, S042, S053, S069, S101, S105, S107, S109, S110, S112, S114, S156
7	Algorítmica + Estadística	1	S128
8	Estimación de Expertos + Algorítmica + Machine Learning	1	S023



Esta PI revela tendencias significativas en las técnicas de estimación utilizadas en el desarrollo de software. Estas tendencias reflejan la diversidad y evolución en la forma en que los profesionales y los investigadores abordan el desafío de predecir y planificar los recursos en proyectos de software. Se identifican las siguientes observaciones clave:

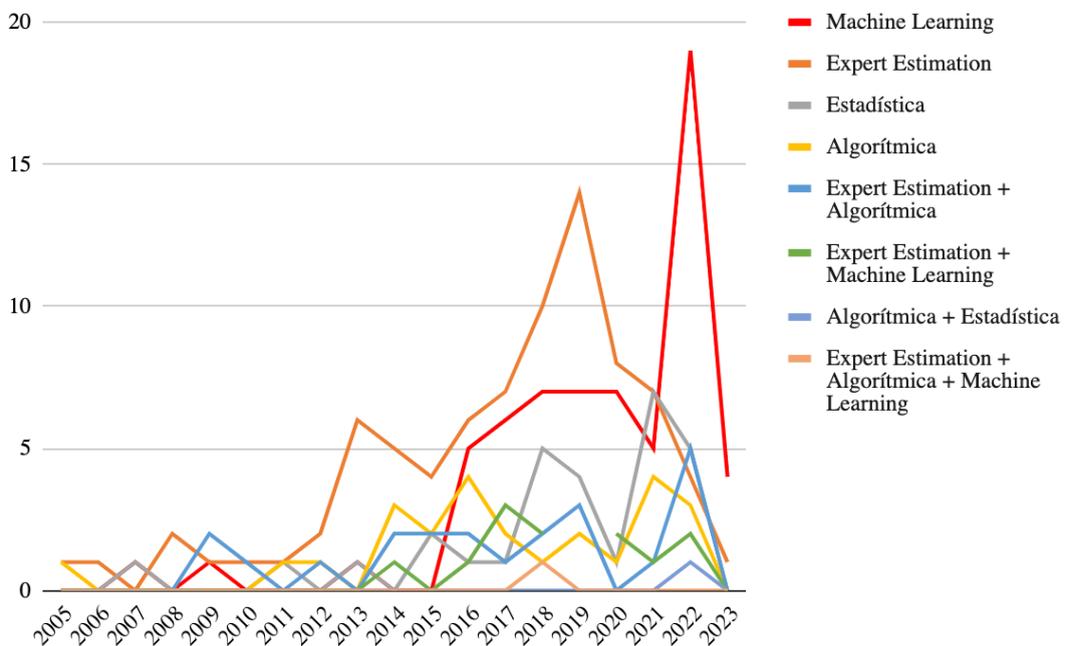
Prevalencia de Estimación de Expertos

Las basadas en el juicio de expertos son las técnicas más predominantes en el campo de desarrollo de software. Estas técnicas representan un 37.32% del total de técnicas encontradas, sin embargo, la estimación de expertos suele utilizarse en combinación con otras técnicas, frecuentemente con algorítmicas o de aprendizaje automático. Esto sugiere que a pesar de los avances tecnológicos, la experiencia humana sigue siendo un componente crítico en la estimación de proyectos de software. Los métodos de estimación de expertos son utilizados en una amplia gama de contextos, lo que subraya su relevancia continua.

Creciente Adopción de Aprendizaje Automático

Como se visualiza en el [Gráfico 1](#), los artículos dedicados al aprendizaje automático presentan estas técnicas como una fuerza importante en la estimación de software. La aplicación de algoritmos de aprendizaje automático sugiere una tendencia hacia la utilización de enfoques automatizados y datos históricos para mejorar la precisión de la estimación. Esto también refleja la creciente disponibilidad de datos en la industria del desarrollo de software.

Gráfico 1. Tendencia de los tipos de técnicas de estimación





Balance entre Estadísticas y Algoritmos

Las técnicas estadísticas (26 instancias) y las técnicas algorítmicas (24 instancias) demuestran un equilibrio entre el enfoque analítico y el enfoque basado en procedimientos. Esto sugiere que los profesionales están considerando tanto la información histórica como los modelos metodológicos para realizar estimaciones efectivas.

Convergencia de Enfoques

Se observa una convergencia entre diferentes categorías de técnicas. Por ejemplo, hay 22 instancias de "Estimación de Expertos + Algorítmica", indicando una búsqueda de precisión al combinar la intuición humana con los procedimientos algorítmicos. Del mismo modo, hay 12 instancias de "Estimación de Expertos + Machine Learning", mostrando la combinación de la experiencia humana con la capacidad de aprendizaje automático.

Enfoques Híbridos y Avanzados

Se identifican varias instancias de técnicas que combinan enfoques múltiples, como "Estimación de Expertos + Algorítmica + Machine Learning" y "Algorítmica + Estadística". Estos enfoques híbridos reflejan el deseo de aprovechar las fortalezas de diferentes técnicas para lograr estimaciones más precisas y confiables.

Necesidad de Flexibilidad

La presencia de diversas técnicas sugiere que no hay un enfoque único y universal para la estimación de software. La elección de la técnica depende del contexto, los datos disponibles y los objetivos del proyecto. Las investigaciones evidenciaron que existe una flexibilidad y consideración de múltiples enfoques para abordar las complejidades de la estimación.

En resumen, la estimación en el desarrollo de software es un campo diverso y en evolución, donde las técnicas tradicionales, algorítmicas, estadísticas y de aprendizaje automático están convergiendo para mejorar la precisión y la eficacia en la planificación y ejecución de proyectos. La combinación de la experiencia humana y el poder computacional está definiendo el futuro de la estimación en la industria.

4.5 PI5: ¿Qué evidencia hay acerca de la precisión de las estimaciones en metodologías ágiles?

Se muestra en la [Tabla 14](#) la evidencia hallada sobre la precisión en las estimaciones en metodologías ágiles.



Tabla 14. Evidencia sobre precisión en metodologías ágiles

ID	Métrica	Valor aceptable	Rango/Valor hallado	Cantidad	Artículos seleccionados
1	Ninguna	N/A	N/A	78	S005, S009, S011, S013, S014, S015, S018, S021, S026, S027, S029, S030, S031, S033, S036, S039, S040, S041, S045, S047, S053, S054, S056, S057, S060, S063, S064, S065, S068, S069, S070, S071, S072, S074, S076, S079, S085, S086, S087, S088, S090, S091, S095, S096, S097, S098, S099, S100, S104, S105, S106, S111, S113, S117, S121, S123, S124, S125, S126, S129, S130, S131, S134, S135, S136, S138, S141, S146, S149, S150, S155, S160, S162, S165
2	MMRE	<25%	2.6% - 204.4%	37	S003, S007, S016, S017, S022, S023, S028, S035, S042, S043, S046, S049, S050, S051, S061, S062, S066, S077, S078, S083, S084, S089, S092, S093, S107, S110, S116, S118, S120, S128, S132, S137, S140, S147, S156, S159, S166
3	PRED(25)	>= 75%	37% - 99.8%	23	S003, S006, S016, S023, S025, S028, S035, S043, S050, S051, S061, S077, S078, S083, S089, S092, S093, S116, S128, S137, S140, S144, S147
4	PA	cercano a 100%	24% - 100%	16	S012, S032, S034, S037, S048, S052, S067, S078, S102, S109, S119, S143, S154, S156, S157, S158
5	MAE	cercano a cero	0.2 - 21.494	13	S004, S006, S024, S025, S035, S075, S080, S092, S094, S108, S112, S114, S164
6	MRE	cercano a 0	9.5% - 284%	12	S001, S022, S050, S059, S080, S081, S082, S115, S122, S133, S139, S143
7	R ²	cercano a 1	0.054 - 0.9913	12	S007, S012, S038, S046, S062, S092, S107, S114, S143, S145, S156, S159
8	RMSE	valores más bajos	0.065 - 0.81388	11	S012, S024, S035, S051, S073, S075, S083, S094, S108, S143, S156
9	MdMRE	cercano a 0	3.33% - 140%	10	S003, S007, S022, S050, S061, S093, S116, S118, S128, S159
10	SA	>= 70%	0% - 98%	6	S003, S004, S103, S112, S153, S164
11	MMER	valores más bajos	0.0564 - 11.77	5	S006, S025, S061, S093, S140
12	MMAE	valores más bajos	1.744 - 3.81	4	S092, S103, S148, S151
13	MSE	valores más bajos	0.0059 - 0.042	4	S012, S051, S092, S107
14	MdBREbias	cercano a 0	-0.1169 - 0.05	4	S044, S055, S127, S142
15	MBREbias	0 - 0.49	0.01 - 0.33	3	S044, S055, S127
16	MBRE	cercano a 0	0.5051 - 1.26	3	S034, S055, S127
17	Pearson Correlation	> 0.5	0.0515 - 0.832	3	S002, S008, S152
18	PRED(50)	variable	70.02% - 95.23%	3	S061, S116, S128



