Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 162 (2019) 712–719

Procedia Computer Science

www.elsevier.com/locate/procedia

7th International Conference on Information Technology and Quantitative Management (ITQM 2019)

Looking Over the Research Literature on Software Engineering from 2016 to 2018

Ruben Heradio[a], David Fernandez-Amoros[a], Daniel Galan[a], Francisco Javier Cabrerizo[b], Enrique Herrera-Viedma[b]

[a]*ETSI Informatica, Universidad Nacional de Educacion a Distancia, Madrid 28040, Spain*
[b]*Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain*

Abstract

This paper carries out a bibliometric analysis to detect (i) what is the most influential research on software engineering at the moment, (ii) where is being published that relevant research, (iii) what are the most commonly researched topics, (iv) and where is being undertaken that research (i.e., in which countries and institutions). For that, 6,365 software engineering articles, published from 2016 to 2018 on a variety of conferences and journals, are examined.

Keywords: Bibliometric analysis; Software engineering; Scopus

1. Introduction

The primary goal of this paper is to provide guidance on the vast literature on software engineering that it is being published nowadays. In particular, it targets the following Research Questions (RQs) for the period 2016-2018 (i.e., the last three years):

- RQ1: What articles are the most impacting?
- RQ2: Where have those articles been published?
- RQ3: What are the upward research trends?
- RQ4: Which countries and institutions are playing the principal role?

In 2007, Kitchenham and Charters published an influential technical report proposing some guidelines about how to perform systematic software engineering reviews, covering *surveys, systematic literature reviews, systematic mapping studies*, and *meta-analyses*. These guidelines were extended in Wohlin et al.'s book [1], and many software engineering reviews have applied them successfully, for instance, [2, 3, 4].

*Ruben Heradio Tel.: +31-913988242.
E-mail address: rheradio@issi.uned.es.

10.1016/j.procs.2019.12.042

Nevertheless, the review strategies mentioned above work for small-to-medium literature samples of a few hundreds of articles. In contrast, this paper analyzes 6,365 documents gathered from Elsevier's Scopus. To do so, it uses *bibliometric techniques*, which have already proven their usefulness in the software product line domain by analyzing thousands of papers automatically [5].

This article is organized in three sections. Section 2 describes the methodology and tools we have used to undertake the literature review. Section 3 presents the main results of our analysis: Subsection 3.1 summarizes the most influential papers and publication sources (conferences and journals); Subsection 1 identifies some hot topics; and Subsection 3.3 presents the most prolific countries and research institutions. Finally, Section 4 gives some concluding remarks.

2. Materials and methods

2.1. Document sample

In order to obtain a representative *sample* of the software engineering literature *population* for the last three years, our corpus is composed of the Scopus records for all the documents published from 2016 to 2018 in the following **journals**: (i) Automated Sw. Eng., (ii) IEEE Sw., (iii) IEEE Trans. on Sw. Eng., (iv) IEICE Trans. on Information and Systems, (v) IET Sw., (vi) Information and Sw. Tech., (vii) Innovations in Systems and Sw. Eng., (viii) Int. Journal of Sw. Eng. and Its Applications, (ix) Int. Journal of Sw. Eng. and Knowledge Eng., (x) Int. Journal on Adv. Science Eng. and Information Tech., (xi) Empirical Sw. Eng., (xii) Sw. and Systems Modeling, (xiii) Sw Practice and Experience, (xiv) Sw. Quality Journal, (xv) Theoretical Computer Sci.; and **conference proceedings**: (i) APSEC: Asia Pacific Sw. Eng. Conf. (APSEC) (ii) ASE: IEEE ACM Int. Conf. on Automated Sw. Eng., (iii) COMPSAC: Int. Computer Sw. and App. Conf., (iv) ESEM: Int. Symposium on Empirical Sw. Eng. and Measurement, (v) FSE: ACM SIGSOFT Symposium on the Foundations of Sw. Eng., (vi) ICODSE: Int. Conf. on Data and Sw. Eng., (vii) ICSE: Int. Conf. on Sw. Eng., (viii) ICSME: IEEE Int. Conf. on Sw. Maintenance and Evolution, (ix) ICST: IEEE Int. Conf. on Sw.. Testing Verification and Validation, (x) ISSRE: Int. Symposium on Sw. Reliability Int., (xii) JCSSE: Int. Joint Conf. on Computer Sci. and Sw. Eng., (xii) MOBILESoft: Int. Conf. on Mobile Sw. Eng. and Systems, (xiii) MODELSWARD: Int. Conf. on Model Driven Eng. and Sw. Dev., (xiv)MSR: IEEE Int. Working Conf. on Mining Sw. Repositories, (xv) QRS: IEEE Int. Conf. on Sw.e Quality Reliability and Security Companion, (xvi) RE: IEEE Int. Requirements Eng. Conf. (xvii) SEAA: EUROMICRO Conf. on Sw. Eng. and Adv. App.

