

Requisitos sostenibles en DevOps

Julio Silverio Suastegui, Facultad de ingeniería/Universidad autónoma de Guerrero, Chilpancingo de los Bravo, Guerrero, México, 13404241@uagro.mx

Rogelio Fernando Hernández Alarcón, Facultad de ingeniería/Universidad autónoma de Guerrero, Chilpancingo de los Bravo, Guerrero, México, 15529@uagro.mx

Eric Rodríguez Peralta, Facultad de ingeniería/Universidad autónoma de Guerrero, Chilpancingo de los Bravo, Guerrero, México, 09063@uagro.mx

Jorge Vázquez Galarce, Facultad de ingeniería/Universidad autónoma de Guerrero, Chilpancingo de los Bravo, Guerrero, México, 13216@uagro.mx

RESUMEN

La gestión de requisitos sostenibles en DevOps presenta la importancia de la integración de las prácticas de gestión de requisitos con DevOps para garantizar la sostenibilidad de los requisitos de software durante todo el ciclo de vida del software. En este artículo destaca la relevancia de la sostenibilidad de los requisitos, es decir, su capacidad de mantener su valor a lo largo del tiempo, más allá de la fase de desarrollo del software, se examinan los problemas comunes en proyectos de software, como la falta de visibilidad y el incumplimiento de los requisitos, y cómo la integración de las mejores prácticas de gestión de requisitos con las prácticas de DevOps puede abordar estos problemas. El enfoque en la gestión de requisitos en el ciclo de vida de DevOps permite mejorar la calidad del software y reducir su impacto ambiental, lo que se traduce en un menor costo del ciclo de vida del software y en una mayor satisfacción del usuario. Finalmente, se describen algunas herramientas y enfoques que permiten la integración de la gestión de requisitos en un ciclo de vida de DevOps, como el uso de historias de usuarios, el análisis de impacto y la automatización de pruebas. En resumen, el artículo concluye que la gestión de requisitos sostenibles en DevOps es una alternativa importante y relevante para garantizar la sostenibilidad de los requisitos de software.

ABSTRACT

Sustainable Requirements Management in DevOps presents the importance of integrating requirements management practices with DevOps to ensure the sustainability of software requirements throughout the software lifecycle. This article highlights the relevance of requirements sustainability, i.e., their ability to maintain their value over time, beyond the software development phase, examines common problems in software projects, such as lack of visibility and non-compliance of requirements, and how the integration of requirements management best practices with DevOps practices can address these problems. The focus on requirements management in the DevOps lifecycle allows for improved software quality and reduced software environmental impact, resulting in lower software lifecycle cost and higher user satisfaction. Finally, some tools and approaches that enable the integration of requirements management into a DevOps lifecycle are described, such as the use of user stories, impact analysis and test automation. In summary, the article concludes that sustainable requirements management in DevOps is an important and relevant alternative to ensure the sustainability of software requirements.

PALABRAS RESERVADAS

Requisitos, sostenibilidad, DevOps, ecológico, desarrollo de software.

KEYWORDS

Requirements, Sustainability, DevOps, Green, Software development.

INTRODUCCIÓN

DevOps (Developer + Operation) tiene origen en el año 2009 cuando se realizó la primera conferencia DevopsDays en Gante Bélgica, se habló y se remarcó el tema de la entrega de software con mayor frecuencia, así como una mejora de la colaboración por parte del personal de desarrollo y de operaciones, la implementación de actividades continuas de pruebas, integración y retroalimentación. Todo esto con el objetivo de minimizar el tiempo entre el inicio de un proyecto hasta la conclusión, haciendo uso de metodologías y herramientas según las necesidades de la empresa, en que los equipos pequeños y multifuncionales con objetivos definidos colaboran utilizando herramientas como Scrum o Kanban incrementando la Integración Continua (CI) y el Despliegue Continuo (CD). Para que DevOps sea eficaz, una organización debe seleccionar y estructurar adecuadamente los equipos, los flujos de trabajo, los requisitos de información y los criterios de evaluación. Intentar seguir las prácticas de DevOps mientras se realiza la planificación y evaluación de equipos y proyectos socava el proceso de DevOps y priva a la organización de muchos de sus beneficios. [5]



Figura 1. Ciclo DevOps (se extrajo de Google)

