A Systematic Mapping of Software Engineering Approaches to Develop Big Data Systems

Rodrigo Nunes Laigner, Marcos Kalinowski, Sérgio Lifschitz
Informatics Department
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro, Brazil
{laigner, kalinowski, sergio}@inf.puc-rio.br

Rodrigo Salvador Monteiro, Daniel de Oliveira
Computing Institute
Fluminense Federal University (UFF)
Niterói, Brazil
{salvador, danielcmo}@ic.uff.br

***Abstract*—[Context] Data is being collected at an unprecedented scale. Data sets are becoming so large and complex that traditionally engineered systems may be inadequate to deal with them. While software engineering comprises a large set of approaches to support engineering robust software systems, there is no comprehensive overview of approaches that have been proposed and/or applied in the context of engineering big data systems. [Goal] This study aims at surveying existing research on big data software engineering to unveil and characterize the development approaches and major contributions. [Method] We conducted a systematic mapping study, identifying 52 related research papers, dated from 2011 to 2016. We classified and analyzed the identified approaches, their objectives, application domains, development lifecycle phase, and type of contribution. [Results] As a result, we outline the current state of the art and gaps on employing software engineering approaches to develop big data systems. For instance, we observed that the major challenges are in the area of software architecture and that more experimentation is needed to assess the classified approaches. [Conclusion] The results of this systematic mapping provide an overview on existing approaches to support building big data systems and helps to steer future research based on the identified gaps.**

Keywords—data intensive systems, big data systems, software engineering, systematic mapping

I. INTRODUCTION

In the early 2000s, due to the impact of e-commerce, at the time an innovative model of transactions performed through the world wide web, challenges on data management were identified by the industry. Typically, organizations dealt with the increase in data volume with the acquisition of more storage space. This process proved to be inefficient when it came to integrate databases from different systems, often causing inconsistent databases through the operation [48]. Laney [48] discussed appropriate initiatives for handling data management challenges in large corporations and introduces the concept of "3Vs": Volume, Velocity and Variety. According to Laney [48], volume is about the amplitude and depth of the data available in a transaction or any point of interaction. Velocity is identified as the speed employed in the usage of data to support interactions and the way these generate data. Finally, variety is based on the degree of compatibility between varied data formats, an important factor for effective data management.

The concept "3Vs" is known as the predecessor of big data. According to TechAmerica Foundation [71], the term big data describes large data volume, which, given its complex and variable characteristics, require advanced techniques and technologies to enable capturing, storing and analyzing information. In addition, according to the Big Data Software Engineering (BDSE) workshop [12] big data systems concern the process of extracting high-value information in order to revolutionize decision-making in business, science, and society.

The term big data has gained particular interest since 2011, when IBM has positioned itself as a provider of big data solutions. Since then, researchers and organizations, such as SAS [65], have introduced other dimensions to the three original ones. For instance, Value and Veracity, as proposed by Kazman [41]. Burbank [15] defines value as the ability to comprehend, manage, and integrate from distinct sources in order to get previously unknown information. In addition, Burbank [15] also argues that veracity is the level of exactness and accuracy of the information.

According to a survey by Capgemini in conjunction with Informatica [17], two challenges to turn big data an effective business resource are a lack of technical experience and difficulty in data integration. Another survey carried out by Capgemini [16] verified that the adoption of a systematic implementation approach is a factor of success in big data initiatives. The report also indicates that only about 70% of organizations have a well-defined process to identify and select appropriate technologies.

In the context of developing systems geared to process large datasets, according to Chen *et al*. [25], some risks are the difficulty in selecting big data technologies, the complex integration between legacy systems with new systems and, as a new field, practitioners have little or no knowledge. In another study [24], the same authors argue that the development of systems that deal with data on a smaller scale is traditionally based on relational databases or data warehouses and explain the importance of an specific architectural design process for designing big data systems. In addition, Gorton and Klein [36] argue that big data systems must be able to sustain high write rates, varying loads and types of requests, and high availability. Although such studies represent a step forward, it is hard to the reader to be aware of the “Big Picture” of software engineering approaches that support big data systems.

The goal of this paper is to present a systematic mapping of the literature conducted to answer the following research question: "Which software engineering approaches have been proposed to support developing big data systems?". We analyze the approaches, their objectives, application domains in which they have been applied, the life cycle phases they support, and what kind of contribution they provide. The types of studies that have been undertaken are also analyzed.

Based on the mapping, it was possible to observe that the main challenges are in the area of software architecture, where the construction of an adequate infrastructure, involving data modeling, application design, integration between different database technologies and applications, often arise as barriers to the development of asolution to deal with intense data processing. In addition, there is a major focus of studies on the architecture and design lifecycle phase. Moreover, it is possible to observe that more experimentation is needed to assess the identified approaches to better understand the situations in which they really work and their limitations.

The remainder of this paper is organized as follows. Section II describes the background and related work concerning secondary studies on big data field. Section III describes the systematic mapping protocol and how the mapping was conducted. Section IV presents the results to our research questions. Finally, Section V concludes the paper.

