

Estratégia de Produto de Software a partir da perspectiva da Gestão do Conhecimento: um Mapeamento Sistemático

Ana Katarine F. S. Santos¹, Ricardo A. C. Souza^{1,2}

¹Programa de Pós-graduação em Informática Aplicada (PPGIA) – Universidade Federal Rural de Pernambuco – (UFRPE) Rua Dom Manoel de Medeiros s/n – Dois Irmãos – 52.171-900 – Recife – PE – Brasil

²Departamento de Computação (DC) – Universidade Federal Rural de Pernambuco (UFRPE) – Recife, PE – Brasil

anakatarines@gmail.com, ricardo.souza@ufrpe.br

Abstract. *The Software Product Strategy is concerned with the constraints and internal (business) and external (market) perspectives to the organization that should be considered in software product planning. Knowledge Management can be used in the design of a software product through socialization in which the tacit knowledge of the different stakeholders (partners, clients, sponsors, etc.) is accumulated and shared so that this knowledge is articulated in a product concept through externalization. This paper presents an overview about software product development strategies supported by knowledge management mechanisms resulting from a systematic mapping of the literature.*

Resumo. *A Estratégia do Produto de Software se preocupa com as restrições e perspectivas internas (do negócio) e externas (do mercado) à organização que devem ser consideradas no planejamento de um produto de software. Já a Gestão do Conhecimento pode ser usada na concepção de um produto de software através da socialização no qual o conhecimento tácito dos diferentes stakeholders (parceiros, clientes, patrocinadores, etc.) é acumulado e compartilhado para, então, este conhecimento ser articulado em um conceito de produto por meio da externalização. Este trabalho apresenta um panorama sobre estratégias de desenvolvimento de produtos de software apoiadas por mecanismos de gestão do conhecimento, resultante de um mapeamento sistemático da literatura.*

1. Introdução

A busca por qualidade de software tem se tornado objetivo estratégico de várias empresas que almejam atender as demandas de mercado [Ferreira, 2010]. Nesse sentido, existem diversas propostas de soluções para auxiliar na qualidade do software, tais como, modelos, padrões e métodos de desenvolvimento de software. Muitos trabalhos propostos na literatura da engenharia de software e sistemas da informação são dedicados à análise de projetos que não cumpriram cronogramas, excederam seus orçamentos e, em alguns casos, entregaram produtos de software com baixa qualidade.

Outras soluções têm sido propostas por acadêmicos e profissionais para melhorar essa produtividade, principalmente com recomendações de mudança nos processos de criação do software [Fayad *et al.*, 2000; Marciniak, 2001]. Entre as soluções conhecidas, muitas são baseadas em ferramentas, abordagens, métodos, processos e notações de desenvolvimento de software [Martin *et al.*, 2007]. No entanto, o escopo dessas soluções pode ser limitado e as melhorias que elas fornecem nem sempre são significativas, isso porque essas abordagens são baseadas em uma visão mecanicista com rotinas fixas que muitas vezes não refletem a complexa realidade de todas as organizações. Além disso, a visão sistemática da engenharia de software induzida pelos métodos e abordagens conhecidos pode não considerar todas as dimensões do produto de software nas perspectivas econômicas, organizacionais e humanas [Ilavarasan *et al.*, 2003; Toffolon, 2000].

Borba *et al.* (2016) apontam que geralmente os processos de software enfatizam a implantação de funcionalidades exclusivamente a partir de requisitos dos usuários, sem levar em consideração questões relacionadas ao alinhamento com a estratégia de entrega de valor da organização e aos fatores de competição que regem o mercado. Segundo o *The Standish Group International*¹, entre os principais fatores de valor que contribuem para o sucesso de projetos de tecnologia da informação está a otimização de valor para o negócio. Nesse contexto surge a Gestão de Produto de Software (SPM do inglês *Software Product Management*) que é a disciplina que governa um produto (solução ou serviço) desde a concepção até a entrega para o cliente ou mercado visando gerar o maior valor possível para o negócio [Ebert, 2014].