ID	Métrica	Valor aceptable	Rango/Valor hallado	Cantidad	Artículos seleccionados
19	MdAE	cercano a 0	0.78 - 3.51	3	S112, S161, S164
20	MdBRE	cercano a 0	0.33 - 0.69	2	S055, S127
21	MdMER	cercano a 0	0.0344 - 0.2033	2	S061, S093
22	PRED(8)	variable	57.14% - 66.67%	2	S007, S093
23	PRED(100)	variable	90.47% - 95.23%	2	S061, S116
24	PRED(75)	variable	80.95% - 95.23%	2	S061, S116
25	SEE	cercano a 0	49.7% - 127%	2	S046, S062
26	BRE	cercano a 0	0.2154 - 0.5	2	S010, S142
27	MER	cercano a 0	0.25 - 8.95	2	S101, S144
28	Mean	cercano al real	0.75 - 13.265	2	S058, S144
29	SD	cercano a 0	0.04 - 2,9465	2	S058, S144
30	Spearman Correlation Coefficient	cercano a 1	0.335 - 0.6395	2	S038, S152
31	Explained Variance	>= 60%	96.40% - 99.53%	2	S143, S156
32	PRED(30)	>= 75%	31% - 97.5%	2	S049, S159
33	Average Variance	cercano a 0	3.8 - 18	2	S110, S163
34	Correlation Coefficient	cercano a 1	0.974 - 0.9961	1	S108
35	Coefficient of Variation	cercano a 0	0.22	1	S046
36	MAR	cercano a 0	3.12 - 4	1	S007
37	MBE	cercano a 0	2.82 - 5.34	1	S024
38	Median	cercano al real	0.329	1	S103
39	MedianRE	cercano a 0	0	1	S022
40	Overestimated	cercano a real	25.48%	1	S019
41	PRED(10)	variable	100%	1	S035
42	RAE	cercano a 0	9.71	1	S035
43	RRSE	cercano a 0	17.81%	1	S035
44	Scalar Variability	valores más bajos	90.85 - 139.91	1	S020
45	SDBRE	valores más bajos	1.48	1	S055
46	SDBREbias	valores más bajos	1.93	1	S055
47	Underestimated	cercano al real	54.90%	1	S019
48	R ² Adjusted	valores más altos	0,0405	1	S145
49	Kendall	cercano a 1	0.275	1	S152
50	Mean RE (Mean of Relative Error)	cercano a 0	-0.08 - -0.26	1	S022



Es revelador el hecho de que el 46.98% de los artículos hallados que tratan el tema de la estimación, no revelan mediciones sobre el grado de precisión que las técnicas poseen.

Por su parte, es importante aclarar que los valores aceptables en la tabla son a modo orientativo, ya que para conocer si un nivel de precisión es aceptable o no, es importante considerar el contexto del proyecto y las expectativas del problema en particular, y a menudo se compara con otros modelos o métricas para determinar su calidad y aceptabilidad.

Algunos estudios (S028, S133, S140, S143, S145, S156) estimaron costo, tiempo o esfuerzo por separado. Con fines prácticos, se realiza el promedio de todos para cada métrica en la tabla.

El único caso donde se reporta una precisión del 100% es en el artículo S157 con el uso de la técnica de SVM, sin embargo es evidente que lograr tal precisión es sólo circunstancial y que no se reporta este grado en el resto de los estudios analizados.

Se halla que las precisiones reportadas están fuertemente correlacionadas a las características de los proyectos donde se aplicaron, ya que se evidenciaron casos donde las mismas técnicas reportaban buenas precisiones en un estudio, y valores inaceptables en otro. A su vez, los márgenes de error hallados dejan en evidencia un nivel de precisión que requiere ser mejorado para lograr un grado aceptable en todos los proyectos de ASD.

Por último, se mantiene la tendencia hallada por Usman et al. [23] de usar la métrica MMRE para medir la precisión de las estimaciones, aunque MRE quedó un poco más relegada a favor de PRED(25) que ganó más aceptación.

4.6 PI6: ¿Qué diferencia en la precisión de las estimaciones reportan las metodologías ágiles frente a las metodologías tradicionales?

Se hallan las diferencias indicadas en la [Tabla 15](#).

No se reportan en los 161 estudios restantes, testimonios sobre diferencias halladas en las precisiones de estimaciones entre metodología ágil y tradicional.

El nivel de precisión hallado es equilibrado tanto en proyectos ágiles como en tradicionales, donde hay técnicas como COCOMO II que arroja una mejor estimación para proyectos tradicionales, pero técnicas basadas en aprendizaje automático en la mayoría de los casos parecen predecir mejor en proyectos ágiles.

La metodología usada en el proyecto parece no estar fuertemente correlacionada al nivel de precisión que ofrece la técnica de estimación de experto.



Tabla 15. Diferencias de estimaciones entre ágil y tradicional

ID	Artículo	Técnica	Ágil	Tradicional
1	S001	Expert Estimation	MRE=9.75%	MRE=11.8%
2	S003	Dynamic Bayesian Network	MMRE=16% SA=0.94 MdMRE=0.14 PRED(25)=93%	MMRE=17% SA=0.93 MdMRE=0.17 PRED(25)=92%
3	S003	Dynamic Bayesian Network - Antlion Optimization	MMRE=0.06 SA=0.98 MdMRE=0.05 PRED(25)=96%	MMRE=0.04 SA=0.98 MdMRE=0.03 PRED(25)=95%
4	S016	Change Effort Estimation Model	MMRE=22% PRED(25)=73.13%	MMRE=18% PRED(25)=88.24%
5	S082	Change Effort Estimation Model	MRE=22.15%	MRE=18,29%
6	S082	COCOMO II	MRE=29.17%	MRE=20,84%
7	S107	LOC + Neural Networks	MMRE=6.220651 MSE=17.03561 R ² =0.973897	MMRE=19.886671 MSE=0.600919 R ² =0.944562
8	S107	LOC + Genetic Algorithm	MMRE=6.742859 MSE=21.46326 R ² =0.967112	MMRE=24.03720 MSE=2.185249 R ² =0.798399

4.7 PI7: ¿Cuáles de las técnicas halladas fue reportado su uso en la industria?

Se evidencia el uso de las siguientes técnicas en la industria, listadas en [Tabla 16](#).

Tabla 16. Técnicas con reporte de uso en la industria

Técnica	Tipo	Cantidad artículos	Artículos seleccionados
Ninguna	N/A	120	S003, S004, S005, S007, S009, S010, S011, S013, S014, S015, S017, S018, S019, S021, S023, S024, S027, S029, S030, S031, S032, S033, S035, S036, S037, S039, S040, S041, S042, S045, S047, S048, S052, S053, S054, S056, S057, S060, S061, S063, S064, S065, S068, S069, S070, S071, S072, S074, S076, S077, S079, S081, S083, S085, S086, S087, S088, S089, S090, S091, S095, S096, S097, S098, S099, S100, S102, S103, S104, S105, S106, S110, S111, S112, S113, S114, S115, S117, S118, S119, S120, S121, S122, S123, S124, S125, S126, S127, S129, S130, S131, S132, S134, S135, S136, S138, S139, S141, S142, S143, S144, S145, S146, S147, S148, S149, S150, S151, S152, S153, S154, S155, S156, S157, S160, S161, S162, S163, S165, S166



Técnica	Tipo	Cantidad artículos	Artículos seleccionados
PP	EE	11	S001, S006, S022, S025, S026, S028, S044, S050, S067, S078, S094
EE Pura	EE	7	S001, S008, S020, S026, S050, S055, S082
DT	Machine Learning	6	S012, S066, S092, S093, S108, S116
KNN	Machine Learning	5	S043, S066, S073, S092, S108
LR	Estadística	4	S012, S059, S073, S108
Método algorítmico propio	Algorítmica	4	S046, S062, S078, S133
RF	Machine Learning	4	S012, S092, S093, S116
Wideband Delphi	EE	4	S001, S028, S050, S078
COSMIC FSM	Algorítmica	4	S034, S078, S094, S159
GRNN	Machine Learning	3	S092, S093, S140
MLP	Machine Learning	3	S073, S092, S108
SVM	Machine Learning	3	S051, S066, S092
CCNN	Machine Learning	2	S092, S093
CEEM	Algorítmica	2	S016, S082
GBA	Machine Learning	2	S025, S092
NB	Machine Learning	2	S066, S092
Pair-Estimation	EE	2	S002, S008
PNN	Machine Learning	2	S092, S093
RBFN	Machine Learning	2	S059, S140
SGB	Machine Learning	2	S093, S116
SVR + RBFN	Machine Learning	2	S092, S018
NPC	Machine Learning	1	S073
Autoencoders	Machine Learning	1	S058
AB	Machine Learning	1	S012
AHN	Machine Learning	1	S073
ANFM	Machine Learning	1	S140
ABC-PSO	Machine Learning	1	S093
BoW	Machine Learning	1	S092
Average Productivity	Estadística	1	S049
CBR	Machine Learning	1	S012
Case-based reasoning	Estimación de Expertos	1	S050



Técnica	Tipo	Cantidad artículos	Artículos seleccionados
DN	Machine Learning	1	S092
COCOMO II	EE + Algorítmica	1	S082
COSMIC FSM + SLR	Algorítmica + Estadística	1	S128
COSMIC FSM + MLR	Algorítmica + Estadística	1	S128
Deep-SE	Machine Learning	1	S164
Event Points	Algorítmica	1	S159
Estimation by Analogy	Estadística	1	S038
ECS-DBN	Machine Learning	1	S051
XGB	Machine Learning	1	S012
EE + Planning Game	EE	1	S067
ENN	Machine Learning	1	S051
EE + Blitz Planning	EE	1	S067
ELM	Machine Learning	1	S051
FLANN-WOA	Machine Learning	1	S116
FFBP	Machine Learning	1	S051
EE using HyEEASe	EE	1	S025
GLM	Estadística	1	S051
FNN	Machine Learning	1	S073
FP	EE + Algorítmica	1	S050
IRDSS	Machine Learning	1	S028
ICR	Estadística	1	S073
GMDH	Machine Learning	1	S092
GMDH-PNN	Machine Learning	1	S093
HKO	Machine Learning	1	S034
Knowledge4Scrum	EE	1	S075
LDBNM	Machine Learning	1	S084
LOC	Estadístico	1	S159
LOC + NN	Estadístico + Machine Learning	1	S107
LOC - GGA	Estadístico + Machine Learning	1	S107
Mean	Estadística	1	S164
Median	Estadística	1	S164