La comunicación es fundamental para que el proceso DevOps pueda ser de máxima utilidad, también hay que tener en cuenta los requisitos los cuales son indispensables para que pueda iniciar de forma satisfactoria la adopción de una nueva metodología, así también tener en cuenta la sostenibilidad que es especialmente de las características del desarrollo que asegura las necesidades del presente sin comprometer las necesidades futuras antes de embarcarse en la adopción de esta metodología. Por ejemplo, en el campo de la ingeniería de software [7] nos dice que, la calidad de la recopilación de requisitos influye directamente en el éxito de la ingeniería de software. Permite a los desarrolladores, programadores e ingenieros de software obtener una comprensión más profunda y decisiva de los problemas que hay que resolver y las funcionalidades que hay que implementar.

Las 5 dimensiones de la sostenibilidad en DevOps requieren un enfoque integrado y una visión a largo plazo que considere los impactos a nivel ambiental, económico, social, institucional y tecnológico. Al hacerlo, se puede lograr una práctica de DevOps más sostenible y equilibrada tal y como se muestra en la siguiente figura.



Figura 2. Las cinco dimensiones de la sostenibilidad

Existen manifiestos para la sostenibilidad como el manifiesto de karskalova que destaca la necesidad de adoptar un enfoque integral y holístico para abordar la sostenibilidad en la tecnología, considerando tanto los impactos ambientales y sociales como los económicos y tecnológicos. Además, el manifiesto destaca la importancia de la transparencia y la responsabilidad en el proceso de desarrollo y adopción de tecnologías, así como la importancia de la inclusión y la diversidad en el sector tecnológico. Por otro lado, la sostenibilidad es adoptada en las empresas en servicios en la nube por su versatilidad a la hora de poder virtualizar distintos entornos sin la necesidad de equipo físico de esa manera [1] analiza la computación en la nube como facilitadora de la sostenibilidad en las dimensiones social, medioambiental y económica. La computación en la nube se considera un factor importante para lograr la sostenibilidad mediante la promoción de la investigación académica, la facilitación del desarrollo empresarial y el fortalecimiento de la competencia empresarial. A lo largo de este artículo abordaremos tres temas importantes para DevOps en el capítulo uno se hablará de las metodologías que son más empleadas por las empresas así como los distintos casos de uso que se les dan, en el capítulo dos se hablará de las distintas herramientas para cada una de las fases del ciclo DevOps desde el inicio del proyecto hasta el producto final y para terminar en el capítulo tres se hará mención a los retos que se enfrenta DevOps en los cuales no solo se encuentran retos por falta de herramientas específicas sino también de recursos humanos tal como la buena coordinación entre los equipos a cargo de un proyecto y la administración de los recursos.

PLANTEAMIENTO DEL PROBLEMA

La gestión de requisitos sostenibles en DevOps es un enfoque que busca integrar los requisitos de sostenibilidad en el ciclo de vida del desarrollo de software. El problema en la gestión de requisitos sostenibles en DevOps es asegurar que los requisitos de sostenibilidad se integren de manera efectiva en el ciclo de vida del desarrollo de software, desde la planificación hasta la implementación y operación. Esto implica considerar aspectos como el impacto ambiental, la eficiencia energética, la accesibilidad, inclusión en el diseño y desarrollo de software, la monitorización y la mejora continua de estos requisitos a medida que el software evoluciona. Además, es importante asegurar que estos requisitos sostenibles se mantengan y se mejoren continuamente a medida que el software evoluciona y cambia.

Además, la gestión de requisitos sostenibles en DevOps se enfoca en la colaboración y la comunicación entre equipos de desarrollo, operaciones y otros stakeholders relevantes, para asegurar que los requisitos de sostenibilidad se integren de manera efectiva en la cultura y los procesos de DevOps.

En resumen, la gestión de requisitos sostenibles en DevOps es un enfoque integral que busca integrar los requisitos de sostenibilidad en el ciclo de vida del desarrollo de software, mejorar la colaboración y la comunicación entre equipos, y garantizar la monitorización y la mejora continua de estos requisitos.

MÉTODO

Protocolo

Para la realización de este artículo se ha seguido la metodología que marcan[6]. Esta metodología se organiza en cuatro etapas: (1) factibilidad, (2) planificación, (3) clasificación y (4) presentación de resultados, como se observa en la Figura 3.