II. BACKGROUND AND RELATED WORK

A. Big Data Systems - Terminology

Sources of information where there is no peer review of content such as blogs and articles use to position ultra-large-scale systems as big data systems. Although ultra-large-scale systems tend to have a large amount of data stored, accessed, and manipulated, the dimension of extracting Value for decision making may often not be present. Another example of wrong positioning is when referring to big data systems as "big data analytics", which according to Russom [63], is described as the application of advanced analytic techniques to large data sets. Big data systems may support specific analytics techniques to provide Value, but do not replace the whole analysis process. Finally, big data systems are also often referred to as data-intensive systems in the literature [6]. The word intensive concerns the development of enterprise applications that can make use of big data technologies, such as Hadoop, Apache Spark, and NoSQL systems that store and process data in large scale. It is important to highlight that technologies such as Spark and Hadoop are not designed for a specific domain or purpose. They are general purpose frameworks to provide parallel capabilities and thus are not considered in this paper as Big Data Systems.

B. Big Data Systems and Software Engineering

Big data can be identified as the process of acquisition and storage of large datasets in order to support data analysis and provide knowledge for the decision-making process. In addition, this work focuses specifically on the software engineering dimension for big data, that is, analyzing approaches in the field of software engineering that enable or support the development of big data systems. This way, big data systems are any software solution that undertakes the collection, storage and processing of a large volume of data. Lastly, according to Gandomi and Haider [33], size dimensions in big data refer to multiple terabytes and petabytes of data, so this work also makes use of this pattern when it refers to large volume of data or intensive data processing. Therefore, big data, in the context of this work, is intrinsically linked to the development of software systems.

C. Related Work

O'Donovan *et al.* [57] introduced the first secondary study in the big data field, intending to provide an overview of the studies about big data in manufacturing. The research questions covered the type of analytics and technologies employed on big data and applicability on manufacturing areas.

Akoka *et al.* [2] undertook a mapping study on big data research aiming to investigate publication trends, including the study objectives, application domains, most active authors and the hot topics in the study area. Alayyoub et al. [3] analyzed studies about stream processing frameworks for big data, providing the classification and analysis of these studies. Recently, Ortega et al. [58] searched for methods to evaluate database management systems focusing on quality criteria.

Finally, Kumar and Alencar [28] conducted a literature survey focusing on the software development life cycle phases in the context of projects that have the potential to utilize big data technologies. The authors observed the most popular application domains and Software Development Lifecycle Phases (SDLCP) that concentrate research efforts. However, their study does not focus on software engineering approaches. Therefore it does not providing a comprehensive overview on this topic. The mapping study described hereafter addresses this gap, focusing specifically on software engineering approaches with strict inclusion criteria and a classification scheme designed based on this focus.

III. SYSTEMATIC MAPPING

A. Research Questions

A primary research question was defined: *"(RQ1) Which types of software engineering approaches have been proposed to support developing big data systems?*". Based on this primary question, complementary research questions were defined to characterize the identified approaches.

- *RQ1a. Which are the objectives of the studies presenting the approaches?*
- *RQ1b. Which type of research has been conducted on the approaches?*
- *RQ1c. Which type of empirical evaluation has been performed on the approaches?*
- *RQ1d. Which application domains have the proposed approaches been applied to?*
- *RQ1e. Which SDLCP the approaches relate to?*
- *RQ1f. Which type of contributions do the approaches represent?*

- *RQ1g. What is the level of collaboration between industry and academia on proposing the approaches?*

B. Search Strategy

First, a set of control papers was selected to provide input into the search for primary studies [25][36][9]. According to Kitchenham and Charters [43], control papers aim at helping in the definition of the search string, providing input for the adjustment of the search string until the search retrieves the control papers.

Besides the control papers, an initial set of 10 studies was identified through exploratory searches in the Scopus database. This set was used to obtain an overview of the area, main challenges, classifications, terms and keywords before starting the more formal and complete mapping study.

Reading the control papers also allowed further understanding of the main terms used in the area and basic knowledge concerning some important studies.

After some analyses of possible search strings, the following one was selected: "big data" AND "software engineering". Applying this search string to titles and abstracts in the chosen digital libraries enabled retrieving the control papers. Indeed, we believe that it appropriately reflects our purpose, given that we were looking for software engineering approaches (intervention) within the population of papers concerning big data systems (population). Nevertheless, besides conducting the search on several digital libraries our search strategy also involved applying backward and forward snowballing on the initially included papers (seed set) to identify additional relevant papers that, for any reason, did not match our search string [74][53].

Data sources were selected based on Dyba et al. [29], which recommends the following sources for software engineering area: Scopus, IEEEXplore, ACM, ScienceDirect, EI Compendex, and Web of Science. The search on digital libraries and the snowballing were carried out in January 2017 and this study contemplates primary studies published until the end of 2016.

C. Study Selection

According to Petersen et al. [42], only studies that are relevant to answering the research questions should be included. Therefore, this research excluded studies that present concepts and proposals outside the scope of research questions or outside the domain of software engineering. The inclusion and exclusion criteria is presented in Table I.

D. Selection Process

The identification and filtering of the articles was divided into 5 steps, using a combination of searches in digital libraries and forward and backward snowballing.

