

Método de gestión de requisitos para promover la sostenibilidad en DevOps: CRETS4DevOps

Rogelio Fernando Hernández Alarcón¹, Begoña Moros Valle², Joaquín Nicolás Ros²

¹Facultad de Ingeniería – Universidad Autónoma de Guerrero – Chilpancingo, México

²Departamento de Informática y Sistemas – Universidad de Murcia – España

rogeliofh@uagro.mx¹, bmoros@um.es², jnr@um.es²

Resumen. *La adopción de prácticas ágiles en contextos DevOps impacta positivamente en el desarrollo de software, aunque existen preocupaciones respecto a la gestión de los requisitos y la durabilidad de los desarrollos en estos contextos. Las organizaciones software enfrentan el desafío de entregas rápidas, continuas, junto con necesidades de sostenibilidad a largo plazo. El Manifiesto de Karlskrona ha subrayado la necesidad de integrar la sostenibilidad en todas las etapas del desarrollo de software, pero se carece de una comprensión compartida de la sostenibilidad entre todos los actores involucrados en el desarrollo, y además no existen metodologías de Ingeniería de Requisitos (IR) que consideren los factores de sostenibilidad en un contexto de prácticas continuas. Por todo ello presentamos CRETS4DevOps (Continuous Requirements Engineering and Technical Sustainability for DevOps), un método que integra la gestión de la sostenibilidad técnica en la IR en el marco de DevOps. CRETS4DevOps combina prácticas tradicionales de IR con prácticas ágiles en el marco de la IR continua, apoyándose en la reutilización de requisitos para fomentar la sostenibilidad técnica.*

1. Introducción

La Ingeniería de Requisitos (IR) está en constante evolución y se enfrenta a desafíos entre los que destaca la sostenibilidad del software. Los ingenieros de requisitos tienen la tarea de analizar los impactos del software en todas las dimensiones de la sostenibilidad (social, medioambiental, económica, técnica y humana), incluyendo el diseño y las posibles consecuencias (Becker et al., 2015; Silveira et al., 2022). Surge así el concepto de requisitos de sostenibilidad, "requisitos que pueden utilizarse para especificar el comportamiento del sistema (por ejemplo, requisitos que reduzcan el consumo de energía de un sistema), así como para influir en el comportamiento de los usuarios" (Venters et al., 2017).

Mireles et al. (2017) muestran propuestas para considerar la sostenibilidad en el campo de la IR, pero los ingenieros de requisitos suelen carecer de los conocimientos, la experiencia y el apoyo metodológico necesarios para actuar como facilitadores de un debate sobre los efectos de la sostenibilidad, especialmente si tenemos en cuenta contextos DevOps: *¿Qué aspectos de la sostenibilidad puede soportar el software? ¿Cómo podemos priorizar los requisitos de sostenibilidad?* (Duboc et al., 2020).

Un desafío presente en la mayoría de los equipos de ingeniería de software es que carecen de expertos en sostenibilidad medioambiental, lo que hace difícil abordar la complejidad de los problemas de sostenibilidad en los sistemas software. En ocasiones se

propone el uso de patrones de requisitos de sostenibilidad, que explican cuándo y cómo aplicar prácticas sostenibles en el desarrollo de software. Estos patrones recogen su relevancia, objetivos, ejemplos de implementación, debates y conexiones con otros patrones de requisitos de sostenibilidad (Roher & Richardson, 2013).

Bambazek et al. (2023) analizan las propuestas de integración de la sostenibilidad en la IR y determinan que una de las principales carencias es la ausencia de evaluación de cómo estas propuestas se pueden aplicar a los procesos de desarrollo ágiles. La carencia de prácticas sostenibles en proyectos ágiles motiva la necesidad de definir un framework de Scrum consciente de la sostenibilidad, que integre los aspectos de la sostenibilidad en las tareas de IR para favorecer la sostenibilidad del software desarrollado. La elección de Scrum está justificada al ser el framework de desarrollo ágil predominante en la actualidad (Garscha, 2021).