Técnica	Tipo	Cantidad artículos	Artículos seleccionados
MLNN	Machine Learning	1	S092
MR	Estadística	1	S049
Normalized Sprint Estimation	EE	1	S137
PP + Checklist	EE	1	S044
PP (Averaging)	EE	1	S080
PP (Consensus)	EE	1	S080
PP using HyEEASE	EE + Machine Learning	1	S006
PP using HyEEASE GBT	EE + Machine Learning	1	S006
PREP	EE + Algorítmica	1	S158
RBFN + WAO	Machine Learning	1	S116
RG	Estadística	1	S164
SP	EE	1	S038
Sprint Points + ML	EE + Machine Learning	1	S109
SP + GA	EE + Machine Learning	1	S107
SP + Neural Networks	EE + Machine Learning	1	S107
SR	Estadística	1	S049
SVR	Machine Learning	1	S094
SVR + Polynomial	Machine Learning	1	S108
SVR: Polynomial, RBFN and Sigmoid	Machine Learning	1	S092
Unstructured group	EE	1	S022
UCP	EE	1	S050
UCP test effort estimation model	EE	1	S050
UCP + MLNN	EE+ Machine Learning	1	S101
ZeroR	Machine Learning	1	S066
Zia et al.'s regression	Estadística	1	S093

Se destaca que el 72.28% de los estudios seleccionados, no evidencia el uso de técnicas de estimación en la industria, lo que vislumbra un campo por explorar en trabajos futuros.

De los 42 estudios que reportan evidencia del uso de técnicas de estimación en la industria, en 35 se trata de la misma muestra en las estimaciones reportadas en metodologías



ágiles (PI5), es decir, que en sólo 7 estudios se reportó una nueva evidencia al utilizar las técnicas en la industria (S001, S016, S034, S066, S082, S092 y S107).

Es interesante destacar que numerosos estudios reportan evidencias de uso de técnicas basadas en SVR, sobre todo complementadas con técnicas de aprendizaje automático.

4.8 PI8: ¿Qué resultados fueron reportados del uso de estas técnicas en la industria?

Se muestran en la [Tabla 17](#) los beneficios reportados del uso de las principales técnicas de estimación usadas en la industria y en la [Tabla 18](#) las desventajas del uso en la industria.

Beneficios

Las técnicas planning poker, CEEM, K-nearest neighbors, el método desarrollado en [S038], PP using HyEEASe GBT y RBFN evidencian una mejora en el nivel de precisión en comparación con el obtenido a través de la estimación de expertos pura. A su vez, en [S002] se halla que tanto la estimación por pares como el método propuesto por Adnan et al. en [S078] ofrecen una precisión mejor que la obtenida a través de PP.

En proyectos XP, las técnicas de regresión lineal y RBFN logran buenos rendimientos en su precisión.

Tanto las técnicas de PP como la propuesta por Vetrò et al. mejoran la precisión frente a wideband delphi.

Por su parte, las técnicas COSMIC FSM, CEEM y PP son sugeridas para aplicar en proyectos ágiles para estimar el esfuerzo.

Por último, las técnicas fuzzy neuronal network, multi-layer perceptron otorgan una buena previsibilidad del esfuerzo requerido en un proyecto ágil.

Desventajas

Lógicamente se reporta una fuerte dependencia de la madurez de los estimadores en las técnicas de estimación de expertos pura, PREP y PP, aunque en el caso de este último, se evidencia la desventaja de la técnica al participar de la estimación los profesionales menos experimentados.

Se halla también la fuerte correlación que existe entre las técnicas de deep-se y HKO con el modo de redacción de las historias de usuario o requisitos.

En cuanto al tiempo del proceso de estimación, se presentan evidencias que las técnicas de PP y PREP tienen duraciones mayores a otras técnicas conocidas.

A la hora de estimar tareas chicas, la técnica CEEM arroja estimaciones poco precisas.

Por último, se identifica la necesidad de crear estándares para las técnicas de estimación de expertos y PP.



Tabla 17. Evidencia de beneficios de las principales técnicas de estimación usadas en la industria

Categoría	Técnica	Beneficio
Precisión en las estimaciones	Planning Poker, CEEM, K-Nearest Neighbors, Método propio (S038),, PP using HyEEASe GBT, RBFN	Mejora significativa en la precisión de estimaciones respecto a estimación de expertos (S001) (S082)(S043)(S038)(S025)(059)
	Estimación por pares, Método propio (S078)	Mejor precisión en las estimaciones que Planning Poker (S002)(078)
	FLANN-WOA y RBFN-WOA	Excelentes resultados en términos de precisión cuando se integran con la técnica de metaheurística WOA (S116).
	Linear Regression, RBFN	Buen rendimiento para proyectos XP (S059)
	CEEM	Mejora significativa en la precisión de estimaciones respecto a COCOMO II (S082)
	Método propio (S038), Planning Poker	Mejor precisión en las estimaciones que Delphi (S078) (S001)
	Deep-SE	Mejor precisión que media, mediana y RG. (S164)
	COSMIC FSM	Mejora la precisión de esfuerzo en proyectos SCRUM (S094) Mejor correlación entre el tamaño de mejora de COSMIC y el esfuerzo que con Story Points. (S094) (S049)
	Decision Tree	Buena correlación entre el valor estimado y el real (S108)
	ECS-DBN	Precisión de casi el 99% en las estimaciones. (S051)
	Estimación de Expertos Pura	Buena correlación positiva cuando se estima el esfuerzo en horas contra lo real (S020) Suficientes para un rango chico de tareas a estimar (S020)
	Estimación por pares	Mejor precisión en las estimaciones que la autoestimación (S002)
	Event Points	Mejores estimaciones que LOC. (S159)
	Event Points	Mejores estimaciones que COSMIC. (S159)
	HKO	Mejor precisión que técnicas de clasificación basada en número de pasos (S034).
	Multi-Layer Perceptron	Produce estimaciones satisfactorias y precisas con bajos valores de error y alta precisión de predicción. (S108)
	Planning Poker	Reducción de las subestimaciones (S001)
	PP using HyEEASe GBT	Más precisión que las estimaciones basadas en modelos (S025)
	PREP	Reducción de errores y aumenta la precisión de las estimaciones en el desarrollo de productos . (S158)
	Support Vector Machine	Muy adecuado para la categorización de texto (S066).
Support Vector Machine	Mejor precisión en las estimaciones que NB (S066)	
Support Vector Machine	Mejor precisión en las estimaciones que KNN (S066)	
Support Vector Machine, SVM + RBFN + GS	Mejor precisión en las estimaciones que DT (S066)(S093)	



Categoría	Técnica	Beneficio
	SVM + RBFN + GS	Mejor precisión en las estimaciones que GRNN (S093)
	SVM + RBFN + GS	Mejor precisión en las estimaciones que el método de Zia et al. (S093)
	SVM + RBFN + GS	Mejor precisión en las estimaciones que PNN (S093)
	SVM + RBFN + GS	Mejora en gran medida el rendimiento del modelo SVR-RBF en términos de precisión (093)
Financieros	Método propio	Útil para evaluar propuestas de costos de contratos en etapa de licitación o antes. (S046)
	Planning Poker	Aumento de beneficios financieros (S001)
Preferencia de uso	COSMIC FSM, CEEM, Planning Poker	Medida adecuada para estimar el esfuerzo en proyectos ágiles. (S049)(S082)(S026)
	Planning Poker	Mayor preferencia frente a otras técnicas para desarrollos móviles (S026)
	PP using HyEEASe GBT	Percepción positiva sobre la utilidad y comprensibilidad del método por parte de los profesionales (S025)
Equipo y confianza	Estimación de Expertos Pura	El uso de predictores proporciona mayor confianza a los estimadores (S008)
	Estimación por pares	Evita que desarrolladores no especializados se involucren en tareas avanzadas (S002) Ayuda al equipo de desarrollo cuando trabaja con historias de usuario. (S002)
	Planning Poker	Garantiza que todos los desarrolladores participen en el proceso de estimación (S094)
Complejidad de uso	Autoencoders	No hay diferencias significativas en las estimaciones de diferentes tipos de Autoencoders. (S058)
	CEEM	Aplicable para metodologías de desarrollo de software tradicionales (S082)
	COSMIC FSM	Requiere poco esfuerzo si las historias de los usuarios incluyen suficientes detalles (S049)
	COSMIC FSM + SLR/MLR	Ayuda a estimaciones tempranas del esfuerzo. (S128) Puede utilizarse en la industria del software móvil para predecir el esfuerzo de prueba. (S128)
	Deep-SE	El modelo pudo aprender con relativo éxito la semántica de la jerga industrial (S164)
	ECS-DBN	Mínimo impacto en la agilidad. (S051)
	Event Points	Medición más rápida que COSMIC (S159)
	Linear Regression, RBFN	No hay necesidad de de datos históricos (S059)
	Linear Regression	Buena adaptación al desarrollo iterativo (S059)
	Multi-Layer Perceptron	Ahorro de tiempo en los procesos de estimación (S108)
	Fuzzy Neuronal Network, Multi-Layer Perceptron	Otorga buena previsibilidad del esfuerzo (S073)S108)
Gestión del conocimiento	Método propio	Otorga facilidades para la gestión del conocimiento en proyectos XP y SCRUM (S078).