The first step consisted performing the search in each of the selected digital libraries and removing duplicates. From 441 articles obtained by the search, after the removal of duplicate articles, 305 were left to be analyzed.

In the second step, filtering is applied based on criteria E2, E3, E4. The objective of applying this criteria was to avoid redundancy in research contributions and to establish a quality level in the included publications.

TABLE I. INCLUSION AND EXCLUSION CRITERIA.

Criteria	Description
IC1	Published studies describing approaches or strategies for developing big data systems. If the study only mentions a technique and does not provide a detailed explanation of its applicability and context, then it should not be included.
EC1	Papers where the focus on software engineering is not identified.
EC2	Papers not written in English.
EC3	Grey literature, including white papers, theses, and papers that were not peer reviewed.
EC4	Papers that are only available in the form of abstracts/posters and presentations.

In the third step a second filter is applied verifying title, abstract and keywords against criteria I1 and E1. The whole filtering process conducted in this step was peer reviewed. If the inclusion of a study gave rise to doubt, a third researcher analyzed the study in order to reach consensus. Whenever the decision was not possible, the study proceeded to the next step, where the full text was read. After this step 55 candidate studies remained.

In the fourth step, the third and final filter was applied. This time, the full text of the remaining unclassified articles was read. At the end of this stage, another 12 articles were excluded, resulting in 43 included studies.

The fifth step concerned applying snowballing using these included studies as seed set. According to Wohlin [74], snowballing is an iterative process where the references (backward snowballing) and citing studies (forward snowballing) of a seed set of studies are checked in order to identify other relevant studies. To enable forward snowballing the Scopus feature of showing citing studies was used for each selected study. During this process 9 additional papers (6 by backward snowballing, and 3 by forward snowballing) that met the inclusion criteria were identified. At the end, the articles selected by the snowballing process were combined with the initially selected articles, totaling 52 included primary studies.

E. Classification Scheme

The information extracted from each of the 52 selected papers and the corresponding classification scheme are described in Table II.

IV. RESULTS

This section presents the results of the mapping study, based on the analysis of the information extracted from the 52 selected papers, organized by research question.

RQ1. Which types of software engineering approaches have been proposed to support developing big data systems?

Table III lists all the types of approaches identified by analyzing the primary studies. Out of the 52 selected studies in this mapping, the majority of papers (10) refer to *development methodology proposal*, indicating efforts on identifying barriers and gaps on existing development

methodologies in the context of big data systems. Next, there is also a high concentration of approaches (9) concerning software architecture. This behavior can be explained by the need of non-conventional architectural solutions for big data systems when compared with traditional systems [22]. Finally, there are 8 approaches concerning *software solutions* (e.g., verification tool to evaluate design [11]) to solve particular problems within the scope of building big data systems.

TABLE II. DATA EXTRACTION FORM.

Information	Description
Approach Type (RQ1)	Type of technique or strategy for developing big data systems. We undertook a classification about type of contribution by using open coding [70].
Objective (RQ1a)	Study objective. Again, open coding was used [70].
Type of Research (RQ1b)	Classification of research type, according to Wieringa et al. [63], including the following categories: *evaluation research, proposal of solution, validation research, philosophical paper, opinion paper,* or *experience paper.*
Empirical Evaluation (RQ1c)	Classification of the empirical strategy, according to Wohlin et al. [18], including the following categories: *experiment*, *case study*, or *survey.*
Application Domain (RQ1d)	Application domain on which the approach was applied. According to Evans [24], application domains can influence constraints such as risk and quality, depending on the complexity of the application. Only application domains in which empirical studies were applied (e.g. case study, survey or experiment) are considered in this category.
SDLCP (RQ1e)	SDLCP, comprising the following categories: requirements, architecture and design, implementation, testing, deploy, and maintenance. It is noteworthy that to enable classifying the approaches independently of its specific development methodology, we defined generic categories using a high-level of abstraction.
Contribution Type (RQ1f)	Contribution the study provided. We undertook a classification about type of contribution by using coding [70] based on the categories defined by O'Donovan et al. [59].
Type of Author (RQ1g)	Industry, Academia or both. Studies can be conducted jointly by researchers and industry professionals or conducted in an individual setting without partnership. The objective of this category is to characterize this degree of intersection.
Study Metadata	Includes the paper title and information on the authors, venue, and year of publication.

RQ1a. Which are the objectives of the studies presenting the approaches?

Table IV lists all the study objectives extracted from the primary studies. It is possible to observe a concentration of studies on systems design and development methodology, indicating again a trend on improving methodologies for big data systems development.

