

Open Innovation and CrowdRE for Requirements Change Management in Software Ecosystems

Paulo Malcher
UNIRIO & UFRA
Rio de Janeiro, Brasil
malcher@edu.unirio.br

Davi Viana
(coorientador)
UFMA
Maranhão, Brasil
davi.viana@ufma.br

Rodrigo Pereira dos Santos
(orientador)
UNIRIO
Rio de Janeiro, Brasil
rps@uniriotec.br

ABSTRACT

Software ecosystems (SECO) have introduced complexity in requirements change management due to their open and dynamic nature. In SECO, multiple actors collaborate across organizational boundaries and can form distinct crowds that communicate requirements changes. This work proposes a method to assist professionals in requirements change identification in SECO (SECO-RCI) based on open innovation and crowd-based requirements engineering (CrowdRE). We applied Design Science to build SECO-RCI and evaluated its acceptance via a focus group with experts and its feasibility via a case study in a large real-world scenario. The focus group indicated positive acceptance of SECO-RCI, highlighting its ease of use, usefulness, and intention to use. The case study revealed the absence of a standardized approach for requirements change identification and emphasized the importance of centralized visibility across feedback channels. These findings suggest that SECO-RCI assists in addressing key challenges of requirements change in SECO. Consequently, SECO-RCI can improve software quality in complex and distributed development environments, promoting transparent and collaborative management of evolving requirements.

KEYWORDS

Requirements Change Management, Software Ecosystems, Open Innovation, CrowdRE

1 Introdução

Mudanças de requisitos são inevitáveis no desenvolvimento de software e impactam diretamente o custo, a qualidade e o cronograma dos projetos [1]. No contexto dos ecossistemas de software (ECOS), essas mudanças se tornam ainda mais complexas, pois alterações em um produto podem gerar efeitos em cadeia sobre diversos outros, provocando conflitos e resultados inesperados [2]. Os ECOS são ambientes abertos e dinâmicos que envolvem múltiplos componentes, plataformas e desenvolvedores [9].

A gestão de mudanças de requisitos em ECOS enfrenta desafios como escalabilidade e dificuldades de comunicação e colaboração [4, 15]. Métodos tradicionais de controle centralizado de mudanças não se adequam à natureza distribuída e dinâmica desses ambientes, o que evidencia a necessidade de novas abordagens que envolvam diversos atores nas atividades de engenharia de requisitos (ER) [2]. Nesse cenário, é proposto o SECO-RCI, um método para apoiar a identificação de mudanças de requisitos em ECOS com base na inovação aberta e na ER baseada em multidão (CrowdRE).

A inovação aberta promove o compartilhamento de conhecimento entre os atores do ecossistema [9], enquanto a CrowdRE

possibilita analisar requisitos de uma base heterogênea de usuários [7]. Integradas no SECO-RCI, essas abordagens ampliam a participação dos *stakeholders*, aumentam a validade dos requisitos e favorecem a detecção precoce de mudanças, contribuindo para a qualidade do software por meio de maior rastreabilidade, consistência e transparência no processo de ER.

2 Procedimentos Metodológicos

A pesquisa adotou os princípios da Design Science [17], especificamente a Design Science Research Methodology (DSRM) [16]. A Figura 1 ilustra o desenho do método de pesquisa segundo DSRM.

Na etapa de problema e motivação, foram realizadas seis iterações para compreender o problema e sua relevância: uma revisão informal da literatura, um estudo terciário, um mapeamento sistemático da literatura (MSL) sobre gestão de requisitos em ECOS e três estudos de campo. A partir dessas atividades, foram definidos os objetivos de uma solução destinada a apoiar a identificação de mudanças de requisitos em ECOS.

Na etapa de concepção e desenvolvimento, foi elaborado o método SECO-RCI, descrito na Seção 3. Um método pode ser entendido como uma abordagem para conduzir projetos de software ou sistemas, baseada em diretrizes, regras e heurísticas [3, 8]. Também foram desenvolvidos dois outros artefatos: (i) o modelo conceitual SECO-RM [14], que forneceu a base teórica do método; e (ii) a ferramenta SECO-RCR [6], que o instanciou por meio de técnicas de mineração de repositórios, processamento de linguagem natural e aprendizado de máquina para apoiar a identificação de mudanças.