Tabla 18. Evidencia de desventajas de las principales técnicas de estimación usadas en la industria

Categoría	Técnica	Desventaja
Equipo	Planning Poker	Participación de miembros no especialistas en la estimación (S008)
	Estimación de Expertos Pura	Los equipos inmaduros incurren en mayores excesos de esfuerzo (S055)
	Estimación de Expertos Pura, PREP	Fuertemente dependiente de la madurez de los estimadores (S055)(S158)
Gestión del conocimiento	Método propio	Requiere la captura y gestión de lecciones aprendidas del equipo (S078)
Precisión en las estimaciones	CEEM	Mala precisión en las estimaciones para tareas chicas(S082)
	Deep-SE	Aumento del error en las historias que tenían story points altos (S164)
	Estimación de Expertos Pura	Las estimaciones se vuelven más ruidosas cuando los rangos del esfuerzo real se vuelven más grandes (S020) La subestimación es la tendencia dominante (S055) Las prioridades de los requisitos afectan la precisión de las estimaciones de esfuerzo (S055)
	HKO	Su precisión es sensible al lenguaje utilizado en las descripciones de los requisitos (034)
	Planning Poker	Aumento de error en la estimación en casos extremos (S022) Reducción del rendimiento de la estimación cuando el equipo carece de experiencia (S022)
Rendimiento y versatilidad	Deep-SE, HKO	El rendimiento del modelo está correlacionado con ciertas características de las historias, como el nivel de detalle y la frecuencia de términos vagos en las historias. (S164)(034)
	Planning Poker, PREP	Largo tiempo del proceso de estimación (S008)(S158)
	Regression-based aprendizaje automático (own method S092)	Sólo pueden proporcionar una estimación del esfuerzo para el software desarrollado con metodología Scrum (S092)
	Regression-based aprendizaje automático (own method S092)	Requiere una preparación correcta de los problemas a crear, asignando los problemas en las iteraciones correctas y también poniendo las iteraciones en la fase correcta. (S092)
	SVM + RBFN + GS	Se pueden obtener diferentes resultados al usar diferentes métodos de validación y diferentes conjuntos de datos.
Robustez	Estimación de Expertos Pura, Planning Poker	Necesidad de crear estándares para estimar grandes proyectos (S020)(S026)
Tamaño	Planning Poker	No proporciona información directa del tamaño del software (S094)

4.9 PdP1: ¿Dónde han sido publicados los estudios?

Con esta pregunta se pretende mostrar dónde han sido publicados los estudios encontrados. La [Tabla 19](#) muestra esta distribución.



Tabla 19. Número de estudios por tipo de sede

Sede	Cantidad	Artículos seleccionados
Conference	87	S002, S008, S009, S011, S012, S013, S014, S015, S016, S019, S022, S026, S029, S031, S033, S034, S039, S040, S042, S045, S047, S048, S050, S052, S054, S057, S058, S060, S063, S064, S065, S066, S068, S069, S070, S071, S072, S074, S076, S079, S080, S081, S087, S088, S091, S093, S095, S098, S099, S101, S102, S103, S104, S105, S106, S109, S111, S112, S114, S115, S118, S119, S122, S123, S124, S129, S130, S132, S133, S134, S135, S136, S139, S140, S151, S152, S155, S156, S157, S158, S159, S160, S162, S163, S164, S165, S166
Journal	71	S001, S003, S004, S005, S006, S007, S010, S018, S021, S023, S024, S025, S027, S028, S032, S035, S036, S037, S038, S041, S043, S044, S051, S053, S055, S056, S061, S062, S067, S073, S075, S077, S078, S083, S084, S085, S086, S089, S090, S092, S094, S096, S097, S100, S107, S108, S110, S116, S117, S120, S121, S125, S126, S127, S128, S131, S137, S138, S141, S142, S143, S144, S145, S146, S147, S148, S149, S150, S153, S154, S161
Symposium	6	S017, S030, S046, S059, S082, S113
Workshop	2	S020, S049

Casi la mitad de los artículos seleccionados (52.40%) se publicaron en conferencias. Esto sugiere que las conferencias son una fuente significativa de publicación para la estimación de proyectos de ASD. También hay un número considerable de artículos publicados en revistas científicas, con un 42.77% del total de artículos. Las revistas científicas suelen ofrecer un formato más detallado y revisado por pares para la presentación de investigaciones.

Por último se han incluido algunos artículos de simposios (Symposium) y talleres (Workshop), lo que indica una diversidad en los lugares donde se presentaron estos estudios.

4.10 PdP2: ¿En qué año fueron publicados?

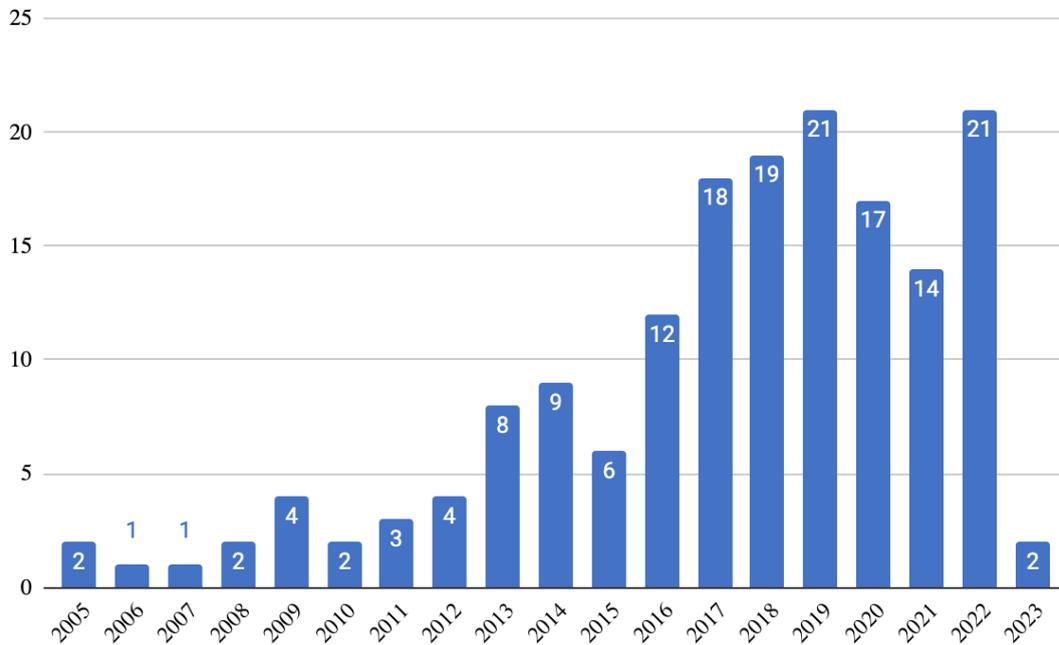
Con esta pregunta se quiere dilucidar el año de publicación de los estudios hallados. La [Gráfico 2](#) muestra esta distribución.

A pesar de que la agilidad moderna a partir del manifiesto ágil, comenzó en 2001, se observa que el interés por las estimaciones y su precisión en este tipo de proyectos comenzó recién en el año 2005.

Se aprecia un aumento del interés en la temática reflejada en la cantidad de artículos hallados en los últimos años. Como se evidencia en la PI 8, en los últimos años, la proliferación de técnicas de aprendizaje automático han generado un área aún en proceso de análisis.



Gráfico 2. Años de publicación



4.11 PdP3: ¿Cuáles son los países más activos?

Se muestra la cantidad de artículos por país en la [Tabla 20](#).

Tabla 20. Cantidad de artículos por país

ID	País	Cantidad
1	India	42
2	Estados Unidos	17
3	Alemania	14
4	Reino Unido, Suecia	9
5	Malaysia	8
6	Australia, Italia, Pakistán, Turquía	7
7	Brasil, Canadá, Noruega	6
8	México	5
9	Arabia Saudita, Indonesia, Mauricio, Polonia	4
10	Colombia, Egipto	3
11	España, Croacia, Irán, Marruecos, Ucrania, Vietnam	2
12	Corea, Eslovenia, Estonia, Finlandia, Letonia, Nueva Zelanda, Nigeria, Palestina, Países Bajos, Portugal, Suiza, Taiwán, Tailandia, Tunes	1



Los artículos seleccionados provienen de una amplia gama de países en todo el mundo, lo que indica un interés global en esta área. India lidera en términos de cantidad de artículos, lo que sugiere un alto nivel de actividad de investigación en el país en relación al estudio de las estimaciones de software. Países como Estados Unidos, Alemania, Reino Unido, y Suecia también están bien representados, lo que refleja la contribución significativa de países occidentales a la investigación en este campo.

En general, estos datos reflejan que la investigación relacionada con la estimación de software en la agilidad es global y se lleva a cabo en una amplia variedad de países. Esto puede proporcionar una perspectiva más amplia y diversa sobre el tema, con diferentes enfoques y perspectivas regionales.

5. Amenazas a la validez

A continuación, se presentan las acciones de mitigación que se han considerado para reducir el impacto de las amenazas a la validez, categorizadas por Petersen [10], que podrían afectar el estudio. Las mismas consideran la validez descriptiva (sección 5.1), validez teórica (sección 5.2), generalización (sección 5.3), validez interpretativa (sección 5.4) y repetibilidad (sección 5.5).

5.1 Validez descriptiva

Esta validez tiene como objetivo garantizar que las observaciones se describen de manera objetiva y precisa.