The corpus is available at:

`https://github.com/rheradio/SwEngScopus2016-18`

2.2. Analysis tools

The document sample was analyzed using the following R packages:

- `bib2df` [6] to parse the Scopus data (in Bibtex format) to an R data frame.
- `tidyverse` [7] for processing the corpus.
- `wordcloud` [8] for identifying the most common article keywords (see Figure 1).
- `ggplot2` [9] for obtaining a visual representation of the countries whose research institutions are the most prolific (see Figure 2).

3. Results

The following sections summarize the main results of our analysis. In particular, Section 3.1 answers Research Questions RQ1 and RQ2 (what papers are the most impacting, and where have they been published?); Section 3.2 deals with RQ3 (what topics should be considered trendy?), and Section 3.3 answers RQ4 (which countries and institutions are playing the principal role?).

3.1. Hot Papers (RQ1 and RQ2)

Table 1 summarizes the papers that, according to the information provided by Scopus on July 23rd, 2019, were in the 99th citation percentile of their corresponding year. Most influential papers have been published in a few sources: the International Conference on Software Engineering - ICSE (16.67% of the papes in Table 1), and the journals Empirical Software Engineering (16.67%), Information and Software Technology (16.67%), IEEE Transactions on Software Engineering (15.15%), and IEEE Software (10.61%).

2016		2017		2018	
Article	**#Citations**	**Article**	**#Citations**	**Article**	**#Citations**
Mechtaev et al. [10]	111	*Tantithamthavorn et al.* [11]	67	*Morschheuser et al.* [12]	24
Gu et al. [13]	87	*Bröring et al.* [14]	61	*Dingsoyr et al.* [15]	24
Wang et al. [16]	84	*Xuan et al.* [17]	57	*Panichella et al.* [18]	21
Tantithamthavorn et al. [19]	80	*Beller et al.* [20]	55	*Falessi et al.* [21]	21
Sajnani et al. [22]	74	*Kitchenham et al.* [23]	50	*Leemans et al.* [24]	16
Stol et al. [25]	72	*Taivalsaari et al.* [26]	44	*Taibi et al.* [27]	15
White et al. [28]	71	*Groen et al.* [29]	40	*García et al.* [30]	14
Ye et al. [31]	71	*Munaiah et al.* [32]	35	*Kula et al.* [33]	14
Zhang et al. [34]	71	*Jongeling et al.* [35]	35	*Chen et al.* [36]	14
Xia et al. [37]	64	*Lu et al.* [38]	34	*Bennin et al.* [39]	13
Villarroel et al. [40]	64	*Le et al.* [41]	34	*Palomba et al.* [42]	13
Sorbo et al. [43]	63	*Jing et al.* [44]	33	*Calefato et al.* [45]	13
Segura et al. [46]	61	*Borges et al.* [47]	33	*Di Nucci et al.* [48]	13
Kosar et al. [49]	59	*Fernández et al.* [50]	31	*Tantithamthavorn et al.* [51]	12
LaToza et al. [52]	59	*Hoda et al.* [53]	30	*Szarnyas et al.* [54]	12
Alves et al. [55]	58	*Yang et al.* [56]	29	*Herbold et al.* [57]	12
Gousios et al. [58]	55	*Beller et al.* [59]	29	*Tong et al.* [60]	12
Hilton et al. [61]	53	*Ciurumelea et al.* [62]	29	*Hosseini et al.* [63]	12
Maalej et al. [64]	52	*Zhang et al.* [65]	29	*Hadar et al.* [66]	12
Treude et al. [67]	51	*Sadeghi et al.* [68]	28	*Hosseini et al.* [69]	12
-	-	-	-	*Hadar et al.* [70]	12
-	-	-	-	*Honghao et al.* [71]	11
-	-	-	-	*Amritanshu et al.* [72]	11
-	-	-	-	*Calefato et al.* [73]	11
-	-	-	-	*Vallon et al.* [74]	11
-	-	-	-	*Carvalho et al.* [75]	11

Table 1. Top 1% cited papers per year.