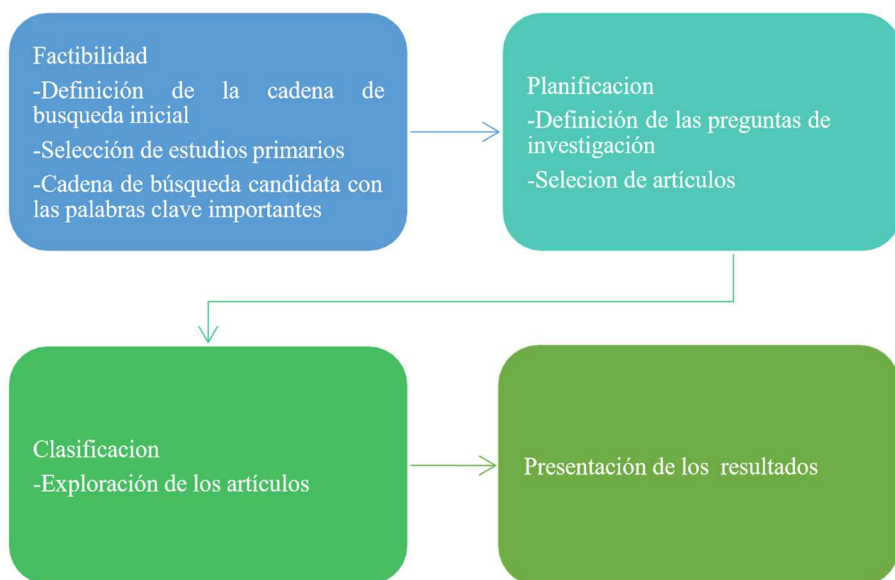


Figura 3. Fases del estudio de mapeo sistemático

2.1 Factibilidad

En primer lugar, se realizó un estudio de factibilidad para determinar si era de interés la realización de este artículo. El análisis se realizó tomando en cuenta las siguientes tareas:

- Determinar búsquedas iniciales para verificar si hay trabajos que hablen sobre el tema central de esta investigación.
- Identificar las palabras claves a partir de los resultados de las búsquedas iniciales, con el objetivo de tener una idea para definir la cadena de búsqueda.
- Definir la cadena de búsqueda inicial.
- Aplicar la cadena inicial a las bases de datos bibliográficas, para conocer el número de estudios.

La primera búsqueda previas y revisiones sistemáticas en el área de ingeniería de requisitos en DevOps y la sostenibilidad fue negativa, no se encontraron trabajos similares, lo que hace que esta propuesta sea única y pionera.

Con base en los trabajos identificados y que hacen referencia explícita a los términos DevOps, requisitos y sustentabilidad, se desarrolló y probó la cadena de búsqueda inicial que se presenta en la siguiente sección. De acuerdo con esta cadena, se procedió a buscar exclusivamente en los títulos de los trabajos, sin embargo, esto limitó considerablemente el número de trabajos mostrados a no más de cinco en algunas bases de datos, a lo que se recurrió a buscar los términos primarios (DevOps, requisitos y sostenibilidad) en el abstract y palabras clave, existieron casos que se tuvo que buscar en el cuerpo del trabajo para identificar si tenía la orientación en el área de DevOps, requisitos y sostenibilidad. De esta forma fue posible obtener suficientes trabajos que realmente contribuyeron.

2.2 Planificación

En la etapa de planificación se propone el protocolo de búsqueda y recopilación de artículos basado en tres tareas: (1) definición de las preguntas de investigación, (2) definición de una estrategia de búsqueda para la selección de estudios primarios y (3) selección de estudios utilizando criterios de inclusión y exclusión. [6] recomiendan registrar la información de todas las etapas, y, por tanto, en esta etapa es necesario definir las herramientas (hojas de cálculo y gestor de referencia para este caso se utilizó Mendeley).

2.2.1 Establecimiento de las preguntas de investigación

De acuerdo con los objetivos de este estudio, las preguntas de investigación se centraron en categorizar y estructurar los trabajos relacionados con DevOps y requisitos. Para ello, se identificaron cinco preguntas de investigación, como se muestra en la Tabla 1.

Preguntas de investigación

RQ1 ¿Qué metodologías se utilizan para la gestión de requisitos en DevOps con visión de la sostenibilidad?
RQ2 ¿Qué herramientas son utilizadas para la gestión de requisitos en DevOps usando empleando la sostenibilidad?
RQ3 ¿A qué retos se enfrenta la gestión de requisitos en la sostenibilidad usando DevOps?