- La información a recolectar ha sido organizada por medio de un par de formularios de extracción de datos, para las PI y las PdP, presentados en la [Tabla 7](#) y en la [Tabla 8](#) para apoyar un registro de datos uniforme y objetivar el proceso de extracción de datos.
- Además, todos los investigadores participaron en una reunión inicial, con el fin de estandarizar conceptos y criterios, responder a cualquier pregunta y mostrar, a través de ejemplos, cómo llevar a cabo el proceso de extracción de datos.
- Finalmente, también se publica el formulario de extracción de datos.

5.2 Validez teórica

La validez teórica depende de la capacidad de hacerse de la información que se pretende obtener.

- Se comienza con una cadena de búsqueda ([Tabla 3](#)) adaptada a las tres bibliotecas digitales más populares sobre bases de datos en línea de informática e ingeniería de software.



- Un experto proporciona un conjunto de artículos para verificar si se recuperaron con la cadena de búsqueda.
- Se definen un conjunto de criterios de exclusión ([Tabla 5](#)) para objetivar el proceso de selección.
- Los estudios se distribuyen entre cuatro investigadores, trabajando de forma independiente y una superposición de estudios asegurando que cada uno de los estudios haya sido verificado al menos por dos investigadores ([Tabla 4](#)).
- Se combinan dos métodos de investigación diferentes: una búsqueda automática y una búsqueda manual (bola de nieve hacia adelante), para reducir el riesgo de no encontrar toda la evidencia disponible.

5.3 Generalización

Esta validez está relacionada con la capacidad de generalizar los resultados a todo el dominio.

- El conjunto de las PI es suficientemente general para identificar y clasificar los hallazgos sobre técnicas de estimación en metodologías ágiles de desarrollo de software, independientemente de casos específicos, el tipo de industria, etc. [12].
- La búsqueda inicial de trabajos similares colabora con la definición de las PI generalizada.

5.4 Validez interpretativa

Esta validez se logra cuando las conclusiones son razonables considerando los datos.

- Al menos dos investigadores validan cada conclusión.
- Dos investigadores, con experiencia en el dominio del tema, ayudan con la interpretación de los datos.

5.5 Repetibilidad

El proceso de investigación debe ser lo suficientemente detallado para asegurar una repetición completa.

- Este protocolo se diseña con suficiente detalle como para permitir la repetición del proceso seguido.
- El protocolo, así como los resultados del estudio, se publican en línea para que otros investigadores puedan replicar el proceso y confirmar los resultados.



6. Conclusiones

En el contexto de un proyecto de software, es independiente de la metodología utilizada, la necesidad de lograr una estimación temprana correcta. Correcta implica que el margen de error con el aspecto estimado real incurrido en el proyecto, se encuentre entre los márgenes considerados como aceptables. Esto determina el nivel de precisión que estas estimaciones tienen y que como se ha evidenciado, es sumamente relevante para garantizar el éxito del proyecto.

El Mapeo Sistemático de la Literatura llevado a cabo para encontrar los niveles de precisión de estimaciones reportados en proyectos de desarrollo de software ágiles, arrojó 4315 artículos, que luego de aplicado el proceso de selección resultó en 166 artículos que tratan la estimación y precisiones en ASD. Se definieron 8 preguntas de investigación y 3 preguntas de publicación.

A continuación se presenta en la sección 6.1 los aspectos más relevantes que se desprenden del trabajo realizado, y en la sección 6.2 las futuras líneas de investigación resultantes.

6.1 Aspectos más relevantes resultantes del estudio

Estos son los aspectos más relevantes del mapeo sistemático llevado a cabo.

Diversidad metodológica en el desarrollo ágil

La investigación revela que el enfoque ágil es el más discutido y ampliamente investigado en la literatura de desarrollo de software cuando se trata de estimaciones. SCRUM también es popular y objeto de estudio frecuente. Sin embargo, las metodologías tradicionales aún despiertan interés, y otras metodologías ágiles como XP, KANBAN y TDD también tienen su espacio. Estas metodologías proporcionan una base sólida para la investigación y la práctica en el desarrollo ágil de software.

Amplia gama de técnicas de estimación

Las técnicas de estimación en el desarrollo de software son diversas y evolucionan constantemente. Desde el uso de métodos de estimación de expertos hasta la aplicación de enfoques de aprendizaje automático, se observa una tendencia hacia la combinación de paradigmas para lograr estimaciones más precisas y confiables.

Enfoque en estimación de esfuerzo

La mayoría de las técnicas se centran en estimar el esfuerzo de un proyecto de software, pero también se estiman otros componentes como el costo financiero o el tamaño del proyecto. Esto demuestra la relevancia y la complejidad de la estimación en el desarrollo de software.



Tendencia hacia el aprendizaje automático

Se observa una creciente adopción de técnicas de aprendizaje automático en la estimación de software. Esto refleja la disponibilidad de datos en la industria del desarrollo de software y la búsqueda de enfoques automatizados para mejorar la precisión de la estimación.

Convergencia y flexibilidad

La combinación de diferentes técnicas, como la estimación de expertos junto con enfoques algorítmicos o de aprendizaje automático, demuestra una convergencia en la búsqueda de precisión. Además, se reconoce la necesidad de flexibilidad al elegir la técnica de estimación adecuada según el contexto y los datos disponibles.

Oportunidades de investigación futura

La investigación también identifica áreas menos estudiadas en la estimación, como la mantenibilidad y la cobertura de pruebas, que podrían representar oportunidades de investigación futura.

Uso en la industria y beneficios reportados

Aunque la mayoría de los estudios no informan sobre el uso de técnicas de estimación en la industria, aquellos que lo hacen (por ejemplo técnicas basadas en aprendizaje automático) destacan beneficios en términos de mejora de la precisión en comparación con la estimación de expertos. Esto sugiere un potencial para aplicar estas técnicas en la práctica.

Desventajas y desafíos identificados

Se señalan desventajas y desafíos, como la dependencia de la madurez de los estimadores en las técnicas de estimación de expertos, la correlación entre las técnicas de deep-se y la redacción de historias de usuario, y la necesidad de estándares para las técnicas de estimación de expertos y planning poker.

6.2 Futuras líneas de investigación

En resumen, esta investigación arroja luz sobre la complejidad y la diversidad de las estimaciones de software en proyectos ágiles. La combinación de metodologías, técnicas y enfoques demuestra un campo dinámico en constante evolución. Las técnicas de aprendizaje automático están ganando terreno, y la convergencia de enfoques ofrece oportunidades para mejorar la precisión en la estimación. Además, la identificación de desafíos y áreas menos estudiadas señala posibles direcciones futuras para la investigación en este campo en constante evolución.

En línea con esto, como futura línea de trabajo, se prevé la construcción de una técnica de estimación de esfuerzo para proyectos de software ágiles, basada en inteligencia artificial, más precisamente utilizando lógica difusa. El objetivo de esta técnica es mejorar los niveles de precisión de estimaciones de esfuerzo en proyectos ágiles de software.



REFERENCIAS

- [1] S. McConnell, "Software Estimation: Demystifying the Black Art", Microsoft, Redmond, Washington, 2006.
- [2] F. Masroor, V. Bhattacharjee and M. Debjani, "A Fuzzy Logic Approach for Software Cost Prediction", Proceedings of the Fourth International Conference on Microelectronics, Computing and Communication Systems, 2019.
- [3] J. Alostad, L. Abdullah and L. Aali, "A Fuzzy based Model for Effort Estimation in Scrum Projects", International Journal of Advanced Computer Science and Applications, 2017.
- [4] S. Iqbal and K. Saghar, "Improving Software Cost Estimation With Function Points Analysis Using Fuzzy Logic Method", LC International Journal of STEM, 2020.
- [5] M. Jørgensen, "A review of studies on expert estimation of software development effort", Journal on System and Software, Vol. 70, No. 1-2, pp. 37-60, 2004.
- [6] R. N. Charette, "Why software fails [software failure]", IEEE spectrum, 42(9), pp. 42-49, 2005.
- [7] Y. S. Kwak, "Project risk management: lessons learned from software development environment", Technovation 24, no 11, pp. 915-920, 2004.
- [8] The Standish Group, "CHAOS Report 2017", <https://www.standishgroup.com>, 2018, Accessed 03/2020.
- [9] B. A. Kitchenham, T. Dyba, and M. Jorgensen, "Evidence-based software engineering", Proceedings of the 26th International Conference on Software Engineering, ICSE, pp. 273–281, 2004.
- [10] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update", Information Software Technology, vol. 64, pp. 1– 18, 2015.
- [11] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering", in Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, p. 38, 2014.
- [12] K. Petersen, "Worldviews, Research Methods, and their Relationship to Validity in Empirical Software Engineering Research", in The Joint Conference of the 23rd International Workshop on Software Measurement (IWSM) and the 8th International Conference on Software Process and Product Measurement, 2013.
- [13] J. Bailey, C. Zhang, D. Budgen, M. Turner, and S. Charters, "Search engine overlaps: Do they agree or disagree?", Proceedings of the Second International Workshop on Realising Evidence-Based Software Engineering REBSE'07, pp. 1–6, 2007.
- [14] B. Kitchenham, Z. Li, and A. Burn, "Validating search processes in systematic literature reviews", in Proceedings of the 1st International Workshop on Evidential Assessment of Software Technologies, vol.1: EAST, pp.3-9, 2011.
- [15] L. Chen, M. A. Babar, and H. Zhang, "Towards Evidence-Based Understanding of Electronic Data Sources", Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering, EASE'10, pp. 135–138, 2010.
- [16] J. Santos, A. Santos, and M. Mendonça, "Investigating bias in the search phase of Software Engineering secondary studies", Proceedings of the 12th Workshop Experimental Software Engineering, 2015.



