

Desafios da Gestão de Requisitos e seus pesquisadores principais: construindo pontes para o sucesso do produto na Engenharia de Software

O presente texto objetiva apresentar a engenharia de requisitos e seus conceitos essenciais, alguns de seus desafios para a engenharia de software e em seguida os autores mais relevantes para essa área do conhecimento científico. Para identificar os principais pesquisadores do campo foi feita uma análise bibliométrica, a qual também permitiu a identificação dos principais periódicos da área e as palavras-chave mais importantes. Ainda, apresentamos a produção científica relacionada a engenharia de requisitos em números e por país, permitindo uma visão global sobre a relevância do tema.

A Gestão de Requisitos e seus desafios principais

A engenharia de requisitos é definida como um subconjunto da engenharia de sistemas que possui o objetivo de descobrir, desenvolver, rastrear, analisar, qualificar, comunicar e gerenciar requisitos que definem o sistema em níveis sucessivos de abstração (ALSANOOSY, SPICHKOVA, HARLAND, 2020). Essa definição deriva da própria ideia de engenharia de requisitos que é definida, segundo Laplante (2017, p.2), como o ramo da engenharia que se preocupa com os objetivos, as funções e as restrições para desenvolver sistemas para o mundo real. Também se preocupa com as especificações do comportamento do sistema e sua evolução ao longo do tempo e como os sistemas relacionados se comportam “em família”, ou seja, como se relacionam e funcionam em conjunto com outros sistemas.

O requisito, portanto, é objeto central da engenharia de requisitos de software e parte do desafio neste campo do conhecimento está em entender o que realmente significa um "requisito". Os requisitos podem variar desde declarações abstratas e esboços simples até especificações formais e matematicamente rigorosas. Essa diversidade nas formas de representação ocorre porque as partes interessadas (os stakeholders) possuem necessidades em diferentes níveis e, consequentemente, dependem de diferentes níveis de abstração. Além disso, as partes interessadas têm capacidades variadas para criar e interpretar essas representações (por exemplo, um cliente empresarial em comparação com um engenheiro de projeto), resultando em uma qualidade diversificada dos requisitos (LAPLANTE, 2017, p. 3).

Reinehr (2020) define um requisito como uma condição, uma habilidade necessária para que um usuário resolva um problema e alcance um objetivo. Pode ser entendido também como uma condição ou característica que um sistema ou componente deve atender ou possuir para cumprir um contrato, norma, especificação ou outro documento formalmente imposto. Além disso, um requisito pode ser uma representação documentada de requisitos estabelecidos a partir destes critérios.

A mesma autora ainda tipifica os requisitos. Um *requisito funcional* é uma funcionalidade exigida pelos stakeholders para atingir um objetivo de negócio, indicando o que os desenvolvedores devem implementar, geralmente expressa em frases como "o

sistema deve". Por exemplo, "o sistema deve permitir que o usuário pague com cartão de débito ou crédito". Já os requisitos de qualidade, ou *não funcionais*, referem-se a como o software operará sob certas circunstâncias, abrangendo aspectos como desempenho, disponibilidade, usabilidade, portabilidade e escalabilidade. Estes são frequentemente negligenciados no início do projeto, apesar de seu impacto significativo na arquitetura da aplicação e da dificuldade de correção posterior. Atender a requisitos específicos de desempenho, por exemplo, requer escolhas arquiteturais antecipadas, sendo complicado ajustar depois que a implementação já está avançada (REINEHR, 2020).

Existem ainda requisitos que são classificados como requisitos de produto e requisitos de projeto e de processo. Os requisitos de produto de software são aqueles desenvolvidos pela equipe e integrados ao produto final. Os requisitos de projeto e de processo são sempre não funcionais. Existe também o conceito de requisito de sistema e é mais abrangente, englobando hardware, software e processos. Assim, requisitos de sistema são de alto nível e devem ser atendidos pelo produto como um todo, enquanto os requisitos de software são específicos para o componente de software do sistema. Um exemplo de requisito de sistema seria: "O sistema deve identificar uma pessoa a partir de uma imagem gerada por webcam", envolvendo tanto sensores físicos (câmera digital) quanto software para identificar pessoas (REINEHR, 2020).