A SPM faz a ponte entre a engenharia de software e o modelo de negócio, preocupando-se com a estratégia (análise do mercado, modelo de negócio, etc.) e planejamento (ciclo de vida, engenharia de requisitos, etc.) do produto de software [Ebert *et al.*, 2014; Fricker, 2012; Maglyas *et al.*, 2017]. A estratégia visa: especificar o valor entregue pelo produto de software em consonância com o modelo de negócios da organização, direcionar a evolução do produto de software em alinhamento com a estratégia da organização e especificar os requisitos e restrições de negócio (perspectiva interna da organização) e mercado (perspectiva externa à organização) que devem ser considerados no processo de software. Assim, a SPM, de acordo com a *International Software Product Management Association (ISPMA)*², contempla a integração entre a estratégia organizacional e a estratégia do produto de software.

Nesse contexto, onde a estratégia do produto de software depende do conhecimento e compreensão das perspectivas dos diferentes *stakeholders* (parceiros, clientes, patrocinadores, etc.) internos e externos a Gestão do Conhecimento (GC) pode ser usada na concepção do produto de software. Isto porque traz mecanismos que possibilitam a socialização e externalização do conhecimento acumulado pelos diferentes atores que influenciam na criação do conceito do produto. A gestão do conhecimento reflete assim um conjunto de tecnologias e processos cujo objetivo é apoiar a criação, a transferência e a aplicação do conhecimento nas organizações, alavancando os ativos humanos em favor da organização [Alavi and Leidner, 2001].

¹ <https://www.standishgroup.com>

² <http://community.ispma.org>

A gestão do conhecimento pode contribuir na mitigação de alguns obstáculos encontrados na SPM, principalmente diminuir os silos no trabalho que dificultam a colaboração interfuncional e às deficiências na estratégia. Deste modo, este trabalho tem como objetivo realizar um mapeamento sistemático da literatura para verificar o panorama das estratégias de produtos de software e os processos apoiados por mecanismos de gestão do conhecimento. Além desta seção introdutória, têm-se a Seção 2 que descreve a metodologia de pesquisa, a Seção 3 apresenta os resultados e na Seção 4 são apresentadas as considerações finais.

2. Metodologia da Pesquisa

O método de mapeamento sistemático da literatura sintetiza e analisa Estudos Primários (EP) e provê uma visão geral de uma determinada área, questão de pesquisa e/ou fenômeno. O mapeamento sistemático segue um método orientado pelas fases de: planejamento, execução e comunicação [Wohlin *et al.*, 2013].

A fase de planejamento envolve a criação de um protocolo detalhado para descrever o processo e os métodos que serão aplicados, diminuindo a possibilidade de viés do pesquisador [Kitchenham, 2004]. É nesta etapa que acontece a identificação da questão de pesquisa e limitação de seu escopo [Costa, 2010]. O protocolo é o instrumento que consolida todas as definições desta fase e deve abordar pontos como: objetivo da revisão, questões de pesquisa, fontes dos estudos primários, identificação da *string* de busca critério de inclusão e exclusão na seleção dos estudos, entre outros. A fase de execução do mapeamento sistemático envolve as etapas de seleção dos estudos primários, além de extração e avaliação dos dados [Wohlin *et al.*, 2013]. Já a fase de comunicação consiste em compartilhar os resultados obtidos e respectiva análise com outros interessados. Nas subseções a seguir são descritos os procedimentos metodológicos utilizados para conduzir este estudo.

2.1. Questão de Pesquisa

A Questão de Pesquisa (QP) direciona o estudo, sendo responsável por delimitar a abrangência da investigação. A QP consiste em um problema que requer discussão, análise e solução ao final da pesquisa [Prodanov, 2013]. A QP que norteou esse estudo foi a seguinte: QP1: Quais aspectos da gestão do conhecimento são utilizados para apoiar estratégias de desenvolvimento de produtos de software?

2.2. Processo de Busca

Para encontrar trabalhos relevantes relacionados à questão de pesquisa, foram definidos os repositórios, a *string* de busca, o idioma (português ou inglês) e o intervalo de tempo (2013 a 2018) considerando as publicações mais recentes e o panorama mais atual em relação aos temas de pesquisa.