TABLE III. TYPES OF APPROACHES FOR DEVELOPMENT OF BIG DATA SYSTEMS

Approach Type	Description	Reference
Development methodology proposal	Introduction of changes/adaptations in a development methodology or proposal of life cycle model	[10][23] [24][25] [26][37] [39][45] [52][60]
Software architecture proposal	Architecture definition proposal or methodology for architecture definition	[5][13][19] [46][47] [54][55] [72][77]
Software solution	Software systems aiming to support/improve processes on system development lifecycle	[8] [11][14] [21][44] [50] [64] [76]
Systems design method proposal	Solution proposal related to the design of software system	[7][22][27] [34][38] [67][69]
Software architecture design method proposal	Solution proposal related to the design of software architecture	[4][36][49] [59]
Experience sharing	Report of knowledge acquired	[6][68]
Requirements engineering approach proposal	Methodologies or strategies to address requirements	[31][56]
Modeling language extension proposal	Inclusion of visual resources in a modeling language	[35][40]
Performance analysis model proposal	Mathematical models to identify performance measures	[1][30]
Problem mapping	Research of latent problems on industry or academia	[51][61]
IDE support tool development	Plugins development to support development on IDE	[20]
Development team monitoring	Monitoring of development team in order to collect adopted practices	[62]
Data migration method proposal	Development of method or guideline to enable data migration	[66]

TABLE IV. STUDY OBJECTIVES

Objective	Count
Provide a solution to systems design	12
Improve or define a development methodology	10
Provide a solution to large data set processing	8
Provide a solution to support architecture design	5
Evolve software architecture body of knowledge	5
Support application execution/development	4
Provide a solution to address requirements	3
Identify gaps and opportunities	3
Support deploy process	1
Support data logging process	1

RQ1b. Which type of research has been conducted on the approaches?

Fig. 1 presents the distribution of research types among the selected primary studies. As depicted, *proposal of solution* leads the number of studies (20), followed by philosophical paper with 14 studies. It is possible to observe that *evaluation research*, i.e., research concerning empirical evaluations, is presented in only 9 studies.

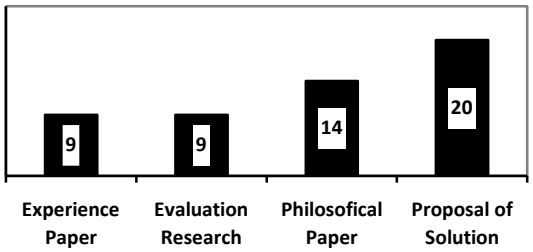

Fig. 1. Type of Research Distribution over Primary Studies

RQ1c. Which type of empirical evaluation has been performed on the approaches?

Fig. 2 depicts the number of studies distributed by empirical evaluation. Almost half of studies did not use an empirical evaluation to validate the introduced approach. In addition, case studies show up as the major used empirical evaluation strategy. This conjecture indicates that more experimentation is needed in the field in order to better assess the proposed approaches.

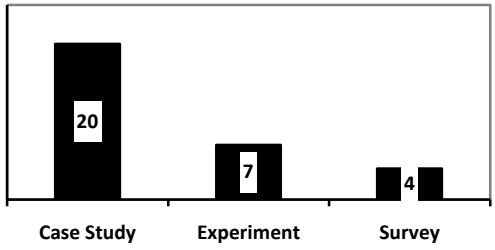

Fig. 2. Empirical Study Distribution over Primary Studies

RQ1d. Which application domains have the proposed approaches been applied to?

Table V exhibits the distribution of primary studies by application domain. Social Networks, Network and E-Commerce are the application domains with most studies. A possible explanation for this conjecture is the typical characteristic of applications in these domains: large datasets, data generation and data processing in high scale.

RQ1e. Which SDLCP the approaches relate to?

The SDLCP to which the identified approaches relate are shown in Table VI. It is possible to identify a great number of approaches on the architecture and design phase. There are several (14) approaches that do not relate to a specific SDLCP (e.g., approaches that deal with methodologies for developing big data applications [10]).

RQ1f. Which type of contributions do the approaches represent?

The results in Table VII show a large number of studies (20) regarding *methodology*. It can be explained by studies proposing development and design methodologies. Next, tool and architecture are categories with 8 studies each. Again, the number of studies on software solutions and software architecture explains this conjecture.

TABLE V. APPLICATION DOMAIN CLASSIFICATION

Application Domain	Count
Social Networks	5
Network Monitoring/Security	4
E-Commerce	3
Bioinformatics	2
Healthcare	2
Electrical Sector	1
Geospatial Data Processing	1
Internet of Things	1
Telecommunications	1
Public Sector	1
Cyber-physical Systems	1

TABLE VI. SDLCP DISTRIBUTION OVER APPROACHES

Life Cycle Phase	Count
Architecture and Design	23
Implementation	5
Requirements	4
Maintenance	3
Deploy	2
Testing	1

TABLE VII. TYPE OF CONTRIBUTION

Classification	Description	Count
Methodology	Study that presents an approach or method to solve a problem	20
Architecture	Study that describes theory view or implementation choices for software architecture	8
Tool	Study that aims the development software tool to address a problem	8
Process	Study that present a process or a set of processes to solve a problem	6
Theory	Study that provide guidelines to solve a given problem	4
Platform	Study that provides a system to support execution of applications	3
Framework	Study that describes a proposal or library development of software to solve a problem	2
Model	Study that proposes a mathematical model to solve a problem	1

RQ1g. What is the level of collaboration between industry and academia on proposing the approaches?

Fig. 3 depicts the distribution of studies by type of authors. It is observed that the majority of studies are from academic authors. Hence, while the topic is highly relevant to industry, apparently most research is being conducted in isolated academic initiatives.

