

USO DE INTELIGÊNCIA ANALÍTICA NA ENGENHARIA DE SOFTWARE – UM MAPEAMENTO SISTEMÁTICO DA LITERATURA

Bruno Rodrigues, Fernando Parreiras

Universidade FUMEC, Brasil

bruno.rodrigues@kroton.com.br, fernando.parreiras@fumec.br

Resumo: Informações existentes em repositórios de software podem auxiliar engenheiros de software em suas atividades durante todas as fases do desenvolvimento. O uso da inteligência analítica em repositórios de software beneficia engenheiros de software a obterem informações relevantes a respeito do projeto que está sob sua responsabilidade. Com a finalidade de fomentar os estudos do uso da inteligência analítica na Engenharia de Software, este artigo realizou um mapeamento sistemático da literatura, para descobrir a aplicabilidade da inteligência analítica na Engenharia de Software. Foram analisados 235 artigos extraídos de cinco bases científicas da computação e de duas conferências da Engenharia de Software. Com base nesses artigos, foram categorizados os tipos de análise, as áreas e as necessidades dos engenheiros de software que estão utilizando a inteligência analítica. Descobriu-se que métodos são estão sendo propostos com a finalidade de quantificar as mudanças dos artefatos de software na prática de desenvolvimento, seja relacionado ao código-fonte do projeto ou como na estimativa de esforço e risco do projeto. Percebe-se um forte interesse dos engenheiros de software em utilizar a inteligência analítica para entender as alterações durante a manutenção de software. A inteligência analítica está fortemente relacionada à prática de manutenção de software, mas sua utilização vem crescendo na gestão de projetos e no gerenciamento de equipes. Pode-se verificar uma demanda ainda pouco explorada pelas pesquisas em relação à prática de testes, nos processos e ciclo de vida do desenvolvimento de software.

Palavras-chave: Inteligência Analítica; Mineração de Repositórios de Software; Engenharia de Software.

Abstract: The existing information about a software repository can assist software engineers in their activities during all phases of software development. The use of software analytics is benefiting software engineering practitioners to obtain relevant information in the software repository. In order to promote studies about software analytics, this paper conducted a systematic literature mapping about software analytics usage in software engineering. 235 papers were analyzed on five computation's bases and two software engineering conferences. Based on these papers we categorized the types of analysis, the areas and the needs of software engineers who are using software analytics. This paper found that methods are often proposed in order to quantify the changes of software artifacts in practice development, in which is related to the project's source code or as in the estimation of effort and project risk. We realized a strong interest of software engineers to use software analytics to understand the changes during software maintenance. Software analytics is strongly related to software maintenance, but its use is

growing in project management and management teams. We observed a demand little explored around the testing practice, software development process and software development life cycle.

Keywords: Software Analytics; Mining Software Repositories; Software Engineering.

I. INTRODUÇÃO

A inteligência analítica é descrita como uma técnica que faz uso de análises, dados e raciocínio sistemático para tomar decisões [1]. Demonstrou ser uma técnica valiosa para a análise de processos de negócios [2]. Na Engenharia de Software seu uso tem crescido de forma substancial na identificação de padrões e no apoio à tomada de decisão em cada fase do desenvolvimento de software [3] [4].

Estudos utilizando dados de repositórios de desenvolvimento software têm coberto várias áreas da Engenharia de Software, como arquitetura, processo de desenvolvimento e reuso, entre outras [5]. A inteligência analítica está permitindo aos gerentes e engenheiros de software melhor compreensão a respeito dos sistemas sob sua responsabilidade [6].

Neste contexto, emerge o problema de pesquisa: Quais são as aplicações da inteligência analítica na Engenharia de Software? Este artigo visa identificar como a inteligência analítica está sendo utilizada para embasar decisões nas áreas da Engenharia de Software.

Para responder ao problema de pesquisa, procedeu-se a um mapeamento sistemático da literatura sobre inteligência analítica na Engenharia de Software. Para isso, foram pesquisados artigos em cinco bases científicas da computação - *ACM Digital Library*, *IEEE Xplore*, *Springer*, *Elsevier* e *Wiley* - e realizada a busca manual em duas conferências da área: *International Conference on Software Engineering (ICSE)* e *International Conference on Mining Software Repositories (MSR)*. Os artigos resultantes deste mapeamento foram classificados com base em aspectos encontrados na literatura, apresentando uma dimensão da produção científica nessa área.

Este artigo contribui para uma melhor compreensão da utilização da inteligência analítica na engenharia de software, tornando possível identificar as áreas de maior predominância das técnicas, bem como as áreas carentes de pesquisa. É possível verificar, também, para quais propósitos a inteligência analítica está sendo empregada na Engenharia de Software e como as pesquisas nessa área estão

suprindo as necessidades dos engenheiros de software. Como contribuição secundária dessa pesquisa, foi possível identificar termos utilizados para tratar de inteligência analítica na Engenharia de Software, na qual se espera que sejam úteis em futuras pesquisas.

O artigo se encontra estruturado em sete seções, incluindo esta introdução. A Seção II contém a fundamentação teórica deste artigo. A Seção III apresenta os trabalhos relacionados. A Seção IV descreve a metodologia utilizada para este mapeamento sistemático e os descritores de busca. A Seção V apresenta os resultados do mapeamento sistemático e responde às perguntas deste trabalho. A Seção VI as discussões dos resultados. A Seção VII trata das ameaças à validade desta pesquisa. A Seção VIII as conclusões e sugestões para trabalhos futuros.

II. FUNDAMENTAÇÃO TEÓRICA

Repositórios de software contêm dados do sistema que, muitas vezes, podem ser difíceis de serem interpretados, porém podem ser utilizados pelos engenheiros de software para embasar decisões. A inteligência analítica na Engenharia de Software analisa e cruza dados disponíveis em repositórios de software, para descobrir informações sobre o sistema [7]. Como base para este mapeamento sistemático, foram considerados os conceitos de inteligência analítica apresentados nesta Seção. Esta Seção contempla os conceitos de inteligência analítica, os tipos de análise que podem ser feitos com base na inteligência analítica, as necessidades de utilização da inteligência analítica por engenheiros de software e as definições das áreas do conhecimento da Engenharia de Software.

A. *Inteligência Analítica*

A inteligência analítica permite aos engenheiros de software realizarem a exploração e análise de dados, a fim de obter informações detalhadas e baseadas em dados do próprio software, auxiliando-os a tomar decisões sobre a condução de seus projetos e seus serviços [8]. Pesquisas têm relatado o uso da inteligência analítica para entender sistemas de software, propagação de mudanças, previsão e identificação de defeitos e entender a dinâmica de equipes de desenvolvimento, melhorar a experiência do usuário, a reusabilidade de código, a automação de técnicas de mineração de repositórios [7] e a qualidade do software [9] e capacitar os indivíduos e equipes de desenvolvimento [6], entre outras atividades.

A inteligência analítica na Engenharia de Software alcançou sucesso substancial tanto em pesquisa quanto na prática de desenvolvimento [4]. Um dos papéis da inteligência analítica na Engenharia de Software é oferecer recomendações práticas para melhorar o projeto [6]. Ou seja, uma ferramenta de inteligência analítica pode recomendar aos programadores que realizem uma ação baseada no histórico de versões, por exemplo, “programadores que alteraram esta função também alteraram...” [10]. Dessa maneira, tem-se a intenção de reduzir erros por alterações

incompletas no código e nos problemas de acoplamento.