A *string* de busca é uma sequência de caracteres com os termos principais derivados das questões de pesquisa [Kitchenham *et al.*, 2009]. A construção da *string* se deu por meio da união das palavras-chaves mais abrangentes, uma vez que a ligação entre esses temas mostrou ser algo novo. Foram utilizados os operadores lógicos AND e OR com necessidade de adaptação sintática de acordo com a base de dados. A *string* de busca resultante foi:

"software product" AND "knowledge management" AND ("software process" OR "business process")

Os termos escolhidos e agrupados mostraram-se eficazes para retorno de trabalhos coerentes com a linha de pesquisa. Foi utilizado “*software product*” para encontrar trabalhos que se referiam a produtos de *software* junto ao termo *knowledge management* para que houvesse relação com a gestão do conhecimento. Já os termos *software process* e *business process* relacionam-se com os processos realizados na construção de um produto de software, já que o trabalho propunha encontrar processos que apoiam-se em gestão do conhecimento.

O processo de busca foi realizado em repositórios de trabalhos relevantes para a grande área de engenharia de software: IEEE Xplore Digital Library; Science Direct; Springer Link; e Scopus. A partir da aplicação da *string* de busca nos repositórios, foram retornados 95 Estudos Primários (EP), catalogados e organizados por meio da ferramenta de software Mendeley. O Mendeley é uma plataforma de redes sociais científicas que também funciona como gestor de conteúdo, possui uma versão online e software para computadores que conta com funcionalidades como: anotações, comentários e integração a motores de busca [Sombra, 2018].

2.3. Extração dos Dados

Nesta etapa foram estabelecidos Critérios de Inclusão (CI) e Critérios Exclusão (CE) para os EP, conforme apresentado na Tabela 1, visando restringir e selecionar apenas de trabalhos mais alinhados ao tema de pesquisa. Os critérios de inclusão e exclusão são importantes para nortear a seleção dos estudos, determinar o rigor da pesquisa e minimizar vieses [Patino, 2004].

Tabela 1. Critérios de Inclusão e Exclusão

Critérios de Inclusão	Critérios de Exclusão
CI1: Estudos que abordam em seus objetivos, hipóteses ou em seus resultados correlação com o tema da pesquisa; CI2: Estudos dentro do intervalo de tempo escolhido para a análise (ano de publicação superior a 2013); CI3: Estudos oriundos de conferências ou <i>journals</i> ; CI4: Estudos primários que estejam nos idiomas inglês e português;	CE1: Estudos primários que não envolvam o tema da questão de pesquisa; CE2: Estudos que não são artigos completos; CE3: Estudos duplicados; CE4: Estudos que expressam pontos de vista pessoais ou opiniões;

A seleção dos estudos consistiu na leitura e análise do título e resumo dos EP retornados no processo de busca, após foram aplicados os CI e CE definidos. Dos 95 EP foram selecionados 41, organizados em uma pasta específica do Mendeley. Nas tabelas 2 e 3 é possível visualizar a quantidade de EP selecionados após a aplicação dos CI e CE por ano de publicação e por repositório de busca, respectivamente.

Tabela 2. Quantidade de EP por ano de publicação

Ano de Publicação	2013	2014	2015	2016	2017	2018	Total
Quantidade de artigos	6	13	3	9	5	5	41

Tabela 3. Quantidade de EP por repositório

Repositório de busca	IEEE	Scopus	Scienc Direct	Springer Link	Total
Total de Artigos antes dos CI e CE	6	5	50	34	95
Total de Artigos depois dos CI e CE	1	2	22	15	41

2.4. Avaliação dos Estudos

Definir e avaliar a qualidade dos estudos primários está além dos critérios de inclusão e exclusão [Kitchenham, 2009]. Os critérios de qualidade ajudam na melhor interpretação dos resultados da pesquisa. Desta forma, nesta etapa foram aplicados Critérios Qualitativos (CQ) para garantir que os EP selecionados possuem uma contribuição relevante relacionada à pergunta de pesquisa que norteia o trabalho. Os seguintes CQ foram considerados: CQ1: O problema de pesquisa está claramente definido? CQ2: A estratégia de execução do estudo foi adequada aos objetivos da pesquisa? CQ3: O estudo responde ao problema de pesquisa? CQ4: O estudo apresenta relação com estratégias de produtos de software? CQ5: O estudo apresenta aplicação de mecanismos de gestão do conhecimento no processo de software?