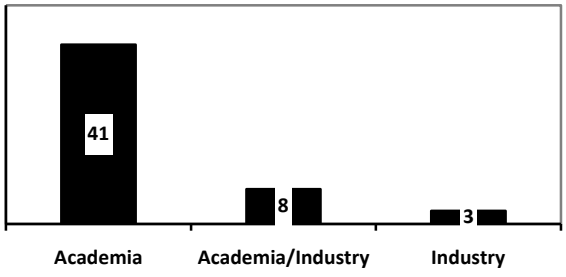

Fig. 3. Type of Author of Primary Studies

V. DISCUSSION

A. Systhesis of Results

Based on the results, it is possible to observe that most research efforts on software engineering approaches are concentrated on development methodologies and software architecture (*cf.* Table III). On behalf of development methodologies, efforts focus on addressing particularities of big data systems, such as their data volume and complexity (e.g., an agile process adaptation employing an architecture-centric approach [23]). Consequently, there seems to be an understanding, from industry and academia, that traditional software development methodologies do not support all the needs to develop big data systems.

Regarding software architecture, the building of an infrastructure, involving data modeling, application design, integration among different database technologies and applications, can imply on barriers to the development of an adequate solution. This way, there is a significant number of studies concerned with defining, prototyping, and implementing software architectural assets. In this context, there are also studies where the focus remains methodologies specifically addressing the definition of software architectures for big data system.

When analyzing the SDLCP of the identified approaches, we observed a large number of studies focusing on architecture and design. It means that there is a high interest from researchers in providing solutions to this specific phase. This context is leveraged by the high number of studies focused on solving problems related to definition and implementation of software architecture in data intensive systems.

Regarding the type of research, the identified papers mainly concern philosophical papers and solution proposals. Further evaluation research is needed in order to better understand the situations in which the identified approaches really work, their limitations and how they can be evolved. However, it is noteworthy that evaluation research has evolved over the years and that some recent papers concern this type of research (e.g., [21]).

Finally, it is noteworthy that the publication landscape in the area is composed of recent papers (2011 to 2016) and, although our mapping study provides the overall distribution of the identified approaches (e.g., types of approaches and contributions), the short publication period did not allow us to observe significant trends concerning the evolution over the years.

B. Threats to Validity

The results of this systematic mapping can potentially be affected by the scope of the study search, study selection process, and bias on synthesis of results. This section address the efforts employed to mitigate threats to validity.

Internal validity: We followed the guidelines provided by Kitchenham and Charters [43]. Additionally, the mapping protocol was discussed and validated among the researchers. In addition, the precise inclusion and exclusion criteria favored a strict selection of primary studies. Finally, in order to mitigate the risk of bias on selection and categorization studies, the filtering and categorization steps (including coding) were peer reviewed. The data including intermediate and auditable results is available online[1].

External validity: We have invested effort to produce complete and valid results, using only peer reviewed sources. Therefore, we believe that some degree of external validity has been achieved. Moreover, our detailed mapping study protocol and the extracted data are auditable and available for replications to validate and reinforce our results.

Reliability: When searching for primary studies we used all the databases recommended by Dyba et al. [29] and complemented the results by applying forward and backward snowballing. One minor limitation could concern the usage of Scopus for forward snowballing. Nevertheless, this limitation concerns a specific part of the search process and Scopus is well-known for indexing the most important software engineering conferences and journals. Furthermore, our mapping considered only peer-reviewed studies published until the end of 2016.

VI. CONCLUDING REMARKS

In this paper we presented a mapping study aiming at providing an overview on existing software engineering approaches to support building big data systems. We defined a precise mapping protocol and applied it, allowing us to identify 52 papers presenting such approaches. Thereafter we classified them according to their type of approach, objective, type of contribution, application domain and SDLCP.

The results provide an overview on existing software engineering approaches to support the development of big data systems. The classification scheme enables identifying the categories that concentrate most of the research and related gaps. Such information is particularly valuable to enable researchers to ground future research based on existing efforts. Another research opportunity concerns conducting further evaluation research investigating the feasibility of applying the proposed approaches.

REFERENCES

[1] S. H. Aboutorabi, M. Rezapour, M. Moradi, and N. Ghadiri, "Performance evaluation of SQL and MongoDB databases for big e-commerce data," in Computer Science and Software Engineering (CSSE), International Symposium on, IEEE, 2015.

[2] J. Akoka, I. ComynWattiau, N. Laoufi, "Research on Big Data – A systematic mapping study," in Computer Standards and Interfaces, vol. 54, pp. 105-115, 2017.

[3] M. Alayyoub, A. Yazici, and Z. Karakaya, " A Systematic Mapping Study for Big Data Stream Processing Frameworks," in Digital Information Management (ICDIM), International Conference on, 2016.

[4] K. Anderson, A. Schram, A. Alzabarah, and L. Palen, "Architectural Implications of Social Media Analytics in Support of Crisis Informatics Research," in IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW), IEEE, 2013.