Dado que las prácticas ágiles son la base sobre la que se construye DevOps (Ebert et al., 2016) podríamos inferir que las carencias detectadas para las metodologías ágiles son trasladables al contexto de DevOps. Así, un estudio previo (Hernández et al., 2023) reveló que en DevOps no existe una metodología específica que permita llevar el seguimiento de todos los aspectos relacionados con los requisitos. Además, en este estudio también se concluye que es necesario un método específico de IR para DevOps que apoye la entrega continua y sostenible de requisitos, la reutilización de requisitos, la gestión adecuada de requisitos de calidad y la provisión de documentación de requisitos clara y actualizada en el marco de un proceso de IR continuo integrado en la cadena de herramientas DevOps. La trazabilidad de los requisitos es necesaria para mejorar la comunicación de los requisitos e implicar no sólo a los equipos de desarrollo, sino también a los equipos de operaciones y negocio, lo cual permitiría mejorar la obtención de requisitos de las áreas de negocio y operaciones.

En este estudio se propone CRETS4DevOps (*Continuous Requirements Engineering and Technical Sustainability for DevOps*), un método diseñado para gestionar los requisitos que promuevan la dimensión técnica de la sostenibilidad en entornos DevOps. CRETS4DevOps propone combinar las prácticas de IR tradicionales y las ágiles para paliar: (1) el problema de requisitos no explícitos en las especificaciones, al confiar las compañías en la comunicación directa entre los interesados (*stakeholders*) en lugar de apoyarse en la documentación (Moreira & Schneider, 2022); y (2) las limitaciones de las historias de usuario para la especificación de requisitos (Kasauli et al., 2021). CRETS4DevOps promueve la retroalimentación e integración continua de requisitos, otorgando a los interesados la capacidad de analizar y evaluar el progreso del proyecto en las diferentes etapas de su desarrollo, alineado a los requisitos.

2. Contexto de CRETS4DevOps

2.1 Ingeniería de requisitos continua y DevOps

Es importante que los métodos de IR consideren los procesos del negocio, faciliten la comunicación entre diferentes roles y promuevan la reutilización de requisitos en procesos continuos (Hernández et al., 2023). Son necesarios procesos continuos de planificación, construcción, operación, despliegue y evaluación para desarrollar productos que cumplan con las necesidades y expectativas de los clientes, faciliten la toma de decisiones y permitan identificar oportunidades de negocio (Barcellos, 2020; Silveira et al., 2022).

La entrega continua es vital en ingeniería de software, y un factor a tener en cuenta al adoptar la entrega continua es que exista un canal de comunicación entre los equipos, de modo que no exista desalineación entre departamentos e inflexibilidad en la comunicación, que puede afectar al éxito de las prácticas DevOps (Azad, 2022).

En el proceso de planificación continua se deben integrar múltiples fuentes de información, como los comentarios de los clientes y datos del mercado. Una parte crucial de este proceso es la retroalimentación continua, que incorpora las opiniones de los clientes en la planificación futura del producto. Para monitorear este proceso, se sugiere utilizar: (1) métricas como el tiempo de desarrollo; (2) tamaño del producto en desarrollo; (3) tiempo medio de recuperación en caso de fallo; y (4) rendimiento del producto desarrollado (Klotins et al., 2022). La implementación de *Quality Feedback Loop* (QFL) ayuda a que los equipos combinen herramientas de análisis de datos con la planificación del proyecto, para permitir a los gerentes supervisar el proceso completo y tomar decisiones informadas (Lopez et al., 2021).

La integración continua es el proceso que busca eliminar brechas entre el desarrollo y la entrega continua. Además, es necesario ampliar la visión de los requisitos a otros niveles, como el negocio. En esta línea BizDevOps indica que la continuidad debe existir no sólo en el proceso de software, sino también entre el software y los procesos estratégicos de la organización (Barcellos, 2020; Klotins et al., 2022). El analista de negocio incluye entre sus responsabilidades la comunicación continua (Forbrig, 2018).

Ya existen propuestas de flujo constante de gestión de requisitos, como la de Michailidis et al. (2021), que utilizan herramientas específicas para fomentar la colaboración y la creación de documentación de requisitos que permiten su reutilización, mejora y la comunicación continua dentro de la organización. Este método de gestión de requisitos permite la priorización, seguimiento y renegociación de requisitos, siendo crucial para adaptarse a los rápidos cambios en los entornos del proyecto. El resultado de esta propuesta es ayudar a las organizaciones a conocer la estructura del proceso de transformación digital desde el principio y tener una visión global de los pasos que hay que dar. Además, el enfoque adopta una perspectiva de continuidad y apertura al cambio.