A gestão de requisitos envolve identificar, documentar e acompanhar os requisitos do sistema desde a concepção até a entrega. Isso inclui compreender o verdadeiro significado dos requisitos e gerenciar as expectativas dos clientes e partes interessadas ao longo do ciclo de vida do sistema. Um processo sólido de gestão de requisitos é fundamental para o sucesso de um projeto (LAPLANTE, 2017, p. 221).

Fatores culturais têm um impacto profundo na maneira como as pessoas se comunicam, e, conseqüentemente, as atividades de engenharia de requisitos podem ser fortemente influenciadas pelas culturas dos indivíduos. Uma vez que a gestão de requisitos é uma atividade centrada nas pessoas, possuindo um caráter social e técnico, requer comunicação intensa com as várias partes interessadas no software (clientes, usuários finais, especialistas no domínio, proprietários de projetos, etc.) para determinar os requisitos de um projeto (ALSANOOSY, SPICHKOVA, HARLAND, 2020). Desse modo, a gestão de requisitos deve levar em consideração os aspectos culturais que influenciam todo o desenvolvimento do produto.

As etapas necessárias para a gestão de requisitos dentro da engenharia de requisitos e a sua relação com o desenvolvimento de software estão apresentadas na Figura 1. A Elicitação é a primeira etapa e compreende a busca, a identificação e a descoberta dos requisitos necessários para o desenvolvimento do produto. Em seguida temos a etapa de Análise, na qual os possíveis conflitos são levantados e são identificadas as relações destes com os requisitos e o contexto no qual o software será aplicado. A Especificação compreende a documentação e o detalhamento das especificações dos requisitos. Por fim temos a etapa de Validação, na qual os requisitos são validados aos propósitos do produto de software.

Figura 1 - Etapas da Engenharia de requisitos.



FONTE: Extraído de Reinehr (2020).

Laplante (2017) aponta cinco principais desafios na gestão de requisitos. O primeiro é que poucas organizações possuem um processo bem definido de gestão de requisitos, resultando em pouca experiência nesse campo. O segundo desafio é a dificuldade de distinguir entre os requisitos dos usuários ou stakeholders e os requisitos de sistema. O terceiro problema é que as organizações gerenciam os requisitos de maneira diferente, dificultando a disseminação e troca de experiências e melhores práticas. As dificuldades no monitoramento do progresso do desenvolvimento dos produtos representam outro problema. Estabelecer um processo bem definido de gestão de requisitos é essencial para enfrentar esses desafios (LAPLANTE, 2017, p. 221).

Vários trabalhos apontam as consequências de não fazer a gestão de requisitos de modo adequado. Em alguns cenários é possível que ocorra a falha total do modelo de negócio.

Para o cenário de uma startup, por exemplo, a gestão de requisitos requer ainda a interação contínua com os seus clientes para entender suas necessidades, permitindo à equipe da startup reunir informações sobre os problemas reais dos clientes e desenvolver modelos conceituais sobre a sua solução, permitindo o desenvolvimento do produto. Isso ajuda a determinar as características prioritárias que atraem os primeiros consumidores a aderir e testar o produto oferecido. Como as startups são formadas para buscar um modelo de negócio repetível e escalável e entregar produtos inovadores a partir de recursos limitados (matéria prima, mão de obra capacitada) elas baseiam as suas suposições de mercado em fatos reais obtidos através de interações contínuas com os clientes. Uma engenharia de requisitos deficiente, portanto, pode levar ao fracasso da startup devido à falta de entendimento sobre a demanda dos clientes, já que os requisitos não são passíveis de serem definidos a ponto de revelar as necessidades dos consumidores (GUPTA et al., 2020).

No cenário de grandes empresas, os projetos de desenvolvimento de software que ocorrem em modelos de terceirização de serviços também são afetados caso a gestão de requisitos não ocorra a contento. Muitas empresas europeias terceirizam o desenvolvimento de software para países como Índia, Vietnã e China, por exemplo. A terceirização traz algumas vantagens para a contratante como a economia de gastos, o acesso a melhores e mais atuais tecnologias e a superação de problemas internos da organização, como problemas organizacionais, que impedem a implantação de uma linha produtiva de

software. Para as empresas que oferecem o serviço terceirizado o lucro vem através do aumento de sua expertise em aprender em como atender aos requisitos dos clientes (IQBAL et al., 2020). Como a cultura nesse cenário pode aparecer como uma barreira na gestão dos requisitos (ALSANOOSY, SPICHKOVA, HARLAND, 2020) o sucesso dos projetos podem atingir apenas 50% do desejado, o que é reflexo das falhas de software (IQBAL et al., 2020; SHAFIQ et al., 2020). Desse modo, Iqbal et al. (2020) ressaltam que requisitos não gerenciados adequadamente podem fazer com que o modelo terceirizado de desenvolvimento de software perca totalmente a capacidade de fornecer as vantagens competitivas que lhe são atribuídas.