Projetos de software tendem a continuar crescendo em tamanho e complexidade. Assim, o esforço para analisar os dados disponíveis em repositórios de software é crescente [11]. A dificuldade de interpretar dados tem sido solucionada mediante a extração das métricas de software e a utilização de técnicas estatísticas [12].

Pesquisadores do campo da Engenharia de Software têm desenvolvido abordagens para extrair informações pertinentes e descobrir relações e tendências de repositórios no contexto da evolução do software [13]. A inteligência analítica no desenvolvimento de software torna possível a utilização de indicadores para a tomada de decisão. Sem esses indicadores, desenvolvedores e gestores contam apenas com intuição e a experiência para embasar decisões. Ferramentas online e ou integradas ao ambiente de desenvolvimento de software estão sendo utilizadas para esse fim [4]. Ou seja, oportunidades estão sendo criadas para serem exploradas nesta área.

B. *Tipos de Análise da Inteligência Analítica*

A inteligência analítica fornece aos engenheiros de software a possibilidade de analisar os projetos sob sua responsabilidade. Essas análises são úteis para saber o que está acontecendo, como está acontecendo, quais são melhores ações a serem tomadas e o que pode acontecer ao projeto de software.

Nesta Seção, apresentam-se os tipos de análise fornecidos pelas ferramentas de inteligência analítica aos engenheiros de software, com base no trabalho de Buse e Zimmermann [1].

Buse e Zimmermann [1] realizaram um *survey* com desenvolvedores e gerentes de software das diversas áreas da Microsoft, com a finalidade de descobrir como a inteligência analítica pode contribuir para embasar decisões no desenvolvimento de sistemas. Os autores categorizaram os tipos de análise, os quais se encontram descritos conforme o Quadro 1.

Os tipos de análise são caracterizados de acordo com o tempo (passado, presente e futuro) e a técnica aplicada (exploração, análise e experimentação). Com base nessa categorização, é possível responder a questões como: O que está acontecendo? Como e por que isso aconteceu? O que está acontecendo agora? Qual a melhor ação? O que acontecerá? O que de melhor ou pior pode acontecer? [1].

Quadro 1. tipos de análise da inteligência analítica [1]

	Passado	Presente	Futuro
Exploração	Tendência Quantifica como os artefatos estão mudando.	Alerta Informa mudanças incomuns nos artefatos quando elas acontecem.	Previsão Prevê eventos baseados em tendências.
Análise	Sumarização Características sucintas de artefatos-chaves ou grupos de artefatos.	Sobreposição Compara artefatos ou históricos de desenvolvimento de forma interativa.	Objetivos Descobre como os artefatos estão mudando com os respectivos objetivos.
Experimentação	Modelagem Caracteriza o comportamento normal do desenvolvimento.	Comparação Compara artefatos para obter as melhores práticas.	Simulação Testa condições antes de acontecerem.

C. Necessidades dos Engenheiros de Software

Com o intuito de descobrir quais são os questionamentos sobre o projeto de software, a inteligência analítica pode ajudar a responder aos engenheiros de software. Begel e Zimmermann [14] realizaram dois *surveys* na Microsoft e categorizaram 145 questões que os engenheiros de software gostariam que os cientistas de dados respondessem. Dessa maneira, torna-se possível verificar como a inteligência analítica pode auxiliar gerentes e desenvolvedores em suas atividades. Essas 145 questões foram categorizadas em 12 grupos. O resultado desta categorização é apresentado no Quadro 2.

Quadro 2. Necessidades dos Engenheiros de Software com a Inteligência Analítica [14]

Medições de defeitos (Bugs)	Prática de Desenvolvimento	Melhores Práticas
Onde os defeitos são encontrados, os mais comuns, seu ciclo de vida e o custo para consertar defeitos.	Relacionado ao código, depuração, perfil de desempenho, refatoração, <i>branching</i> em repositório, revisão de código, comentários e documentação de código. Estimativa de esforço e riscos devido as mudanças no código.	Melhores conduções no projeto de software. Como técnicas adequadas de migrações entre as versões de software, a melhor maneira de controlar os itens de trabalho, o momento certo para usar métodos formais para a análise de software ou quais critérios devem influenciar a decisão de usar uma linguagem de programação

		específica ou API
Prática de Teste Cobre automação de teste, estratégia de teste, teste unitário, desenvolvimento orientado a testes, processos de teste, escritas, infraestruturas, medidas e impactos nos defeitos.	Avaliação da Qualidade Trata sobre otimização de código, melhores métricas, código duplicado e problemas de qualidade de software.	Serviços Forte relação com o desenvolvimento na <i>nuvem</i> , os efeitos sobre a retenção de clientes e monetização causadas nas versões do software.
Clientes e Requisitos Relacionado aos interesses do cliente, gosto do cliente, compatibilidade com versões anteriores. Investimento em especificações técnicas, custo com o aumento de clientes.	Ciclo de Vida do Desenvolvimento de Software Como o tempo de desenvolvimento deve ser alocado entre planejamento, projeto, codificação e teste, impacto dessas alocações no software.	Processo de Desenvolvimento de Software Funcionamento das metodologias de processos de software e seus impactos na empresa.
Produtividade Investigar a métricas de qualidade para medir produtividade dos desenvolvedores, como monitorar a produtividade da equipe ou do indivíduo.	Equipe e Colaboração Formação de equipes, funções de desenvolvimento, quantidade de recursos e práticas para gestão do conhecimento.	Reuso e Compartilhamento de Componentes Reuso de código, seus componentes, custo de reaproveitamento de código.

D. Áreas do Conhecimento da Engenharia de Software

Com a finalidade de reconhecer as áreas da Engenharia de Software em que estão sendo aplicadas as técnicas de inteligência analítica, este artigo utilizou como base quinze áreas do conhecimento da engenharia de software descritas no guia SWEBOK (*Software Engineering Body of Knowledge*) [15]. Cada área pode ser descrita como:

- **Requisitos de Software** - concentra a elicitação, análise, especificação, validação e gerenciamento de requisitos de software.
- **Projeto de Software** - define a arquitetura, componentes, interfaces, características do sistema e seus componentes.
- **Construção de Software** – refere-se aos detalhes de criação de software, codificação, verificação, testes de unidade, integração e *debugging*.

- **Teste de Software** - atividades de teste de software, objetivos de teste, técnicas, avaliação, performance e medidas de teste, processos de teste e ferramentas.
- **Manutenção de Software** - recorrente às modificações no software, preservando sua integridade.
- **Gerenciamento de Configuração de Software** - configuração física e funcional do sistema de software que armazena e organiza artefatos arquivados do projeto de software, controlando suas alterações.
- **Gestão da Engenharia de Software** - gerencia as atividades de planejamento, coordenação, medidas, monitoramento, controle e relatórios dos produtos e serviços de software, eficiência das entregas e os benefícios às partes interessadas.
- **Processos de Engenharia de Software** - consistem em atividades inter-relacionadas da Engenharia de Software durante o ciclo de vida do software. Facilitam o entendimento, comunicação e coordenação dos envolvidos para o gerenciamento de projetos de software.
- **Modelos e Métodos da Engenharia de Software** - impõem estrutura sobre a Engenharia de Software, com o objetivo de fazer as atividades sistemáticas com maior sucesso.
- **Qualidade de Software** - garante que as especificações do produto sejam satisfeitas.
- **Prática Profissional em Engenharia de Software** – concentra os conhecimentos, habilidades, atitudes, responsabilidades e maneiras éticas dos profissionais de software, bem como a dinâmica do grupo e sua comunicação.
- **Economia em Engenharia de Software** - relacionada à tomada de decisão no contexto de negócio. Relaciona-se ao custo e ao risco de produtos de software e dos ecossistemas do projeto de software.
- **Fundamentos da Computação** - propõem as práticas computacionais para a Engenharia de Software. Ou seja, é o núcleo da Engenharia de Software com o entendimento da Ciência da Computação.
- **Fundamentos Matemáticos** - cobrem as técnicas básicas para identificar o conjunto de regras para o raciocínio no contexto de sistemas.
- **Fundamentos da Engenharia** - resultam em habilidades e técnicas usadas sistematicamente na Engenharia de Software. Relacionadas a estatística, medidas, modelos, simulações, protótipos e análises de causa raiz.