Foi realizada a análise dos 41 EP (cuja referência pode ser encontrada no link: <http://urlshortener.me/ml6Ntn9>) através da leitura completa e aplicação dos CQ. Após a leitura para cada CQ foi atribuído o valor: AT (Atende) ou NT (Não Atende). Ao final da análise, foi definido o Índice de Aceitação (IA) do EP de acordo com o somatório de CQ com valor AT. Os EP com IA igual ou maior que três foram considerados aceitos (AC), os demais foram rejeitados (RT). A Tabela 4 apresenta a análise dos estudos com a identificação, valores atribuídos aos CQ, Índice de Aceitação (IA) e resultado. Ao final da análise foram considerados 26 EP aceitos e 15 não aceitos.

Tabela 4. Análise de EP através de CQ

ID	CQ1	CQ2	CQ3	CQ4	CQ5	IA	Res.
01	AT	AT	AT	AT	AT	5	AC
02	AT	AT	AT	AT	AT	5	AC
03	AT	AT	NA	NA	NA	2	RT
04	AT	AT	NA	AT	NA	3	AC
05	AT	AT	AT	AT	AT	5	AC
06	AT	AT	NA	NA	NA	2	RT
07	AT	AT	NA	NA	NA	2	RT
08	AT	AT	AT	AT	AT	5	AC
09	AT	AT	AT	AT	NA	4	AC
10	AT	AT	AT	AT	AT	5	AC
11	AT	AT	AT	AT	AT	5	AC
12	AT	AT	AT	AT	NA	4	AC
13	AT	AT	AT	AT	AT	5	AC
14	AT	AT	NA	NA	NA	2	RT
15	AT	AT	NA	NA	NA	2	RT
16	AT	AT	NA	NA	NA	2	RT
17	AT	AT	AT	AT	NA	4	AC
18	AT	AT	AT	AT	AT	5	AC
19	AT	AT	NA	NA	NA	2	RT
20	AT	AT	AT	AT	NA	4	AC
21	AT	AT	NA	AT	NA	3	AC

22	AT	AT	NA	NA	NA	2	RT
23	AT	AT	NA	NA	NA	2	RT
24	AT	AT	AT	AT	AT	5	AC
25	AT	AT	AT	AT	AT	5	AC
26	AT	AT	AT	AT	AT	5	AC
27	AT	AT	NA	AT	NA	3	AC
28	AT	AT	AT	AT	AT	5	AC
29	AT	AT	NA	NA	NA	2	RT
30	AT	AT	NA	NA	NA	2	RT
31	AT	AT	NA	NA	NA	2	RT
32	AT	AT	NA	NA	NA	2	RT
33	AT	AT	NA	AT	NA	3	AC
34	AT	AT	AT	NA	AT	4	AC
35	AT	AT	AT	AT	AT	5	AC
36	AT	AT	NA	NA	NA	2	RT
37	AT	AT	AT	AT	AT	5	AC
38	AT	AT	NA	AT	NA	3	AC
39	AT	AT	AT	AT	NA	4	AC
40	AT	AT	AT	AT	AT	5	AC
41	AT	AT	NA	NA	NA	2	RT

3. Resultados

A Figura 1 apresenta um Mapa Mental para ilustrar as principais temáticas abordadas nos EP considerados aceitos. Dessa forma, é possível observar os ativos de Gestão de Produto de Software e de Gestão do Conhecimento mais evidenciados nos EP. Na Tabela 5 está a síntese das temáticas abordadas pelos EP aceitos (Tabela 4).