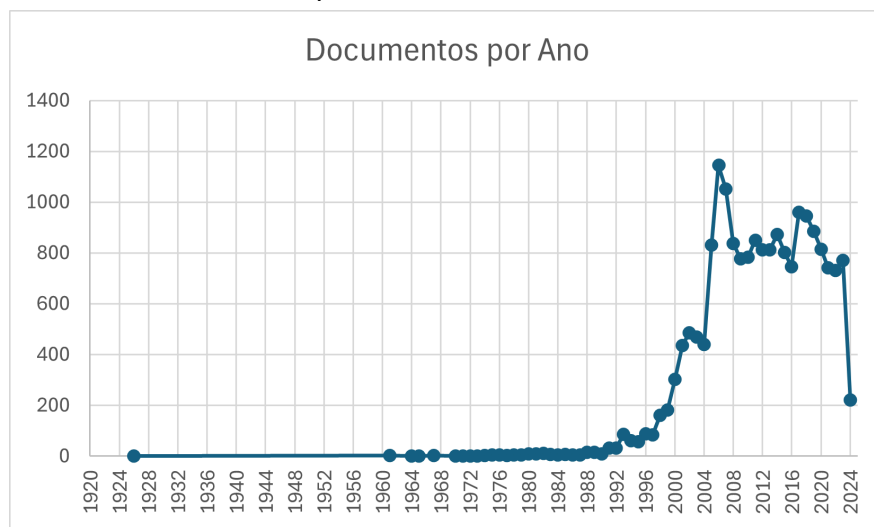
A relevância da engenharia de requisitos de software é bastante discutida em âmbito educacional. A área de Educação em Engenharia de Requisitos, ou *Requirements engineering education* (REE), se preocupa em formar profissionais preparados para enfrentar desafios industriais ao mesmo tempo em que incorporam os conceitos teóricos subjacentes à engenharia de requisitos (DAUN et al., 2023). A crítica que a literatura científica faz a respeito da formação dos profissionais classifica o ensino do tema como “na melhor das hipóteses, uma reflexão tardia em muitos currículos universitários de engenharia de software, concentrando-se no ensino em estilo de aula [expositivas] com poucos ou nenhum exemplo realista” (DAUN et al., 2023, p. 145, tradução livre). Os autores ressaltam que os alunos geralmente saem dos cursos de engenharia de software com uma visão rudimentar da engenharia de requisitos exigidas por este campo do conhecimento, cabendo às empresas formar adequadamente os profissionais para atuarem em gestão de requisitos.

Análise bibliométrica para identificação dos nomes relevantes

Na base de dados Scopus no dia 31/05/2024 às 21:00 utilizou-se o tipo da busca como TITLE-ABS-KEY, o que significa uma busca no título, resumo e palavras-chave respectivamente. O termo buscado foi: “requirements engineering”. O total de documentos encontrados foi 19.448.

No Gráfico 1 é possível observar a evolução dos trabalhos ao longo dos anos. É notável um crescimento das publicações nos anos entre 1990 a 2010, desde então nota-se uma regularidade nas publicações.

