



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL
TÍTULO DE:**

INGENIERO DE SISTEMAS

**CARRERA:
INGENIERÍA DE SISTEMAS**

**TEMA:
"ESTUDIO DE TENDENCIAS Y HERRAMIENTAS
METODOLOGICAS PARA EL DESARROLLO DE SOFTWARE
EN GUAYAQUIL"**

**AUTORES:
ISAIAS DAVID VIVAS FLORES y ANDREE HECTOR SALVADOR HERRERA**

**TUTOR:
Msg. VANESSA JURADO VITE**

**Abril 2021
GUAYAQUIL-ECUADOR**

DECLARATORIA DE RESPONSABILIDAD

Yo, **ISAIAS DAVID VIVAS FLORES** y **ANDREE HECTOR SALVADOR HERRERA**, declaramos que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad del/los autor/es.


Firma del autor

Nombre: Andreé Héctor Salvador Herrera
CI. 0930113386



Nombre: Isaias David Vivas Flores.
CI. 0954819579



FIRMA TUTOR
Mgt. Vanessa Jurado

ESTUDIO DE TENDENCIAS Y HERRAMIENTAS METODOLOGICAS PARA EL DESARROLLO DE SOFTWARE EN GUAYAQUIL

Abstract. The general objective of this scientific article was to carry out a study of the methodological and practical tools used by software development companies in Guayaquil, to achieve compliance with these, the data obtained from other published scientific articles and surveys carried out to different companies in the city of Guayaquil were used to understand the trends, frequency or preferences of developer companies on tools or software development methods. Analysis of documents, such as books, scientific journals or articles, is necessary to ensure customer satisfaction, which is another important factor, as this creates future possibilities of working with the same customer and / or recommendations with others. Similarly, understanding customer needs is critical. Before starting to work with the customer, it is important that the development team understand the requirements of the software product. Based on the analysis of primary studies, it is revealed that Agile and Lean Startup methodologies are the popular software development methods used by new software companies. Finally, it is concluded that studies have shown that there is little information on software development practices. Engineering practices in new companies are not fully explored.

Abstract. El presente artículo científico tuvo como objetivo general realizar un estudio de las herramientas metodológicas y prácticas que utilizan las empresas desarrolladoras de software en Guayaquil, para lograr el cumplimiento de estos, se utilizaron los datos obtenidos de otros artículos científicos publicados y encuestas realizadas a diferentes empresas en la ciudad de Guayaquil para comprender las tendencias, frecuencia o preferencias de las empresas desarrolladoras sobre herramientas o métodos de desarrollo de software. Análisis de documentos, como libros, revistas científicas o artículos, es necesario para asegurar la satisfacción del cliente, que es otro factor importante, ya que esto crea posibilidades futuras de trabajar con el mismo cliente y / o recomendaciones con otros. Del mismo modo, comprender las necesidades del cliente es fundamental. Antes de comenzar a trabajar con el cliente, es importante que el equipo de desarrollo comprenda los requisitos del producto de software. Basado en el análisis de estudios primarios, se revela que las metodologías ágiles y Lean Startup son los métodos populares de desarrollo de software utilizados por las nuevas empresas de software. Finalmente, se concluye que los estudios han demostrado que

hay poca información sobre las prácticas de desarrollo de software. Las prácticas de ingeniería en las nuevas empresas no están completamente exploradas.

Keywords: Tendencias – herramientas – diseño – software – Guayaquil

Introducción

“El software es el facilitador esencial para la ciencia y la nueva economía. Sin embargo, las herramientas y metodologías actuales siguen siendo caras y aún no lo suficientemente confiables para un mercado cambiante y en evolución, mientras que muchos enfoques que parecían prometedores han demostrado ser poco más que soluciones orientadas a casos” [1].

“Por lo general, un ciclo de vida de desarrollo de software incluye varias etapas, desde análisis de desarrollo preliminares hasta pruebas y evaluación de software posteriores al desarrollo. Para manejar estas actividades, se utilizan varias metodologías de desarrollo de software hoy en día, es decir, secuenciales, incrementales, evolutivas, ágiles. Algunas compañías tienen su propia metodología de desarrollo de software personalizada, pero la mayoría utiliza metodologías tradicionales o ágiles” [2].

Las metodologías tradicionales, también conocidas como metodologías pesadas o desarrollo basado en planes, apoyan a los diseñadores que usan planificación integral, documentación detallada y métodos estructurados. Estas metodologías, como la cascada, el modelo V y el proceso unificado racional, están destinadas a proyectos a gran escala, que involucran múltiples sistemas, para quienes los enfoques detallados y los métodos de control ofrecidos son cruciales [3]. Sin embargo, los cambios emergentes más adelante en el proceso de desarrollo tienen un gran costo.

El desarrollo ágil de software (ASD) o las metodologías livianas, en contraste con los enfoques tradicionales, emplean ciclos iterativos cortos y se basan en el conocimiento tácito, existente de manera informal dentro de un equipo, en lugar de la documentación. En la literatura, se pueden encontrar diferentes comparaciones de estas dos metodologías principales [4].

“ASD es una filosofía o una forma de pensar sobre el desarrollo de software y no hay una metodología ágil unificada a seguir. ASD se refiere a varias metodologías de desarrollo de software iterativas e incrementales diferentes, tales como programación extrema, scrum y desarrollo dirigido por funciones. que comparten principios y prácticas comunes” [5].

Estos principios están delimitados en el manifiesto ágil, que se centra en el proceso de desarrollo de una manera centrada en el ser humano. “Las cuatro características clave son: individuos e interacciones sobre procesos y herramientas, software de trabajo sobre documentación exhaustiva, colaboración con el cliente sobre negociación de contratos y respuesta al cambio sobre el seguimiento de un plan” [1].

Los ASD tienen períodos de lanzamiento cortos, son flexibles, requieren una documentación mínima, dependen de individuos y usan equipos autoorganizados. Sin embargo, ASD no promueve un proceso prescriptivo, es incremental a medida que el sistema se desarrolla (y libera) en partes pequeñas.