Figura 1. Mapa Mental

Tabela 5. Síntese das temáticas abordadas

ID dos EP	Temáticas abordadas
02, 04, 09, 37, 10, 24, 25, 28, 34, 38	Gestão do conhecimento no planejamento e execução da gestão do produto de software
01, 24, 41	Engenharia de requisitos
33, 40	Modelos de maturidade de software

08, 18	Ecosistemas de software
11, 13, 17, 20, 21, 25, 32, 34, 35	Métodos para melhoria no processo de desenvolvimento software
02, 20, 26, 35	Importância dos fatores humanos e conhecimento tácito para vantagem competitiva da empresa
28	Usabilidade do software e como a gestão do conhecimento pode ajudar o especialista dessa área
26, 39	Conhecimento como ativo de valor para tomada de decisões
05	Processo de planejamento colaborativo
10	Teste de software
02, 12	Compartilhamento do conhecimento dentro das empresas de software

Os EP aceitos trazem diversas propostas relacionadas ao tema da pesquisa, a Tabela 6 sintetiza as principais contribuições e propostas que estes estudos trouxeram.

Tabela 6. Contribuições dos EP aceitos

ID	Síntese da Contribuição
01	Método padronizado para priorizar requisitos, utilizando conhecimento tácito. Unifica conceitos da Engenharia de Requisitos e Gestão do Conhecimento.
02	Criação de um modelo para avaliar o nível e características da aprendizagem organizacional de uma empresa de desenvolvimento de software. Utilização do conhecimento como ativo para vantagem competitiva dentro da empresa.
04	Coleta de abordagens baseadas em conhecimento para arquitetura de software. Faz uma classificação e análise temática sobre essas abordagens para identificar as lacunas na aplicação do conhecimento às atividades de arquitetura de software.
05	Propõe um modelo de maturidade de software e uma metodologia para avaliação dos elementos estruturais no processo de planejamento colaborativo.
08	Através de entrevistas qualitativas, traz como resultado um conjunto de características comuns de desenvolvimento e evolução de produtos de software, apesar das diferenças de tamanho, tipo de software e modelos de negócios.
09	Revisão sobre como abordagens baseadas no conhecimento são empregadas na documentação de software, suas influências na qualidade da documentação do software e os custos e benefícios do uso dessas abordagens.
10	Revisão sobre as pesquisas existentes em iniciativas de Gestão do Conhecimento (GC) em testes de software. Como problema mais citado em relação à aplicação da GC em testes de software é citado a inadequação dos sistemas de GC existentes.
11	Modelo para organizações de software emergentes realizarem avaliações de habilidades de força de trabalho baseadas em funções alinhadas a processos de software, acoplando modelos SW-CMM e SWEBOK.
12	Análise qualitativa feita em empresas de software para identificar quais ferramentas de compartilhamento do conhecimento são mais utilizadas e com que frequência os funcionários as utilizam. Aponta que embora as empresas conheçam ferramentas de compartilhamento de conhecimento, elas não são utilizadas por se considerar que elas não organizam informações de maneira eficiente.
13	Proposta de um framework para integração entre ambientes multi-modelo e tecnologias da informação, construído em torno da integração de ontologias para gestão de conhecimento.