- [17] D. Badampudi, C. Wohlin, and K. Petersen, "Experiences from using snowballing and database searches in systematic literature studies", *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, pp. 1–10, 2015.
- [18] S. Kumar, M. Arora, Sakshi, "A Review of Effort Estimation in Agile Software Development using Machine Learning Techniques", *ICIRCA*, 2022.
- [19] S. Bilgaiyan, S. Mishra, M. Das, "A Review of Software Cost Estimation in Agile Software Development Using Soft Computing Techniques", *International Conference on Computational Intelligence and Networks*, 2016.
- [20] M. Bingamawa, M. Kamalrudin, "A Review of Software Cost Estimation: Tools, Methods, and Techniques", *researchgate.net*, 2016.
- [21] M. Vyas, A. Bohra, C. Lamba, A. Vyas, "A review on software cost and effort estimation techniques for agile development process", *International Journal of Recent Research Aspects*, 2018.
- [22] S. Bilgaiyan, S. Sagnika, S. Mishra, M. Das, "A Systematic Review on Software Cost Estimation in Agile Software Development.", *Journal of Engineering Science and Technology Review*, 2017.
- [23] M. Usman, E. Mendes, F. Weidt, R. Britto, "Effort Estimation in Agile Software Development: A Systematic Literature Review", *Proceedings of the 10th International Conference on Predictive Models in Software Engineering*, 2014.
- [24] E. Dantas, E. Dilorenzo, M. Perkusich, D. Santos, "Effort Estimation in Agile Software Development: An Updated Review", *International Journal of Software Engineering and Knowledge Engineering*, 2018.
- [25] M. Fernández-Diego, E. Méndez, F. González-Ladrón-de-Guevara, S. Abrahao, E. Insfran, "An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review", *IEEE Access*, 2020.
- [26] A. Sembhoo, B. Gobin-Rahimbux, "A SLR on Deep Learning Models Based on Textual Information For Effort Estimation in Scrum", *Research Square*, 2023.
- [27] M. Arora, S. Chopra, "A systematic literature review of machine learning estimation approaches in scrum projects", *AISC*, 2020.
- [28] Y. Mahmood, N. Kama, A. Azmi, "A systematic review of studies on use case points and expert-based estimation of software development effort", *Journal of Software: Evolution and Process*, 2020.
- [29] E. Canedo, D. Aranha, M. Cardoso, R. da Costa, L. Leite, "Methods for estimating agile software projects: Systematic literature review", *30th International Conference on Software Engineering and Knowledge Engineering*, 2018.
- [30] M. Durán, R. Juárez-Ramírez, S. Jiménez, C. Tona, "Taxonomy for complexity estimation in agile methodologies: A systematic literature review", *CONISOFT*, 2019.
- [31] A. Kaur, K. Kaur, "Investigation on test effort estimation of mobile applications: Systematic literature review and survey", *Information and Software Technology*, 2019.
- [32] B. Alsaadi, K. Saeedi, "Data-driven effort estimation techniques of agile user stories: a systematic literature review", *Artificial Intelligence Review*, 2022.
- [33] M. Azzeh, Al Nassif, I. Attili, "Predicting software effort from use case points: A systematic review", *Science of Computer Programming Journal*, 2020.



-
- [34] Y. Mahmood, N. Kamaa, A. Azmia, A. Khanb, M. Alic, "Software Effort Estimation Accuracy Prediction of Machine Learning Techniques: A Systematic Performance Evaluation", *Journal of Software: Practice and Experience*, 2021.
- [35] P. Sharma, J. Singh, "Systematic literature review on software effort estimation using machine learning approaches", *ICNGCIS*, 2017.
- [36] M. Salamea, L. González-Palacio, M. Oriol, C. Farré, "Systematic Literature study on estimation and prioritization of quality requirements in software development", *CISTI*, 2020.
- [37] W. Carvalho Almeida, F. Furtado, L. de Aguiar Monteiro, F. Escobar, S. Gomes e Silva, "Systematic Review on the Use of Metrics for Estimating the Effort and Cost of Software Applicable to the Brazilian Public Sector", *ICSEA*, 2020.
- [38] A. Altaleb, A. Gravell, "Effort estimation across Mobile app platforms using agile processes: a systematic literature review", *Journal of Software*, 2018.
- [39] C. Carbonera, K. Farias, V. Bischoff, "Software development effort estimation: A systematic mapping study", *IET Software*, 2020.
- [40] E. Fávero, A. Pimentel, R. Pereira, D. Casanova, "Analogy-based Effort Estimation: A Systematic Mapping of Literature", *INFOCOMP*, 2018.
- [41] A. Najm, A. Zakrani, A. Marzak, "Decision Trees Based Software Development Effort Estimation: A Systematic Mapping Study", *ICCSRE*, 2019.
- [42] P. Klimczyk, L. Madeyski, "Technical debt aware estimations in software engineering: A systematic mapping study", *e-Informatica Software Engineering Journal*, 2020.
- [43] K. Beck et al., "Manifesto for Agile Software Development". Agile Alliance, <https://agilemanifesto.org>, 2001.



APÉNDICE A. Listado de estudios primarios seleccionados

Se detallan a continuación los 166 estudios primarios que se utilizaron en la realización del MSL.

- [S001] Gandomani, T.J., Wei, K.T., Binhamid, A.K., "A case study research on software cost estimation using experts' estimates, Wideband Delphi, and Planning Poker technique"
- [S002] A. Altaieb; H. Alhashimi; A. Gravell, "A Case Study Validation of the Pair-estimation Technique in Effort Estimation of Mobile App Development Using Agile Processes"
- [S003] Kaushik, Anupama; Tayal, Devendra Kr.; Yadav, Kalpana, "A Comparative Analysis on Effort Estimation for Agile and Non-agile Software Projects Using DBN-ALO"
- [S004] Choetkiertikul, M., Dam, H.K., Tran, T., Pham, T., Ghose, A., Menzies, T., "A Deep Learning Model for Estimating Story Points"
- [S005] Kaushik, A., Tayal, D.Kr., Yadav, K., "A fuzzified story point approach for agile projects"
- [S006] Tanveer, Binish; Vollmer, Anna Maria; Braun, Stefan, "A Hybrid Methodology for Effort Estimation in Agile Development: An Industrial Evaluation"
- [S007] Khuat, T.T., Le, M.H., "A Novel Hybrid ABC-PSO Algorithm for Effort Estimation of Software Projects Using Agile Methodologies"
- [S008] Altaieb, Abdullah; Altherwi, Muna; Gravell, Andy, "A Pair Estimation Technique of Effort Estimation in Mobile App Development for Agile Process: Case Study"
- [S009] Braga, Mirla R. R.; Bezerra, Carla I. M.; Monteiro, José Maria S.; Andrade, Rossana M. C., "A Pattern Language for Agile Software Estimation"
- [S010] Ghazali, S.N.H., Salirti, S.S., Inayat, I., Ab Hamid, S.H., "A risk poker based testing model for scrum"
- [S011] R. Popli; N. Chauhan, "A sprint-point based estimation technique in Scrum"
- [S012] M. Arora; A. Sharma; S. Katoch; M. Malviya; S. Chopra, "A State of the Art Regressor Model, A comparison for Effort Estimation of Agile software"
- [S013] H. Zahraoui; M. A. Janati Idrissi, "Adjusting story points calculation in scrum effort & time estimation"
- [S014] L. Kompella, "Advancement of Decision-Making in Agile Projects by Applying Logistic Regression on Estimates"
- [S015] R. Popli; N. Chauhan, "Agile estimation using people and project related factors"
- [S016] S. Basri; N. Kama; H. M. Sarkan; S. Adli; F. Haneem, "An Algorithmic-Based Change Effort Estimation Model for Software Development"



- [S017] Miranda, Pedro; Faria, J. Pascoal; Correia, Filipe F.; Fares, Ahmed; Graça, Ricardo; Moreira, João Mendes, "An Analysis of Monte Carlo Simulations for Forecasting Software Projects"
- [S018] Gupta, N., Mahapatra, R.P., "An effective agile development process by a hybrid intelligent effort estimation protocol"
- [S019] R. Britto; E. Mendes; J. Börstler, "An Empirical Investigation on Effort Estimation in Agile Global Software Development"
- [S020] H. H. Arifin; J. Daengdej; N. T. Khanh, "An Empirical Study of Effort-Size and Effort-Time in Expert-Based Estimations"
- [S021] Alshayeb, M., Li, W., "An empirical study of system design instability metric and design evolution in an agile software process"
- [S022] N. C. Haugen, "An empirical study of using planning poker for user story estimation"
- [S023] Raslan, A.T., Darwish, N.R., "An enhanced framework for effort estimation of agile projects"
- [S024] Malgonde, O., Chari, K., "An ensemble-based model for predicting agile software development effort"
- [S025] Tanveer, Binish; Vollmer, Anna Maria; Braun, Stefan; Ali, Nauman bin, "An evaluation of effort estimation supported by change impact analysis in agile software development"
- [S026] A. Altaieb; M. Altherwi; A. Gravell, "An Industrial Investigation into Effort Estimation Predictors for Mobile App Development in Agile Processes"
- [S027] Weflen, E., MacKenzie, C.A., Rivero, I.V., "An influence diagram approach to automating lead time estimation in Agile Kanban project management"
- [S028] Hamid, M., Zeshan, F., Ahmad, A., Ahmad, F., Hamza, M.A., Khan, Z.A., Munawar, S., Aljuaid, H., "An Intelligent Recommender and Decision Support System (IRDSS) for Effective Management of Software Projects"
- [S029] J. Lopez-Martinez; A. Ramirez-Noriega; R. Juarez-Ramirez; G. Licea; Y. Martinez-Ramirez, "Analysis of Planning Poker Factors between University and Enterprise"
- [S030] Kovags, Diego; Falchi, Fábio Luíz; Rivas, André Ronaldo, "Analysis of the Utilization of Scrum Framework Effort Estimation Metrics in Federal Public Administration"
- [S031] A. Shams; S. B√∂hm; P. Winzer; R. D√∂rner, "App Cost Estimation: Evaluating Agile Environments"
- [S032] Rola, P., Kuchta, D., "Application of fuzzy sets to the expert estimation of Scrum-based projects"
- [S033] Chatzipetrou, Panagiota; Ouriques, Raquel; Gonzalez-Huerta, Javier, "Approaching the Relative Estimation Concept with Planning Poker"