3.2. Hot Topics (RQ3)

The word-clouds in Figure 1 represent the most repeated *keywords* in the articles published in 2016, 2017, and 2018. According to the figure, three upward trends can be identified:

- Researching on open-source software: see the keywords “open-source software”, “open systems”, and “open-source projects”.
- The application of artificial intelligence techniques to software engineering: see the keywords “deep learning”, “artificial intelligence”, “leaning systems”, “learning algorithms”, “genetic algorithms”, “natural language processing”, and “classification of information”.
- Undertaking empirical software engineering: see the keywords “empirical studies”, “empirical software engineering”, “benchmarking”, and “mining software repositories”.

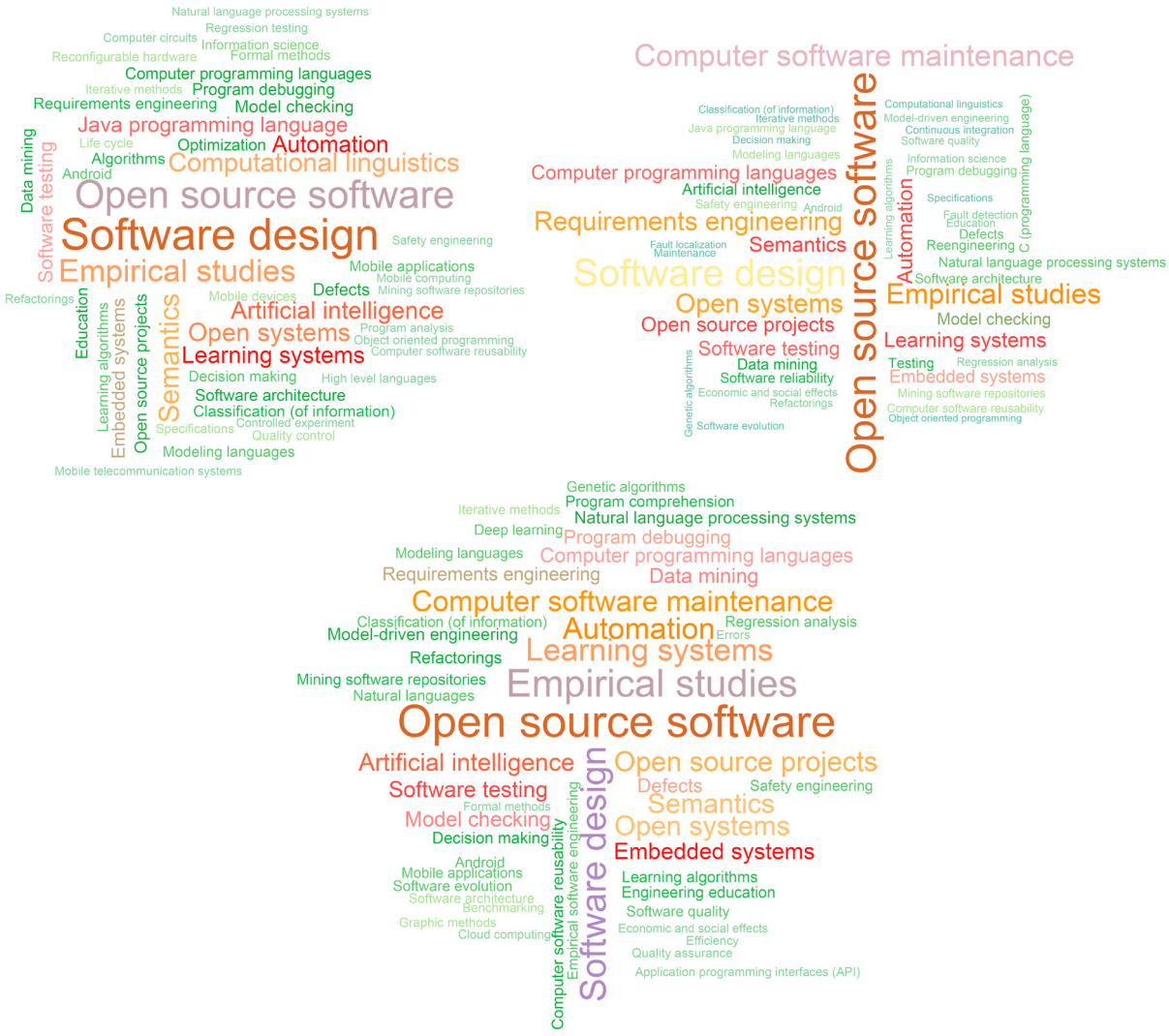

Fig. 1. Most repeated keywords in (a) 2016, (b) 2017, and (c) 2018.