Gráfico 1 - Número de Documentos por Ano

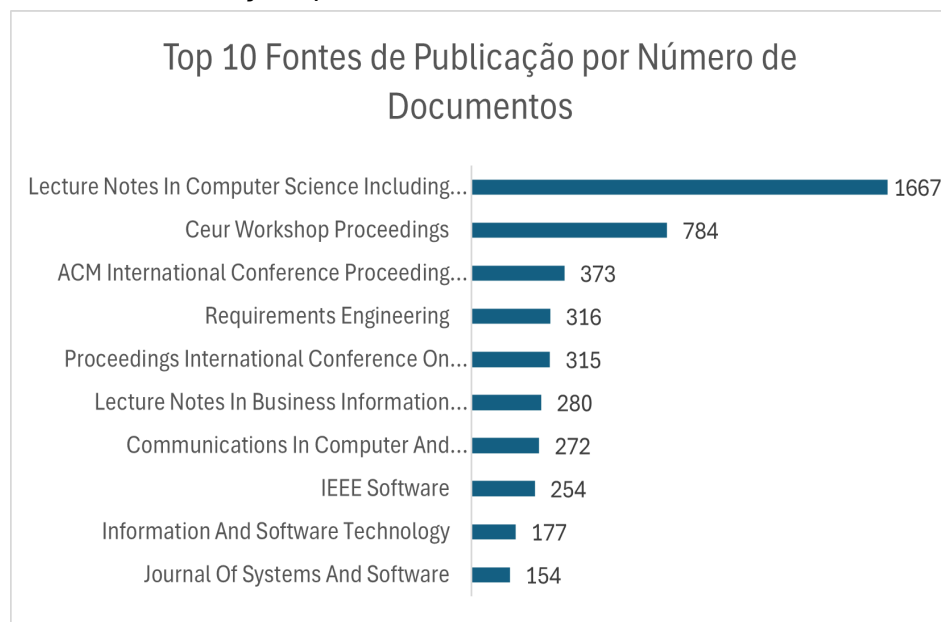


Fonte: Os autores (2024)

Para saber quais as fontes que contribuíram mais em número para as publicações, realizou-se uma análise nos documentos e um total de 107 documentos publicados ao longo dos anos foram retirados das 10 maiores fontes publicadoras. O Gráfico 2 mostra o nome das fontes e o número de documentos publicados. A fonte que mais se destaca é a *Lecture Notes In Computer Science Including Subseries Lecture Notes In Artificial Intelligence And Lecture Notes In Bioinformatics* contando com um total de 1667 documentos publicados. As demais fontes também demonstram números expressivos.

A fim de se ter uma noção dos autores que mais contribuíram para o avanço na temática realizou-se um levantamento dos autores contidos dessas publicações. Um total de 159 autores encontram-se nas publicações, separando-se então os 10 autores que mais contribuíram em número de publicações.

Gráfico 2 - Fonte de Publicações por Número de Documentos

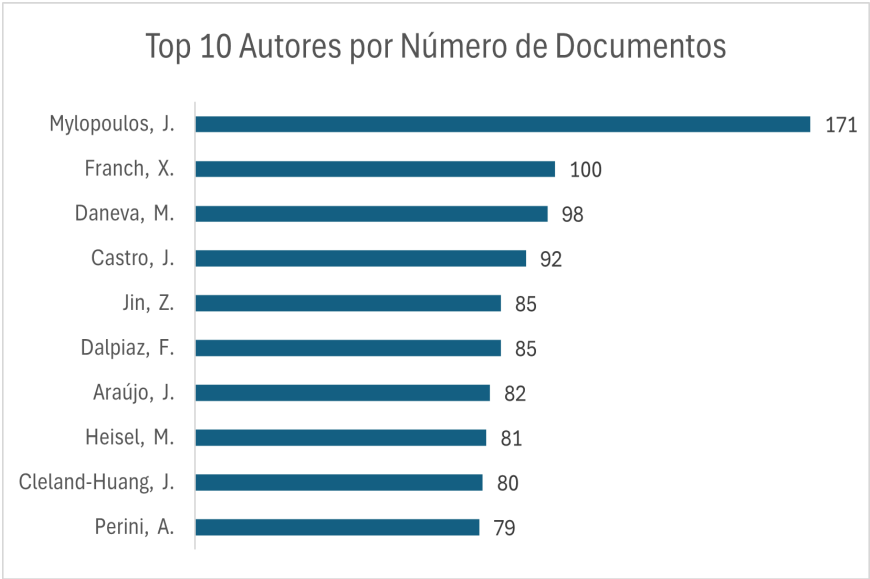


Fonte: Os autores (2024)

O Gráfico 3 apresenta em ordem crescente por número de publicações. O autor com maior número de publicações é o Mylopoulos contando com um total de 171 documentos publicados.