- [S034] M. Ochodek, "Approximation of COSMIC Functional Size of Scenario-Based Requirements in Agile Based on Syntactic Linguistic Features, A Replication Study"
- [S035] Dragicevic, S., Celar, S., Turic, M., "Bayesian network model for task effort estimation in agile software development"
- [S036] Tøndel, I.A., Jaatun, M.G., Cruzes, D.S., Williams, L., "Collaborative security risk estimation in agile software development"
- [S037] Conoscenti, M., Besner, V., Vetrò, A., Fernández, D.M., "Combining data analytics and developers feedback for identifying reasons of inaccurate estimations in agile software development"
- [S038] Vetrò, A., Dürre, R., Conoscenti, M., Fernández, D.M., Jørgensen, M., "Combining Data Analytics with Team Feedback to Improve the Estimation Process in Agile Software Development"
- [S039] Z. Racheva; M. Daneva; L. Buglione, "Complementing Measurements and Real Options Concepts to Support Inter-iteration Decision-Making in Agile Projects"
- [S040] A. Magazinius; R. Feldt, "Confirming Distortional Behaviors in Software Cost Estimation Practice"
- [S041] Soni, D., Kohli, P.J., "Cost estimation model for web applications using agile software development methodology"
- [S042] K. Moharrerri; A. V. Sapre; J. Ramanathan; R. Ramnath, "Cost-Effective Supervised Learning Models for Software Effort Estimation in Agile Environments"
- [S043] Karna, H., Gotovac, S., Vicković, L., "Data mining approach to effort modeling on agile software projects"
- [S044] Usman, M., Petersen, K., Börstler, J., Santos Neto, P., "Developing and using checklists to improve software effort estimation: A multi-case study"
- [S045] R. Tamrakar; M. Jørgensen, "Does the use of Fibonacci numbers in planning poker affect effort estimates?"
- [S046] W. Rosa; R. Madachy; B. Clark; B. Boehm, "Early Phase Cost Models for Agile Software Processes in the US DoD"
- [S047] Fehlmann, T.M., Kranich, E., "Early software project estimation the six sigma way"
- [S048] A. W. M. M. Parvez, "Efficiency factor and risk factor based user case point test effort estimation model compatible with agile software development"
- [S049] Salmanoglu, Murat; Hacıoglu, Tuna; Demirors, Onur, "Effort estimation for agile software development: comparative case studies using COSMIC functional size measurement and story points"
- [S050] Britto, R., Usman, M., Mendes, E., "Effort estimation in agile global software development context"



- [S051] Premalatha, H.M., Srikrishna, C.V., "Effort estimation in agile software development using evolutionary cost- sensitive deep Belief Network"
- [S052] Usman, Muhammad; Mendes, Emilia; Börstler, Jürgen, "Effort Estimation in Agile Software Development: A Survey on the State of the Practice"
- [S053] Tanveer, Binish; Guzmán, Liliana; Engel, Ulf Martin, "Effort estimation in agile software development: Case study and improvement framework"
- [S054] M. Usman; R. Britto, "Effort Estimation in Co-located and Globally Distributed Agile Software Development: A Comparative Study"
- [S055] Usman, M., Britto, R., Damm, L.-O., Börstler, J., "Effort estimation in large-scale software development: An industrial case study"
- [S056] Kaur, A., Kaur, K., "Effort estimation in traditional and agile mobile application development & testing"
- [S057] A. Saini; L. Ahuja; S. K. Khatri, "Effort Estimation of Agile Development using Fuzzy Logic"
- [S058] R. G. F. Soares, "Effort Estimation via Text Classification And Autoencoders"
- [S059] P. Abrahamsson; R. Moser; W. Pedrycz; A. Sillitti; G. Succi, "Effort Prediction in Iterative Software Development Processes -- Incremental Versus Global Prediction Models"
- [S060] M. Owais; R. Ramakishore, "Effort, duration and cost estimation in agile software development"
- [S061] Satapathy, S.M., Rath, S.K., "Empirical assessment of machine learning models for agile software development effort estimation using story points"
- [S062] Rosa, W., Clark, B.K., Madachy, R., Boehm, B., "Empirical Effort and Schedule Estimation Models for Agile Processes in the US DoD"
- [S063] H. Sheemar; G. Kour, "Enhancing User-Stories Prioritization Process in Agile Environment"
- [S064] M. M. Hohman, "Estimating in actual time [extreme programming]"
- [S065] Wang, Zhe, "Estimating Productivity in a Scrum Team: A Multi-Agent Simulation"
- [S066] Porru, Simone; Murgia, Alessandro; Demeyer, Serge; Marchesi, Michele; Tonelli, Roberto, "Estimating Story Points from Issue Reports"
- [S067] Torrecilla-Salinas, C.J., Sedeño, J., Escalona, M.J., Mejías, M., "Estimating, planning and managing Agile Web development projects under a value-based perspective"
- [S068] P. Sudarmaningtyas; R. B. Mohamed, "Extended Planning Poker: A Proposed Model"
- [S069] Tanveer, Binish, "Guidelines for utilizing change impact analysis when estimating effort in agile software development"



- [S070] L. Buglione; A. Abran, "Improving the User Story Agile Technique Using the INVEST Criteria"
- [S071] R. Litoriya; N. Sharma; A. Kothari, "Incorporating Cost driver substitution to improve the effort using Agile COCOMO II"
- [S072] Bhalerao, S., Ingle, M., "Incorporating vital factors in agile estimation through algorithmic method"
- [S073] de Campos Souza, P.V., Guimaraes, A.J., Araujo, V.S., Rezende, T.S., Araujo, V.J.S., "Incremental regularized Data Density-Based Clustering neural networks to aid in the construction of effort forecasting systems in software development"
- [S074] Pasuksmit, Jirat, "Investigating Documented Information for Accurate Effort Estimation in Agile Software Development"
- [S075] Sungkur, R.K., Ramasawmy, M., "Knowledge4Scrum, a novel knowledge management tool for agile distributed teams"
- [S076] T. Hacaloglu; O. Demirors, "Measureability of Functional Size in Agile Software Projects: Multiple Case Studies with COSMIC FSM"
- [S077] Adnan, M., Afzal, M., "Ontology based multiagent effort estimation system for scrum agile method"
- [S078] Adnan, M., Afzal, M., Asif, K.H., "Ontology-Oriented Software Effort Estimation System for E-commerce Applications Based on Extreme Programming and Scrum Methodologies"
- [S079] Choudhari, Jitender; Suman, Ugrasen, "Phase Wise Effort Estimation for Software Maintenance: An Extended SMEEM Model"
- [S080] T. J. Gandomani; H. Faraji; M. Radnejad, "Planning Poker in cost estimation in Agile methods: Averaging Vs. Consensus"
- [S081] M. Alhamed; T. Storer, "Playing Planning Poker in Crowds: Human Computation of Software Effort Estimates"
- [S082] Basri, Sufyan; Kama, Nazri; Haneem, Faizura; Ismail, Saiful Adli, "Predicting Effort for Requirement Changes during Software Development"
- [S083] Vyas, M., Hemrajani, N., "Predicting effort of agile software projects using linear regression, ridge regression and logistic regression"
- [S084] Hearty, P., Fenton, N., Marquez, D., Neil, M., "Predicting project velocity in XP using a learning dynamic Bayesian network model"
- [S085] Singh, J., Singh, K., Singh, J., "Reengineering cost estimation using scrum agile methodology"
- [S086] Singh, J., Singh, K., Singh, J., "Reengineering framework to enhance the performance of existing software"