Desde sus inicios hasta la actualidad, el desarrollo de programas informáticos ha experimentado un proceso evolutivo con marcas notorias, característica propia de cualquier campo técnico. En Ecuador, los programas informáticos se consideran necesarios para que las empresas aumenten la productividad y, a medida que la demanda de software y profesionales de alta calidad continúa creciendo, se ha implementado infraestructura en campos y departamentos cada vez más diversos.

“El campo de desarrollo superó los ejercicios de codificación iniciales, pasó muchas veces y superó con creces las expectativas del usuario, y luego pasó por modelos de codificación y corrección, y luego adoptó una metodología más rigurosa, que se centró en procesos y actividades. Tener un marco secuencial con la misma dirección, con el objetivo de completar el proyecto dentro de todas las limitaciones de contenido, tiempo y presupuesto” [6].

“La industria del software es un pilar importante de la economía de la información, estudia el papel que juegan los sistemas de información en la toma de decisiones económicas y, en un aspecto, se relaciona cada vez más con la economía de un país”[7]. Continúan aumentando la productividad, la creación de empleo, la calidad de los procesos, la inversión y las conexiones con otras actividades y áreas de la empresa.

Este estudio intenta utilizar los datos obtenidos de otros artículos científicos que se muestran en Internet para comprender las tendencias, la frecuencia o las preferencias de las herramientas o métodos de desarrollo de software por parte de las empresas dedicadas a esta industria.

Se cree que aun considerando que la mayoría de los gerentes locales han utilizado los modelos matemáticos propuestos internacionalmente, es difícil adaptarse a la situación actual del proyecto de software ecuatoriano (excepto que es muy complicado de manejar) [8]. Tener la oportunidad de discutir este tema directamente y estimar su proyecto en base a su experiencia en el campo, sus puntos de vista personales o sus propios modelos (en lugar de basarse en el desarrollo teórico formal) [9].

Esta investigación se realizó para ayudar a los propietarios de empresas de desarrollo de software a nivel nacional a diseñar estrategias de software adecuadas, ya que se han propuesto modelos de predicción matemática y una metodología completa en todo el mundo. “Los planes de software, especialmente aquellos diseñados para acortar el tiempo de diseño y optimizar los presupuestos, deben considerar estos factores al diseñar dichos proyectos y cómo controlar y corregir los cambios durante la ejecución” [5].

En cuanto al nivel técnico de este artículo, el primer punto es el análisis de la aparición de las computadoras. La primera generación de computadoras se originó en la década de 1940. Con la llegada de la década de 1970, los métodos de desarrollo fueron generalmente informales. No fue hasta 1972 que Dijkstra publicó su libro titulado "El programador modesto", que sentó las bases para la creación de métodos conocidos que

todavía se utilizan en la actualidad. Posteriormente, nació el "modelo tradicional de desarrollo de sistemas" entre 1970 y 1988.

Gracias a estándares y normativas, las etapas de desarrollo mayormente se pueden resumir en:

- Análisis de requerimientos
- Diseño
- Codificación
- Implementación
- Pruebas
- Documentación
- Mantenimiento

La ingeniería de software es el proceso de analizar las necesidades del usuario, como diseñar, construir y probar aplicaciones de usuario final que satisfagan estas necesidades mediante el uso de lenguajes de programación de software. Es la aplicación de principios de ingeniería en el desarrollo de software” [9]. A diferencia de la programación simple, la ingeniería de software se utiliza para sistemas de software más grandes y complejos, que se utilizan como sistemas clave para empresas y organizaciones. Generalmente, el marco describe o proporciona recomendaciones generales para los pasos o etapas que deben continuarse cuando se usa la satisfacción, sin tener que ingresar datos sobre qué ocupaciones están involucradas en cada etapa. Son varios procesos o métodos seleccionados para el desarrollo del proyecto de acuerdo con el propósito y objetivos del proyecto.

Materiales y Métodos / Metodología

Se emplea una investigación de tipo documental por ser un análisis de varios documentos relacionados con la temática en cuestión, tales como: libros, fuentes bibliográficas electrónicas, folletos, entre otras fuentes.

El diseño de la investigación es bibliográfico, cuyo concepto se fundamenta como el resultado final de un proceso metodológico, estructurado y sistemático que conlleva a la elección de una serie de datos documentales válidos, útiles y relevantes para la comprensión de un tema de investigación o área específica [10]. Este artículo es bibliográfico, puesto que, se acudieron a diferentes fuentes teóricas que sustenten el desarrollo del presente estudio.

Es decir, se revisaron informes, ensayos y artículos sobre el liderazgo, de manera general para luego entrar en el ejercicio de sus funciones dentro de las organizaciones independientemente de cualquier naturaleza.

En simples palabras es una herramienta fundamental para la construcción y generación de un nuevo conocimiento, además, también este artículo se concibe desde una perspectiva dialógica debido a que se selecciona percepciones de expertos académicos en la parte de la administración sobre el liderazgo en las organizaciones. Siendo así, el

enfoque es reflexivo sobre los fenómenos encubiertos en una realidad social para un mejor entendimiento de los mismos.

Dentro de las implicaciones de la investigación, la forma de recolección de información es de orden cualitativa, correspondiendo a los aportes teóricos de diversos autores que han sido publicados a través de diferente material bibliográfico o artículos científicos. Por lo tanto, corresponden como fuentes de información dichas publicaciones, y, por otro lado, las fuentes son los expertos con las percepciones del tema.

Identificación de la necesidad de revisión sistemática

La necesidad de realizar una revisión sistemática surge del interés de resumir el conocimiento científico existente sobre los métodos y prácticas de desarrollo de software utilizados en las empresas. Como se indicó anteriormente, existe una brecha identificada en el desarrollo de software en las empresas. Se necesitan más estudios para explorar el estado del arte en la investigación de inicio de software. Para llenar este vacío, esta SLR es necesaria para extraer y analizar las metodologías de desarrollo de software y las prácticas de trabajo utilizadas en las empresas.