2.2 Ingeniería de Requisitos y sostenibilidad

La sostenibilidad es la capacidad de perdurar, por lo que la sostenibilidad de un sistema describe qué continuará existiendo y funcionando, incluso cuando cambian las circunstancias. La sostenibilidad normalmente se compara con aspectos ambientales, pero cada vez es más claro que requiere la consideración simultánea de los recursos ambientales, el bienestar social e individual, la estabilidad económica y la durabilidad a largo plazo de la infraestructura técnica (Becker et al., 2016). Al considerar la sostenibilidad se tiene que pensar en las siguientes preguntas: *¿Qué se debe sostener? ¿Para quién? ¿Por cuánto tiempo? ¿A qué costo?* (Venters et al., 2017).

La IR debe tener un papel importante para promover la sostenibilidad. Sin embargo, existe una falta de directrices concretas que ofrezcan una descomposición tangible de la sostenibilidad. Actualmente no existe un soporte específico para analizar, diseñar, medir, manejar y respaldar de manera efectiva los requisitos de sostenibilidad (Saputri & Lee, 2020). Para fomentar un cambio en las prácticas de diseño sostenible en IR es importante transmitir a los interesados la importancia de la sostenibilidad y su influencia en el sistema en desarrollo, así como identificar y evaluar los desafíos actuales

en el software, con el objetivo de definir estrategias que permitan abordar la sostenibilidad a medio y largo plazo (Chitchyan et al., 2016; Karita et al., 2022; Silveira et al., 2022).

Existen estudios que muestran desafíos en la sostenibilidad (Mireles et al., 2017) que se centran en cómo el software puede impulsar el desarrollo sostenible (efectos de segundo orden) y en definir requisitos al inicio del desarrollo del software. Sin embargo, hay una falta de investigación en la validación de estos requisitos. En consecuencia, se tiene la necesidad de que los métodos de IR contemplen la sostenibilidad y proporcionen una estructura metodológica para formular requisitos de software basados en metas sostenibles. Se debe considerar la sostenibilidad desde múltiples perspectivas, asegurando que los sistemas no solo sean eficientes y amables con el medio ambiente, sino también duraderos para evolucionar de acuerdo con las necesidades futuras (Christoph, 2014).

Dentro de la sostenibilidad técnica, uno de los desafíos es la elicitación continua de requisitos, para mantener alineados los requisitos cambiantes de los usuarios. Así mismo, existe la preocupación de que los enfoques de IR actuales no abordan explícitamente la sostenibilidad en la evolución de los sistemas (Seyff et al., 2018). Requisitos de calidad que influyen en la sostenibilidad técnica son la continuidad, durabilidad, seguridad, usabilidad, eficiencia energética, escalabilidad, mantenibilidad, buenas prácticas en la evolución del sistema, optimización y uso de código abierto (Karita et al., 2022). En esta línea, Moreira et al. (2023) definen un catálogo de requisitos reutilizables de sostenibilidad técnica y social, que se ha representado utilizando modelos iStar; los aspectos relacionados con la sostenibilidad técnica que se proponen están alineados con el modelo de calidad del estándar ISO/IEC 25010:2011.

3. CREST4DevOps

De acuerdo con Karita et al. (2022), para alcanzar los objetivos de desarrollo sostenible, en la IR se deben incorporar actividades que permitan: (1) analizar el contexto de sostenibilidad en proyectos de software; (2) identificar a los interesados relacionados con la sostenibilidad, que se consideran “expertos en sostenibilidad” y se encargan de documentar los efectos del desarrollo del producto en las dimensiones de la sostenibilidad; y (3) conseguir los objetivos de sostenibilidad. Por un lado, modelar los objetivos de sostenibilidad proporciona una visión clara del sistema y permite verificar si se implementa según lo planeado; por otro lado, se puede incluir una sección dedicada a la sostenibilidad dentro del documento de especificación de requisitos del software. Todo esto permitirá que los requisitos contribuyan a los objetivos de sostenibilidad, ofreciendo un marco de referencia durante todo el proyecto y asegurando que la sostenibilidad permanezca como una prioridad central en todo el ciclo de vida del software.

CRETS4DevOps es un método que se propone para una gestión de requisitos continua en DevOps que fomente la sostenibilidad. Las actividades de CRETS4DevOps se representan en un infinito, como se muestra en la Figura 1, siguiendo la analogía de la filosofía de DevOps, poniendo de relieve el enfoque de continuidad. En esta propuesta juega un papel importante la reutilización de requisitos, en concreto los requisitos de sostenibilidad de la dimensión técnica. El soporte a la reutilización se propone mediante el uso de catálogos de requisitos reutilizables de sostenibilidad que preserven el conocimiento de los expertos en este campo.