Tabla 1. Preguntas de investigación

2.2.2 Definición de las estrategias de búsqueda de estudios primarios

La cadena de búsqueda final se compone de dos partes: (1) un término “requirement*” que se refiere a la gestión de requisitos contemplando todas las actividades que existen en dicho proceso, (2) un segundo término DevOps para particularizar los métodos, técnicas y herramientas de requisitos a contextos DevOps y (3) sustainability. Por consiguiente, la cadena de búsqueda se compone de la siguiente manera:

Requirements* AND Sustainability and DevOps And Green

En las búsquedas no se impuso restricciones de tiempo para poder contar con todos los trabajos publicados y no omitir algún artículo que sea de importancia para el presente estudio. Para la búsqueda de la información se recurrió a las bases de datos científicas de ACM Digital Library, IEEE Xplore, ScienceDirect, Springer, Wiley online library, Google Scholar.

Las bases de datos consultadas aseguran la calidad de sus artículos porque siguen un proceso riguroso de revisión y publicación. Cabe hacer mención que en algunas bases de datos científicas se tuvo que adaptar la cadena de búsqueda, porque cada base de datos científica tiene su propia sintaxis de búsqueda, en algunos casos se recurrió al uso de la búsqueda avanzada, pero se siguieron los términos mencionados con anterioridad.

El factor que influyó para la selección de los artículos fue que las palabras mencionadas en la cadena de búsqueda estuvieran presentes en alguna de las secciones de los trabajos seleccionados, las cuales fueron: (1) título del artículo, (2) introducción, (3) palabras clave o (4) conclusiones.

2.2.3 Criterios de inclusión y exclusión

Los criterios de inclusión son las características que deben tener los artículos candidatos a estudiar, mientras que los criterios de exclusión son las características que hacen que se rechace alguno de los artículos candidatos. Los criterios de inclusión y exclusión para este estudio son fáciles de catalogar y no requieren mayor complejidad para su interpretación. Los criterios para la selección de los artículos deben de garantizar información que aporte valor a las respuestas a las preguntas de investigación presentadas. A continuación, se describen los criterios para realizar la evaluación de los artículos: I.C.1.- Estudio que tenga relación con la gestión de requisitos en entornos de DevOps.

I.C.2.- Trabajo publicado en alguna revista de investigación indexada o en alguna conferencia relevante en el área de estudio.

I.C.3.- Artículo publicado en inglés.

Los siguientes criterios son los que sirvieron para determinar los estudios que se excluyeron: E.C.1.- El artículo tiene menos de 4 páginas.

E.C.2.- El artículo extiende otro trabajo, en cuyo caso, se seleccionó solamente la versión extendida.

E.C.3. El artículo está duplicado.

Resultado de la búsqueda se identificaron 1966 artículos como se muestra en la Figura 4, se identificaron 56 trabajos potencialmente relevantes, finalmente se seleccionaron 19 trabajos después de superar los criterios de inclusión y exclusión como se muestra en la Tabla 2.

Scientific database	Search string	Filters used	No. Papers obtained	candidates	useful
ACM Digital Library	Requirements* AND Sustainability AND DevOps AND Green		41	7	4
IEEE xplore	Requirements* AND DevOps		49	26	6
ScienceDirect	Requirements AND Sustainability and DevOps And Green	Se usó búsqueda avanzada	48	6	3
Springer	Requirements* AND Sustainability and DevOps And Green	Ingles, chapter,article	43	10	3

Wiley online library	Requirements* AND Sustainability and DevOps And Green	ninguno	25	7	3
Google Scholar	Requirements* AND Sustainability and DevOps And Green	ninguno	1760		
Total			1996	56	19

Tabla 2. Tabla de estudios seleccionados

Realizada la lectura de los artículos con base a los puntos mencionados anteriormente, se determinó un total de 56 estudios candidatos que superaron los criterios de inclusión. Paso seguido se aplicaron los criterios de exclusión para al final descartar 37 trabajos, obteniendo 19 trabajos finales.

2.3 Esquema de clasificación- estrategia de extracción de datos.

Según el esquema de clasificación sugerido por [4] La estrategia de extracción de datos se basa en la exploración de determinadas partes de cada artículo (resumen, introducción y resultados). Solo en aquellos casos en los que la información no es clara o no está bien estructurada, se realizó una lectura completa del trabajo.