III. TRABALHOS RELACIONADOS

Trabalhos de revisão e mapeamento sistemáticos da literatura têm contribuído para entender o que está

sendo pesquisado na Engenharia de Software. Heckman e Williams [16] realizaram uma revisão sistemática da literatura para identificar as técnicas de análise estática automatizada que identifiquem potenciais anomalias em código-fonte. De 17.571 artigos, selecionaram 21 trabalhos relevantes para a análise das técnicas de análise estática automatizada. Semelhante ao trabalho de Heckman e Williams [16], este mapeamento sistemático da literatura procurou identificar técnicas que auxiliem os engenheiros de software. Porém, o foco deste mapeamento está associado à utilização da inteligência analítica como técnica que apoie as atividades dos engenheiros de software.

Demeyer et al. [17] aplicaram a abordagem de mineração de textos N-Gram sobre os artigos da Conferência MSR, analisando as tendências que estão seguindo os artigos da conferência de 2004 a 2012. Os autores verificaram os termos mais frequentes e os menos frequentes, os casos mais utilizados e os menos utilizados, as infraestruturas utilizadas e informações práticas que estão sendo estudadas. Como resultados, descobriram que muitos estudos sobre evolução de software estão sendo realizados e poucos estudos sobre padrões de projeto. Nas pesquisas, são frequentemente utilizados softwares livres, como o Eclipse, que dominam as pesquisas. Porém, poucos casos da indústria são relatados. O CVS e o Bugzilla são geralmente utilizados, sendo que o SVN, o Git e o Jira estão ganhando popularidade. Existe a tendência de obter dados de repositórios de comunicação, como e-mail e o *Stackoverflow*. O uso de técnicas de desenvolvimento e testes são amplamente discutidas nas pesquisas, contudo o ganho da utilização das técnicas para as organizações nem sempre é considerada. Ao contrário do trabalho de Demeyer et al. [17], o presente mapeamento não focou em uma única conferência, ampliando a busca para várias bases. Ele teve a intenção de descobrir o direcionamento das pesquisas nessa área. Contudo, a ideia deste trabalho é revelar a aplicabilidade da inteligência analítica na Engenharia de Software, e não apenas visualizar as tendências da conferência.

Novais et al. [18] mapearam 146 artigos sobre a visualização da evolução de software, descrevendo as características da visualização de software utilizadas na Engenharia de Software. Novais et al. [18] perceberam a utilização de múltiplas métricas, estratégias e perspectivas e da carência da validação das abordagens propostas. Já o presente mapeamento sistemático direcionou seus esforços em descobrir as carências de pesquisas sobre inteligência analítica na Engenharia de Software.

A taxonomia na mineração de repositórios de software foi tratada por Kagdi [13]. Ele analisou 80 trabalhos para investigar as taxonomias da mineração de repositório na evolução de software. Esse trabalho teve a intenção de facilitar as comparações das abordagens na mineração de repositórios, para entender a evolução de sistemas de software. O entendimento da evolução de sistemas de software é

um tópico relevante na mineração de repositórios. A mineração de repositórios de software é a base da inteligência analítica na Engenharia de Software. O presente mapeamento sistemático analisou trabalhos de mineração repositórios de software, para compreender a técnica de inteligência analítica na Engenharia de Software.

Robles [19] realizou uma revisão dos artigos publicados no *Workshop on Mining Software Repositories* (MSR) entre 2004 a 2006 e na *Working Conference on MSR* de 2007 a 2009. O intuito dessa revisão foi verificar os experimentos que pudessem ser replicados. Para isso, ele analisou os artigos empíricos, verificando se os dados e as ferramentas utilizadas nos artigos estavam disponíveis ao público. Do total de 171 artigos analisados, apenas 2 se mostraram aptos para replicação. No presente mapeamento foram analisados trabalhos empíricos e teóricos, não se preocupando com os dados para replicação.

IV. METODOLOGIA DE PESQUISA

Como suporte no mapeamento sistemático [20] empreendido contou-se com o uso do protocolo de mapeamento de estudos proposto por Kitchenham [21], [22], o qual auxilia no direcionamento das pesquisas de revisões sistemáticas e dos mapeamentos sistemáticos da Engenharia de Software. Nesse protocolo, foram declaradas: necessidades e escopo da pesquisa, estratégia de busca dos artigos, os critérios de seleção e exclusão, dados extraídos e limitações e validações do trabalho.

A. Questões de Pesquisa

Como objetivo desta pesquisa, pretende-se descobrir o que está sendo produzido sobre a inteligência analítica no âmbito da Engenharia de Software. Para investigar as pesquisas realizadas nessa área, este artigo propõe-se a responder as seguintes questões de pesquisa:

Questão de pesquisa 1 (QP1): Quais são os tipos de análises a inteligência analítica está proporcionando à Engenharia de Software? Esta questão tem por objetivo apontar os tipos de análise dos dados de repositórios de software que têm sido frequentes na literatura. Pretende-se, por meio desta questão, descobrir como essas informações ajudam na compreensão de projetos de software.

Questão de pesquisa 2 (QP2): Quais são as áreas de conhecimento da Engenharia de Software que têm maior predominância no uso da inteligência analítica? A Engenharia de Software é composta por diversas áreas do conhecimento. Para compreender o modo como a inteligência analítica está sendo utilizada na Engenharia de Software, é importante direcionar a área de predominância desta técnica no desenvolvimento de software. Esta questão tem por objetivo nortear a utilização da inteligência analítica nas áreas de conhecimento da Engenharia de Software.

Questão de pesquisa 3 (QP3): As pesquisas em inteligência analítica têm suprido as necessidades de informações dos engenheiros de software? Por meio

desta questão, pretende-se descobrir se as pesquisas disponíveis na literatura estão alinhadas com as necessidades reais dos engenheiros de software apresentadas por Begel e Zimmermann [14] e descrita na Seção II deste trabalho.

B. Estratégia de Busca

Este mapeamento buscou pelos descritores *software analytics*, *software intelligence*, *mining software repositories* e *software development analytics* nas bases da *ACM Digital Library*, *IEEE Xplore*, *Springer*, *Elsevier* e *Wiley*. Tais descritores foram referenciados por Zhang [23] em *Software analytics in practice: approaches and experiences*, o termo *software analytics* foi tratado por Buse e Zimmermann [11] em seu trabalho *Analytics for Software Development*. O termo *analytics* foi também usado por Hullett et al. [24] para tratar os dados que direcionam o desenvolvimento de software, mais especificamente o desenvolvimento de jogos digitais. Inspirados no termo *business intelligence*, Hassan e Xie [4] chamaram de *software intelligence* o uso de dados no desenvolvimento de software para a tomada de decisão. Devido ao fato de as pesquisas de inteligência analítica na Engenharia de Software tratarem sobre mineração de repositório software, o termo *mining software repositories* foi utilizado para ampliar as possibilidades de encontrar trabalhos relevantes sobre o uso da inteligência analítica na Engenharia de Software. Dessa maneira, os descritores de busca utilizados neste artigo foram formulados de acordo com o Quadro 3.