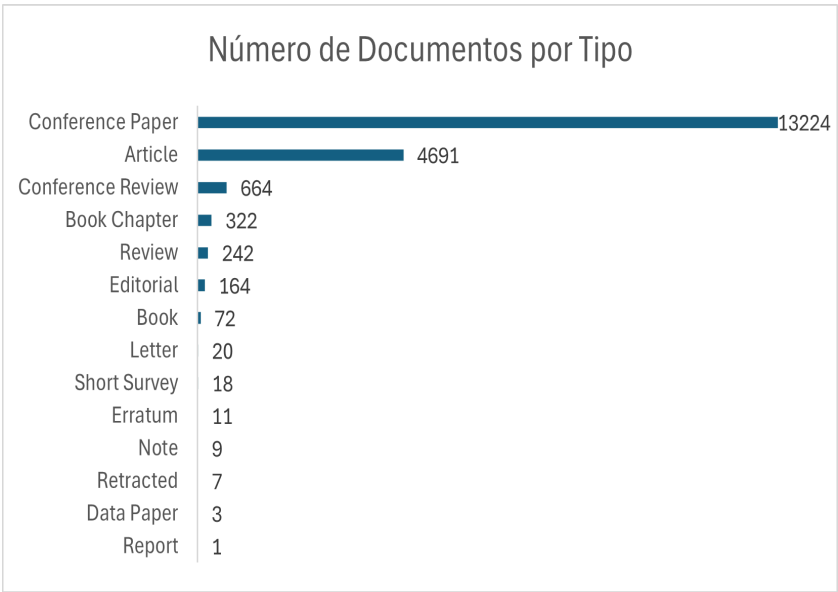
O Gráfico 4 apresenta o tipo dos documentos que apareceram na busca, os tipos que mais aparecem são Conference Paper e Article com 13.224 e 4.691 documentos publicados respectivamente.

Gráfico 3 - Top 10 Autores por Número de Documentos



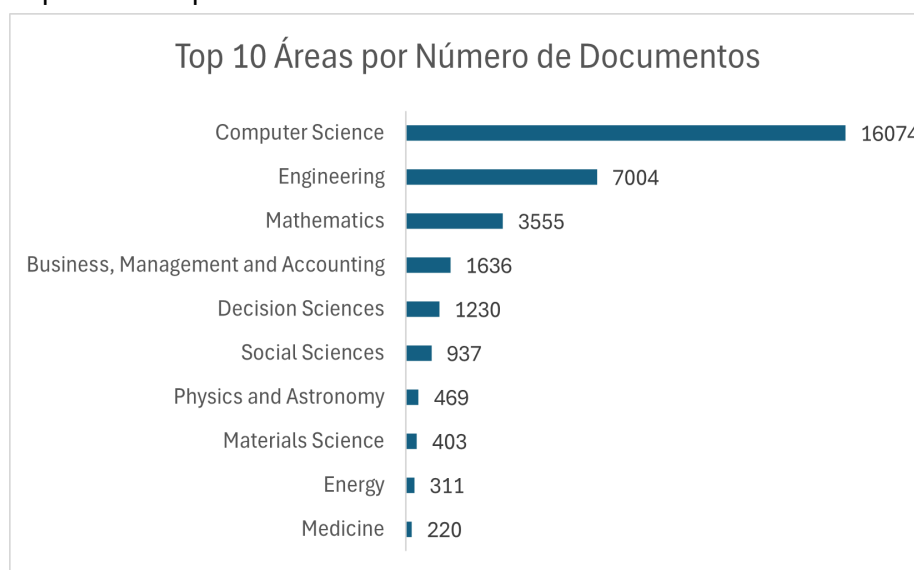
Fonte: Os autores (2024)

Gráfico 4 - Número de Documentos por Tipo



Fonte: Os autores (2024)