La estrategia de extracción de información consistió en catalogar cada pregunta de investigación en criterios más específicos. De esta manera, se estableció un procedimiento mediante el cual se crearon términos más específicos para cada pregunta de investigación para facilitar la extracción y categorización de los artículos.

DESARROLLO

METODOLOGIAS

Hoy en día existen varias tecnologías competentes y contrastadas en el ámbito de las TI, por lo que la selección de las tecnologías y metodologías como nos dice [7] No sólo hay que elegir las tecnologías, sino también las metodologías. En otras palabras, la incorporación y el uso de la tecnología deben hacerse con toda seriedad y sagacidad; de lo contrario, aunque las tecnologías elegidas sean sólidas, los proyectos no alcanzarán el éxito inicialmente previsto. Muy pocas tecnologías pudieron sobrevivir y contribuir copiosamente durante mucho tiempo. En primer lugar, la complejidad intrínseca de la utilización global de las tecnologías y la falta de innovaciones de renombre son las principales razones de su abyecto y estrepitoso fracaso y del consiguiente destierro en el aire. Así pues, hay que tener muy en cuenta factores como la idoneidad, adaptabilidad, sostenibilidad, sencillez y extensibilidad de las tecnologías a la hora de elegir tecnologías y herramientas para proyectos a escala empresarial, transformacionales y de misión crítica. A continuación, se presenta una tabla con algunas de las metodologías comunes utilizadas en DevOps y su descripción:

Metodología	Descripción
Scrum	Una metodología ágil que se centra en el desarrollo de software en equipos auto-organizados. Se basa en sprints cortos, una planificación diaria y una revisión continua.
Kanban	Un enfoque visual para la gestión de tareas que se centra en la mejora continua del flujo de trabajo. Se basa en un tablero Kanban y limitaciones de WIP (trabajo en proceso).
Agile	Un enfoque para el desarrollo de software que se centra en la entrega continua, la adaptabilidad y la colaboración. Se basa en principios y valores como el desarrollo ágil y el pensamiento ágil.

Tabla 3. Metodologías usadas en DevOps.

Es de importancia saber las metodologías usadas en DevOps, no obstante, también es de vital importancia conocer en qué consisten las diferentes partes que integran el ciclo de vida DevOps tal y como se puede observar en la figura 1, donde se hace alusión a estas. A continuación, se presenta una tabla con las partes del ciclo de vida de DevOps y su explicación:

Parte del ciclo de vida	Explicación
Planificación y diseño	Identificar y planificar los requisitos del software y los objetivos de negocio, diseño de la arquitectura del sistema, elección de tecnologías y herramientas.
Integración continua	Automatización de la integración y compilación de código para detectar problemas temprano en el proceso de desarrollo.

Entrega continua	Automatización del proceso de despliegue para poner nuevas características en producción rápidamente y con confianza.
Infraestructura como código	Utilizando herramientas para automatizar la configuración y el escalamiento de la infraestructura.
Monitoreo y supervisión	Recopilando y analizando datos de rendimiento en tiempo real para detectar y resolver problemas de forma rápida.
Gestión de configuración	Utilizando herramientas para controlar y automatizar la configuración de la infraestructura y los servicios
Pruebas automatizadas	Automatizando las pruebas para asegurar la calidad del software antes de desplegarlo en producción.
Análisis de métricas y retroalimentación	Utilizando métricas para evaluar el rendimiento del sistema y retroalimentar los cambios en el proceso de desarrollo.

Tabla 4. Metodologías usadas en DevOps.

HERRAMIENTAS

Para entender el uso de las herramientas usadas durante el ciclo de vida de DevOps tenemos que tener en cuenta lo que dice [10] Cuando se habla de DevOps, se menciona el término de "pipeline". Este proceso no es específico de DevOps y se aplica a cualquier desarrollo y despliegue de aplicaciones. Dependiendo de su fuente, este pipeline tendrá de tres a cinco partes diferentes, en esta ocasión hablaremos de cuatro partes o pasos,

1. Development
2. Build
3. Test
4. Release.

Los pasos 1 y 2 son la parte Dev de DevOps. Los pasos 3 y 4 son la parte Ops de DevOps. En el proceso clásico de desarrollo de aplicaciones, se trata de equipos separados y a menudo aislados entre sí. Estos silos son a menudo diseñados porque los dos equipos pueden tener objetivos y prioridades contradictorios. Los desarrolladores están incentivados para producir nuevas características y corregir errores; es decir, el cambio es generalmente aceptable. Las operaciones de TI, por otro lado, están incentivadas para mantener las cosas en funcionamiento y estables; es decir, el cambio es malo. DevOps es, en su mayor parte, una solución a este conflicto.