Quadro 3. Descritores de Busca

Base de Busca	Descritores
ACM Digital Library	(Abstract:"software analytics" or Abstract:"software intelligence" or Abstract:"Mining Software Repositories" or Abstract:"Software Development Analytics") or (Title:"software analytics" or Title:"software intelligence" or Title:"Mining Software Repositories" or Title:"Software Development Analytics") for: ((Abstract:"software analytics" or Abstract:"software intelligence" or Abstract:"Mining Software Repositories" or Abstract:"Software Development Analytics") or (Title:"software analytics" or Title:"software intelligence" or Title:"Mining Software Repositories" or Title:"Software Development Analytics")) and ("software analytics" or "software intelligence" or "Mining Software Repositories" or "Software Development Analytics")
IEEE Xplore	((("software analytics") OR "software intelligence") OR "Mining Software Repositories") OR "Software Development Analytics")
Springer	("software analytics" OR "software intelligence" OR "Mining Software Repositories" OR "Software Development Analytics")
Elsevier	("software analytics") or ("software intelligence") or ("Mining Software Repositories") or ("Software Development Analytics")
Wiley	"software analytics" in All Fields OR "software intelligence" in All Fields OR "Mining Software Repositories" in All Fields OR "Software Development Analytics" in All Fields

Visto que grande parte das pesquisas na busca automática retornou das conferências MSR (*Working Conference on Mining Software Repositories*) e a ICSE (*International Conference on Software Engineering*), foram adicionadas a esta pesquisa os seus artigos premiados como mais influentes para ampliar a possibilidade de encontrar artigos relevantes.

C. Critérios de Inclusão/Exclusão

Como critério de inclusão, foram verificados os artigos que tratavam de temas da Engenharia de Software e que utilizaram inteligência analítica; ou seja, artigos que usaram as técnicas de análise dos dados de repositórios de software ou artigos teóricos que tratavam sobre o tema. Neste artigo, contou-se com apenas artigos escritos na língua inglesa. Foram incluídos os artigos que continham os descritores de busca no título, resumo e palavras-chaves e que no corpo do texto tratavam explicitamente da Engenharia de Software. Devido à falta de padronização dos termos para se tratar o tema, os artigos colhidos manualmente foram incluídos pela interpretação do título e do resumo, e não pelos descritores de busca. Foram analisados artigos completos ou que exibam resultados parciais da pesquisa.

Foram excluídos artigos que não tratavam de forma explícita da Engenharia de Software e cujos descritores de busca estavam apenas citados na referência bibliográfica, no cabeçalho ou no rodapé do artigo. Foram excluídos também livros, seções de livros, entrevistas, resumos e chamadas para conferências e workshops e editoriais em geral. Também, foram eliminadas as duplicidades de artigos que estavam em mais de uma base. Apurou-se que 5 dos artigos retornados na busca da base *Wiley* e 1 na base da *Springer* estavam com problemas no redirecionamento dos documentos, não sendo possível ter acesso a eles, mesmo procurando-os em outros sistemas de busca. Por essa razão não foram considerados na análise deste mapeamento.

D. Seleção dos Artigos

A quantidade de artigos encontrados nas bases de busca por meio da estratégia de busca e após a aplicação dos critérios de seleção está retratada na Tabela 1. A Figura 1 apresenta o fluxo como os artigos foram refinados. Ressalta-se que os artigos das conferências MSR e ICSE indicados na Tabela 1 fazem parte de artigos premiados pelas conferências cujos descritores desta pesquisa não puderam capturar na busca automática. Muitos dos artigos dessas conferências fazem parte do IEEE Xplore e ACM Digital Library.

Tabela 1. Quantidade de Artigos por Base de Dados

Base de Busca	Quantidade de Artigos encontrados	Quantidade de Artigos Após Critério de Seleção
ACM Digital Library	78	29
IEEE Xplore	350	107
Springer	182	27
Elsevier	436	27
Wiley	228	12
MSR	31	30
ICSE	27	3
Total	1332	235

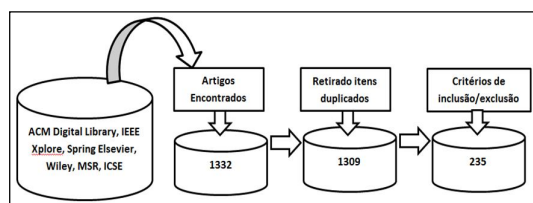


Figura 1. Refinamento dos Artigos para o Mapeamento Sistemático da literatura.

E. Extração dos Dados

Para melhor analisar os artigos selecionados pelo critério de inclusão deste mapeamento sistemático, foram extraídos dos artigos os seguintes dados: título da publicação, autores, local de publicação, ano de publicação, tipo de questão da pesquisa, área de conhecimento da Engenharia de Software, condução da pesquisa, tipos de artefatos produzidos, tipos de análise da inteligência analítica e necessidades dos engenheiros de software.

Como suporte para a classificação dos artigos deste mapeamento foram utilizadas categorias disponíveis na literatura. Os trabalhos de Shaw [25] [26] forneceram base para classificar os tipos de questões de pesquisa. Estes trabalhos citam que as pesquisas em Engenharia de Software respondem às perguntas sobre método ou meio de desenvolvimento, projeto, avaliação ou análise de um caso particular; generalização ou caracterização, método para análise ou avaliação e estudo de viabilidade ou exploração.

Para realizar a classificação das áreas de conhecimento da Engenharia de Software foram utilizadas suas 15 áreas referenciadas pelo SWEBOK v3 no *Guide to the Software Engineering Body of Knowledge* [15], descritos na Seção II desse trabalho.

Para determinar as conduções das pesquisas dos artigos analisados, foram utilizados os conceitos apresentados no trabalho *Experimentation in Software Engineering* de Wohlin et al. [27], no qual os autores citam as conduções de pesquisa em Engenharia de Software, como experimento, estudo de caso, *survey*, revisão sistemática da literatura e teoria em Engenharia de Software.

A classificação dos tipos de artefatos produzidos foi baseada no trabalho *Design Science Research Evaluation*, de Peffers et al. [28], no qual os autores descrevem que os artefatos podem ser classificados como: algoritmo, construtor, framework, instanciação, método ou modelo.

Buse e Zimmermann [1] caracterizaram os tipos de análise mais comuns realizados com as ferramentas de inteligência analítica descritos no Quadro 1. Begel e Zimmermann [14] categorizaram as necessidades dos engenheiros de software que podem ser supridas pela inteligência analítica. As descrições dessas necessidades podem ser vista conforme o Quadro 2.

V. RESULTADOS

A. Análises Descritivas

Os artigos analisados no mapeamento apontam o aumento do número de pesquisas sobre a inteligência analítica na Engenharia de Software nas últimas décadas, ou seja, entre anos de 2004 a 2014. O primeiro workshop sobre mineração repositórios de software foi realizado em conjunto com a conferência ICSE, em 2004 [17]. Sete artigos que constam na base deste mapeamento datam do início de 2015, porém foram retirados da análise da Figura 2. No momento em que este artigo foi escrito, ainda não era possível analisar a real tendência das pesquisas de inteligência analítica na Engenharia de Software no ano de 2015.