Para identificar los requisitos de sostenibilidad son necesarios artefactos que ayuden a determinar el impacto de la sostenibilidad (Penzenstadler & Femmer, 2013).

Dado que los requisitos se suelen documentar en lenguaje natural en la industria (Franch et al., 2023), en esta propuesta preferimos esta aproximación en lugar de otras basadas en un lenguaje de dominio, ontologías o modelos iStar. En particular, en los catálogos se usará una especificación basada en el formato habitual de las historias de usuario.



Figura 1 Actividades CREST4DevOps

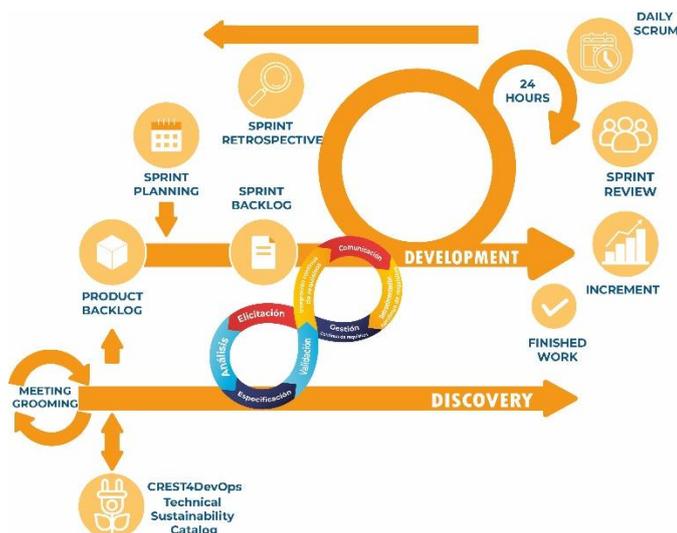


Figura 2 Integración de CREST4DevOps en Scrum

Siguiendo las recomendaciones de Karita et al. (2022), en CRETS4DevOps se incorpora el rol de *experto en sostenibilidad*, que será responsable de: (1) gestionar y mantener un catálogo de requisitos reutilizables de sostenibilidad; (2) ayudar a la definición de los objetivos de sostenibilidad del proyecto actual; y (3) alinear los requisitos de sostenibilidad con la legislación o estándares aplicables. Por otro lado, también se propone el rol de *analista del negocio*, integrando de este modo el enfoque BizDevOps (Forbrig, 2018) en la propuesta. Este rol será el responsable de mantener la visión global del proyecto y armonizar las necesidades de todos los interesados. Por último, el rol del *ingeniero de requisitos* será el responsable de la integración y entrega continua de requisitos a lo largo del proceso de desarrollo.

CRETS4DevOps incorpora las actividades típicas de la IR (SWEBOK 2022): *elicitación*, *análisis*, *especificación*, y *validación* de requisitos, considerando la sostenibilidad desde el principio, bien reutilizando los requisitos de sostenibilidad incluidos en el catálogo o bien definiendo requisitos nuevos con la ayuda del experto en sostenibilidad. Además, en la actividad de *integración continua de requisitos*, el experto en sostenibilidad, el analista del negocio y el ingeniero de requisitos colaboran conjuntamente para actualizar y validar de forma continua los requisitos del proyecto, incluyendo los requisitos de sostenibilidad, asegurando su integración continua en el desarrollo. La actividad de *comunicación continua* con los interesados del proyecto garantiza una comprensión común de los requisitos, lo cual permite un trabajo colaborativo entre los diferentes roles. La actividad de *retroalimentación continua* de requisitos implica recopilar comentarios, opiniones y sugerencias de los interesados, con el objeto de mejorar y refinar tanto los requisitos del proyecto como los requisitos reutilizables contenidos en el catálogo. Estas propuestas de cambio serán integradas durante la actividad de *gestión continua de requisitos*, encargada de controlar y

administrar los cambios en los requisitos garantizando su alineación con los objetivos del proyecto incluyendo los objetivos de sostenibilidad establecidos.