[5] K. Anderson and A. Schram, "Design and Implementation of a Data Analytics Infrastructure in Support of Crisis Informatics Research

[1] http://www.inf.puc-rio.br/~kalinowski/seaa2018

(NIER Track)," in Software Engineering (ICSE), 33rd International Conference on, IEEE, 2011.

[6] K. Anderson, "Embrace the Challenges: Software Engineering in a Big Data World," in Big Data Software Engineering (BIGDSE), International Workshop on, IEEE, 2015.

[7] J. Anderson, R. Soden, K. Anderson, M. Kogan, and L. Palen, "EPIC-OSM: A Software Framework for OpenStreetMap Data Analytics," in System Sciences (HICSS), 49th Hawaii International Conference on, IEEE,2016.

[8] N. C. Audsley, Y. Chan, I. Gray, and A. J. Wellings, "Real-Time Big Data the JUNIPER Approach," in Real-time and distributed computing in emerging applications, 3rd IEEE International Workshop on, IEEE, 2014.

[9] A. B. Bener, I. Gorton, A. Mockus, "Software Engineering for Big Data Systems," in IEEE Software, vol. 33, no. 2, IEEE, 2016.

[10] S. Bazargani, J. Brinkley, and N. Tabrizi, "Implementing conceptual search capability in a cloud-based feed aggregator," in Innovative Computing Technology (INTECH), International Conference on, IEEE, 2013.

[11] F. Bersani and M. Erascu, "A tool for verification of big-data applications," in Quality-Aware DevOps, Proceedings of the 2nd International Workshop on, ACM, 2016, pp. 44-45.

[12] BIGDSE, "Proceedings of the 2nd International Workshop on BIG Data Software Engineering," https://sse.uni-due.de/bigdse16/, 2018.

[13] J. Bodorik and D. N. Jutla, "PAUSE: A Privacy Architecture for Heterogeneous Big Data Environments," in Big Data (Big Data), IEEE International Conference on, IEEE, 2015.

[14] W. Brewer, W. Scott, and J. Sanford, "An Integrated Cloud Platform for Rapid Interface Generation, Job Scheduling, Monitoring, Plotting, and Case Management of Scientific Applications," in Cloud Computing Research and Innovation (ICCCRI), International Conference on, IEEE,2015.

[15] D. Burbank, "The 5 V's of Big Data," https://www.ealearning.com/resources/blog/the-5-v%E2%80%99s-of-big-data.html, 2018.

[16] Capgemini. Cracking the Data Conundrum: How Successful Companies Make Big Data Operational. 2015.

[17] Capgemini and Informatica. The Big Data Payoff: Turning Big Data into Business Value. 2016.

[18] G. Casale, D. Ardagna, M. Artac, F. Barbier, E. D. Nitto, A. Henry, G. Iuhasz, C. Joubert, J. Merseguer, V. I. Munteanu, J. F. Pérez, D. Petcu, M. Rossi, C. Sheridan, I. Spais, and D. Vladuic, "DICE: Quality-Driven Development of Data-Intensive Cloud Applications," in Modeling in Software Engineering, Proceedings of the Seventh International Workshop on, IEEE Press, 2015, pp. 78-83.

[19] C. Cecchinel, M. Jimenez, S. Mosser, and M. Riveill, "An Architecture to Support the Collection of Big Data in the Internet of Things," in Services (SERVICES), World Congress on, IEEE,2014.

[20] T. Cerqueus, E. C. D. Almeida, and S. Scherzinger, "Safely Managing Data Variety in Big Data Software Development," in Big Data Software Engineering, IEEE/ACM International Workshop on, IEEE,2015.

[21] S. Chen, G. Bronevetsky, L. Peng, B. Li, and X. Fu, "Soft error resilience in Big Data kernels through modular analysis," in The Journal of Supercomputing, vol. 72, no. 4, March 2016, pp. 1570-1596.

[22] H. Chen, R. Kazman, and O. Haziyev, "Big Data System Development: An Embedded Case Study with a Global Outsourcing Firm," in Big Data Software Engineering (BIGDSE), IEEE/ACM 1st International Workshop on, IEEE, 2015.

[23] H. Chen, R. Kazman, and S. Haziyev, "Agile Big Data Analytics Development: An Architecture-Centric Approach," in System Sciences, 49th Hawaii International Conference on, IEEE, 2016.

[24] H. Chen, R. Kazman, and S. Haziyev, "Agile Big Data Analytics for Web-Based Systems: An Architecture-Centric Approach," in IEEE Transactions on Big Data, vol. 2, no. 3, May 2016, pp. 234-248.

[25] H. Chen, R. Kazman, and S. Haziyev, "Strategic Prototyping for Developing Big Data Systems," in IEEE Software, vol. 33, no. 2, February 2016, pp. 36-43.

[26] H. Chen, R. Kazman, J. Garbajosa, and E. Gonzalez, "Toward Big Data Value Engineering for Innovation," in BIG Data Software Engineering, International Workshop on, IEEE, 2016.

[27] K. Chen, X. Li, and H. Wang, "On the model design of integrated intelligent big data analytics systems," in Industrial Management & Data Systems, vol. 115, no. 9, 2015, pp. 1666-1682.