Desarrollo del protocolo de revisión

El protocolo de revisión describe el conjunto de procedimientos que se seguirán para realizar una revisión sistemática específica. Es un plan detallado para obtener material de estudio relevante para responder las preguntas de investigación dadas. El protocolo de revisión predefinido es necesario para reducir los posibles sesgos de los investigadores. El protocolo se desarrolla en base al proceso de revisión descrito en las pautas para realizar la SLR que se adapta de Kitchenham & Charters:

- Antecedentes de la SLR
- Formule las preguntas de investigación de la revisión.
- Definición de la estrategia de búsqueda, los términos de búsqueda y las bases de datos utilizadas para la búsqueda.
- Detalle de los criterios de selección (criterios de inclusión y exclusión) y procedimientos de selección a aplicar.
- Listas de verificación y procedimientos para la evaluación de la calidad de los estudios individuales.
- Definición de la estrategia de extracción de datos para obtener la información requerida de cada estudio primario.
- Definición de la estrategia de análisis de datos: qué técnica se utiliza para analizar y resumir los resultados de los estudios primarios.

Estrategia de búsqueda

El objetivo de la estrategia de búsqueda es encontrar tantos estudios relacionados con las preguntas de investigación como sea posible. Para lograr esto, se desarrolla una estrategia de búsqueda y se aplica a varias bases de datos electrónicas. Para tener una

idea general sobre la cantidad de artículos y luego formular cadenas de búsqueda, se realiza una búsqueda piloto.

Búsqueda piloto

Kitchenham y Charters “han propuesto realizar búsquedas piloto para identificar los estudios potenciales utilizando las cadenas de búsqueda y los recursos definidos en el protocolo de revisión. Para tener una idea general sobre la cantidad disponible de literatura, se realizó un piloto en Google Scholar. Se elige Google Scholar porque incluye literatura de diversos campos de estudio” [11]. La búsqueda piloto se aplicó en Google Scholar con opciones predeterminadas utilizando la palabra clave de entrada "desarrollo de software en el inicio" sin comillas.

Se encontró un resultado de 189,000 que incluyen artículos, libros, patentes, citas, etc. Luego se modificó la palabra clave colocando comillas alrededor de la palabra clave y ejecutando la búsqueda nuevamente, luego de lo cual se encontraron 107 resultados. Esto indica que el “desarrollo de software en la puesta en marcha” como concepto único ha despertado considerablemente menos interés que el software, el desarrollo y la puesta en marcha como palabras clave distintas. También indica que el uso de "desarrollo de software en el inicio" no proporciona una cantidad suficiente de artículos para nuestra búsqueda. En base a esto, se aplican palabras clave separadas a Google Scholar. Se aplicó la palabra clave 'inicio de software' sin comillas para la búsqueda y se encontraron 258.000 resultados.

Para refinar los resultados de la búsqueda, se agrega la palabra "desarrollo de software" sin comillas junto con "inicio de software". Como resultado, se encontraron un total de 183.000 accesos. En el siguiente paso, la palabra 'métodos y prácticas' sin comillas se agrega a la cadena utilizada anteriormente y proporciona un resultado de 58.800 artículos. Finalmente, al eliminar una duplicación de la palabra "software", se encontraron 50.200 resultados. El resultado indica que las palabras clave separadas proporcionan más resultados de búsqueda que un solo término y el autor se enteró de que existe una cantidad considerable de literatura sobre el tema.

Según los resultados de la búsqueda piloto de Google Scholar, las cadenas se modifican nuevamente agregando sinónimos y términos relacionados para cada concepto y se aplican a las bases de datos seleccionadas. Google Scholar busca en recursos como artículos, libros, tesis y resúmenes, que no están relacionados con la ingeniería de software. Como resultado, las palabras clave de búsqueda se aplican a varias bases de datos que se considera que contienen la mayoría de los estudios en la disciplina de ingeniería de software para obtener resultados más precisos y relevantes. Estas bases de datos seleccionadas incluyen IEEE, Scopus y Web of Science. Al aplicar la búsqueda, se aplica el refinamiento en las bases de datos a nuestra búsqueda piloto para obtener más estudios relacionados con el tema. Los criterios de refinamiento incluyen la selección por año (artículos publicados entre enero de 2006 y diciembre de 2016), área temática (Ingeniería; Informática y Negocios, administración y contabilidad) e idioma (inglés).

La búsqueda piloto de las tres bases de datos indica que la adición de sinónimos y términos relacionados a cada palabra clave resultó en estudios más relevantes y relacionados. Aprendiendo de los resultados de las búsquedas piloto, se definieron criterios de búsqueda adicionales para las cadenas de búsqueda reales. Esto incluye la adición

de sinónimos, conceptos similares, términos relacionados y ortografías alternativas para cada concepto central al definir las cadenas de búsqueda reales. Los mecanismos de búsqueda avanzada que utilizan operadores de búsqueda (como OR, AND) también se aplican a cada base de datos seleccionada. Las cadenas de búsqueda y las bases de datos reales se describen a continuación.

Bases de datos

Representan fuentes electrónicas que se utilizan para adquirir estudios relevantes. La cadena de búsqueda se aplica a cada base de datos que sea relevante para los métodos y prácticas de desarrollo de software en nuevas empresas. Se seleccionan tres bases de datos científicas considerando su cobertura y uso en el dominio de la ingeniería de software, y su capacidad para manejar consultas avanzadas, siguiendo el ejemplo de Pater-noster [12], Petersen, Svahnberg, Aurum & Barney [7]. Las bases de datos científicas seleccionadas incluyen:

- IEEE Xplore
- Scopus
- ISI Web of Science

En SLR, se considera fundamental evaluar la "calidad" de los estudios primarios además de los criterios generales de inclusión / exclusión. Los criterios de evaluación de la calidad se utilizan como un instrumento para proporcionar criterios de inclusión / exclusión más detallados y ponderar la importancia de los estudios individuales cuando se sintetizan los resultados [13]. Las evaluaciones de calidad se miden en función de listas de verificación que deben evaluarse para cada estudio. Según Kitchenham & Charters, no existe consenso sobre la definición de calidad. Sin embargo, para evaluar la calidad y reducir cualquier forma de sesgo, el autor de la tesis aplicó una lista de verificación de criterios de calidad [14]. Se formularon cinco preguntas para evaluar la calidad de los estudios primarios seleccionados. La lista de verificación de preguntas está estructurada para garantizar que los artículos seleccionados aborden las preguntas de investigación. Con base en las preguntas a continuación, se eligieron todos los estudios primarios que abordaron las preguntas (calidades). Todos los artículos deben satisfacer los criterios de inclusión y la lista de verificación de calidad para ser considerados en la revisión.

