

Uma Abordagem para Gerência de Requisitos em Ecosystemas de Software

Paulo Malcher^{1,3} (Doutorando), Rodrigo Santos¹ (Orientador),
Davi Viana² (Coorientador)

¹Programa de Pós-Graduação em Informática (PPGI)
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Rio de Janeiro – RJ – Brasil

²PPGCC & DCCMAPI
Universidade Federal do Maranhão (UFMA)
São Luís – MA – Brasil

²Universidade Federal Rural da Amazônia (UFRA) – Campus Capitão Poço
Capitão Poço – PA – Brasil

Ano de ingresso: 2020 – Previsão Qualificação: 12/2021 – Previsão Defesa: 02/2024

paulo.malcher@edu.unirio.br, rps@uniriotec.com, davi.viana@ufma.br

Resumo. *A gerência de requisitos tem se tornado cada vez mais crítica devido a mudanças ocorridas nas últimas décadas no desenvolvimento de software. Novas abordagens de desenvolvimento de software têm surgido, entre elas os Ecosystemas de Software (ECOS). Em ECOS, são desenvolvidos múltiplos produtos em torno de uma plataforma tecnológica comum mantida por uma organização central que se relaciona com redes de atores e artefatos. As relações entre a organização central, seus atores externos e os usuários finais resultam em variados e emergentes fluxos de requisitos, importantes para a evolução do ecossistema e da sua plataforma. Isso faz com que gerenciar requisitos em ECOS seja considerado um desafio. Esta pesquisa de doutorado propõe uma abordagem de apoio a gerência de requisitos específica para ECOS. Essa abordagem conta com a definição de um método e sua sistematização por meio de uma ferramenta tecnológica de apoio. Para o desenvolvimento da pesquisa, está sendo utilizada a metodologia Design Science Research (DSR). Além disso, um estudo terciário, um estudo secundário e uma pesquisa de opinião estão sendo realizados. Por fim, estudos experimentais serão conduzidos aplicados para a avaliação dos artefatos propostos. Como contribuição, pretende-se oferecer uma abordagem de apoio específica para ECOS que auxilie gestores de requisitos e a comunidade de ECOS a entender como os requisitos emergem, são priorizados, modificados e rastreados.*

Palavras-chave: *Gerência de Requisitos, Ecosystemas de Software, Design Science Research*

Evento CbSoft: *SBES*

1. Caracterização do Problema

O desenvolvimento de um sistema de software é uma atividade complexa que precisa da coordenação de muitas pessoas para produzir produtos de trabalho interconectados [Wysocki 2011]. Lai e Ali (2013) consideram a gerência de requisitos como um dos processos mais dependentes de colaboração no desenvolvimento de software. A gerência de requisitos é um processo organizado de documentação, análise, rastreabilidade, priorização, controle de mudanças e comunicação dos requisitos. Problemas relacionados à gerência de requisitos como a existência de requisitos incompletos e/ou ocultos, ausência de rastreabilidade e falhas de comunicação foram identificados na literatura e são citados no trabalho de Fernández et al. (2017). Os autores ainda afirmam que as diferentes formas que a engenharia de requisitos é realizada a torna uma área de processo difícil de melhorar.

Lai e Ali (2013) apontam fatores cruciais para o sucesso da gerência de requisitos, tais como a rastreabilidade e a gerência de mudanças. Estes fatores têm se tornado ainda mais complexos em novas abordagens de desenvolvimento de software, tais como os Ecossistemas de Software (ECOS) [Linåker et al. 2020]. Em ECOS, são desenvolvidos múltiplos produtos, derivados de uma plataforma tecnológica comum baseada em uma arquitetura central integrados com outros sistemas por meio de redes de atores e artefatos [Manikas 2016]. Jansen (2009) destaca a existência de redes de influência e interoperabilidade em ECOS. Dessa forma, os requisitos surgem de maneira descentralizada e podem ser influenciados pelos múltiplos atores do ecossistema [Linåker et al. 2020]. Além disso, o fluxo tradicional de requisitos é complementado por fluxos variados e emergentes de requisitos, o que dificulta a definição, priorização, rastreabilidade e gerência de mudanças de requisitos por meio de abordagens tradicionais.