[28] V. D. Kumar and P. Alencar, "Software Engineering for Big Data Projects: Domains, Methodologies and Gaps," Big Data (Big Data), IEEE International Conference on, IEEE, 2016.

[29] T. Dyba, B. A. Kitchenham, M. Jorgensen, "Evidence-based Software Engineering for Practitioners," IEEE Software, v. 22, 1, p. 58-65, IEEE, jan. 2014.

[30] L. E. B. Villalpando and A. April, "Performance analysis model for big data applications in cloud computing," in Journal of Cloud Computing: Advances, Systems and Applications, 2014, pp. 3.

[31] H. Eridaputra, B. Hendradjaya, and W. D. Sunindyo, "Modeling the requirements for big data application using goal oriented approach," in Data and Software Engineering (ICODSE), International Conference on, IEEE, 2014.

[32] E. Evans, Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley, 2004.

[33] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," International Journal of Information Management, vol. 35, pp. 137-144, 2015.

[34] D. Gil and I. Song, "Modeling and Management of Big Data Challenges and opportunities," in Future Generation Computer Systems, vol. 63, October 2016, pp. 96–99.

[35] A. Gómez, J. Merseguer, E. D. Nitto, and D. A. Tamburri, "Towards a UML Profile for Data Intensive Applications," in Quality-Aware DevOps, Proceedings of the 2nd International Workshop on, pp. 18-23, 2016.

[36] I. Gorton and J. Klein, "Distribution, Data, Deployment Software Architecture Convergence in Big Data Systems," in IEEE Software, IEEE, 2015, pp. 32.

[37] M. Guerriero, S. Tajfar, D. A. Tamburri, and E. D. Nitto, "Towards a model-driven design tool for big data architectures," in BIG Data Software Engineering, IEEE/ACM International Workshop on, IEEE 2016.

[38] Z. L. He, X. H. Xiao, and Y. H. He, "A software design model based on big data," in Applied Mechanics and Materials, vol. 644-650, pp. 2821-2825, 2014.

[39] H. Hu, Y. Wen, T. Chua, and X. Li, "Toward Scalable Systems for Big Data Analytics: A Technology Tutorial," in IEEE Access, 2,June 2014, pp. 652-687.

[40] D. N. Jutla, P. Bodorik, and S. Ali, "Engineering Privacy for Big Data Apps with the Unified Modeling Language," in Big Data (BigData Congress), IEEE International Congress on, IEEE, 2013.

[41] R. Kazman, "Prototyping for developing big data systems," https://insights.sei.cmu.edu/sei_blog/2016/07/prototyping-for-developing-big-datasystems.html, 2018.

[42] B. Kitchenham, K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, "Systematic Mapping Studies in Software Engineering," in Evaluation and Assessment in Software Engineering, Proceedings of the 12th international conference on, BCS Learning & Development Ltd., pp. 68-77, June 2008.

[43] B. Kitchenham, S. Charters, "Guidelines for performing systematic literature reviews in software engineering." Keele University and Durham University Joint Report, Technical Report EBSE 2007-001, 2007.

[44] J. Klein and I. Gorton, "Design Assistant for NoSQL Technology Selection," in the Future of Software Architecture Design Assistants, International Workshop on, IEEE, 2015.

[45] J. Klein, I. Gorton, N. Ernst, P. Donohoe, K. Pham, and C. Matser, "Application-Specific Evaluation of No SQL Databases," in Big Data, IEEE International Congress on, IEEE, 2015.

[46] J. Klein, R. Buglak, D. Blockow, T. Wuttke, and B. Cooper, "A reference architecture for big data systems in the national security domain," in BIG Data Software Engineering, Proceedings of the 2nd International Workshop on, pp. 51-57, ACM, 2016.

[47] M. Kraemer and I. Senner, "A modular software architecture for processing of big geospatial data in the cloud," in Computers & Graphics, vol. 49, June 2015, pp. 69-81.

[48] D. Laney, "3D Data Management: Controlling Data Volume, Velocity, and Variety", Meta Group, 2001.

[49] C. Li, L. Huang, and L. Chen, "Breeze graph grammar: a graph grammar approach for modeling the software architecture of big data-oriented software systems," in Software: Practice and Experience, vol. 45, no. 8, pp. 1023-1050, 2015.

[50] M. Mirakhorli, H. Chen, and R. Kazman, "Mining Big Data for Detecting, Extracting and Recommending Architectural Design Concepts," in Big Data Software Engineering (BIGDSE), International Workshop on, IEEE, 2015.

[51] A. Miranskyy, A. Hamou-lhadj, E. Cialini, and A. Larsson, "Operational-Log Analysis for Big Data Systems Challenges and Solutions," in IEEE Software, vol. 33, no. 2, pp. 52- 59, 2016.

[52] A. Mockus, "Engineering Big Data Solutions," in Proceedings of the on Future of Software Engineering, pp. 85-99, ACM, 2014.

[53] E. Mourão, M. Kalinowski, L. Murta, E. Mendes, and C. Wohlin, "Investigating the Use of a Hybrid Search Strategy for Systematic Reviews." International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 193-198, 2017.