A etapa de demonstração e avaliação foi conduzida no contexto de um ECOS de um instituto europeu de pesquisa aplicada, composto por mais de 70 unidades e 28 mil colaboradores. A avaliação não teve como objetivo comparar o SECO-RCI com outras abordagens, uma vez que não foram identificados métodos com os mesmos propósitos, mas verificar sua aceitação e viabilidade. Foram conduzidos dois estudos: um grupo focal com cinco especialistas, que possibilitou a coleta de percepções qualitativas sobre o método após uma demonstração, e um estudo de caso único, no qual um participante aplicou o SECO-RCI com apoio da ferramenta SECO-RCR e posteriormente foi entrevistado. Os resultados desta pesquisa foram disseminados em publicações científicas listadas na Seção 5.

3 Método SECO-RCI

O SECO-RCI é um método para identificar mudanças de requisitos em ECOS, fundamentado na inovação aberta e CrowdRE. A inovação aberta é incorporada por meio da análise de *feedback* de usuários externos provenientes de múltiplos canais, enquanto a CrowdRE

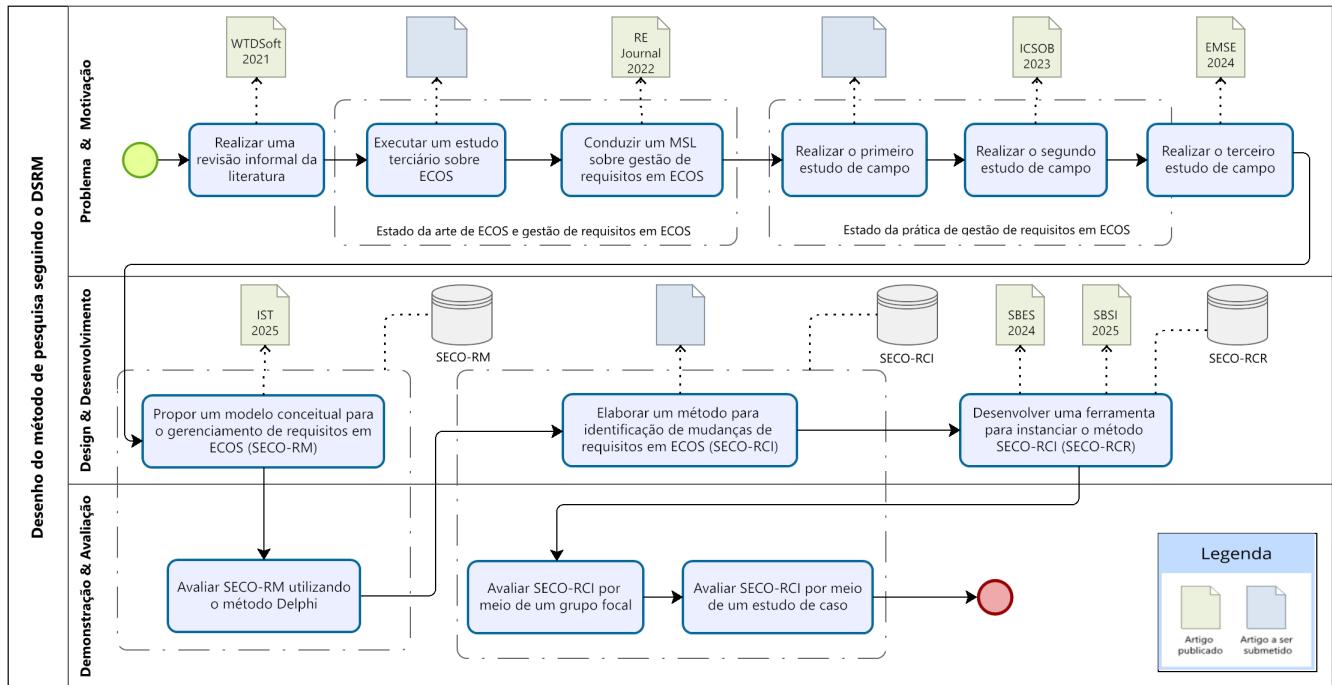


Figura 1: Desenho do método de pesquisa seguindo o DSRM.

permite processar contribuições de diferentes grupos de usuários (*niche players*) por meio de abordagens (semi)automatizadas.