En el proceso de extracción de datos, los principales conceptos relacionados con las metodologías de desarrollo de software y las prácticas en las empresas de software se identifican de cada estudio primario, utilizando los términos del autor original. Luego, esos conceptos principales se organizaron en forma de tabla para permitir comparaciones entre diferentes estudios, de modo que los hallazgos se extraigan para las preguntas de investigación. En resumen, el análisis de datos se logra mediante:

- Documentación de un conjunto de prácticas de desarrollo informadas en empresas de cada estudio primario
- Elaboración de brechas sobre la aplicabilidad de prácticas laborales en contextos de empresas, si las hubiera.

La fase final de una SLR implica informar los resultados de la revisión. Los resultados de la SLR se llevan a cabo escribiendo los hallazgos de la revisión sistemática. Durante la revisión, se seleccionaron los estudios primarios relevantes y se extrajeron los datos relevantes de los estudios primarios seleccionados mediante los formularios de extracción de datos. Los datos recopilados se sintetizaron utilizando enfoques apropiados de síntesis de datos.

Resultados

En este apartado, se mostrará los resultados obtenidos de la encuesta realizada a las empresas desarrolladoras de software en la ciudad de Guayaquil.

1. ¿Cuántos empleados por perfil profesional conforman su equipo de desarrollo de software?

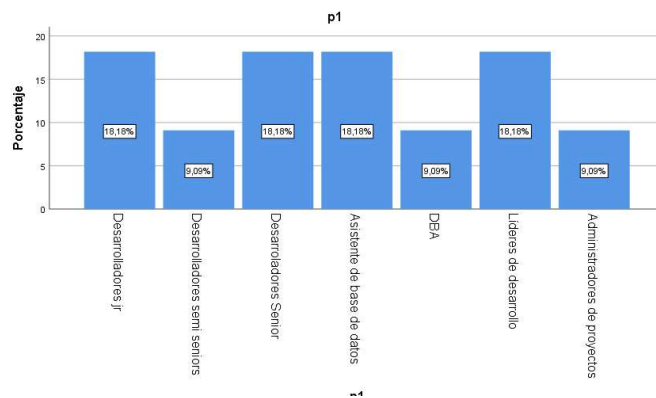


Figura 1 Empleados por perfil profesional en el equipo

Los datos recopilados de la encuesta muestran que el 18.18% de los encuestados tienen a desarrolladores junior como perfil profesional de trabajadores para el desarrollo de softwares, en exacta similitud de porcentaje se encuentran aquellos con asistentes de base de datos.

2. ¿Cuáles entornos de desarrollo utilizan en la empresa?

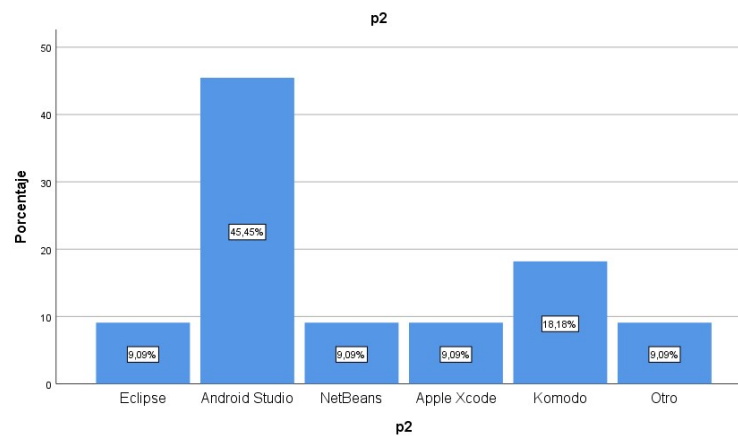


Figura 2 Entornos de desarrollo usados en la empresa

El resultado expuesto de esta interrogante, muestra que el Android studio es el principal entorno de desarrollo que usan las empresas desarrolladoras de softwares en la ciudad de Guayaquil, seguido de aquellas empresas que prefieren usar el sistema Apple xcode.

3. ¿Cuáles lenguajes de programación son utilizados para el desarrollo en la empresa donde usted labora?

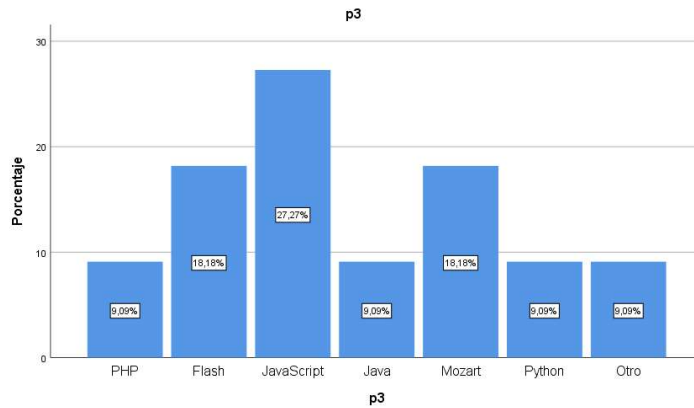


Figura 3 Lenguajes de programación utilizados

El JavaScript es el principal lenguaje de programación utilizada por las empresas de desarrollo de software en la ciudad de Guayaquil con el 27.27% de preferencia, lo sigue el sistema Mozart y flash con el 18.18% cada uno, la diferencia entre las más usadas es amplia, por lo tanto, se puede concluir que el JavaScript es el lenguaje de programación informática por excelencia para el desarrollo de softwares.

4. ¿Qué metodologías de desarrollo utilizan en la empresa?