Axelsson e Skoglund (2016) e Manikas (2016) afirmam que a definição de requisitos é considerada um desafio, uma vez que as partes interessadas são múltiplas e distantes do gerenciamento do ECOS. Vegendla et al. (2018) explicam que ECOS devem possibilitar a comunicação entre uma rede de desenvolvedores e usuários que podem estar distribuídos globalmente e que demandam a todo momento novos requisitos. Por essa razão, os autores consideram que gerenciar requisitos em ECOS é desafiador. Knauss et al. (2014), por sua vez, sugerem que um bom ponto de partida para enfrentar esses desafios é o desenvolvimento de métodos e ferramentas para que as organizações comerciais otimizem suas engenharias de requisitos. Diante dos desafios, a questão que norteia esta pesquisa de doutorado é: “Como gerenciar requisitos em ECOS?”. Alinhado à questão de pesquisa, o objetivo geral deste trabalho consiste em: “propor uma abordagem para gerência de requisitos em ECOS que leve em consideração a influência exercida pelos múltiplos atores e os fluxos emergentes de requisitos”. No contexto dessa pesquisa, entende-se como abordagem o conjunto dos artefatos (método e ferramenta) a serem desenvolvido para auxiliar pesquisadores e profissionais na resolução de problemas específicos da área.

2. Fundamentação Teórica

2.1. Gerência de Requisitos

A gerência de requisitos contempla atividades que garantem que os requisitos sejam identificados, documentados, mantidos, comunicados e rastreados ao longo de todo o

ciclo de vida de um sistema, produto ou serviço [IEEE 2017]. Song (2017) define a gerência de requisitos como um processo que acompanha o planejamento e desenvolvimento de um sistema, capturando e mapeando a origem e o contexto das mudanças. Para Wiegers e Beatty (2013), a gerência de requisitos engloba as atividades que estão relacionadas ao controle de mudanças e de versão, à assistência diante do estado em que os requisitos se encontram e à rastreabilidade dos requisitos. Gerir os requisitos garante que as alterações sejam refletidas nos planos, atividades e produtos de trabalho do projeto [SEI 2010]. Yaseen et al. (2019) declaram que a gestão incorreta de requisitos está ligada a mais da metade dos motivos de falhas em projetos.

2.2. Ecossistemas de Software

Nas últimas décadas, o estudo de ECOS tem se tornado um campo de pesquisa ativo e definido pelo aparecimento de novos modelos de arquitetura de software, de colaboração de desenvolvedores e de negócios abertos [Hanssen e Dybå 2012]. Para Hanssen e Dybå (2012), a abordagem de ECOS representa um salto radical em como a engenharia de software está sendo feita, de maneira que o desenvolvimento está se tornando um processo aberto em um complexo ambiente distribuído. Santos et al. (2020) apresentam ECOS como uma nova forma de desenvolver sistemas oferecida pela engenharia de software. Por sua vez, Axelsson e Skoglund (2016) explicam que ECOS são baseados em um ambiente aberto e possuem uma gama de atores de diferentes organizações, interagindo em torno de uma plataforma aberta ou semiaberta, prosperando entre si de maneira cooperativa e competitiva e produzindo diversas soluções ou serviços de software.

A noção de ECOS se originou de conceitos de Ecossistemas da Ecologia e Manikas (2016) faz uma atualização na definição de ECOS como: “a interação de software e atores em relação a uma infraestrutura tecnológica comum, que resulta em um conjunto de contribuições e influencia direta ou indiretamente o ecossistema”. O autor ainda afirma que a atividade de cada ator é motivada pela criação de valor tanto para o ator quanto para o ecossistema. Além disso, cada ator tem um ou vários papéis no ecossistema e que este papel pode influenciar o ecossistema.

3. Metodologia

A metodologia de pesquisa utilizada neste projeto é inspirada na abordagem *Design Science Research* (DSR), que é um método experimental (baseado em evidência) para criação sistemática de soluções inovadoras [Hevner et al. 2004]. Em DSR, fluxo da pesquisa é direcionado do espaço-problema ao espaço-solução [Wieringa 2014]. Além disso, é uma metodologia que tem auxiliado diversas pesquisas nas áreas de Sistemas de Informação e Engenharia de Software [Horita et al. 2018].

Para Hevner (2007), três ciclos devem estar presentes e claramente identificáveis em um trabalho que utilize DSR como método de pesquisa. O autor os define como ciclos de relevância, do rigor e de *design*. Os ciclos de relevância atuam no espaço ambiente, ou seja, estão relacionados ao contexto da pesquisa. Os ciclos do rigor se relacionam com o espaço da base de conhecimento que, por sua vez, deve incorporar todo um corpo de conhecimento sobre a temática em questão. Por fim, os ciclos de *design* suportam as atividades relacionadas à construção e à avaliação do artefato [Hevner 2007].