- [S087] J. Dumas-Monette; S. Trudel, "Requirements Engineering Quality Revealed through Functional Size Measurement: An Empirical Study in an Agile Context"
- [S088] S. Downey; J. Sutherland, "Scrum Metrics for Hyperproductive Teams: How They Fly like Fighter Aircraft"
- [S089] Miranda, E., Bourque, P., Abran, A., "Sizing user stories using paired comparisons"
- [S090] Jain, P., Sharma, A., Ahuja, L., "Software maintainability estimation in agile software development"
- [S091] Brezočnik, Lucija; Fister, Iztok; Podgorelec, Vili, "Solving Agile Software Development Problems with Swarm Intelligence Algorithms"
- [S092] Gultekin, M., Kalipsiz, O., "Story Point-Based Effort Estimation Model with Machine Learning Techniques"
- [S093] A. Zakrani; A. Najm; A. Marzak, "Support Vector Regression Based on Grid-Search Method for Agile Software Effort Prediction"
- [S094] Sakhrawi, Z., Sellami, A., Bouassida, N., "Support vector regression for enhancement effort prediction of Scrum projects from COSMIC functional size"
- [S095] B. Seetharaman; Z. Mansor, "The development of agile cost management tool"
- [S096] Angara, J., Prasad, S., Sridevi, G., "Towards benchmarking user stories estimation with cosmic function points-a case example of participant observation"
- [S097] J. Pasuksmit; P. Thongtanunam; S. Karunasekera, "Towards Just-Enough Documentation for Agile Effort Estimation: What Information Should Be Documented?"
- [S098] B. Tanveer; L. Guzmán; U. M. Engel, "Understanding and Improving Effort Estimation in Agile Software Development, An Industrial Case Study"
- [S099] S. K. Khatri; S. Malhotra; P. Johri, "Use case point estimation technique in software development"
- [S100] López-Martínez, J., Ramírez-Noriega, A., Juárez-Ramírez, R., Licea, G., Jiménez, S., "User stories complexity estimation using bayesian networks for inexperienced developers"
- [S101] A. E. D. Hamouda, "Using Agile Story Points as an Estimation Technique in CMMI Organizations"
- [S102] A. Ramirez-Noriega; R. Juarez-Ramirez; R. Navarro; J. Lopez-Martinez, "Using Bayesian Networks to Obtain the Task's Parameters for Schedule Planning in Scrum"
- [S103] Scott, Ezequiel; Pfahl, Dietmar, "Using Developers' Features to Estimate Story Points"
- [S104] Desharnais, Jean-Marc; Buglione, Luigi; Kocatürk, Buundefinedra, "Using the COSMIC Method to Estimate Agile User Stories"
- [S105] B. Tanveer; A. M. Vollmer; U. M. Engel, "Utilizing Change Impact Analysis for Effort Estimation in Agile Development"



- [S106] S Chopra, A Malik, "90 Deep learning inspired continuous estimation framework for Scrum projects"
- [S107] A Sharma, N Chaudhary, "Analysis of Software Effort Estimation Based on Story Point and Lines of Code using Machine Learning"
- [S108] MA Ramessur, SD Nagowah, "A predictive model to estimate effort in a sprint using machine learning techniques"
- [S109] K Periyasamy, J Chianelli, "A Project Tracking Tool for Scrum Projects with Machine Learning Support for Cost Estimation"
- [S110] N Gaffar, H Moussa, A Kamel..., "A Proposed Framework for Enhancing Story Points in Agile Software Projects"
- [S111] MA Ramessur, SD Nagowah, "Factors affecting sprint effort estimation"
- [S112] V Tawosi, A Al-Subaihin, F Sarro, "Investigating the Effectiveness of Clustering for Story Point Estimation"
- [S113] JH Kuo, YH Li, HC Jiau, "Revealing Coordination Needs from Task Allocation for Team Performance Estimation"
- [S114] B Marapelli, A Carie, SMN Islam, "RNN-CNN MODEL: A Bi-directional Long Short-Term Memory Deep Learning Network For Story Point Estimation"
- [S115] W Septian, W Gata, "Software development framework on small team using Agile Framework for Small Projects (AFSP) with neural network estimation"
- [S116] A Kaushik, DK Tayal, K Yadav, "The role of neural networks and metaheuristics in agile software development effort estimation"
- [S117] S Dhir, D Kumar, VB Singh, "An estimation technique in agile archetype using story points and function point analysis"
- [S118] V Lenarduzzi, D Taibi, "Can Functional Size Measures Improve Effort Estimation in SCRUM?"
- [S119] J Grabis, V Minkēviča, B Haidabrus..., "Is team always right: Producing risk aware effort estimates in agile development"
- [S120] M Arora, S Chopra, P Gupta, "Estimation of regression test effort in Agile projects"
- [S121] A Zarour, S Zein, "Software development estimation techniques in industrial contexts: An exploratory multiple case-study"
- [S122] NFBA Kadir, HBM Sarkan, AB Azmi..., "Specification of a Hybrid Effort Estimation System using UML"
- [S123] E Weflen, K Korniejczuk, S Lau, S Kryk..., "Application of Bayesian Belief Network for Agile Kanban Backlog Estimation"
- [S124] TE Fægri, "Adoption of team estimation in a specialist organizational environment"



- [S125] A Altaieb, A Gravell, "An empirical investigation of effort estimation in mobile apps using agile development process"
- [S126] M MAJCHRZAK, L Madeyski, "Factors influencing user story estimations: an industrial interview and a conceptual model"
- [S127] K Moløkken-Østvold, NC Haugen..., "Using planning poker for combining expert estimates in software projects"
- [S128] A Kaur, K Kaur, "A COSMIC function points based test effort estimation model for mobile applications"
- [S129] PGF Matsubara, I Steinmacher..., "Buying time in software development: how estimates become commitments?"
- [S130] H Yuliansyah, SN Qudsiyah, L Zahrotun..., "Implementation of use case point as software effort estimation in Scrum Framework"
- [S131] M Łabędzki, P Promiński, A Rybicki..., "Agile effort estimation in software development projects-case study"
- [S132] S Dhir, D Kumar, VB Singh, "Feedforward and feedbackward approach-based estimation model for agile software development"
- [S133] SA Butt, S Misra, G Piñeres-Espitia..., "A Cost Estimating Method for Agile Software Development"
- [S134] S Bhalerao, M Ingle, "Agile estimation using caea: A comparative study of agile projects"
- [S135] R Popli, N Chauhan, "An agile software estimation technique based on regression testing efforts"
- [S136] S Bahlerao, M Ingle, "Generalized agile estimation method"
- [S137] S Kumar, "Normalized Sprint Estimation: An Agile Project Estimation Method"
- [S138] R Popli, N Chauhan, "A sprint point based tool for agile estimation"
- [S139] J Shah, N Kama, SA Ismail, "An empirical study with function point analysis for software development phase method"
- [S140] C Prasada Rao, P Siva Kumar, S Rama Sree..., "An agile effort estimation based on story points using machine learning techniques"
- [S141] A Trendowicz, R Jeffery, "Planning Poker"
- [S142] Sharma S.; Kumar D.; Fayad M.E., "On the sprint length estimation technique in agile software development using planning poker"
- [S143] Rodríguez Sánchez E.; Vázquez Santacruz E.F.; Cervantes Maceda H., "Effort and Cost Estimation Using Decision Tree Techniques and Story Points in Agile Software Development"



- [S144] Alshammari F.H., "Cost estimate in scrum project with the decision-based effort estimation technique"
- [S145] Butt S.A.; Khalid A.; Ercan T.; Ariza-Colpas P.P.; Melisa A.-C.; Piñeres-Espitia G.; De-La-Hoz-Franco E.; Melo M.A.P.; Ortega R.M., "A software-based cost estimation technique in scrum using a developer's expertise"
- [S146] Teslyuk V.; Batyuk A.; Voityshyn V., "Method of Software Development Project Duration Estimation for Scrum Teams with Differentiated Specializations"
- [S147] L. Cao, "Estimating Efforts for Various Activities in Agile Software Development: An Empirical Study"
- [S148] V. Tawosi; R. Moussa; F. Sarro, "Agile Effort Estimation: Have We Solved the Problem Yet? Insights From a Replication Study"
- [S149] Feng X, Periyasamy K, "A Cost Estimation Model for Scrum Projects"
- [S150] Butt SA, Ercan T, Binsawad M, Ariza-Colpas PP, Diaz-Martinez J, Piñeres-Espitia G, De-La-Hoz-Franco E, Melo MA, Ortega RM, De-La-Hoz-Hernández JD, "Prediction Based Cost Estimation Technique in Agile Development"
- [S151] Phan H, Jannesari A, "Heterogeneous Graph Neural Networks for Software Effort Estimation"
- [S152] V Tawosi, R Moussa, F Sarro, "On the relationship between story points and development effort in Agile open-source software"
- [S153] J Pasuksmit, P Thongtanunam..., "Story points changes in agile iterative development: An empirical study and a prediction approach"
- [S154] Govil N.; Sharma A., "Estimation of cost and development effort in Scrum-based software projects considering dimensional success factors"
- [S155] Ritu; Y. Garg, "Comparative Analysis of Machine Learning Techniques in Effort Estimation"
- [S156] E. R. Sánchez; H. C. Maceda; E. V. Santacruz, "Software Effort Estimation for Agile Software Development Using a Strategy Based on k-Nearest Neighbors Algorithm"
- [S157] Y. Ramchurreetoo; V. Hurbungs, "A multiclass classification model to estimate Agile user stories"
- [S158] G. R. Madya; E. K. Budiardjo; K. Mahatma, "PREP: A Post-Requirements Effort Estimation Method in Scrum's Sprint Grooming"
- [S159] H. Ünlü; T. Hacaloglu; F. Büber; K. Berrak; O. Leblebici; O. Demirörs, "Utilization of Three Software Size Measures for Effort Estimation in Agile World: A Case Study"
- [S160] P. Hansen; H. Timinger, "Concept of a Fuzzy Expert System for Story Point Estimations in Agile Projects"



[S161] M. Fu; C. Tantithamthavorn, "GPT2SP: A Transformer-Based Agile Story Point Estimation Approach"

[S162] R. Popli; N. Chauhan, "Cost and effort estimation in agile software development"

[S163] F. Raith; I. Richter; R. Lindermeier; G. Klinker, "Identification of Inaccurate Effort Estimates in Agile Software Development"

[S164] M. Abadeer; M. Sabetzadeh, "Machine Learning-based Estimation of Story Points in Agile Development: Industrial Experience and Lessons Learned"

[S165] S. Kang; O. Choi; J. Baik, "Model-Based Dynamic Cost Estimation and Tracking Method for Agile Software Development"

[S166] S. Garg; D. Gupta, "PCA based cost estimation model for agile software development projects"