Gráfico 5 - Top 10 Áreas por Número de Documentos

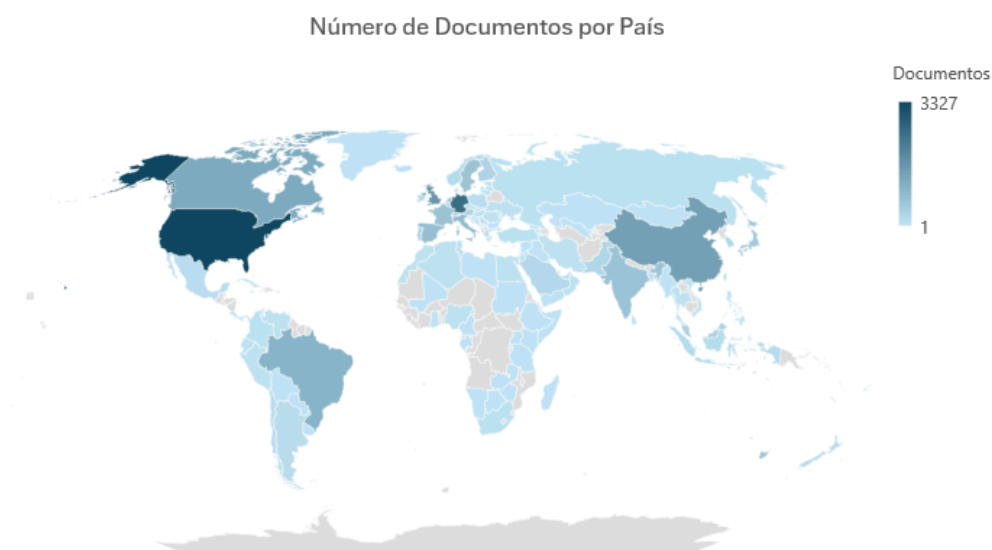


Fonte: Os autores (2024)

O Gráfico 5 apresenta as 10 principais áreas do conhecimento de um total de 26 áreas que foram encontradas na busca. O maior destaque está nas áreas de Computer Science, Engineering and Mathematics, que possuem expressivos números, sendo eles 16.074, 7.004 e 3.555 respectivamente.

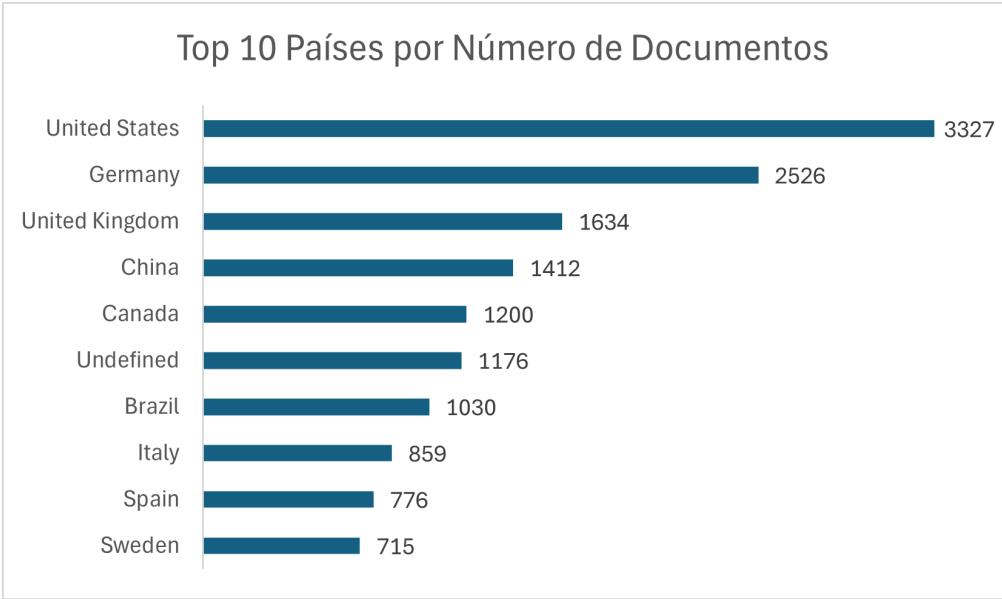
Os Gráficos 6 e 7 apresentam o número de documentos publicados nos países, sendo que o Gráfico 6 apresenta um mapa coroplético e o Gráfico 7 os 10 principais países publicadores de um total de 127 retornados na consulta, um destaque maior para os Estados Unidos com o maior número de publicações contando com 3.327. O Brasil aparece na 7ª posição, contando com 1.030 documentos na busca realizada.