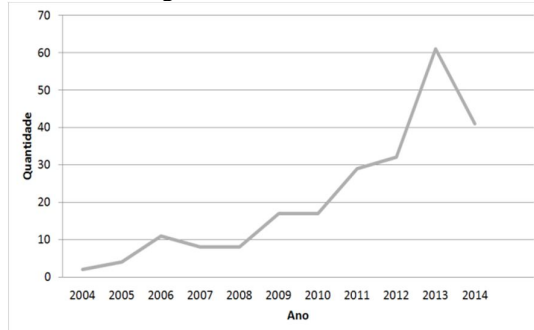


Figura 2. Quantidade de Publicações por Ano.

Dentre os tipos de questões de pesquisas utilizados nos artigos analisados 35% foram caracterizados como generalização ou caracterização, 18% procuraram um método para análise ou avaliação e 16% um método ou meio de desenvolvimento, estudo de viabilidade ou exploração e projeto, avaliação ou análise de um caso particular.

O tipo de artefato mais frequente nas pesquisas foram os métodos. A condução de pesquisa mais utilizadas para validar esses artefatos foi o estudo de caso, como exibido pela Figura 3.

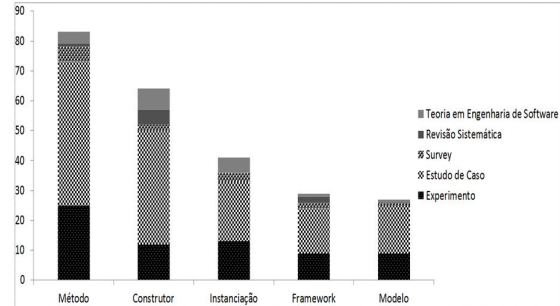


Figura 3. Artefato versus Validação da Pesquisa.

Novos métodos têm sido validados por meio de estudos de caso para entender os problemas enfrentados por desenvolvedores [29] caracterizar *commits* [30], estimar esforço [31], gerir conhecimento [32] e caracterizar defeitos [33] entre outros.

B. QP1 – Quais são os tipos de análise que a inteligência analítica está proporcionando à Engenharia de Software?

Os tipos de análise de Tendências, referenciados na Seção II, estão presentes nas análises de 48 artigos deste mapeamento e a modelagem em 45 artigos. Entre os 235 artigos analisados, apenas 1 referenciou a inteligência analítica como análise de simulação, conforme a Figura 4. Isso indica um interesse nas pesquisas em entender as mudanças dos artefatos dos repositórios de software, apresentando múltiplas visões ou construindo uma representação do projeto de software. Poucos se arriscaram a realizar análises que detectem eventos que impliquem no funcionamento do software ou indiquem o que aconteceria ao projeto por meio de condições alternativas.

Este mapeamento confirma os resultados do trabalho de Buse e Zimmermann que dizem que tanto desenvolvedores quanto gerentes acham mais importante compreender o passado do que tentar prever o futuro [1].

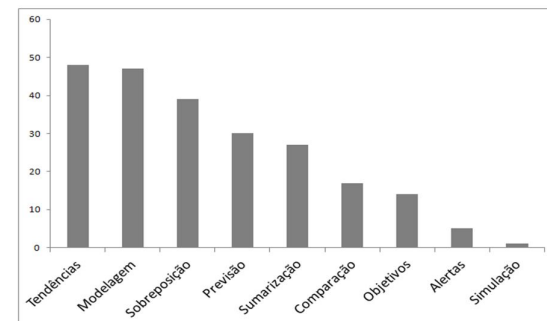


Figura 4. Tipos de Análises na Inteligência Analítica.

C. QP2 – Quais são as áreas de conhecimento da Engenharia de Software que têm maior predominância no uso da inteligência analítica?

A maior parte da utilização da inteligência analítica na Engenharia de Software se concentra na área de conhecimento da Manutenção de Software, seguido pela Construção de Software e Fundamentos da Computação, referenciados na Seção II deste artigo. A utilização da inteligência analítica na área da Manutenção de Software é bastante expressiva quando comparada a outras áreas.

Observa-se uma demanda das pesquisas nas áreas de Gestão de Engenharia de Software e de Práticas

Profissionais em Engenharia de Software, que vem crescendo desde 2008, conforme mostra a Figura 5.

Essas pesquisas contribuem para o gerenciamento de projetos de software e permitem analisar a dinâmica de equipes de desenvolvimento de software. Dessa maneira, a inteligência analítica tem sido utilizada para gerenciar incidentes de serviços online [34], planejar o projeto e otimizar as atividades de correção de defeitos [35], obter conhecimento sobre o risco do projeto de software [36], estimar esforço em projetos de software [37], [38], investigar redes de conhecimento em equipes de desenvolvimento de software [32] e reconhecer as habilidades e conhecimentos dos desenvolvedores de software [29], [39], entre outras atividades.

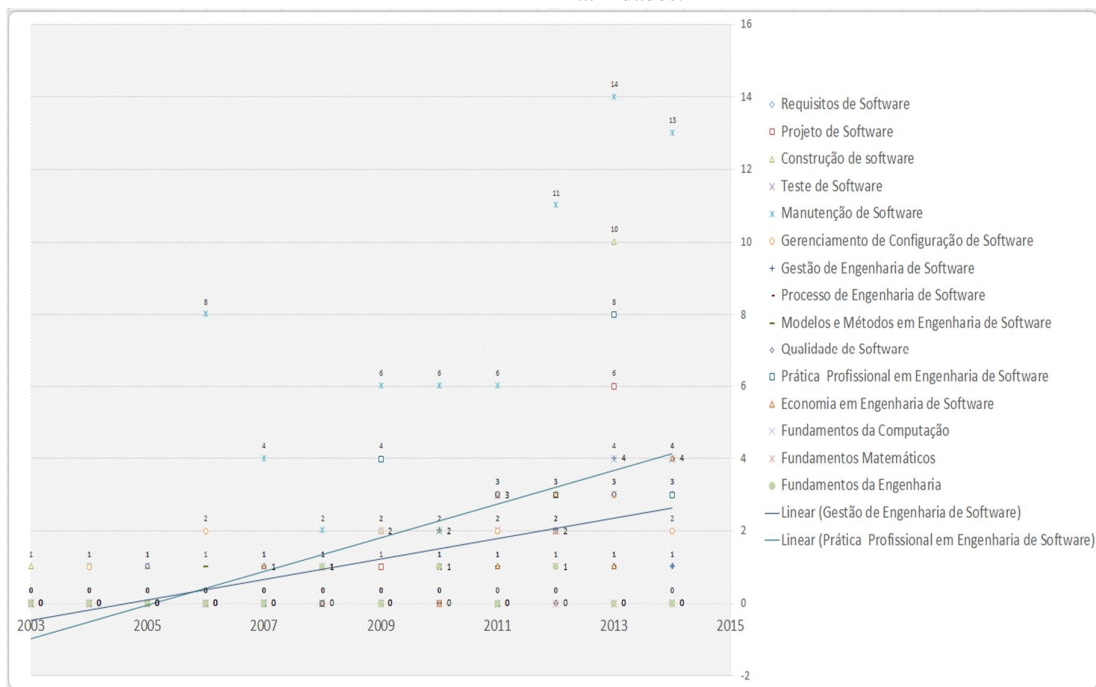


Figura 5. Áreas da Engenharia de Software versus Ano.