O SECO-RCI foi desenvolvido com base na engenharia de métodos [3] e é composto por cinco atividades principais: (i) identificar práticas de inovação aberta e canais de *feedback* do ECOS; (ii) realizar a detecção de padrões em canais de *feedback* com base em CrowdRE; (iii) definir solicitações de mudanças de requisitos a partir dos padrões detectados; (iv) revisar essas solicitações com a participação de *niche players*, por meio de análise, votação e priorização; e (v) redefinir as solicitações com base nas revisões dos *niche players*, possibilitando o refinamento e a repriorização das mudanças propostas. Essas atividades envolvem dois papéis principais: o gerente de requisitos do ECOS, responsável por conduzir o processo, e os *niche players*, que colaboram avaliando e priorizando as mudanças. A descrição completa do método está disponível em [10].

4 Avaliação

O **Estudo #1 (Grupo Focal)** foi conduzido com cinco especialistas, que foram apresentados ao SECO-RCI e avaliaram sua aceitação com base em critérios como clareza, facilidade de uso, utilidade, potencial de redução de esforço, completude, intenção de uso e adaptabilidade. A maioria destacou a clareza e organização lógica das atividades, embora tenha sugerido o uso de diagramas BPMN (do inglês, *Business Process Model and Notation*), bem como a inclusão de cenários, para facilitar a compreensão, especialmente por usuários menos familiarizados com o processo. Também houve percepção positiva sobre a facilidade de uso, ainda que tenham sido apontadas dificuldades na priorização de mudanças propostas por *niche players* e a necessidade de incluir exemplos mais diversos.

Todos os participantes consideraram o método útil, enfatizando sua capacidade de apoiar a identificação de mudanças de requisitos em ECOS e de sistematizar a gestão de requisitos em ambientes complexos. A maioria reconheceu seu potencial para reduzir o esforço necessário ao processo, embora tenha havido dúvidas sobre sua efetividade em diferentes domínios. Houve divergências quanto à completude: enquanto alguns participantes consideraram o método abrangente e flexível, outro o avaliou como excessivamente complexo para determinados contextos, como projetos de software livre. Apesar disso, todos afirmaram que utilizariam o método em futuras oportunidades, reconhecendo sua aderência aos princípios de inovação aberta e CrowdRE.

A adaptabilidade do SECO-RCI recebeu avaliações divergentes: parte dos participantes considerou que suas atividades genéricas permitem aplicação em diferentes tipos de ECOS, enquanto outros indicaram que diferenças de porte, cultura e domínio podem exigir adaptações específicas para garantir sua efetividade. De modo geral, o estudo apontou uma percepção positiva do método, especialmente quanto à clareza, utilidade, facilidade de uso e intenção de uso, destacando que ele apoia de forma sistemática a identificação de mudanças de requisitos e se alinha aos princípios colaborativos e dinâmicos dos ECOS.

Estudo #2 (Estudo de Caso). Foi conduzido com dois profissionais de um instituto europeu de pesquisa aplicada: o chefe de departamento e o gerente de projetos de um ECOS. O chefe de departamento relatou que: (i) não existe um processo padronizado para identificar mudanças de requisitos, pois cada projeto executa essa atividade de forma distinta; (ii) a ausência de uma visão abrangente

das solicitações de mudanças entre projetos e canais de comunicação; e (iii) a dependência de diferentes pessoas para consolidar essas informações. Além disso, indicou que o uso de automação poderia reduzir essa dependência e facilitar a visualização das mudanças em diferentes níveis de detalhe.