En la Figura 2 se representa la integración de CRETS4DevOps en Scrum. Se puede observar cómo el catálogo de requisitos reutilizables interviene en la línea de Descubrimiento (*Discovery track*) tanto para extraer requisitos de sostenibilidad que sean aplicables en el proyecto actual como para modificar y actualizar el contenido del catálogo. El infinito se representa abarcando las líneas de Descubrimiento y Desarrollo para significar que habrá una etapa durante el Descubrimiento, en la que se llevará a cabo la elicitación de requisitos que poblará el *Product Backlog*, pero que, durante el Desarrollo, de forma continua, se podrán definir nuevos requisitos o modificar los existentes, que se tendrán que integrar de forma continua en la especificación de requisitos actual. El ingeniero de requisitos será el encargado de supervisar y garantizar la integración continua de los requisitos a lo largo del proceso de desarrollo.

4. Conclusiones y trabajo a futuro

La integración de la sostenibilidad en IR con procesos de desarrollo ágil, en concreto en DevOps, es un problema que está todavía abierto. Para contribuir a cubrir este vacío, en este trabajo se propone CRETS4DevOps, un método para la gestión de requisitos continua enfocado a promover la dimensión técnica de la sostenibilidad en entornos DevOps. Los elementos que constituyen CRETS4DevOps se pueden integrar en diferentes métodos de desarrollo ágil, como por ejemplo Scrum. El uso de catálogos de requisitos reutilizables de sostenibilidad permite cubrir la ausencia de expertos en sostenibilidad en los equipos de desarrollo. La integración continua de requisitos a lo largo del proceso, asistida por el ingeniero de requisitos, permite mantener en todo momento la visión global del sistema que se está desarrollando, manteniendo la documentación clara y actualizada. No obstante, es necesaria la validación de la propuesta en un entorno real para potenciar su aplicación. Para ello, estamos trabajando en el desarrollo de una herramienta de soporte integrada en Azure DevOps que favorezca la integración continua de requisitos y la comunicación entre los interesados. Una vez desarrollada se intentaría implantar en un proyecto en la industria o, en su defecto, se evaluaría mediante una encuesta a expertos en DevOps.

Agradecimientos

Esta investigación es parte del proyecto OASSIS-UMU (PID2021-122554OB-C32) y la Red de Excelencia en Calidad del Software y Sostenibilidad (RED2022-134656-T), financiados por el Ministerio de Ciencia e Innovación y el Fondo de Desarrollo Regional Europeo.

Referencias

- Azad, N. (2022). Understanding DevOps critical success factors and organizational practices. *Proc. 5th Int. Workshop Softw.-Intens. Bus.: Sustain. Softw. Bus., IWSiB 2022*, 83–90. <https://doi.org/10.1145/3524614.3528627>
- Bambazek, P., Groher, I., & Seyff, N. (2023). Requirements engineering for sustainable software systems: a systematic mapping study. *Requirements Engineering*, 28(3), 481–505. <https://doi.org/10.1007/S00766-023-00402-1/>
- Barcellos, M. P. (2020). Towards a Framework for Continuous Software Engineering. *ACM Int. Conf. Proc. Ser.*, 626–631. <https://doi.org/10.1145/3422392.3422469>