Nesse sentido, as seguintes fases foram definidas para o desenvolvimento dessa pesquisa: (i) Caracterização inicial da pesquisa, a fim de reunir conceitos e definições sobre ECOS, gerência de requisitos e relações de influência; (ii) Estudo exploratório, com o objetivo de realizar uma investigação inicial sobre o senso de influência em ECOS e como ela pode afetar a gerência de requisitos; (iii) Revisões de literatura por meio de um estudo terciário para apresentar o estado atual das pesquisas em ECOS e de um mapeamento sistemático da literatura (MSL) para caracterizar a gerência de requisitos em ECOS; (iv) Pesquisa de opinião online com especialistas para identificar e confirmar características da gerência de requisitos em ECOS; (v) Desenvolvimento dos artefatos por meio da concepção de um método e ferramenta para gerência de requisitos em ECOS; e (vi) Avaliação dos artefatos. A Figura 1 apresenta a metodologia proposta e a relaciona aos ciclos propostos por Hevner (2007), bem como apresenta o estado atual de cada fase.

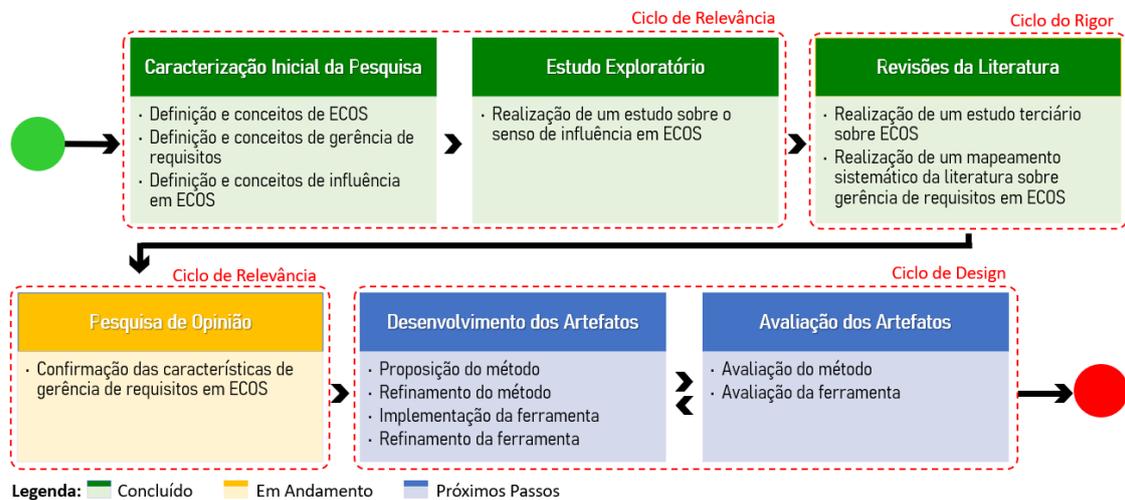


Figura 1. Metodologia proposta.

4. Método para Avaliar Resultados

A avaliação dos artefatos será realizada por meio da aplicação de estudos experimentais visando verificar a abordagem em relação ao seu objetivo [Shull et al. 2001]. A avaliação de um artefato em DSR está relacionada aos critérios de valor e utilidade, ou seja, a verificação do seu desempenho como solução desejada [March e Smith 1995]. Segundo Hevner et al. (2004), DSR não anseia alcançar verdades absolutas, grandes teorias ou leis gerais, mas procura identificar e compreender os problemas do mundo real e propor soluções apropriadas e úteis, fazendo avançar o conhecimento teórico da área. Dessa forma, os artefatos gerados passarão por avaliações intermediárias relacionadas aos requisitos, processo, práticas e modelo para gerência de requisitos em ECOS. Ao final de cada etapa, os resultados da pesquisa serão publicados e utilizados para a realização de ajustes nos elementos criados nas etapas anteriores.