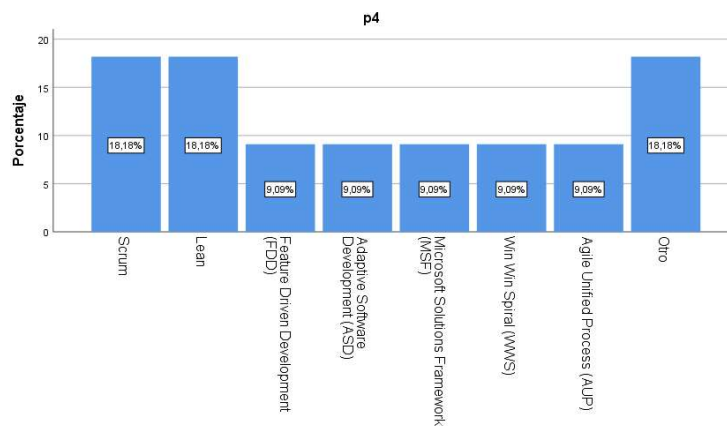


Figura 4 Metodologías de desarrollo utilizados

El scrum es la principal metodología usada para el desarrollo de software por parte de las empresas dedicadas a este sector en la ciudad de Guayaquil, esto se debe a que su uso es relativamente sencillo comparado a otras metodologías, su preferencia está en el 18.18%, sin embargo, existe también la misma preferencia para la metodología Lean.

5. ¿Qué motores de base de datos utilizan en la empresa donde usted labora?

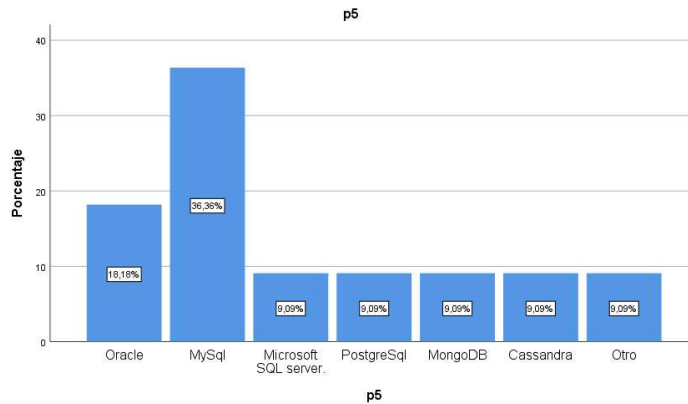


Figura 5 Motores de base de datos utilizados

El principal motor de base de datos para el desarrollo de software es MySQL con el 36.36%, seguido de Oracle con el 18.18%, los siguientes motores de búsquedas no son comunes dentro de las empresas de Guayaquil.

6. ¿Durante el desarrollo de software cuales son los problemas más comunes que se presentan?

a) Con respecto al personal:

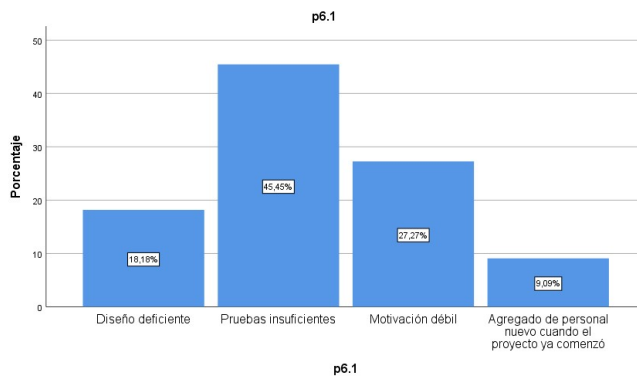


Figura 6 Problemas más comunes del desarrollo de software en lo que respecta al personal

El principal problema más común que se presenta en el desarrollo de softwares se da en el proceso de pruebas, de acuerdo a los encuestados, la cantidad de las mismas es insuficiente, otro de los problemas más comunes es la falta de motivación en los trabajadores.

b) Con respecto a recursos:

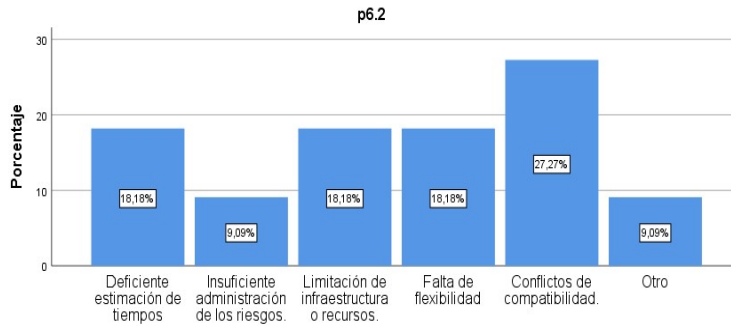


Figura 7 Problemas más comunes del desarrollo de software en lo que respecta al recurso

El problema más común para el desarrollo de software en lo que tiene que ver con recursos son los conflictos de compatibilidad que se presentan al momento de realizar las pruebas, esto corresponde al 27.27%, otro problema común es la limitación de infraestructura de las empresas dedicadas a esta industria con el 18.18%, pues, la mayoría son emprendimientos, lo que dificulta su acceso a una buena instalación para el desarrollo de sus operaciones.

c) Otros:

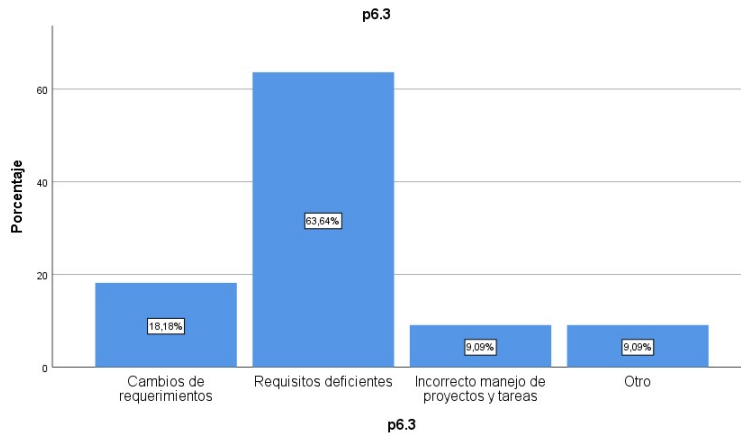


Figura 8 Problemas más comunes del desarrollo de software en lo que respecta a otros factores

Entre otros aspectos de los problemas más comunes para el desarrollo del software, con el 63.64% están ligados a los requisitos deficientes por parte de los clientes, pues piden ciertas funciones que normalmente no suelen desarrollarse, otro 18.18% menciona que el problema común en estos factores ajenos al desarrollo y al recurso son los cambios de requerimientos que usualmente suceden con los clientes.