D. QP3 – As pesquisas em inteligência analítica têm suprido as necessidades de informações dos engenheiros de software?

A maioria das pesquisas sobre inteligência analítica na Engenharia de Software presentes neste mapeamento procurou suprir as necessidades de práticas de desenvolvimento. Conforme a pesquisa de Begel e Zimmermman [14], em que os autores realizaram um *survey* com os profissionais da Microsoft para descobrir as questões dos engenheiros de software que podem ser respondidas com a inteligência analítica. A categoria de maior destaque foi Prática de Desenvolvimento, descrita na fundamentação teórica deste artigo. Em contrapartida, as questões sobre Práticas de Teste, Processos de Desenvolvimento de Software e Ciclo de Vida de Desenvolvimento, bastante apontadas no *survey* de Begel e Zimmermman [14], não foram totalmente refletidas nas pesquisas analisadas neste mapeamento, como aponta a Figura 6. O mapeamento mostra que as pesquisas ainda não estão totalmente alinhadas com as necessidades práticas dos engenheiros de software descritas por Begel e Zimmermman [14].

Ressalta-se que a pesquisa realizada por Begel e Zimmermman [14] focou os engenheiros de software da Microsoft. Os próprios autores apontam esse fator como ameaça à validade da pesquisa. Para reforçar os achados, é encorajada a replicação dessa pesquisa em outras instituições. Contudo, o presente mapeamento apoia-se nos resultados deste trabalho por estar bem embasado, contando com a participação de 203 engenheiros de software no primeiro *survey* e de 607 no segundo *survey*. Contudo, espera-se que o presente

mapeamento possa inspirar novas pesquisas em inteligência analítica para suprir as necessidades dos engenheiros de software que apresentaram maior carência de pesquisa.

Os resultados deste mapeamento e do *survey* [14] podem ser visualizados na Figura 6.

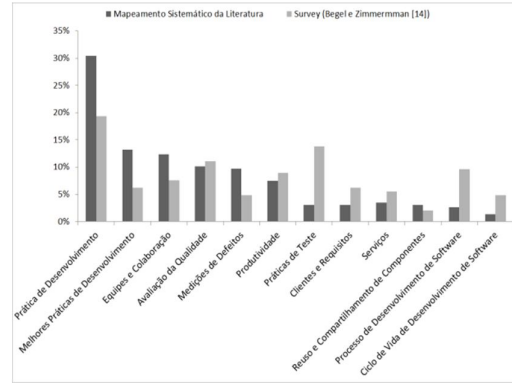


Figura 6. Quantidade de artigos classificados no mapeamento sistemático da literatura versus necessidades dos engenheiros de software com inteligência analítica [14].

E. Interações das Categorias

As interações das categorias analisadas neste mapeamento entre as áreas do conhecimento da Engenharia de Software, os tipos de análises e as necessidades dos engenheiros de software podem ser verificadas na Figura 7.

Medições de Defeitos	Prática de Desenvolvimento	Melhores Práticas de Desenv.	Práticas de Teste	Avaliação da Qualidade	Serviços	Clientes e Requisitos	Ciclo de Vida de Desenvol. de Soft.	Processo de Desenvol. de Soft.	Equipes e Colaboração	Reuso e Compartilh. de Compon.	Produtividade	Tendências	Requisitos de Software	Projeto de Software	Construção de software	Teste de Software	Manutenção de Software	Gerenciamento de Config. De Soft.	Gestão de Eng. de Soft.	Processo de Eng. de Soft.	Modelos e Métodos em Eng. de Soft.	Qualidade de Software	Prática Profissional em Eng. de Soft.	Economia em Eng. de Soft.	Fundamentos da Computação	Fundamentos Matemáticos	Fundamentos da Engenharia
4	21	3	4	5	1	3	0	1	2	1	4	17	0	3	3	4	17	2	1	1	0	4	3	6	2	0	1
5	8	7	0	6	3	2	0	1	2	4	2	13	0	1	8	1	13	2	1	1	3	4	2	1	3	0	0
4	11	7	0	1	1	2	1	1	15	1	4	9	1	2	6	0	9	9	2	3	0	1	9	1	5	0	0
2	8	1	0	4	0	1	0	2	8	1	1	16	1	2	4	0	8	1	1	1	0	1	4	1	2	0	1
0	5	0	0	2	0	0	1	0	1	0	5	4	0	1	1	0	4	0	4	1	2	0	1	0	0	0	0
1	2	8	1	3	2	0	0	0	0	0	0	3	0	2	3	1	3	1	0	0	0	0	1	1	4	0	1
1	1	0	1	0	1	0	0	1	0	0	0	1	0	2	0	1	1	0	1	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 7. Necessidades dos engenheiros de software versus tipos de análises versus áreas do conhecimento da Engenharia de software.

Os tipos de análises de tendências, sobreposição, previsão, sumarização estiveram presentes nas pesquisas com foco na área de Manutenção de Software, atendendo às necessidades de prática de desenvolvimento. O tipo de análise de modelagem se mostrou mais focado a atender às necessidades de

equipe e colaboração utilizadas nas áreas de Gerenciamento de Configuração de Software e na Prática Profissional de Engenharia de Software. O tipo de análise objetivos, além de estar presente na área de manutenção de software e da prática de desenvolvimento está fortemente ligado às pesquisas

na área de Gestão de Engenharia de Software, atendendo às necessidades ligadas a produtividade. O tipo de análise de comparação tem sido pesquisado na área de Fundamentos da Computação, objetivando produzir melhores práticas de desenvolvimento. Nota-se que análises de comparação estão sendo usadas na área de Fundamentos da Computação, além das áreas de Manutenção e Construção. Em Fundamentos da Computação, as pesquisas estão interessadas em melhorar a performance [40], [41], recuperação da informação [42] e o aumento de interoperabilidade em repositórios de software [43].

VI. DISCUSSÃO DOS RESULTADOS

De acordo com os resultados deste mapeamento, 84% dos trabalhos foram classificados como experimento ou estudo de caso, ou seja, percebe-se que muitos trabalhos empíricos estão sendo feitos utilizando inteligência analítica na Engenharia de Software. Devido a aplicabilidade das técnicas de inteligência analítica, torna-se um forte aliado para analisar as mudanças que ocorrem durante a fase de manutenção. Essa aplicabilidade é bastante visível para se entender a evolução de software, tratamento de defeitos e incidentes e análise de código, por exemplo.

De acordo com os resultados deste mapeamento, percebeu-se que a inteligência analítica tem sido bastante explorada para ajudar gerentes de projeto a entenderem melhor as equipes e a colaboração entre os desenvolvedores, bem como sua produtividade. Contudo, ainda é percebida a falta de pesquisas na área de requisitos, testes e processos de software.

Os tipos de análise que permitem compreender melhor o projeto de software têm sido bastante utilizados, como para analisar tendências, modelar e sumarizar o projeto, por exemplo. Podendo estes ser úteis para descobrir características do projeto nas áreas da Engenharia de Software. Tipos de análises que indicam possíveis acontecimentos como Previsão e Objetivos estão sendo utilizados na Engenharia de Software para recomendar equipes, realizar previsões de defeito, esforço, ou para recomendar refatorações, por exemplo. O mesmo não pode ser dito para o tipo de análise de simulação. Por meio deste mapeamento não se pode verificar uso significativo deste tipo de análise ainda pouco explorado na Engenharia de Software.