Categoría	Herramienta
Source tools	GitHub, Apache Subversion
Build	Jenkins, Microsoft Visual Studio
Test	Selenium, Pester, Vagrant
Release	Chef, Puppet

Tabla 5. Herramientas usadas en el Pipeline.

A continuación, se presenta una tabla con algunas herramientas comunes utilizadas en diferentes partes del ciclo de vida de DevOps:

Parte del ciclo de vida	Herramientas
Planificación y diseño	Jira, Trello, Asana, Microsoft Planner
Integración continua	Jenkins, Travis CI, CircleCI, GitLab CI/CD
Entrega continua	Ansible, Chef, Puppet, SaltStack

Infraestructura como código	Terraform, CloudFormation, Ansible Tower
Monitoreo y supervisión	Nagios, Prometheus, Grafana, Datadog
Gestión de configuración	Ansible, Puppet, Chef, SaltStack
Pruebas automatizadas	Selenium, JUnit, TestNG, Cucumber
Análisis de métricas y retroalimentación	New Relic, AppDynamics, Splunk, DataDog

Tabla 6. Ciclo de vida DevOps y herramientas más usadas.

La tabla mostrada anteriormente es solo un ejemplo y no es exhaustiva, hay muchas otras herramientas disponibles en el mercado y la elección de una herramienta específica dependerá de las necesidades específicas de su organización y proyecto.

Desarrollar: Basándose en el plan, los desarrolladores se centran en el desarrollo y la revisión del código del software y/o IaC. Normalmente, en esta fase el código se somete a frecuentes confirmaciones en repositorios de código, así como a pruebas de integración y unitarias basadas en herramientas de automatización de la compilación.

Verificación: La verificación es el proceso para evaluar si los artefactos de software cumplen los requisitos.

Pruebas: en esta fase, se realizan pruebas de automatización de forma continua para garantizar la calidad del artefacto de software. A diferencia de las pruebas tradicionales, esta fase puede incluir la liberación de versiones de prueba a parte de la base de usuarios finales, mediante pruebas canarias.

Despliegue: Esta fase se centra en el redespiegue continuo del software en el entorno de producción. Esta fase conlleva el problema de la gestión de la configuración de las plataformas y recursos de destino.

Operar: La operación en los ciclos DevOps se ocupa de la configuración y gestión de la aplicación de software después del despliegue, por ejemplo, el aprovisionamiento de recursos y el auto escalado. Los orquestadores y otros métodos en tiempo de ejecución pueden utilizarse para instanciar y adaptar automáticamente en tiempo de ejecución la topología y los componentes de la aplicación.

Monitorización: Supervisar el rendimiento de las aplicaciones desplegadas recopilando y analizando datos de uso, lo que puede ayudar a detectar e identificar excepciones y proporcionar información para mejorar iterativamente el software. El seguimiento y el diagnóstico continuos de los problemas de producción son importantes para guiar la evolución de la aplicación a lo largo de los ciclos de lanzamiento.

Las fases anteriores son cualitativamente similares a las que se llevan a cabo en las metodologías tradicionales de ingeniería de software, con la diferencia principal de que el proceso de liberación del artefacto de software es continuo y altamente automatizado (Alnafessah et al., 2021).

RETOS

A pesar de tener herramientas y metodologías aún existen retos que dificultan el camino para un buen proceso DevOps de los cuales no solo se encuentran en el ámbito de las metodologías y herramientas sino también de las empresas que van iniciando en la adopción de DevOps, uno de estos retos nos lo dice [7] La mayoría de los entornos informáticos de las empresas de todo el mundo están sobrecargados, son cerrados, inflexibles, estáticos, complejos y caros. Por tanto, los retos empresariales y de TI (Tecnologías de la información) son cómo hacer que las TI sean elásticas, ampliables, programables, dinámicas, modulares y rentables. Sobre todo, teniendo en cuenta que las empresas de todo el mundo recortan sus presupuestos de TI año tras año, al equipo de TI de la empresa no le queda otra opción que embarcarse en un viaje meticuloso y medido para conseguir más con menos a través de una serie de soluciones tecnológicas pioneras y prometedoras.