5. Estado Atual do Trabalho

Esta pesquisa de doutorado teve início em março de 2020. Para compreensão do tema, foi realizada, no primeiro ano, a caracterização inicial da pesquisa por meio de revisões informais da literatura. Por meio dessa fase, foi possível compreender melhor o problema da pesquisa e seu contexto, além de auxiliar na construção do quadro teórico

da pesquisa e perceber que conceitos acerca de influência foram estudados na literatura de ECOS. Dessa forma, na fase seguinte, foi realizado um estudo exploratório sobre o senso de influência de desenvolvedores em um ECOS *open source* e publicado no Simpósio Brasileiro de Engenharia de Software (SBES) [Condina et al. 2020]. Este estudo auxiliou no entendimento das relações de influência que ocorrem entre os atores em ECOS. Além disso, ajudou a obter *insights* de como tais relações podem impactar a gerência de requisitos. A fase de revisões de literatura foi realizada e seus resultados são detalhados nas subseções seguintes. As demais etapas ainda não foram iniciadas, mas estão em processo de planejamento.

5.1. Estudo Terciário sobre ECOS

O estudo terciário sobre ECOS foi realizado para aprofundar o conhecimento a respeito de ECOS e apresentar o estado atual da área baseado em estudos secundários. O estudo seguiu a mesma metodologia de uma revisão sistemática da literatura e foi desenvolvido seguindo as diretrizes propostas por Kitchenham e Charters (2007) e utilizadas em [Kitchenham et al. 2010; Cruzes e Dybå 2011]. O objetivo do estudo foi fornecer um catálogo categorizado de estudos secundários sobre ECOS; analisar as características, qualidade e cobertura desses estudos; e analisar o impacto potencial dos resultados desses estudos sobre o progresso da pesquisa e da prática em engenharia de software. Para isso, pretendeu-se responder duas questões de pesquisa principais: “**(QP1)** *Quais são as características gerais e metodológicas dos estudos secundários sobre ECOS?*” e “**(QP2)** *Quais são as bases teóricas e os resultados dos estudos secundários sobre ECOS?*”. Um total de 518 estudos foram retornados e 22 estudos secundários foram incluídos. As respostas às questões de pesquisa foram definidas por meio da extração e síntese de dados.

Os resultados do estudo terciário indicam um crescimento na realização de estudos secundários com o passar dos anos, especialmente nos anos 2018-2019 com 4 e 7 estudos, respectivamente. Foi identificado que os estudos se concentraram em 10 tópicos de pesquisa, a saber: preocupações com qualidade e saúde em ECOS; ECOS *mobile*; inovação aberta; ECOS *open source*; arquitetura de software; governança; métricas; modelos de parceria; engenharia de requisitos; e estudos gerais. Além disso, a pesquisa em ECOS vem de outros tipos de ecossistemas, em particular ecossistemas naturais ou biológicos, ecossistemas de negócio, ecossistemas digitais e ecossistemas de negócio digitais. Além disso, segundo os resultados, ECOS influencia outros tipos de ecossistemas derivados, tais como ecossistemas de dados. Como principais variantes de ECOS foram identificadas os seguintes: ecossistemas de sistemas federados legados, ECOS *mobile*, ECOS *open source* e ecossistemas tecnológicos de saúde. Vale destacar que a maioria dos autores utilizou três dimensões para resumir suas conclusões, a saber: dimensão técnica, dimensão de negócios e dimensão social.

5.2. Mapeamento Sistemático sobre Gerência de Requisitos em ECOS

A fim de caracterizar a gerência de requisitos no âmbito de ECOS, um MSL foi realizado e seguiu as diretrizes propostas por Petersen et al. (2015). Para atender o objetivo do estudo, foi definida uma questão de pesquisa principal (QP): “*Como a gerência de requisitos tem sido realizada em ECOS?*”. Para ajudar a responder a QP, subquestões foram definidas: “**(SubQ-1)** *Que abordagens são utilizadas para a realização da gerência de requisitos em ECOS?*”; “**(SubQ-2)** *Como estas abordagens*

são avaliadas?”; e “(SubQ-3) *Quais as diferenças entre as abordagens no contexto de ECOS em relação àquelas utilizadas no cenário tradicional?*”. De um total de 901 estudos retornados, ao final da seleção, foram incluídos 18 estudos primários.

Os resultados obtidos mostraram que a gerência de requisitos em ECOS tem sido realizada com o auxílio de: Ferramentas (8 estudos); Modelos (5 estudos); Métodos/Metodologias (2 estudos); e Práticas (1 estudo). Além disso, os estudos incluídos foram classificados em: os que apresentam alguma forma de avaliação (3 estudos) e os que não apresentam avaliação (11 estudos). Entre os que não apresentam avaliação, alguns citam a necessidade de avaliações futuras (4 estudos). Por fim, foram identificadas algumas características sobre a gerência de requisitos em ECOS. Essas características, apontadas nos estudos incluídos, estão relacionadas à definição, priorização, comunicação, negociação, mudanças e rastreabilidade de requisitos, que tendem a ser diferentes da realizada em ambientes tradicionais.