3.3. Principal institutional actors (RQ4)

Figure 2 shows how many articles the research centers of every country have published for the last three years, summarizing on the right the top ten most prolific institutions per year. It is worth noting that the countries whose institutions published most papers were:

- In 2016, USA (23.96% of all the articles published in 2016), China (12.71%) and South Korea (11.25%).
- In 2017, USA (22.37%), China (14.61%) and Canada (9.58%).
- In 2018, USA (23.16%), China (14.38%) and Germany (11.03%).

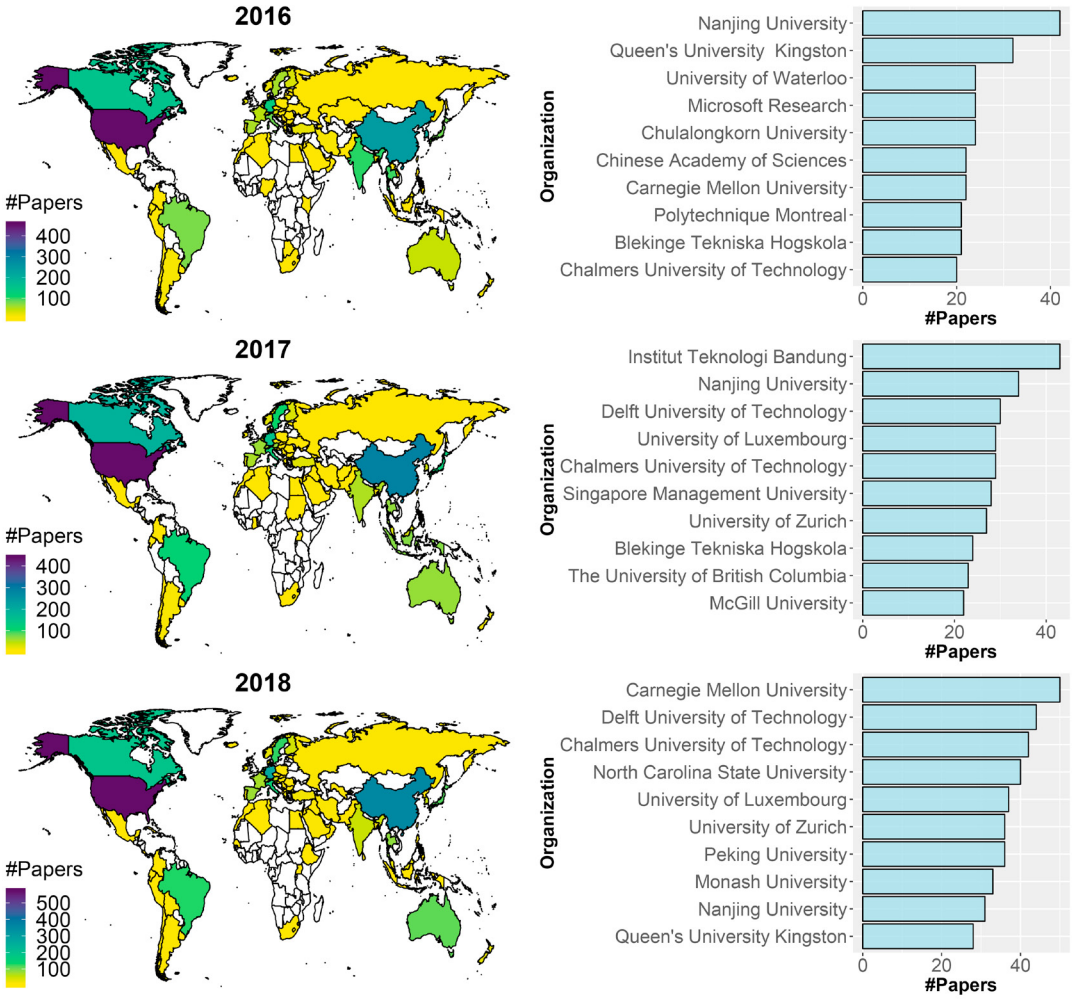

Fig. 2. Most prolific countries and organizations.