7. ¿Cuáles fueron los criterios de selección de las herramientas escogidas para el desarrollo de software? En el caso de ser varios factores detallarnos el orden de prioridad del mayor al menor.

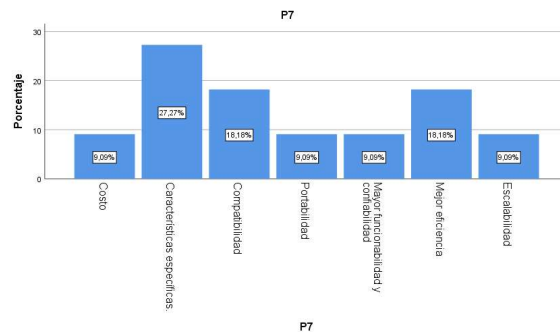


Figura 9 Criterios de selección para las herramientas escogidas

En esta interrogante sobre los criterios de selección de las herramientas con las que trabajan las empresas desarrolladoras de software en la ciudad de Guayaquil, expone que las características específicas con el 27.27% de preferencia, es el factor de más importancia para seleccionar una herramienta de desarrollo de software.

8. ¿Qué tipo de proyecto es más común en su empresa? De ser necesario por favor detallarnos el más común enumerando, siendo el 1 el más recurrente.

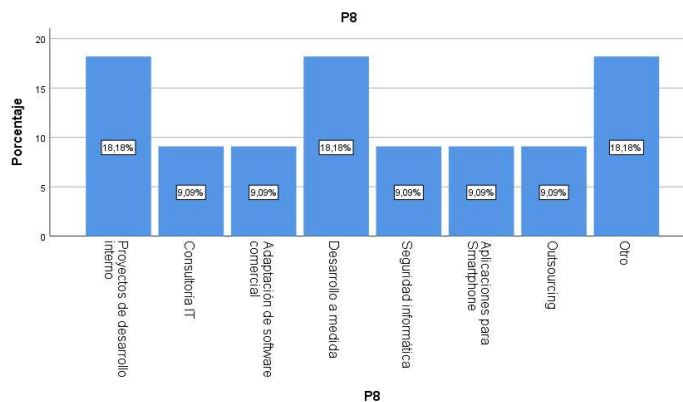


Figura 10 Tipo de proyecto más común en la empresa

Los tipos de proyectos más comunes que se presentan en las empresas desarrolladoras de software en la ciudad de Guayaquil son los proyectos de desarrollo interno y los desarrollos a medidas con el 18.18%, las aplicaciones para Smartphone en menor proporción también son de las más solicitadas en las empresas de desarrollo de software.

9. ¿Qué criterios se tomaron en cuenta para la elección de una metodología de desarrollo? De ser necesario por favor enumere por importancia, siendo 1 el más importante.

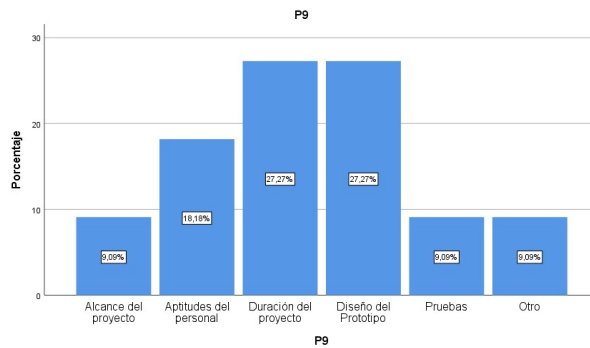


Figura 11 Criterios de selección para la metodología escogida

Dentro de los resultados de esta interrogante surgen algunas respuestas parejas, existen dos criterios principales de selección para la metodología de desarrollo de software, ambos cuentan con un 27.27% de preferencia, las mismas son, el diseño del prototipo y la duración del proyecto, sin embargo, otro de los aspectos de mayor análisis son las aptitudes de los trabajadores, pues, deben estar consciente de las capacidades del talento humano e identificar si pueden realizar o no aquel trabajo.

10. ¿Qué criterios se tomaron en cuenta para la elección de un motor de base de datos? De ser necesario por favor enumere por importancia, siendo 1 el más importante.

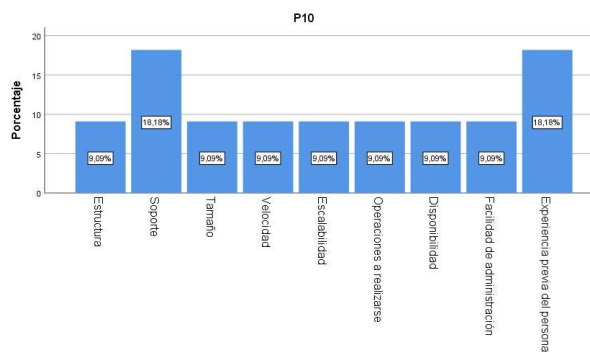


Figura 12 Criterios de elección de un motor de datos

Los criterios más importantes a la hora de la elección de un motor de datos, están basado con el 18.18% en el soporte y la experiencia del personal, seguido de la estructura del motor con el 9.09%, sin dejar de lado la disponibilidad, la facilidad de la administración y las operaciones a realizarse.

11. ¿Qué sistema operativo utilizan en la empresa?