[54] A. Naseer, B.Y. Alkazemi, and E. U. Waraich, "A big data approach for proactive healthcare monitoring of chronic patients," in Ubiquitous and Future Networks (ICUFN), International Conference on, IEEE, 2016.

[55] E. Nitto, P. Jamshidi, M. Guerriero, I. Spais, and D. A. Tamburri, "A Software Architecture Framework for Quality-aware DevOps," in Quality-Aware DevOps, Proceedings of the 2nd International Workshop on, July 2015, pp. 12-17.

[56] I. Noorwali, D. Arruda, and N. H. Madhavji, "Understanding Quality Requirements in the Context of Big Data Systems," in BIG Data Software Engineering, Proceedings of the 2nd International Workshop on, ACM, 2016.

[57] P. O'Donovan, Kevin Leahy, Ken Bruton, and Dominic T. J. O'Sullivan, "Big data in manufacturing: a systematic mapping study," in Jornal of Big Data. Springer, 2015.

[58] M. I. Ortega, M. Genero, M. Piattini, "Big data DBMS assessment: A systematic mapping study," in Model and Data Engineering, International Conference on, pp. 96-110, 2017.

[59] P. Pääkkönen and D. Pakkala, "Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems," in Big Data Research, vol. 2, no. 4, December 2015, pp. 166-186.

[60] A. Rajbhoj, V. Kulkarni, and N. Bellarykar, "Early Experience with Model-driven Development of MapReduce based Big Data Application," in Asia-Pacific Software Engineering Conference (APSEC), IEEE,2014.

[61] A. Ringlstetter, S. Scherzinger, and T. Bissyandé, "Data Model Evolution using Object-NoSQL Mappers: Folklore or State-of-the-Art," in BIG Data Software Engineering, IEEE/ACM 2nd International Workshop on, IEEE, 2016.

[62] S. Rosenthal, S. Mcmillan, and E. G. Matthew, "Developer Toolchains for Large-Scale Analytics: Two Case Studies," in Big Data (Big Data), IEEE International Conference on, IEEE, 2015.

[63] P. Russom, Big Data Analytics. TDWI Research, 2011.

[64] K. S. Yim, "Norming to performing: Failure analysis and deployment automation of big data software developed by highly iterative models," in Software Reliability Engineering, IEEE International Symposium on, IEEE, 2014.

[65] SAS, "Big Data What it is and why it matters," https://www.sas.com/en_us/insights/big-data/what-is-big-data.html, 2018.

[66] M. Scavuzzo, D. A. Tamburri, and E. D. Nitto, "Providing Big Data Applications with Fault-tolerant Data Migration Across Heterogeneous NoSQL Databases," in BIG Data Software Engineering, IEEE/ACM International Workshop on, IEEE, 2016.

[67] A. Schram and K. Anderson, "Design challenges/solutions for environments supporting the analysis of social media data in crisis informatics research," in System Sciences (HICSS), International Conference on, IEEE, 2015.

[68] A. Schram and K. Anderson, "MySQL to NoSQL Data Modeling Challenges in Supporting Scalability," in Systems, Programming Languages and Applications: Software for Humanity, Conference on, ACM, 2012.

[69] M. A. A. D. Silva, A. Sadovykh, A. Bagnato, A. Cheptsov, and L. Adam, "JUNIPER: Towards Modeling Approach Enabling Efficient Platform for Heterogeneous Big Data Analysis," in Software Engineering Conference, ACM, 2014.

[70] K. J. Stol, P. Ralph, B. Fitzgerald, Grounded theory in software engineering research: a critical review and guidelines. IEEE/ACM 38th International Conference on Software Engineering (ICSE). IEEE, pp. 120-131, 2016.

[71] Techamerica Foundation's Federal Big Data Commission, "Demystifying big data: A practical guide to transforming the business of Government," http://www.techamerica.org/ Docs/fileManager.cfm?f=techamericabigdatareport-final.pdf, 2018.

[72] M. Villari, A. Celesti, M. Fazio, and A. Puliafito, "AllJoyn Lambda: an Architecture for the Management of Smart Environments in IoT," in Smart Computing, International Conference on, IEEE, 2014.

[73] R. Wieringa, N. Maiden, N. Mead, and C. Rolland. "Requirements engineering paper classification and evaluation criteria: a proposal and a discussion," Journal of Requir. Eng. vol. 11, no. 1, December 2005, pp. 102-107.

[74] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," International Conference on Evaluation and Assessment in Software Engineering (EASE), 10 p., 2014.

[75] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén. "A Experimentation in Software Engineering," Springer Publishing Company, 2012.

[76] Y. Zhang, F. Xu, E. Frise, S. Wu, B. Yu, and W. Xu, "DataLab: A Version Data Management and Analytics System," in BIG Data Software Engineering, IEEE/ACM 2nd International Workshop on, IEEE,2016.

[77] A. Zimmermann, M. Pretz, G. Zimmermann, D. G. Firesmith, I. Petrov, and E. El-sheikh, "Towards Service-oriented Enterprise Architectures for Big Data," in International Enterprise Distributed Object Computing Conference Workshops, IEEE, 2013.