17	Propõe e avalia a combinação de VSM (análise do fluxo de valor) e FLOW (modelagem de fluxo de informações) para identificar e tratar os desafios relacionados a informação e a comunicação na produção de software em larga escala.
18	Detalha um conceito de ecossistema de software (SECO) e propõe um método baseado nas capacidades endógenas regionais e nas necessidades do país estudado para definir uma estratégia SECO.
20	Estrutura para integrar fatores humanos à melhoria do processo de software. Para tanto, propõe um framework denominado SAMAY com dois grupos: complemento e suporte. O primeiro grupo está diretamente ligado ao processo de desenvolvimento do software, enquanto o segundo tem ações para facilitar o uso do framework.
21	Modelo de avaliação de processos em empresas de desenvolvimento de software, visando melhorar o resultado do software.
24	Apresenta uma visão e um conjunto de requisitos para ferramentas com intuito de superar problemas no ciclo da gestão do conhecimento para melhor apoiar todo o processo de evolução do software.
25	Trata a problemática da gestão das informações para melhorar processos de software. Contém uma análise dos três repositórios de software mais utilizados, e se eles cobrem os principais requisitos para implementar um modelo de análise de dados. Os resultados mostram que não e aponta aspectos a serem melhorados.
26	Propõe o uso de um framework chamado USTA. O USTA consiste em gerenciamento de conhecimento de aspectos, ativos e esquemas de correspondência, e descoberta de ativos. Fornece maior precisão de comparação em relação a abordagens que adotam um processo de descoberta estática para todas as consultas.
27	Análise dos pontos de vista gerencial e político sobre o impacto da capacidade tecnológica, recursos humanos, internacionalização, recursos de mercado e satisfação do cliente sobre as taxas de crescimento de vendas de empresas de software coreanas.
28	Confronta pesquisas existentes sobre os papéis dos especialistas em usabilidade com a literatura sobre gestão do conhecimento.
32	Analisa as lacunas em relação ao uso de processos de melhoria de software, identificando os aspectos que devem ser considerados como áreas potenciais ou fontes de riscos.
33	Proposta de modelo de maturidade para gerenciamento do ciclo de vida do produto.
34	Avalia o efeito da conversão entre conhecimentos tácito e explícito e sua influência na engenharia de software e na melhoria de processos de software no contexto de uma empresa de pequeno porte.
35	Analisa que o sucesso do negócio está ligado a etapa em que o software é projetado e que o conhecimento tácito é ativo para auxiliar a estratégia de negócio.
37	Estudo sobre práticas de profissionais indianos, como estão usando metodologias ágeis e se realmente existem práticas de gestão do conhecimento para auxiliar o processo.
38	Pesquisa realizada através de entrevistas de pequenas empresas de software, com objetivo de verificar o uso de práticas pelos desenvolvedores de software.
39	Detalha o framework VALUE utilizado para melhorar a tomada de decisões relacionadas ao desenvolvimento de produtos e serviços de software.
40	Examina a partir da teoria gerencial moderna, as suposições que regem a gestão e a melhoria organizacional por trás da abordagem <i>Capability Maturity Model</i> (CMM).

Ao final desta etapa buscou-se responder a questão de pesquisa que norteou o desenvolvimento deste mapeamento sistemático da literatura: Quais aspectos da Gestão do Conhecimento são utilizados para apoiar o desenvolvimento da Estratégia para

Produtos de Software?. A Tabela 7 apresenta os aspectos da Gestão do Conhecimento encontrados e seu uso no desenvolvimento de estratégias de produtos de software, bem como a fonte representada pelo(s) identificador(es) do(s) EP.

Tabela 7. Contribuições dos EP aceitos

Aspecto da KM	Como apoia a Estratégia do Produto de Software	ID do EP
Conversão de conhecimento tácito em explícito	Conhecimento explícito (antes tácito), tais como procedimentos (<i>know how</i>) especializados podem representar diferenciais competitivos incorporados ao produto de software.	01, 09, 17, 24, 28, 34, 35
Aprendizagem organizacional	Uso de boas práticas e lições aprendidas como entrada para especificação de processos de negócio a serem considerados no produto de software.	02, 08, 18, 21, 33, 38, 40
Compartilhamento de informações e conhecimento	Integração das diferentes perspectivas internas de negócio por meio do compartilhamento de informações e conhecimento entre diferentes stakeholders.	04, 10, 12, 13, 17, 24, 25, 26, 32, 37, 39
Fatores Humanos	Avaliação do comportamento dos stakeholders e relações humanas informais para definição de processos de negócio a serem considerados no produto de software.	11, 20, 26, 27, 32

4. Considerações Finais

Este trabalho apresenta as atividades e resultados de um mapeamento sistemático da literatura relacionado à integração entre as disciplinas de Gestão do Produto de Software e Gestão do Conhecimento (GC). Mais especificamente, o objetivo foi identificar quais mecanismos de GC são utilizados em estratégias de produtos de software.