Gráfico 6 - Mapa de Países por Número de Documentos



Fonte: Os autores (2024)

Gráfico 7 - Número de Documentos por Países



Fonte: Os autores (2024)

Top 5 documentos com maior número de citações

O número de documentos com maior número de citações foi analisado na base de dados scopus no entanto para se ter uma noção maior da quantidade de citações de uma maior base de indexação foi retirada o número de citações do Google Scholar na data de 01/06/2024 às 21:17. O quadro 1 apresenta o nome do documento, a autoria, o ano e o número de citações dos documentos com maior número de citações encontradas neste estudo.

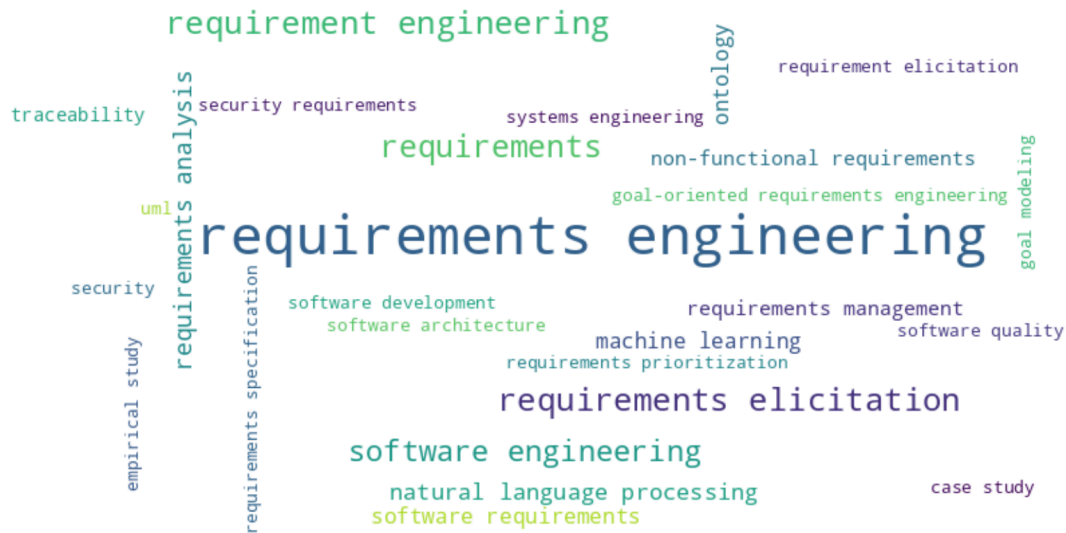
Quadro 1 - Documentos com maior número de citações

Nome	Autoria	Nº Citações
Requirements engineering: a roadmap	NUSEIBEH, Bashar; EASTERBROOK (2000)	3424
Goal-directed requirements acquisition	DARDENNE, Anne; VAN LAMSWEERDE, Axel; FICKAS, Stephen (1993)	2918
Towards modelling and reasoning support for early-phase requirements engineering	YU, Eric SK. (1997)	2213
Graph visualization and navigation in information visualization: A survey	HERMAN, Ivan; MELANÇON, Guy; MARSHALL, M. Scott.(2000)	2448
An analysis of the requirements traceability problem	GOTEL, Orlena CZ; FINKELSTEIN, C. W. (1994)	2010

Fonte: Os autores (2024).

Analizou-se também da busca realizada no Scopus a maior ocorrência de palavras, escolheu-se a critério dos autores as 28 primeiras palavras do ranque para criar uma nuvem de palavras. A nuvem de palavras é mostrada na Figura 2.

Figura 2 - Nuvem de palavras-chave



Fonte: Os autores (2024).

John Mylopoulos

Nesta seção será apresentado um breve histórico de John Mylopoulos, o autor com maior número de publicações de documentos nos conforme apresentado na seção anterior.

John Mylopoulos (nascido em 12 de julho de 1943) é um cientista da computação greco-canadense, professor da Universidade de Toronto, no Canadá, e da Universidade de Trento, na Itália. É conhecido por seu trabalho no campo da modelagem conceitual, especificamente pelo desenvolvimento de uma metodologia de desenvolvimento de software orientada a agentes, chamada TROPOS.