Diante do exposto entende-se que a inteligência analítica ajuda engenheiros de software a tomarem decisões, seja em nível de código, refatoração e teste. Bem como, no nível gerencial, em relação ao tratamento de demandas, defeitos e gestão de equipe. Dessa maneira esta pesquisa espera inspirar novos trabalhos de inteligência analítica tanto para preencher lacunas quanto para ampliar trabalhos em áreas já exploradas. De acordo com o presente estudo a utilização das técnicas de inteligência analítica vem se tornando cada vez mais integrada as práticas da Engenharia de Software.

VII. AMEAÇAS À VALIDADE DA PESQUISA

Alguns fatores podem comprometer os resultados deste mapeamento, tal como os descritores de busca, que, apesar de bem embasados, podem não ter englobado todos os trabalhos relevantes sobre inteligência analítica. Para promover os trabalhos analisados neste mapeamento os descritores de busca foram embasados conforme descrito na Seção IV, ou seja, procurou na literatura os termos mais comuns para encontrar os artigos para esta pesquisa. Em conhecimento da fragilidade dos descritores, foram buscados artigos relevantes nas bases das principais conferências da área.

Outro ponto a se destacar é que neste mapeamento apenas referenciou artigos escritos em língua inglesa. Contudo, entende-se que os trabalhos encontrados nas maiores conferências da Engenharia de Software se encontram escritos na língua inglesa. Ressalta-se ainda que a dificuldade de encontrar trabalhos com os termos utilizados neste trabalho em escritos em português devido a sua taxonomia.

Entende-se que, por se tratar de uma forma interpretativa, a categorização dos artigos pode ser considerada subjetiva. Para evitar essa subjetividade os artigos foram classificados criteriosamente de acordo com os estudos expostos na Seção II deste trabalho.

Apontam-se, ainda, os critérios de seleção dos artigos e a quantidade de bases verificadas, que poderiam ter sido ampliadas para aumentar as chances de encontrar trabalhos relevantes. Neste trabalho a maior parte dos artigos analisados retornou das plataformas de busca automatizadas. Para evitar problemas de não encontrar os trabalhos relevantes nessa pesquisa foram incluídas as conferências *International Conference on Software Engineering (ICSE)* e *International Conference on Mining Software Repositories (MSR)*.

O presente trabalho se baseou nas pesquisas realizadas por Buse e Zimmermann [1] e Begel e Zimmermann [14], para classificar os tipos de análise e as necessidades dos engenheiros de software. Esses trabalhos utilizaram como universo de pesquisa a empresa Microsoft. Apesar de não poder generalizar os resultados desses trabalhos, não foram encontradas outras pesquisas, até o momento em que o presente mapeamento foi realizado, com as propostas e embasamentos utilizadas por Buse e Zimmermann [1] e Begel e Zimmermann [14]. Buse e Zimmermann [1] e Begel e Zimmermann [14].

VIII. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou um mapeamento sistemático da literatura sobre o uso de inteligência analítica na Engenharia de Software. Constatou-se que a maioria das pesquisas procurou generalizar ou caracterizar elementos no desenvolvimento de sistemas, bem como desenvolver métodos para a utilização da inteligência analítica. Dos 235 artigos analisados, houve apenas uma referência de simulação

para analisar dados em repositórios de software. A inteligência analítica é muito utilizada nas atividades de manutenção de software, seja para tratar defeitos ou erros, realizar a análise de impacto de mudanças ou analisar a evolução de software entre outros tópicos relativos à manutenção. Observou-se um interesse cada vez maior nos estudos sobre as práticas profissionais e gestão de projetos. O presente estudo indica que as necessidades reais dos engenheiros de software não estão sendo totalmente pesquisadas, carecendo de maiores investimentos na prática de teste e nos processos e ciclo de vida do desenvolvimento de software.

Para facilitar a reprodução e a análise dos dados deste mapeamento, os artigos analisados podem ser obtidos pelo link: <http://migre.me/pDQXm>.

Durante a leitura dos artigos outros termos que tratam sobre inteligência analítica puderam ser observados como: *software mining*, *software repository mining*, *mining software*, *source repository mining*, *mining software engineering data* e *engineering intelligence analytics*. Desta maneira, como sugestão para a realização de trabalhos futuros, é interessante investigar os artigos retornados com esses descritores com a finalidade de ampliar os achados deste mapeamento. É importante frisar a dificuldade para encontrar termos que se encaixem nesse contexto, devido à utilização de diversos termos ou, até mesmo, a não utilização de termos específicos para se tratar de inteligência analítica na Engenharia de Software. Para expandir os achados deste trabalho recomenda-se, também, que sejam analisados artigos de outras conferências tais como: *The International Symposium on the Foundations of Software Engineering (FSE)*, *International Conference on Software Analysis, Evolution, and Reengineering (SANER)* e *The International Conference on Software Maintenance and Evolution (ICSME)*.

Com este trabalho, espera-se que novas pesquisas sejam incentivadas para explorar a inteligência analítica na utilização da Engenharia de Software.