- Becker, C., Betz, S., Chitchyan, R., & Duboc, L. (2016). Requirements: The key to sustainability. *IEEE Softw.*, 33(1), 56–65. <https://doi.org/10.1109/MS.2015.158>
- Becker, C., Chitchyan, R., Duboc, L., & Penzenstadler, B. (2015). Sustainability Design and Software: The Karlskrona Manifesto. *Proc. Int. Conf. Softw. Eng.*, 2, 467–476. <https://doi.org/10.1109/ICSE.2015.179>
- Chitchyan, R., Becker, C., Betz, S., & Duboc, L. (2016). Sustainability design in requirements engineering: State of practice. *Proc. Int. Conf. Softw. Eng.*, 533–542. <https://doi.org/10.1145/2889160.2889217>
- Christoph, B. (2014). Sustainability and longevity: Two sides of the same quality? *CEUR Workshop Proc.*, 1216, 1–6.
- Duboc, L., Penzenstadler, B., Porras, J., & Akinli Kocak, S. (2020). Requirements engineering for sustainability: an awareness framework for designing software systems for a better tomorrow. *Req. Eng.*, 25(4), 469–492. <https://doi.org/10.1007/s00766-020-00336-y>
- Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. *IEEE Software*, 33(3), 94–100. <https://doi.org/10.1109/MS.2016.68>
- Forbrig, P. (2018). BizDevOps and the role of S-BPM. *ACM Int. Conf. Proc. Ser.*, 1–8. <https://doi.org/10.1145/3178248.3178250>
- Franch, X., Palomares, C., Quer, C., Chatzipetrou, P., & Gorschek, T. (2023). The state-of-practice in requirements specification: an extended interview study at 12 companies. *Requirements Engineering*, 28(3), 377–409. <https://doi.org/10.1007/s00766-023-00399-7>
- Garscha, P. (2021). From Sustainability in Requirements Engineering to a Sustainability-Aware Scrum Framework. *Proc. IEEE Int. Conf. Req. Eng.*, 462–467. <https://doi.org/10.1109/RE51729.2021.00069>
- Hernández, R., Moros, B., & Nicolás, J. (2023). Requirements management in DevOps environments: a multivocal mapping study. *Requirements Engineering*. <https://doi.org/10.1007/s00766-023-00396-w>
- IEEE_Computer_Society. (2022). *SWEBOK Guide Version 4.0*. https://sebokwiki.org/wiki/SWEBOK_Guide_Version_4.0
- Karita, L., Mourão, B. C., & MacHado, I. (2022). Towards a common understanding of sustainable software development. *ACM Int. Conf. Proc. Ser.*, 269–278. <https://doi.org/10.1145/3555228.3555236>
- Kasauli, R., Knauss, E., Horkoff, J., Liebel, G., & de Oliveira Neto, F. G. (2021). Requirements engineering challenges and practices in large-scale agile system development. *Journal of Systems and Software*, 172. <https://doi.org/10.1016/j.jss.2020.110851>
- Klotins, E., Gorschek, T., Sundelin, K., & Falk, E. (2022). Towards cost-benefit evaluation for continuous software engineering activities. *Empir. Softw. Eng.*, 27(6). <https://doi.org/10.1007/s10664-022-10191-w>
- Lopez, L., Bagnato, A., Ahberve, A., & Franch, X. (2021). QFL: Data-Driven Feedback Loop to Manage Quality in Agile Development. *Proc. Int. Conf. Softw. Eng.*, 2021-May, 58–66. <https://doi.org/10.1109/ICSE-SEIS52602.2021.00015>

- Michailidis, K., Strazdina, R., & Kirikova, M. (2021). Continuous requirements engineering for digital transformation. *CEUR Workshop Proc.*, 2991, 26–40.
- Mireles, G. A. G., Moraga, M. A., Garcia, F., & Piattini, M. (2017). A classification approach of sustainability aware requirements methods. *Iber. Conf. Inf. Syst. Technol., CISTI*. <https://doi.org/10.23919/CISTI.2017.7975813>
- Moreira, A., & Schneider, K. (2022). Editorial. *Requirements Engineering*, 27(4), 403–404. <https://doi.org/10.1007/s00766-022-00392-6>
- Penzenstadler, B., & Femmer, H. (2013). A Generic Model for Sustainability with Process- and Product-specific Instances. *First Intl. Workshop on Green In Software Engineering and Green By Software Engineering*. <https://doi.org/10.1145/2451605.2451609>
- Roher, K., & Richardson, D. (2013). Sustainability Requirement Patterns. *IEEE Third International Workshop on Requirements Patterns (RePa)*. <https://doi.org/10.1109/RePa.2013.6602665>
- Saputri, T. R. D., & Lee, S. W. (2020). Addressing sustainability in the requirements engineering process: From elicitation to functional decomposition. *J. Softw.: Evol. Process*, 32(8), 1–25. <https://doi.org/10.1002/smr.2254>
- Seyff, N., Betz, S., Groher, I., & Stade, M. (2018). Crowd-focused semi-automated requirements engineering for evolution towards sustainability. *Proc. 2018 IEEE 26th Int. Reqs. Eng. Conf., RE 2018*, 370–375. <https://doi.org/10.1109/RE.2018.00-23>
- Silveira, C., Santos, V., Reis, L., & Mamede, H. (2022). CRESustain: Approach to Include Sustainability and Creativity in Requirements Engineering. *J. Eng. Res. Sci.*, 1(8), 27–34. <https://doi.org/10.55708/js0108004>
- Venters, C. C., Becker, C., Duboc, L., & Penzenstadler, B. (2017). Characterising sustainability requirements: A new species red herring or just an odd fish? *Proc. 2017 IEEE/ACM 39th Int. Conf. Softw. Eng.: SEIS Track, ICSE-SEIS 2017*, 3–12. <https://doi.org/10.1109/ICSE-SEIS.2017.2>