4. Conclusions

According to the document sample this paper analyzes, most literature on software engineering is produced in two countries: USA and China, with 23.16% and 13.9% of all published articles on average, respectively. The publication sources of the most influential papers are also rather concentrated: one conference and four journals have published 75.77% of the documents in the top 0.1% citation ranking. Finally, we have detected three hot topics: open-source software, the application of artificial intelligence to deal with software engineering problems, and the importance of performing empirical studies about software development.

Acknowledgements

This work has been funded by the Spanish Ministry of Science, Innovation, and Universities under Project DPI2016-77677-P, the Community of Madrid under Grant RoboCity2030-DIH-CM P2018/NMT-4331, and grant TIN2016-75850-R from the FEDER funds.

References

[1] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslen, Experimentation in Software Engineering, Springer-Verlag Berlin Heidelberg, 2012.
[2] V. Berg, J. Birkeland, A. Nguyen-Duc, I. O. Pappas, L. Jaccheri, Software startup engineering: A systematic mapping study, Journal of Systems and Software 144 (2018) 255 – 274.
[3] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Information and Software Technology 64 (Supplement C) (2015) 1 – 18.
[4] M. Souza, L. Veado, R. T. Moreira, E. Figueiredo, H. Costa, A systematic mapping study on game-related methods for software engineering education, Information and Software Technology 95 (2018) 201 – 218.
[5] R. Heradio, H. Perez-Morago, D. Fernandez-Amoros, F. J. Cabrerizo, E. Herrera-Viedma, A bibliometric analysis of 20 years of research on software product lines, Information and Software Technology 72 (2016) 1 – 15.
[6] P. Ottolinger, T. Leeper, M. Salmon, P. Egeler, E. X. Esposito, bib2df: Parse a bibtex file to a data frame (May 2019).
URL `https://CRAN.R-project.org/package=bib2df`
[7] G. Grolemund, H. Wickham, R for Data Science, 2016.
[8] I. Fellows, Word clouds (Aug. 2018).
URL `https://CRAN.R-project.org/package=wordcloud`
[9] H. Wickham, ggplot2: Elegant Graphics for Data Analysis, Springer; 2nd ed., 2016.
[10] S. Mechtaev, J. Yi, A. Roychoudhury, Angelix: Scalable multiline program patch synthesis via symbolic analysis, in: International Conference on Software Engineering, Austin, TX, USA, 2016.
[11] C. Tantithamthavorn, S. McIntosh, A. Hassan, K. Matsumoto, An empirical comparison of model validation techniques for defect prediction models, IEEE Transactions on Software Engineering 43 (1) (2017) 1–18, cited By 67. doi:10.1109/TSE.2016.2584050.
[12] B. Morschheuser, L. Hassan, K. Werder, J. Hamari, How to design gamification? a method for engineering gamified software, Information and Software Technology 95 (2018) 219–237.
[13] X. Gu, H. Zhang, D. Zhang, S. Kim, Deep api learning, in: ACM SIGSOFT International Symposium on Foundations of Software Engineering, Seattle, WA, USA, 2016.
[14] A. Bröring, S. Schmid, C.-K. Schindhelm, A. Khelil, S. Käbisch, D. Kramer, D. Le Phuoc, J. Mitic, D. Anicic, E. Teniente, Enabling IoT Ecosystems through Platform Interoperability, IEEE Software 34 (1) (2017) 54–61, cited By 61. doi:10.1109/MS.2017.2.
[15] T. Dingsoyr, N. Moe, T. Fagri, E. Seim, Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation, Empirical Software Engineering 23 (1) (2018) 490–520.
[16] S. Wang, T. Liu, L. Tan, Automatically learning semantic features for defect prediction, in: International Conference on Software Engineering, Austin, TX, USA, 2016.
[17] J. Xuan, M. Martinez, F. DeMarco, M. Clement, S. Marcote, T. Durieux, D. Le Berre, M. Monperrus, Nopol: Automatic repair of conditional statement bugs in java programs, IEEE Transactions on Software Engineering 43 (1) (2017) 34–55, cited By 57. doi:10.1109/TSE.2016.2560811.
[18] A. Panichella, F. Kifetew, P. Tonella, Automated test case generation as a many-objective optimisation problem with dynamic selection of the targets, IEEE Transactions on Software Engineering 44 (2) (2018) 122–158.
[19] C. Tantithamthavorn, S. McIntosh, A. Hassan, K. Matsumoto, Automated parameter optimization of classification techniques for defect prediction models, in: International Conference on Software Engineering, Austin, TX, USA, 2016.
[20] M. Beller, G. Gousios, A. Zaidman, Travistorrent: Synthesizing travis ci and github for full-stack research on continuous integration, in: IEEE International Working Conference on Mining Software Repositories, Buenos Aires, Argentina, 2017.
[21] D. Falessi, N. Juristo, C. Wohlin, B. Turhan, J. Münch, A. Jedlitschka, M. Oivo, Empirical software engineering experts on the use of students and professionals in experiments, Empirical Software Engineering 23 (1) (2018) 452–489.
[22] H. Sajnani, V. Saini, J. Svajlenko, C. Roy, C. Lopes, SourcererCC: Scaling code clone detection to big-code, in: International Conference on Software Engineering, Austin, TX, USA, 2016.
[23] B. Kitchenham, L. Madeyski, D. Budgen, J. Keung, P. Brereton, S. Charters, S. Gibbs, A. Pohthong, Robust statistical methods for empirical software engineering, Empirical Software Engineering 22 (2) (2017) 579–630, cited By 50. doi:10.1007/s10664-016-9437-5.
[24] S. Leemans, D. Fahland, W. van der Aalst, Scalable process discovery and conformance checking, Software and Systems Modeling 17 (2) (2018) 599–631.
[25] K.-J. Stol, P. Ralph, B. Fitzgerald, Grounded theory in software engineering research: A critical review and guidelines, in: International Conference on Software Engineering, Austin, TX, USA, 2016.
[26] A. Taivalsaari, T. Mikkonen, A roadmap to the programmable world: Software challenges in the iot era, IEEE Software 34 (1) (2017) 72–80, cited By 44.
[27] D. Taibi, V. Lenarduzzi, On the definition of microservice bad smells, IEEE Software 35 (3) (2018) 56–62.
[28] M. White, M. Tufano, C. Vendome, D. Poshyvanyk, Deep learning code fragments for code clone detection, in: IEEE/ACM International Conference on Automated Software Engineering, Singapore, 2016.
[29] E. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Doerr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini, M. Stade, The crowd in requirements engineering: The landscape and challenges, IEEE Software 34 (2) (2017) 44–52, cited By 40. doi:10.1109/MS.2017.33.