Dado que DevOps no es una metodología con plataformas y herramientas estándar para cada uno de los diferentes entornos en los que se puede desplegar es necesario que los equipos de desarrollo tengan una buena comunicación, no solo del lado de desarrollo sino también del equipo de operaciones ya que tal como lo dice [8], la adopción con éxito de DevOps requiere un cambio de mentalidad, apoyo de la dirección y responsabilidad compartida entre todas las partes implicadas en la entrega de software. El apoyo de la alta dirección refuerza el valor y la adopción de DevOps en toda la organización. Sin embargo, la organización puede encontrarse con varios retos durante la adopción de DevOps, incluida la necesidad de rearchivar el software para mejorar el soporte de DevOps en cuanto a capacidad de despliegue, comprobabilidad y flexibilidad. La arquitectura de

microservicios es un diseño arquitectónico que ofrece ventajas como la escalabilidad, la flexibilidad y la portabilidad, y es compatible con DevOps.

Otro aspecto a tomar en cuenta él es lado de la sostenibilidad dado que es un amplio campo que abarca desde iniciativas para reducir costos de producción, ahorro de recursos humanos y de maquinaria, uno de estos aspectos implica la ingeniería de software así como nos dice [1] La arquitectura de sistemas sostenibles en general es uno de los grandes retos de la ingeniería de software moderna también de la necesidad de una visión más completa que integre las distintas dimensiones teniendo en cuenta sus relaciones y posibles compensaciones, y permita investigar más a fondo las relaciones de otras cualidades y métricas del software.

Más allá de estos problemas generales de sostenibilidad en la arquitectura del software, hay otros que surgen debido a la infraestructura, las plataformas y el software que se ofrecen y utilizan como servicio, tanto dentro como fuera de las instalaciones. Para empezar, la inherente difusión de los límites de responsabilidad en las relaciones de pro-suming (pro-suming neologismo que combina las palabras "producing" y "consuming" se refiere a la práctica de producir productos o servicios mientras se consumen, lo que permite una mayor participación y control por parte del consumidor en la producción de bienes y servicios.)

1. Sin embargo, cuando los servicios se consumen como parte de la implementación de dicho servicio, existen dependencias externas obvias que, en principio, están fuera del control de las partes interesadas en el servicio.
2. En muchos casos, existe una tensión entre la responsabilidad ante los interesados/propietarios del servicio en términos de generación de ingresos suficientes como justificación de la existencia continuada del servicio, y las responsabilidades ante la sociedad en general y el medio ambiente.

A continuación, se presenta una tabla con algunos de los desafíos comunes que se enfrentan en DevOps y su sostenibilidad:

Retos	Sostenibilidad
Falta de colaboración y comunicación entre equipos de desarrollo y operaciones	Puede conducir a una falta de eficiencia y eficacia en el proceso de desarrollo y a problemas de sostenibilidad en el software
Falta de automatización de procesos	Puede conducir a errores humanos y a una falta de eficiencia en el proceso de desarrollo y a problemas de sostenibilidad en el software
Falta de monitoreo y supervisión	Puede conducir a problemas de rendimiento y problemas de sostenibilidad en el software
Falta de análisis de impacto ambiental	Puede conducir a problemas de sostenibilidad ambiental en el software
Falta de objetivos y metas sostenibles	Puede conducir a una falta de progreso en la sostenibilidad del software
Falta de implementación de cambios sostenibles en el ciclo de vida del software	Puede conducir a problemas de sostenibilidad en el software a lo largo de su ciclo de vida

Tabla 7. Retos DevOps y su sostenibilidad

CONCLUSIONES

DISCUSIÓN

La discusión en el artículo se centra en cómo la gestión de requisitos sostenibles puede mejorar la calidad del software y reducir los costos del ciclo de vida del desarrollo de software.