O mapeamento sistemático possibilitou identificar que apesar da dificuldade de gerenciar e compartilhar o conhecimento, este se faz cada vez mais reconhecido como ativo para melhoria de estratégias na criação de um produto de software. Nota-se um avanço nos estudos que estão gradativamente associando a gestão do conhecimento e a gestão do produto de software. O aspecto humano mostra-se essencial, pois consiste na principal fonte ativa de conhecimento de uma organização.

Entre as oportunidades de pesquisa identificadas estão: especificação e avaliação de uma abordagem para melhorar a estratégia de um produto de software por meio do uso efetivo de mecanismos de gestão do conhecimento, incluindo a socialização a qual se preocupa com o compartilhamento e troca de conhecimento entre as pessoas, além da especificação de uma abordagem para guiar a estratégia do produto de software a partir da perspectiva do comportamento (engajamento, fidelização, etc.) dos usuários.

Referências

- Alavi, M., and Leidner, D. E. (2001). Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly*, pp. 107-136.
- Borba, A. W. T., Batista, G. H. C., & Souza, R. A. C. (2016). InnoStartup: a Toolbox for Innovation in Software Development Process. *IEEE Latin America Transactions*, 14(8), pp. 3875-3885.

- Ebert, C. (2014). Software product management. *IEEE Software*, 31(3), pp. 21-24.
- Ebert, C., and Brinkkemper, S. (2014). Software product management: An industry evaluation. *Journal of Systems and Software* (95), pp. 10-18.
- Fayad, M. E., Laitinen, M., & Ward, R. P. (2000). Software engineering in the small. *Communications of the ACM*, 43(3), pp. 115-115
- Ferreira, P. J. A. V. (2010). *Rastreabilidade de ponto de função*. Dissertação de Mestrado. PPGI/UNIRIO. Rio de Janeiro.
- Fricker, S. A. (2012). Software product management. In *Software for People*. Springer, Berlin, Heidelberg, pp. 53-81.
- Ilavarasan, P. V. and Sharma, A. K. (2003). Is software work routinized?: Some empirical observations from Indian software industry. *Journal of systems and software*, 66(1), pp. 1-6.
- Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering—a systematic literature review. *Information and software technology*, 51(1), pp. 7-15.
- Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004), pp. 1-26.
- Maglyas, A., Nikula, U., Smolander, K., & Fricker, S. A. (2017). Core software product management activities. *Journal of Advances in Management Research*, 14(1), pp. 23-45.
- Marciniak, J. J. (2001). Process Models in Software Engineering. *Encyclopedia of Software Engineering, 2nd Edition*, John Wiley and Sons, Inc, New York.
- Martin, D., Rooksby, J., Rouncefield, M., & Sommerville, I. (2007). 'Good'organisational reasons for'Bad'software testing: An ethnographic study of testing in a small software company. In *Proceedings of the 29th international conference on Software Engineering*. IEEE Computer Society. pp. 602-611.
- Patino, C. M. and Ferreira, J. C. (2018). Inclusion and exclusion criteria in research studies: definitions and why they matter. *Jornal Brasileiro de Pneumologia*, 44(2), PP. 84-84.
- Prodanov, C. C., and De Freitas, E. C. (2013). *Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª Edição*. Editora Feevale.
- Sombra, T. R. (2018). Reconhecimento de padrões em rede social científica: aplicação do algoritmo Naive Bayes para classificação de papers no Mendeley.
- Teixeira, M. M. and Ferreira, T. A. E. (2014). *The communication model of virtual universe: multipolarity, ICT, cyberculture, education and media manipulation*. GRIN Verlag.
- Toffolon, C. (2000). The software dimensions theory. In *Enterprise information systems*. Springer, Dordrecht. pp. 89-98.
- Wohlin, C. and Prikladnicki, R. (2013). Systematic literature reviews in software engineering. *Information and Software Technology*, 55(6), pp. 919-920.