[30] L. García-Bañuelos, N. Van Beest, M. Dumas, M. La Rosa, W. Mertens, Complete and interpretable conformance checking of business processes, IEEE Transactions on Software Engineering 44 (3) (2018) 262–290.
[31] X. Ye, H. Shen, X. Ma, R. Bunescu, C. Liu, From word embeddings to document similarities for improved information retrieval in software engineering, in: International Conference on Software Engineering, Austin, TX, USA, 2016.
[32] N. Munaiah, S. Kroh, C. Cabrey, M. Nagappan, Curating github for engineered software projects, Empirical Software Engineering 22 (6) (2017) 3219–3253, cited By 35. doi:10.1007/s10664-017-9512-6.
[33] R. Kula, D. German, A. Ouni, T. Ishio, K. Inoue, Do developers update their library dependencies?: An empirical study on the impact of security advisories on library migration, Empirical Software Engineering 23 (1) (2018) 384–417.
[34] F. Zhang, Q. Zheng, Y. Zou, A. Hassan, Cross-project defect prediction using a connectivity-based unsupervised classifier, in: International Conference on Software Engineering, Austin, TX, USA, 2016.
[35] R. Jongeling, P. Sarkar, S. Datta, A. Serebrenik, On negative results when using sentiment analysis tools for software engineering research, Empirical Software Engineering 22 (5) (2017) 2543–2584, cited By 35. doi:10.1007/s10664-016-9493-x.
[36] D. Chen, Y. Jiang, C. Xu, X. Ma, J. Lu, Testing multithreaded programs via thread speed control, in: Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Lake Buena Vista, FL, USA, 2018.
[37] X. Xia, D. Lo, S. Pan, N. Nagappan, X. Wang, Hydra: Massively compositional model for cross-project defect prediction, IEEE Transactions on Software Engineering 42 (10) (2016) 977–998.
[38] Q. Lu, X. Xu, Adaptable blockchain-based systems: A case study for product traceability, IEEE Software 34 (6) (2017) 21–27, cited By 34.
[39] K. Bennin, J. Keung, P. Phannachitta, A. Monden, S. Mensah, Mahakil: Diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction, IEEE Transactions on Software Engineering 44 (6) (2018) 534–550.
[40] L. Villarroel, G. Bavota, B. Russo, R. Oliveto, M. Di Penta, Release planning of mobile apps based on user reviews, in: International Conference on Software Engineering, Austin, TX, USA, 2016.
[41] X.-B. Le, D.-H. Chu, D. Lo, C. Le Goues, W. Visser, S3: Syntax- and semantic-guided repair synthesis via programming by examples, in: Joint Meeting on Foundations of Software Engineering, Paderborn, Germany, 2017.
[42] F. Palomba, G. Bavota, M. Penta, F. Fasano, R. Oliveto, A. Lucia, On the diffuseness and the impact on maintainability of code smells: a large scale empirical investigation, Empirical Software Engineering 23 (3) (2018) 1188–1221.
[43] A. Sorbo, S. Panichella, C. Alexandru, J. Shimagaki, C. Visaggio, G. Canfora, H. Gall, What would users change in my app? summarizing app reviews for recommending software changes, in: ACM SIGSOFT Symposium on the Foundations of Software Engineering, Seattle, WA, USA, 2016.
[44] X.-Y. Jing, F. Wu, X. Dong, B. Xu, An improved sda based defect prediction framework for both within-project and cross-project class-imbalance problems, IEEE Transactions on Software Engineering 43 (4) (2017) 321–339.