O gerente de projetos aplicou o SECO-RCI em um caso real, usando uma ferramenta que dá suporte à execução. Após um treinamento, realizou todas as atividades previstas no método, incluindo a criação do ambiente, a detecção de padrões em repositórios GitHub, a análise dos resultados, o registro e a priorização das solicitações de mudanças de requisitos. Depois, foram coletadas suas percepções sobre o uso do método. O participante considerou a experiência de aprendizagem positiva, embora tenha apontado que a descrição do método é extensa e por vezes repetitiva. Reconheceu a clareza dos objetivos e atividades, mas sugeriu apresentar, para cada atividade, os insumos e produtos esperados, além de melhorias na configuração do ambiente de CrowdRE e na explicação do uso de filtros de palavras-chave para detecção de padrões.

O método foi considerado fácil de usar, principalmente por alinhar suas atividades a práticas já conhecidas, como a análise de *issues* no GitHub. Também foi visto como útil para apoiar a identificação de mudanças de requisitos em ECOS. Entretanto, o participante ressaltou a necessidade de maior interação com a ferramenta para aprender boas práticas de uso. Mencionou ainda que gestores de projetos maiores podem ter dificuldades por não possuírem conhecimento técnico prévio. O participante não soube avaliar o impacto do método no desempenho do trabalho. Embora a ferramenta tenha auxiliado na organização de *feedbacks* e na identificação de possíveis mudanças, o participante relatou dificuldade na configuração dos filtros e incerteza quanto à capacidade de agrupar solicitações sem conhecimento prévio do ECOS. Em síntese, o estudo reforçou a viabilidade prática do SECO-RCI, evidenciando seu potencial para apoiar a identificação de mudanças de requisitos, especialmente quando aliado a uma ferramenta de suporte.

5 Conclusão

A identificação de mudanças de requisitos em ECOS é desafiadora pela complexidade, interconexão entre componentes, diversidade de atores e dinamismo das necessidades dos usuários. Esses fatores afetam diretamente a qualidade do software, exigindo abordagens que considerem as características abertas e distribuídas dos ECOS, promovendo transparéncia, escalabilidade e colaboração.

Este trabalho contribui com três artefatos principais: (i) o modelo conceitual SECO-RM, que organiza os elementos e relações da gestão de requisitos em ECOS; (ii) o método SECO-RCI, que aplica princípios de inovação aberta e CrowdRE em atividades estruturadas para apoiar a identificação de mudanças de requisitos; e (iii) a ferramenta SECO-RCR, que viabiliza a aplicação prática do método. Avaliações com especialistas e um estudo de caso indicaram que o método é viável e útil para melhorar o alinhamento, a comunicação e a transparéncia entre os atores envolvidos.

Os resultados desta pesquisa foram disseminados por meio de publicações nos periódicos *Requirements Engineering* [11], *Empirical Software Engineering* [13] e *Information and Software Technology* [14], além dos anais de conferências como a *International Conference*

on Software Business (ICSOB) [12], o Simpósio Brasileiro de Engenharia de Software (SBES) [6] e o Simpósio Brasileiro de Sistemas de Informação (SBSI) [5]. A versão completa desta pesquisa pode ser consultada em [10] e em: <https://doi.org/10.5281/zenodo.14733348>.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001, CNPq (Procs. 316510/2023-8 e 311533/2025-6), FAPERJ (E-26/204.404/2024), FAPEMA (Procs. UNIVERSAL00745/19 e BEPP-03906/23), UNIRIO e UFRA.