5.3. Fundamentação do Método para a Gerência de Requisitos em ECOS

A partir do conhecimento adquirido por meio das revisões de literaturas, foi possível perceber que há uma falta de abordagens abrangentes para a gerência de requisitos e específicas para o contexto de ECOS. Dessa forma, os resultados das fases anteriores podem ser utilizados no *design* do artefato descrito nesta pesquisa com a hipótese de que ele pode atuar como um tratamento para o problema. O método para gerência de requisitos em ECOS ainda será formalizado e estruturado em etapas. Para isso, as características de ECOS (resultados do estudo terciário) e de gerência de requisitos em ECOS (resultados do estudo secundário) foram definidas e serão confirmadas por meio de uma pesquisa de opinião com especialistas que está em fase de planejamento. Nesse sentido, o método de gerência de requisitos em ECOS deve levar em conta que:

- A **definição e priorização de requisitos** ocorrem de forma colaborativa por meio de discussões e negociações interativas e transparentes comumente conduzidas pela organização central do ECOS e também podem envolver atores externos e dispersos geograficamente;
- A **comunicação de requisitos** é estruturada em rede e afetada pela heterogeneidade de atores desconhecidos e precisa ser realizada dentro e fora do ECOS. Além disso, é realizada por meio de ambientes e/ou canais de comunicação abertos para os múltiplos atores do ECOS;
- A **negociação de requisitos** é orquestrada pela organização central do ECOS e conta com a participação de atores externos, interdependentes e dispersos geograficamente;
- Os requisitos **não são dispostos em um repositório central** e as discussões e implementações de requisitos estão espalhadas por uma série de artefatos de requisitos, cada um com seu próprio repositório. Além disso, os **requisitos da plataforma tecnológica comum** têm prioridade sobre as extensões e aplicações e **requisitos de interface** devem ser definidos e gerenciados;
- A gerência de requisitos é **informal, descentralizada e evolutiva**, além de ser influenciada por grupos de **atores multipartidários** que podem ser externos à organização central e pelas **relações de poder** existentes entre eles; e

de caso no ecossistema Apache Hadoop OSS. O método utiliza construções de redes sociais para medir a influência e as interações dos *stakeholders* e considera as características especiais do *open source* e da engenharia de requisitos. Apesar disso, o estudo não apresenta um método específico para gerência de requisitos e não cita a implementação de uma ferramenta para esse contexto, o que é o objetivo desta proposta de pesquisa de doutorado.

Knauss et al. (2018) apresentam conclusões feitas por meio de um estudo experimental de método misto sobre práticas de gerência de requisitos no ECOS comercial aberto CLM (*Collaborative Lifecycle Management*) da IBM. O estudo mapeia e descreve o fluxo de requisitos no ECOS, além de descrever práticas e desafios relacionados à seleção de *stakeholders*, comunicação e priorização de requisitos. Além disso, o estudo fornece uma descrição detalhada dos processos de engenharia de requisitos e como os requisitos fluem por meio do ECOS CLM. Entretanto, o estudo não apresenta uma abordagem para o processo de engenharia de requisito em ECOS. Para os autores, o desenvolvimento de métodos e ferramentas específicas para ECOS é um bom ponto de partida para enfrentar os desafios da engenharia de requisitos apontados. Esse fato corrobora com o objetivo dessa proposta de pesquisa em propor uma abordagem que apoie a gerência de requisitos em ECOS e leve em consideração a influência exercida pelos os múltiplos atores do ECOS.