[45] F. Calefato, F. Lanubile, N. Novielli, How to ask for technical help? evidence-based guidelines for writing questions on stack overflow, Information and Software Technology 94 (2018) 186–207.
[46] S. Segura, G. Fraser, A. Sanchez, A. Ruiz-Cortes, A survey on metamorphic testing, IEEE Transactions on Software Engineering 42 (9) (2016) 805–824.
[47] H. Borges, A. Hora, M. Valente, Understanding the factors that impact the popularity of github repositories, in: IEEE International Conference on Software Maintenance and Evolution, Raleigh, NC, USA, 2017.
[48] D. Di Nucci, F. Palomba, G. De Rosa, G. Bavota, R. Oliveto, A. De Lucia, A developer centered bug prediction model, IEEE Transactions on Software Engineering 44 (1) (2018) 5–24.
[49] T. Kosar, S. Bohra, M. Mernik, Domain-specific languages: A systematic mapping study, Information and Software Technology 71 (2016) 77–91.
[50] D. Fernández, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetrò, T. Conte, M.-T. Christiansson, D. Greer, C. Lassenius, T. Männistö, M. Nayabi, M. Oivo, B. Penzenstadler, D. Pfahl, R. Prikladnicki, G. Ruhe, A. Schekelmann, S. Sen, R. Spinola, A. Tuzcu, J. de la Vara, R. Wieringa, Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice, Empirical Software Engineering 22 (5) (2017) 2298–2338.
[51] C. Tantithamthavorn, A. Hassan, An experience report on defect modelling in practice: Pitfalls and challenges, in: International Conference on Software Engineering, Gothenburg, Sweden, 2018.
[52] T. LaToza, A. Van Der Hoek, Crowdsourcing in software engineering: Models, motivations, and challenges, IEEE Software 33 (1) (2016) 74–80.
[53] R. Hoda, N. Salleh, J. Grundy, H. Tee, Systematic literature reviews in agile software development: A tertiary study, Information and Software Technology 85 (2017) 60–70.
[54] G. Szarnyas, B. Izso, I. Rath, D. Varro, The train benchmark: cross-technology performance evaluation of continuous model queries, Software and Systems Modeling 17 (4) (2018) 1365–1393.
[55] N. Alves, T. Mendes, M. De Mendonça, R. Spinola, F. Shull, C. Seaman, Identification and management of technical debt: A systematic mapping study, Information and Software Technology 70 (2016) 100–121.
[56] X. Yang, D. Lo, X. Xia, J. Sun, TLEL: A two-layer ensemble learning approach for just-in-time defect prediction, Information and Software Technology 87 (2017) 206–220.
[57] S. Herbold, A. Trautsch, J. Grabowski, A comparative study to benchmark cross-project defect prediction approaches, IEEE Transactions on Software Engineering 44 (9) (2018) 811–833.
[58] G. Gousios, M.-A. Storey, A. Bacchelli, Work practices and challenges in pull-based development: The contributor's perspective, in: International Conference on Software Engineering, Austin, TX, USA, 2016.
[59] M. Beller, G. Gousios, A. Zaidman, Oops, My Tests Broke the Build: An Explorative Analysis of Travis CI with GitHub, in: International Conference on Mining Software Repositories, Buenos Aires, Argentina, 2017.
[60] H. Tong, B. Liu, S. Wang, Software defect prediction using stacked denoising autoencoders and two-stage ensemble learning, Information and Software Technology 96 (2018) 94–111.