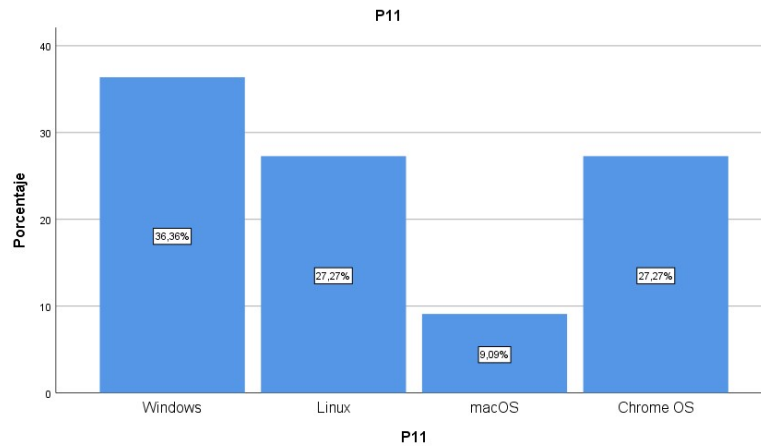


Figura 13 Sistema operativo utilizado en la empresa

El 36.36% de las empresas usan como sistema operativo Windows, seguido de Linux y Chrome con el 27.27%, finalmente tal solo un 9.09% utiliza el sistema operativo macOS.

Todas las empresas de casos bajo adoptaron un enfoque de desarrollo centrado en prototipos. Estuvieron muy involucrados en el desarrollo de prototipos desechables o prototipos evolutivos. Los prototipos desechables se construyen rápidamente en papel / software para visualizar la idea de negocio, mientras que los prototipos evolutivos son la construcción del producto de lanzamiento a través de una planificación detallada y gradual. Por lo tanto, los dos estudios primarios centrados en prototipos describieron que el enfoque de desarrollo de prototipos se utiliza para obtener respuestas rápidas del usuario y validación rápida del producto, seguido de la construcción de un prototipo funcional e iterarlo a lo largo del tiempo. Según el estudio, otras metodologías de desarrollo utilizadas por las empresas incluyen Scrum, desarrollo impulsado por modelos, cascada, hiper-agilidad, proceso unificado racional, iterativo / incremental, y un enfoque ágil y de gestión ad hoc personalizado.

Discusión

Basado en el análisis de estudios primarios, se revela que las metodologías ágiles y Lean Startup son los métodos populares de desarrollo de software utilizados por las nuevas empresas de software. Otros métodos utilizados por las startups incluyen XP,

Prototyping, Scrum, MDD, Waterfall, RUP, Ad hoc management, iterativo / incremental y método ágil personalizado.

Las nuevas empresas de software, particularmente en su etapa inicial, aplican enfoques y prácticas de desarrollo mixtos debido a un entorno que cambia rápidamente. Como resultado de las limitaciones de recursos, personal y tiempo, las startups cambian su metodología de desarrollo para adaptarse a la situación existente. Los resultados indicaron que lo importante para los gerentes de proyectos en la puesta en marcha es superar los desafíos de la incertidumbre y la escasez de recursos, y luego obtener la satisfacción del cliente. Los jefes de proyecto eligen metodologías ligeras que permiten una mayor flexibilidad para adoptar prácticas personalizadas. Como resultado, el equipo de desarrollo puede reaccionar rápidamente al cambiar el producto de acuerdo con las estrategias comerciales.

En el contexto de una startup, donde el entorno es caótico e impredecible, es difícil predecir con precisión los riesgos y las prácticas necesarias para desarrollar un producto [15]. Del mismo modo, es difícil seguir los principios de una metodología de desarrollo específica. En cambio, las metodologías flexibles y reactivas que favorecen los cambios en la toma de decisiones, los procesos de desarrollo y el aprendizaje de las fallas son las más adecuadas en las empresas emergentes. Este es un factor clave para elegir y adoptar de manera efectiva una metodología de desarrollo en una empresa de nueva creación. Los desarrolladores deben tener la libertad de elegir prácticas de desarrollo rápidamente y cambiar o detenerse inmediatamente cuando el desarrollo va mal, corregir el enfoque y aprender de fallas anteriores [16].

El estudio principal muestra que hay prácticas de desarrollo de software diferentes utilizadas por las nuevas empresas de software. Las prácticas más comunes entre las nuevas empresas incluyen el desarrollo de clientes, la adaptación de procesos, el uso de prototipos o producto mínimo viable, la entrega continua de valor y el uso de herramientas fáciles de implementar para facilitar el desarrollo de productos. La SLR también revela que las prácticas de desarrollo de software no se adoptan ni documentan por completo en las nuevas empresas. En cambio, las empresas emergentes eligen prácticas de desarrollo que se adaptan a su entorno de inicio y la mayor parte del proceso se adapta. El empoderamiento de los miembros del equipo se considera un mecanismo de gestión / organización productiva para obtener mejores responsabilidades y una mayor experiencia.

Para sobrevivir, las nuevas empresas deben centrarse en el cliente. Los estudios indican que descubrir clientes es una de la máxima prioridad en las nuevas empresas para mantenerlas en el negocio. Además de que asegurar la satisfacción del cliente es otro factor importante, ya que esto crea posibilidades futuras de trabajar con el mismo cliente y / o recomendaciones con otros. Del mismo modo, comprender las necesidades del cliente es fundamental.

Conclusiones

Los resultados muestran que Agile y Lean Startup son las metodologías de desarrollo más comunes utilizadas para el desarrollo de software. La razón de esto es que permiten

flexibilidad en el proceso, adaptabilidad a un entorno que cambia con frecuencia, una forma fácil de comunicarse con el cliente y un rápido acceso al mercado con bajo costo y entrega rápida. Pero debe tenerse en cuenta que las empresas no siguen estrictamente los principios de la metodología, debido a la limitación de recursos y tiempo. En cambio, adaptan los procesos de acuerdo con su situación eliminando algunas de las características de la metodología existente y agregando nuevas prácticas de otros métodos.

En algunas empresas, incluso es difícil adoptar una determinada metodología de forma estructurada, más bien aplican un mecanismo ad hoc para gestionar los procesos de desarrollo. También se observa que la adopción de un método y prácticas de desarrollo de software depende principalmente de la experiencia de los propietarios y / o de la dirección del proyecto, el nivel de madurez de las nuevas empresas, el tamaño del equipo y la disponibilidad de recursos. En general, las metodologías de desarrollo de software en empresas son informales, a medida y muy ligeras.