REFERÊNCIAS

- [1] R. P. L. Buse e T. Zimmermann, "Information needs for software development analytics", in *2012 34th International Conference on Software Engineering (ICSE)*, 2012, p. 987–996.
- [2] W. Poncin, A. Serebrenik, e M. van den Brand, "Process Mining Software Repositories", in *2011 15th European Conference on Software Maintenance and Reengineering (CSMR)*, 2011, p. 5–14.
- [3] O. Baysal, "Informing development decisions: From data to information", in *2013 35th International Conference on Software Engineering (ICSE)*, 2013, p. 1407–1410.
- [4] A. E. Hassan e T. Xie, "Software Intelligence: The Future of Mining Software Engineering Data", in *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, New York, NY, USA, 2010, p. 161–166.
- [5] A. E. Hassan, "Mining Software Repositories to Assist Developers and Support Managers", in *22nd IEEE International Conference on Software Maintenance, 2006. ICSM '06*, 2006, p. 339–342.
- [6] T. Menzies e T. Zimmermann, "Software Analytics: So What?", *IEEE Softw.*, vol. 30, n° 4, p. 31–37, jul. 2013.
- [7] A. E. Hassan, "The road ahead for Mining Software Repositories", in *Frontiers of Software Maintenance, 2008. FoSM 2008.*, 2008, p. 48–57.
- [8] D. Zhang, Y. Dang, J.-G. Lou, S. Han, H. Zhang, e T. Xie, "Software Analytics As a Learning Case in Practice: Approaches and Experiences", in *Proceedings of the International Workshop on Machine Learning Technologies in Software Engineering*, New York, NY, USA, 2011, p. 55–58.
- [9] D. Zhang, S. Han, Y. Dang, J.-G. Lou, H. Zhang, e T. Xie, "Software Analytics in Practice", *IEEE Softw.*, vol. 30, n° 5, p. 30–37, set. 2013.
- [10] T. Zimmermann, A. Zeller, P. Weissgerber, e S. Diehl, "Mining version histories to guide software changes", *IEEE Trans. Softw. Eng.*, vol. 31, n° 6, p. 429–445, jun. 2005.
- [11] R. P. L. Buse e T. Zimmermann, "Analytics for Software Development", in *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, New York, NY, USA, 2010, p. 77–80.
- [12] Y. Zhang e D. Sheth, "Mining software repositories for model-driven development", *IEEE Softw.*, vol. 23, n° 1, p. 82–90, jan. 2006.
- [13] H. Kagdi, M. L. Collard, e J. I. Maletic, "A survey and taxonomy of approaches for mining software repositories in the context of software evolution", *J. Softw. Maint. Evol. Res. Pr.*, vol. 19, n° 2, p. 77–131, mar. 2007.
- [14] A. Begel e T. Zimmermann, "Analyze This! 145 Questions for Data Scientists in Software Engineering", in *Proceedings of the 36th International Conference on Software Engineering*, New York, NY, USA, 2014, p. 12–23.
- [15] P. Bourque e R.E. Fairley, "Guide to the Software Engineering Body of Knowledge, Version 3.0". IEEE Computer Society, 2014.
- [16] S. Heckman e L. Williams, "A Systematic Literature Review of Actionable Alert Identification Techniques for Automated Static Code Analysis", *Inf. Softw. Technol.*, vol. 53, n° 4, p. 363–387, abr. 2011.
- [17] S. Demeyer, A. Murgia, K. Wyckmans, e A. Lamkanfi, "Happy Birthday! A trend analysis on past MSR papers", in *2013 10th IEEE Working Conference on Mining Software Repositories (MSR)*, 2013, p. 353–362.
- [18] R. L. Novais, A. Torres, T. S. Mendes, M. Mendonça, e N. Zazworka, "Software evolution visualization: A systematic mapping study", *Inf. Softw. Technol.*, vol. 55, n° 11, p. 1860–1883, nov. 2013.
- [19] G. Robles, "Replicating MSR: A study of the potential replicability of papers published in the Mining Software Repositories proceedings", in *2010 7th IEEE Working Conference on Mining Software Repositories (MSR)*, 2010, p. 171–180.
- [20] K. Petersen, R. Feldt, S. Mujtaba, e M. Mattsson, "Systematic Mapping Studies in Software Engineering", in *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, Swinton, UK, UK, 2008, p. 68–77.
- [21] B. A. Kitchenham, D. Budgen, e O. Pearl Brereton, "Using Mapping Studies As the Basis for Further Research - A Participant-observer Case Study", *Inf. Softw. Technol.*, vol. 53, n° 6, p. 638–651, jun. 2011.
- [22] Kitchenham, Barbara, "Guidelines for performing systematic literature reviews in software engineering", Technical report, EBSE Technical Report EBSE-2007-01, 2007.
- [23] D. Zhang, "Software Analytics in Practice: Approaches and Experiences", in *Proceedings of the 9th IEEE Working Conference on Mining Software Repositories*, Piscataway, NJ, USA, 2012, p. 1–1.
- [24] K. Hullett, N. Nagappan, E. Schuh, e J. Hopson, "Data analytics for game development: NIER track", in *2011 33rd International Conference on Software Engineering (ICSE)*, 2011, p. 940–943.
- [25] M. Shaw, "What makes good research in software engineering?", *Int. J. Softw. Tools Technol. Transf.*, vol. 4, n° 1, p. 1–7, out. 2002.

- [26] M. Shaw, "Writing Good Software Engineering Research Papers: Minututorial", in *Proceedings of the 25th International Conference on Software Engineering*, Washington, DC, USA, 2003, p. 726–736.
- [27] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, e A. Wesslén, *Experimentation in Software Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
- [28] K. Peffers, M. Rothenberger, T. Tuunanen, e R. Vaezi, "Design Science Research Evaluation", in *Design Science Research in Information Systems. Advances in Theory and Practice*, K. Peffers, M. Rothenberger, e B. Kuechler, Orgs. Springer Berlin Heidelberg, 2012, p. 398–410.
- [29] A. Barua, S. W. Thomas, e A. E. Hassan, "What are developers talking about? An analysis of topics and trends in Stack Overflow", *Empir Softw. Eng*, vol. 19, n° 3, p. 619–654, jun. 2014.
- [30] A. Alali, H. Kagdi, e J. I. Maletic, "What's a Typical Commit? A Characterization of Open Source Software Repositories", in *The 16th IEEE International Conference on Program Comprehension, 2008. ICPC 2008*, 2008, p. 182–191.
- [31] A. Capiluppi e D. Izquierdo-Cortázar, "Effort estimation of FLOSS projects: a study of the Linux kernel", *Empir Softw. Eng*, vol. 18, n° 1, p. 60–88, fev. 2013.
- [32] C. Del Rosso, "Comprehend and analyze knowledge networks to improve software evolution", *J Softw Maint Evol Res Pr.*, vol. 21, n° 3, p. 189–215, maio 2009.
- [33] R. Souza, C. Chavez, e R. Bittencourt, "Patterns for extracting high level information from bug reports", in *2013 1st International Workshop on Data Analysis Patterns in Software Engineering (DAPSE)*, 2013, p. 29–31.
- [34] J.-G. Lou, Q. Lin, R. Ding, Q. Fu, D. Zhang, e T. Xie, "Software analytics for incident management of online services: An experience report", in *2013 IEEE/ACM 28th International Conference on Automated Software Engineering (ASE)*, 2013, p. 475–485.
- [35] T. T. Nguyen, T. N. Nguyen, E. Duesterwald, T. Klinger, e P. Santhanam, "Inferring developer expertise through defect analysis", in *2012 34th International Conference on Software Engineering (ICSE)*, 2012, p. 1297–1300.
- [36] C.-P. Chang, "Mining software repositories to acquire software risk knowledge", in *2013 IEEE 14th International Conference on Information Reuse and Integration (IRI)*, 2013, p. 489–496.
- [37] E. A. El-Sebakhy, "Functional networks as a novel data mining paradigm in forecasting software development efforts", *Expert Syst. Appl.*, vol. 38, n° 3, p. 2187–2194, mar. 2011.
- [38] G. Robles, J. M. González-Barahona, C. Cervigón, A. Capiluppi, e D. Izquierdo-Cortázar, "Estimating Development Effort in Free/Open Source Software Projects by Mining Software Repositories: A Case Study of OpenStack", in *Proceedings of the 11th Working Conference on Mining Software Repositories*, New York, NY, USA, 2014, p. 222–231.
- [39] P. Morrison e E. Murphy-Hill, "Is programming knowledge related to age? An exploration of stack overflow", in *2013 10th IEEE Working Conference on Mining Software Repositories (MSR)*, 2013, p. 69–72.
- [40] O. Kononenko, O. Baysal, R. Holmes, e M. W. Godfrey, "Mining Modern Repositories with Elasticsearch", in *Proceedings of the 11th Working Conference on Mining Software Repositories*, New York, NY, USA, 2014, p. 328–331.
- [41] W. Shang, B. Adams, e A. E. Hassan, "Using Pig as a data preparation language for large-scale mining software repositories studies: An experience report", *J. Syst. Softw.*, vol. 85, n° 10, p. 2195–2204, out. 2012.
- [42] S. Zamani, S. P. Lee, R. Shokripour, e J. Anvik, "A noun-based approach to feature location using time-aware term-weighting", *Inf. Softw. Technol.*, vol. 56, n° 8, p. 991–1011, ago. 2014.
- [43] G. Ghezzi e H. C. Gall, "Replicating mining studies with SOFAS", in *2013 10th IEEE Working Conference on Mining Software Repositories (MSR)*, 2013, p. 363–372.