[61] M. Hilton, T. Tunnell, K. Huang, D. Marinov, D. Dig, Usage, costs, and benefits of continuous integration in open-source projects, in: IEEE/ACM International Conference on Automated Software Engineerin, Singapore, 2016, pp. 426–437.
[62] A. Ciurumelea, A. Schaufelbuhl, S. Panichella, H. Gall, Analyzing reviews and code of mobile apps for better release planning, in: International Conference on Software Analysis, Evolution and Reengineering, Klagenfurt, Austria, 2017.
[63] S. Hosseini, B. Turhan, M. Mäntylä, A benchmark study on the effectiveness of search-based data selection and feature selection for cross project defect prediction, Information and Software Technology 95 (2018) 296–312.
[64] W. Maalej, M. Nayebi, T. Johann, G. Ruhe, Toward data-driven requirements engineering, IEEE Software 33 (1) (2016) 48–54.
[65] Z.-W. Zhang, X.-Y. Jing, T.-J. Wang, Label propagation based semi-supervised learning for software defect prediction, Automated Software Engineering 24 (1) (2017) 47–69.
[66] I. Hadar, T. Hasson, O. Ayalon, E. Toch, M. Birnhack, S. Sherman, A. Balissa, Privacy by designers: software developers privacy mindset, Empirical Software Engineering 23 (1) (2018) 259–289.
[67] C. Treude, M. Robillard, Augmenting api documentation with insights from stack overflow, in: International Conference on Software Engineering, Austin, TX, USA, 2016.
[68] A. Sadeghi, H. Bagheri, J. Garcia, S. Malek, A taxonomy and qualitative comparison of program analysis techniques for security assessment of android software, IEEE Transactions on Software Engineering 43 (6) (2017) 492–530.
[69] S. Hosseini, B. Turhan, M. Mäntylä, A benchmark study on the effectiveness of search-based data selection and feature selection for cross project defect prediction, Information and Software Technology 95 (2018) 296 – 312.
[70] I. Hadar, T. Hasson, O. Ayalon, E. Toch, M. Birnhack, S. Sherman, A. Balissa, Privacy by designers: software developers' privacy mindset, Empirical Software Engineering 23 (1) (2018) 259–289.
[71] H. Gao, H. Miao, L. Liu, J. Kai, K. Zhao, Automated quantitative verification for service-based system design: A visualization transform tool perspective, International Journal of Software Engineering and Knowledge Engineering 28 (10) (2018) 1369–1397.
[72] A. Agrawal, W. Fu, T. Menzies, What is wrong with topic modeling? and how to fix it using search-based software engineering, Information and Software Technology 98 (2018) 74 – 88.
[73] F. Calefato, F. Lanubile, F. Maiorano, N. Novielli, Sentiment polarity detection for software development, Empirical Software Engineering 23 (3) (2018) 1352–1382.
[74] R. Vallon, B. J. da Silva Estácio, R. Prikladnicki, T. Grechenig, Systematic literature review on agile practices in global software development, Information and Software Technology 96 (2018) 161 – 180.
[75] V. A. Carvalho, J. P. A. Almeida, Toward a well-founded theory for multi-level conceptual modeling, Software & Systems Modeling 17 (1) (2018) 205–231.