El resultado muestra que existe un creciente interés en los métodos y prácticas de desarrollo de software en las empresas de nueva creación. La tendencia de publicación indica que las publicaciones sobre el tema están aumentando en los últimos años. La categorización del método de investigación demostró que el enfoque de investigación más alto utilizado para estudiar las nuevas empresas de software es el estudio de caso. La clasificación de enfoque de investigación también indica que los profesionales están más interesados en estudiar el desarrollo de software y los aspectos de gestión de procesos de una startup que en la estructura administrativa u organizativa de una startup.

Los estudios han demostrado que hay poca información sobre las prácticas de desarrollo de software. Las prácticas de ingeniería en las nuevas empresas no están completamente exploradas. Las prácticas tampoco están documentadas en las empresas debido a un menor enfoque en los procesos y la documentación. Klotins (2015) explica que la literatura existente sobre prácticas de ingeniería en nuevas empresas no es transferible a los profesionales. Paternoster (2014) también ha demostrado que el estado de la técnica existente es escaso.

Esta SLR se realizó de manera sistemática para cubrir todos los estudios posibles relacionados con los métodos y prácticas de desarrollo de software desde el contexto de inicio. La principal limitación de esta SLR está relacionada con la realización de la búsqueda. La siguiente es la lista de limitaciones de la SLR que deben tenerse en cuenta:

La revisión no incluyó libros sobre desarrollo de software en nuevas empresas. La revisión solo incluyó artículos accesibles en las tres bases de datos digitales: IEEE, Scopus y Web of Science. La revisión solo incluyó artículos que estaban disponibles en texto completo.

Trabajos futuros

En esta SLR, los métodos y prácticas de desarrollo de software en empresas de nueva creación se identifican con base en la literatura existente. Se necesitan más estudios empíricos para fortalecer los resultados de este estudio. Debido a la escasez de estudios y al nivel inadecuado de rigor de los informes en los estudios primarios, creemos que

se recomienda un estudio adicional con un enfoque de investigación de alto rigor y con un enfoque en métodos y prácticas de desarrollo de software para transferir el estado del arte a otros practicantes de empresas.

Además, es necesario evaluar la efectividad e idoneidad de cada metodología y práctica de desarrollo para diferentes situaciones y entornos de inicio de software (es decir, etapa inicial o empresas establecidas). Las empresas emergentes cambian su proceso de desarrollo debido a un entorno que cambia rápidamente. Esto sugiere que un tema de investigación qué proceso es adecuado para diferentes situaciones de inicio. También es necesario evaluar la efectividad de cada metodología y práctica sobre el desempeño de las empresas. Las prácticas exploradas son las prácticas de uso común, no las mejores o malas prácticas. Por esa razón, es necesario evaluar el impacto de cada práctica en las empresas y categorizarlas como prácticas 'buenas' o 'malas' en el contexto de la startup.

Agradecimientos

Los autores agradecen a la Universidad Politécnica Salesiana del Ecuador, Grupo de investigación de ingeniería de software e ingeniería del conocimiento (GIISIC).

Bibliografía

- [1] H. Haron and R. Modeling, “New Trends in Software,” no. September, pp. 2–3, 2014.
- [2] B. Montero Molina, H. Cevallos Vite, and J. Dávila Cuesta, “Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,” *Espirales Rev. Multidiscip. Investig. ISSN 2550-6862*, vol. 2, no. 17, pp. 114–121, 2018.
- [3] D. Carrizo and A. Alfaro, “Quality assurance method in a software development methodology: A practice approach,” *Ingeniare*, vol. 26, no. 1, pp. 114–129, 2018.
- [4] V. A. Cicirello, “Student Developed Computer Science Educational Tools as Software Engineering Course Projects,” *J. Comput. Sci. Coll.*, vol. 32, no. 3, pp. 55–61, 2017.
- [5] R. Tinoco Gómez, López, P. Pablo, and U. N. M. de S. M. P. Bacalla, Salas, “Criterios de selección de metodologías de desarrollo de software,” *Ind. Data*, vol. 13, no. 2, pp. 70–74, 2010.
- [6] M. Rujana, N. R. Franco, N. Tortosa, and G. Tomaselli, “Análisis sobre adopción de metodologías ágiles en los equipos de desarrollo en pymes del NEA,” *XVIII Work. Investig. en Ciencias la Comput.*, pp. 646–650, 2016.
- [7] S. Barney, K. Petersen, M. Svahnberg, A. Aurum, and H. Barney, “Software quality trade-offs: A systematic map,” *Inf. Softw. Technol.*, vol. 54, no. 7, pp. 651–662, 2012.
- [8] R. Florea and V. Stray, “A Global View on the Hard Skills and Testing Tools in Software Testing,” *Proc. - 2019 ACM/IEEE 14th Int. Conf. Glob. Softw. Eng.*

- ICGSE 2019*, pp. 143–151, 2019.
- [9] L. G. Molano Cetina, “Modelo De Gestión Para Empresas De Software En El Ecuador,” *Biomédica*, vol. 31, no. sup3.2, p. 233, 2011.
 - [10] M. Campos, “Métodos de investigación académica,” *Univ. Costa Rica*, pp. 8–9, 2017.
 - [11] D. Carrizo and C. Moller, “Estructuras metodológicas de revisiones sistemáticas de literatura en Ingeniería de Software: un estudio de mapeo sistemático,” *Ingeniare. Rev. Chil. Ing.*, vol. 26, pp. 45–54, 2018.
 - [12] J. M. Drake, “Programación orientada a objetos: Lenguajes, Metodologías y Herramientas,” *Santander, 2008*, pp. 0–36, 2008.
 - [13] W. Roel and K. Mens, “Report on the 1st Workshop WASDeTT at ECOOP 2008,” no. May 2014, 2008.
 - [14] B. A. Kitchenham, “Systematic review in software engineering: where we are and where we should be going,” *Proc. 2nd Int. Work. Evidential Assess. Softw. Technol. - EAST '12*, p. 1, 2012.
 - [15] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, “Sentiment Polarity Detection for Software Development,” *Empir. Softw. Eng.*, vol. 23, no. 3, pp. 1352–1382, 2018.
 - [16] D. Savchenko, J. Kasurinen, and O. Taipale, “Smart tools in software engineering: A systematic mapping study,” *2019 42nd Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2019 - Proc.*, pp. 1509–1513, 2019.