El autor [1] argumentan que la gestión de requisitos sostenibles puede ayudar a mejorar la calidad del software mediante la eliminación de requisitos redundantes y contradictorios, lo que a su vez puede reducir el tiempo y los recursos necesarios para el desarrollo y la validación del software. Además, la aplicación de prácticas sostenibles también puede ayudar a reducir los costos del ciclo de vida del desarrollo de software, sin embargo, [3]nos dice que también es importante considerar los requisitos relacionados con las aplicaciones concretas que se están desarrollando ya que se pueden evitar problemas costosos en etapas posteriores del ciclo de vida del software.

Los autores [5]y [9] también discuten la importancia de la colaboración y la comunicación en la gestión de requisitos sostenibles en DevOps. Argumentan que la comunicación efectiva entre los equipos de desarrollo y los interesados en el software es esencial para garantizar que los requisitos sean entendidos y priorizados adecuadamente. Además, la colaboración entre los equipos de desarrollo y los interesados en el software puede ayudar a identificar requisitos redundantes y contradictorios. En resumen, la propuesta presentada en el artículo puede apoyar a la mejorar de la calidad del software y reducir los costos del ciclo de vida del desarrollo de software mediante la aplicación de prácticas sostenibles y la mejora de la comunicación y la colaboración entre los equipos de desarrollo y los interesados en el software. Sin embargo, se requiere una mayor investigación y evaluación de la propuesta para validar su efectividad en entornos de desarrollo de software reales.

CONCLUSIONES

Este artículo destaca la importancia de la integración de las prácticas de gestión de requisitos con DevOps para garantizar la sostenibilidad de los requisitos de software. La sostenibilidad de los requisitos se refiere a su capacidad de mantener su valor a lo largo del tiempo, más allá de la fase de desarrollo del software. En este sentido, se resalta cómo la falta de visibilidad y el incumplimiento de los requisitos son problemas comunes en proyectos de software, y cómo la integración de las mejores prácticas de gestión de requisitos con las prácticas de DevOps puede abordar estos problemas.

El enfoque en la gestión de requisitos en el ciclo de vida de DevOps permite mejorar la calidad del software y reducir su impacto ambiental, lo que se traduce en un menor costo del ciclo de vida del software y en una mayor satisfacción del usuario. Además, se describen algunas herramientas y enfoques que permiten la integración de la gestión de requisitos en un ciclo de vida de DevOps, como el uso de historias de usuarios, el análisis de impacto y la automatización de pruebas.

La gestión de requisitos sostenibles en DevOps es una alternativa importante y relevante para garantizar la sostenibilidad de los requisitos de software. La integración de las prácticas de gestión de requisitos con DevOps permite una mejor colaboración y comunicación entre los equipos de desarrollo y operaciones, lo que a su vez puede llevar a un desarrollo de software más sostenible y exitoso.

REFERENCIAS

- [1] Aiello, M. et al. eds. 2021. *Next-Gen Digital Services. A Retrospective and Roadmap for Service Computing of the Future*. Springer International Publishing.
- [2] Alnafessah, A. et al. 2021. Quality-Aware DevOps Research: Where Do We Stand? *IEEE Access*. 9, (2021), 44476–44489. DOI:<https://doi.org/10.1109/ACCESS.2021.3064867>.
- [3] Ayerdi, J. et al. 2020. Towards a Taxonomy for Eliciting Design-Operation Continuum Requirements of Cyber-Physical Systems. *Proceedings of the IEEE International Conference on Requirements Engineering* (Aug. 2020), 280–290.
- [4] Buckingham Shum, S.J. et al. *LAK 2012 : proceedings of the 2nd International Conference on Learning Analytics and Knowledge, April 29 - May 2, 2012, Vancouver, British Columbia, Canada*.
- [5] Johnson, M. et al. 2020. Continuous testing and deployment for urban air mobility. *AIAA/IEEE Digital Avionics Systems Conference - Proceedings* (Oct. 2020).
- [6] Petersen, K. et al. *Systematic Mapping Studies in Software Engineering*.
- [7] Raj, P. et al. 2017. The requirements elicitation approaches for software-defined cloud environments. *Requirements Engineering for Service and Cloud Computing*. Springer International Publishing. 89–117.
- [8] Rodríguez, P. et al. 2019. Advances in Using Agile and Lean Processes for Software Development. *Advances in*

Computers. Academic Press Inc. 135–224.

- [9] Rodríguez, P. et al. 2019. Advances in Using Agile and Lean Processes for Software Development. *Advances in Computers*. Academic Press Inc. 135–224. [10] *Management and Technical Operations*.