O interesse de pesquisa de Mylopoulos abrange desde técnicas de modelagem de informações, especificamente modelos de dados semânticos, até sistemas baseados em conhecimento e design de sistemas de informações e o campo da engenharia de requisitos. Mylopoulos tem grandes contribuições nesses campos:

- TORUS: Acesso em linguagem natural a bancos de dados, que exigia a representação da semântica dos dados e, portanto, nos levou pela primeira vez a modelos conceituais de tabelas relacionais usando redes semânticas.
- TAXIS: linguagem de programação para aplicativos com uso intensivo de dados que suportava classes de objetos, transações, restrições, exceções e fluxos de trabalho, todos organizados ortogonalmente em hierarquias de subclasses com herança de propriedades.
- TELOS: linguagem de representação do conhecimento de vários tipos diferentes de participantes da engenharia de software, incluindo o domínio do aplicativo e o domínio do desenvolvimento, que explorava metaclasses e tratava as propriedades como objetos

- TROPOS: Aplicação das ideias de requisitos iniciais (orientação a metas, dependência de agentes) a toda a gama de desenvolvimento de software e expansão de seu escopo para muitos tópicos, incluindo segurança e evolução.

Axel van Lamsweerde

Dos documentos analisados, Alex van Lamsweerde é o autor com maior número de citações, contendo um total de 2388 citações em documentos, por isso nesta seção será apresentado uma breve descrição sobre o autor.

Axel van Lamsweerde (nascido em 1947) é um cientista da computação belga e professor de ciência da computação na Université Catholique de Louvain, conhecido por seu trabalho em engenharia de requisitos e pelo desenvolvimento da linguagem de modelagem orientada a objetivos KAOS.

Os interesses de pesquisa de Van Lamsweerde estão nas áreas de técnicas precisas para engenharia de requisitos, modelagem de sistemas, sistemas de alta garantia, métodos formais leves, modelagem e análise de processos, segurança médica e ambientes de desenvolvimento de software baseados em conhecimento.

AUTORIA



Hesau Hugo Cavalcanti Leite

Atua como Engenheiro de Software em uma software house. Possui graduação em Engenharia de Produção pela Universidade do Grande Vale (UGV) e mestrado em Métodos Numéricos em Engenharia pela Universidade Federal do Paraná (UFPR). Atualmente mora em Fraiburgo (SC) e dedica-se aos estudos no campo da Pesquisa Operacional e Engenharia de Software.



Leonardo Antonio Pregnolo

É desenvolvedor de software que atua com backend em Java Spring Boot no setor de seguros e assistências. Possui doutorado em Sustentabilidade pela Universidade de São Paulo, com pesquisa na área de ciência e tecnologia ambiental em parceria com o Centro Nacional de Pesquisa em Energia e Materiais (CNPEM). Possui ampla experiência na área de ensino de Geografia e Ciências Naturais para o Ensino Fundamental. Atualmente mora em Caconde (SP) e dedica-se aos estudos no campo da Engenharia de Software.

REFERÊNCIAS BIBLIOGRÁFICAS

ALSANOOSY, Tawfeeq; SPICHKOVA, Maria; HARLAND, James. Cultural influence on requirements engineering activities: a systematic literature review and analysis. **Requirements Engineering**, v. 25, p. 339-362, 2020.

Axel van Lamsweerde. 2024. Disponível em: https://en.wikipedia.org/wiki/Axel_van_Lamsweerde. Acesso em: 1 jun. 2024.

DAUN, Marian et al. A systematic literature review of requirements engineering education. **Requirements Engineering**, v. 28, n. 2, p. 145-175, 2023.

GUPTA, Varun et al. Requirements engineering in software startups: A systematic mapping study. **Applied Sciences**, v. 10, n. 17, p. 6125, 2020.

IQBAL, Javed et al. Requirements engineering issues causing software development outsourcing failure. **PloS one**, v. 15, n. 4, p. e0229785, 2020.

John Mylopoulos. 2024. Disponível em: https://en.wikipedia.org/wiki/John_Mylopoulos. Acesso em: 1 jun. 2024.

LAPLANTE, Phillip A. **Requirements engineering for software and systems**. Taylor & Francis, CRC Press, 2017.

REINEHR, Sheila. **Engenharia de requisitos**. Porto Alegre: Grupo A, 2020. E-book. ISBN 9786556900674. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786556900674/>. Acesso em: 30 mai. 2024.

SHAFIQ, Muhammad et al. Factors influencing the requirements engineering process in offshore software development outsourcing environments. **IET Software**, v. 14, n. 6, p. 623-637, 2020.