Referências

- Axelsson, J. e Skoglund, M. (2016). Quality assurance in software ecosystems: A systematic literature mapping and research agenda. *Journal of Systems and Software*, v. 114, pp. 69-81.
- Condina, V., Malcher, P., et al. (2020). An Exploratory Study on Developers Opinions about Influence in Open Source Software Ecosystems. *34th Brazilian Symposium on Software Engineering (SBES'20)*, pp. 137–146.
- Cruzes, D. e Dybå, T. Research synthesis in software engineering: a tertiary study. *Information and Software Technology*, v. 53 (2011), pp. 440–455.
- Fernández, D. M., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetrò, A. ... e Männistö, T. (2017). Naming the pain in requirements engineering. *Empirical software engineering*, v. 22 (5), pp. 2298-2338.
- Hanssen, G. e Dybå, T. (2012). Theoretical Foundations of Software Ecosystems. In *4th International Workshop on Software Ecosystems (IWSECO) – 3rd International Conference on Software Business (ICSOB)*, Cambridge, USA, pp. 6-17.
- Hevner, A. R., March, S. T., Park, J., e Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, v. 28 (1), pp. 75-105.
- Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems*, v. 19 (2), pp. 4.
- Horita, F. E. A., Neto, V. V. G., e Santos, R. P. (2018). Design Science Research em Sistemas de Informação e Engenharia de Software: Conceitos, Aplicações e Trabalhos Futuros. In *I Jornada Latino-Americana de Atualização em Informática*, pp. 191-210.
- IEEE. (2017). ISO/IEC/IEEE – System and Software. Engineering – Software Life Cycle Processes. *ISO/IEC/IEEE 12207:2017(E) First edition 2017-11*, pp. 1–157.
- Jansen, S., Brinkkemper, S., e Finkelstein, A. (2009). A Sense of Community: A Research Agenda for Software Ecosystems. In *31st International Conference on Software Engineering*, pp. 187-190.

- Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering, version 2.3. EBSE Technical Report
- Kitchenham, B., Budgen, D., Brereton, O., Turner, M., Niazi, M. e Linkman, S. 2010. Systematic literature reviews in software engineering – a tertiary study. *Information and Software Technology*, v. 52 (8), pp. 792–805.
- Knauss, E., Damian, D., Knauss, A., e Borici, A. (2014). Openness and requirements: Opportunities and tradeoffs in software ecosystems. *In 22nd International Requirements Engineering Conference (RE)*, pp. 213-222.
- Knauss, E., Yussuf, A., Blincoe, K., Damian, D., e Knauss, A. (2018). Continuous clarification and emergent requirements flows in open-commercial software ecosystems. *Requirements Engineering*, v. 23(1), pp. 97-117.
- Lai, R. e Ali, N. (2013). A Requirements Management Method for Global Software Development. *Advances in Information Sciences*, v. 1(1), pp. 38–58.
- Linåker, J., Regnell, B., & Damian, D. (2020). A method for analyzing stakeholders' influence on an open source software ecosystem's requirements engineering process. *Requirements Engineering*, v. 25(1), pp. 115-130.
- Manikas, K. (2016). Revisiting software ecosystems research: A longitudinal literature study. *Journal of Systems and Software*, v. 117, pp. 84-103.
- March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision support systems*, v. 15(4), pp. 251-266.
- Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, v. 64, pp. 1-18.
- Santos, R. P., Maciel, C., e Viterbo, J. (Eds.). (2020). *Software Ecosystems, Sustainability and Human Values in the Social Web: In 8th Workshop of Human-Computer Interaction Aspects to the Social Web, WAIHCWS 2017, Joinville, Brazil, October 23, 2017 and 9th Workshop, WAIHCWS 2018, Belém, Brazil, October 22, 2018, Revised Selected Papers* (Vol. 1081). Springer Nature.
- SEI (2010). CMMI for Development, Version 1.3 (CMU/SEI-2010-TR-033). *Software Engineering Institute, Carnegie Mellon University*. <http://www.sei.cmu.edu/library>. Visitado em maio de 2021.
- Shull, F., Carver, J., e Travassos, G. H. (2001). An empirical methodology for introducing software processes. *Software Engineering Notes*, v. 26(5), pp. 288–296.
- Song, W. (2017). Requirement management for product-service systems: Status review and future trends. *Computers in Industry*, v. 85, pp. 11-22.
- Vegendla A. et al. (2018). A Systematic Mapping Study on Requirements Engineering in Software Ecosystems. *Journal of Information Technology Research*, v. 11(1), pp. 135-142.
- Wieggers, K. e Beatty, J. (2013). *Software requirements*. Pearson Education.
- Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.
- Wysocki, R. K. (2011). *Effective project management: traditional, agile, extreme*. JohnWiley & Sons.
- Yaseen, M., Ali, Z. e Humayoun, M. (2019). Requirements Management Model (RMM): A Proposed Model for Successful Delivery of Software Projects. *International Journal of Computer Applications*, v. 178 (17), pp. 32-36.