REFERÊNCIAS

- [1] Syed Adnan Afaf and Mohammad Faisal. 2021. An Efficient Approach For Software Requirement Change Identification. *Webology* 18, 3 (2021), 1919–1926.
- [2] Christopher Bogart, Christian Kästner, James Herbsleb, and Ferdinand Thung. 2016. How to break an API: cost negotiation and community values in three software ecosystems. In *2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ACM, NY, USA, 109–120. doi:10.1145/2950290.2950325
- [3] Sjaak Brinkkemper. 1996. Method engineering: engineering of information systems development methods and tools. *Information and Software Technology* 38, 4 (1996), 275–280. doi:10.1016/0950-5849(95)01059-9 Method Engineering and Meta-Modelling.
- [4] Daniela Damian, Johan Linåker, David Johnson, Tony Clear, and Kelly Blincoe. 2021. Challenges and Strategies for Managing Requirements Selection in Software Ecosystems. *IEEE Software* 38, 6 (2021), 76–87. doi:10.1109/MS.2021.3105044
- [5] Eduardo Santos Gonçalves, Paulo Malcher, Pablo Oliveira Antonino, and Rodrigo Pereira Santos. 2025. Investigating requirements change requests. In *XXI Simpósio Brasileiro de Sistemas de Informação (SBSI)*. SBC, Porto Alegre, RS, Brasil, 733–742. doi:10.5753/sbsi.2025.246624.
- [6] Eduardo Gonçalves, Paulo Malcher, Laura Moraes, Davi Viana, and Rodrigo Pereira Santos. 2024. SECO-RCR: A Tool to Manage Requirements Change in Software Ecosystems. In *XXXVIII Simpósio Brasileiro de Engenharia de Software* (Curitiba, PR, Brasil). SBC, Porto Alegre, RS, Brasil, 782–788. doi:10.5753/sbes.2024.3606
- [7] Eduard C. Groen, Joerg Doerr, and Sebastian Adam. 2015. Towards Crowd-Based Requirements Engineering: A Research Preview. In *Requirements Engineering: Foundation for Software Quality*. Springer, Cham, Germany, 247–253. doi:10.1007/978-3-319-16101-3_16
- [8] Brian Henderson-Sellers, Jolita Ralyté, Pär J Ågerfalk, and Matti Rossi. 2014. *Situational method engineering*. Springer. doi:10.1007/978-3-642-41467-1
- [9] Slimer Jansen. 2020. A focus area maturity model for software ecosystem governance. *Information and Software Technology* 118 (2020), 106219. doi:10.1016/j.infsof.2019.106219
- [10] Paulo Malcher. 2024. *A Method for Supporting Requirements Change Management in Software Ecosystems Based on Open Innovation and CrowdRE*. Ph.D. Thesis. Federal University of the State of Rio de Janeiro (UNIRIO). doi:10.13140/RG.2.2.27404.53128
- [11] Paulo Malcher, Eduardo Silva, Davi Viana, and Rodrigo Santos. 2023. What do we know about requirements management in software ecosystems? *Requirements Engineering* 28 (2023), 567–593. doi:10.1007/s00766-023-00407-w
- [12] Paulo Malcher, Davi Viana, Pablo Oliveira Antonino, and Rodrigo Santos. 2023. Investigating Open Innovation Practices to Support Requirements Management in Software Ecosystems. In *Software Business*. Springer, Cham, Germany, 35–50. doi:10.1007/978-3-031-53227-6_3
- [13] Paulo Malcher, Davi Viana, Pablo Oliveira Antonino, and Rodrigo Santos. 2024. Investigating User Feedback from a Crowd in Requirements Management in Software Ecosystems. *Empirical Software Engineering* 29, 6 (2024), 152. doi:10.1007/s10664-024-10546-5
- [14] Paulo Malcher, Davi Viana, Pablo Oliveira Antonino, and Rodrigo Pereira Santos. 2025. Towards an understanding of requirements management in software ecosystems. *Information and Software Technology* (2025), 107672. doi:10.1016/j.infsof.2025.107672
- [15] Gabriel Matute, Alvin Cheung, and Sarah Chasins. 2022. Change in Software Ecosystems. In *Plateau Workshop*, 1–6. doi:10.1184/R1/19799314.v1
- [16] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. 2007. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems* 24, 3 (2007), 45–77. doi:10.2753/MIS0742-1222240302
- [17] Roel J. Wieringa. 2014. *What Is Design Science?* Springer Berlin Heidelberg, Berlin, Heidelberg, 3–11. doi:10.1007/978-3-662-43839-8_1