2021 International Conference on Information Technology (ICIT)

Applying Test Driven Development in the Big Data Domain – Lessons From the Literature

Daniel Staegemann
MRCC VLBA
Otto-von-Guericke University
Magdeburg, Germany
daniel.staegemann@ovgu.de

Matthias Volk
MRCC VLBA
Otto-von-Guericke University
Magdeburg, Germany
matthias.volk@ovgu.de

Erik Lautenschläger
Faculty of Economics
FernUniversität in Hagen
Hagen, Germany
erik.laute@gmail.com

Matthias Pohl
MRCC VLBA
Otto-von-Guericke University
Magdeburg, Germany
matthias.pohl@ovgu.de

Mohammad Abdallah
Department of Software Engineering
Al-Zaytoonah University of Jordan
Amman, Jordan
m.abdallah@zuj.edu.jo

Klaus Turowski
MRCC VLBA
Otto-von-Guericke University
Magdeburg, Germany
klaus.turowski@ovgu.de

2021 International Conference on Information Technology (ICIT) | 978-1-6654-2870-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIT52682.2021.9491728

***Abstract*—Big data has evolved to a ubiquitous part of today's society. However, despite its popularity, the development and testing of the corresponding applications are still very challenging tasks that are being actively researched in pursuit of ways for improvement. One newly introduced proposition is the application of test driven development (TDD) in the big data domain. To facilitate this concept, existing literature reviews on TDD have been analyzed to extract insights from those sources of aggregated knowledge, which can be applied to this new setting. After introducing the different studies, lessons for the application of TDD in the big data domain are deducted and discussed. Finally, avenues for future works are proposed.**

Keywords*— *test driven development, TDD, big data, literature review, meta review

I. Introduction

The importance and ubiquity of information technology for our daily lives has been growing for many years [1], without any indication that this trend will stop in the foreseeable future [2]. Subsequently, also the significance of software products and their development has risen [3]. The same applies to the utilization of ever-growing amounts of more and more complex data that has been facilitated through the technological evolution and culminated in the concept of big data analytics (BDA). As a result, big data has found its way into many application domains such as agriculture [4], education [5], governance [6], manufacturing [7], sports [8] or transportation [9]. However, with the increasing technical possibilities and corresponding opportunities, also the complexity of the software itself has reached new levels [10]. Thus, depending on their purpose, the development of new applications can be very expensive, time-consuming, and is also prone to error [11]. As a result, approaches which promise to solve or alleviate those challenges are highly relevant for practitioners and thence also for researchers [12].

One of those strategies is test driven development (TDD), which already has been in the focus of the scientific community for many years [13]. Yet, the proposition to apply it in the domain of big data engineering is rather new [14]. Though, since the underlying principles generally stay the same, examining the existing literature on TDD appears to be a sensible approach to obtain insights that might be valuable for its transfer into the realm of big data. Therefore, the publication at hand seeks to answer the following research question (RQ):

RQ: Which lessons for the application of test driven development in the domain of big data can be learned from the existing literature on test driven development?

The remainder of the text is structured as follows. After the introduction, an overview regarding the topics of big data and TDD is given. Then, the study's review protocol is outlined. Subsequently, the findings are presented and discussed. Finally, a conclusion is given and potential prospects for future research are shown.

II. Background

With the transfer of lessons from TDD to the implementation of big data applications being the focus of the publication at hand, it is at first necessary to provide short introductions of both domains to have a solid foundation for the ensuing considerations.

A. Big Data

While the concept of BDA gained a highly important role in many aspects of today's society, there is no unified definition of the term big data [15]. However, it generally refers to data whose analysis is especially demanding due to their volume, velocity, variety and/or variability, resulting in a need to utilize highly sophisticated solutions that surpass the abilities of traditional techniques [16]. Those characteristics are also described as the four Vs of big data.

The first property, *volume*, refers to the huge amount of data, either in size or concerning the number of files that must be handled. This characteristic is probably the most renowned one, which is also reflected in the denomination. While the definition of the word "big" in the given context is somewhat fuzzy, today's workloads can reach the petabyte-region [17]. Velocity can be used to describe two aspects. On the one hand it refers to the speed with which data are ingested by a system and on the other hand how fast it has to process them to fulfill the use case's respective applicable requirements [11]. Variety denotes the diversity in the data's origins [16] as well as the heterogeneity of data types, languages, ways of formatting and further significant properties of the handled data [18]. Finally, the term variability stands for the fluctuation concerning the previously mentioned characteristics [11]. Those changes can occur temporarily [19] or long-term, which can potentially necessitate a corresponding adaption of the big data application [20].

Yet, while those four characteristics might be the most common ones and are therefore also adopted for the publication at hand, the scientific literature deals with many others as well [21]. Examples for further characteristics are for instance properties like value [22], validity [23] or veracity [24]. As a result of this complexity, the implementation of BDA applications and the corresponding infrastructure is complicated and often prone to error [25–28]. Furthermore, also the integration of big data into an organization's productive workflow is a challenging task with many aspects that have to be considered [11, 29–32]. However, if implemented correctly, it also promises substantial gains [31, 33, 34], which are in many cases realized through an improved decision-making by the respective management. Subsequently, to assure their usage and maximize the benefits, it is necessary to make sure that decision makers are provided relevant, timely and correct information in an easily accessible and comprehensible way [20]. One way towards this vision is the continuous improvement of the utilized applications in conjunction with a rigorous testing regime. This in turn can be facilitated through the adaptation of test driven development to the big data engineering domain [14].

B. Test Driven Development

TDD is an approach that is mostly used in software development and aims at increasing the quality of the implemented product by not only regulating its testing, but also the way it is designed [35, 36]. However, due to the focus of the publication at hand, the following considerations are explicitly only geared towards the development of software, even though most of it might also be applicable to other domains like the development of ontologies [37, 38] or process modeling [39]. When developing software, the traditional way of doing so is to first think of a change in functionality that ought to be integrated, implement it, and then test it. When leveraging the process of TDD, the last two steps are interchanged. This means a change in functionality is envisioned first, subsequently one or multiple tests for the change are written and executed, but supposed to fail without a corresponding software counterpart to run against [40]. Only then, the implementation of the change is conducted. To be valid, it must pass the previously written tests [36]. Subsequently, a refactoring of the written code is conducted, so not only the tests are passed, but an adherence to the environment's standards and best practices is warranted [40].

This approach also severely affects the software design, since an underlying principle of TDD is that a change is planned as a smallest possible unit, instead of a large working package [41]. This allows for small, incremental modifications [42], facilitating the interlocking of testing and development to enable the developer to leverage short testing cycles [43] and therefore achieve a high test frequency. Commonly, most of the tests are specifically written for those units. However, a variety of other tests can be leveraged in TDD, such as integration, system, and acceptance tests [44]. Another method often used in the context of TDD is continuous integration (CI) in combination with test automation [45, 46]. A CI server automatically starts all available tests when new code is committed to a versioning system, thus assuring that newly implemented changes have no negative effect on already existing parts of the software.

III. Review protocol

The publication at hand aims to harness the existing literature on TDD to derive insights that might help facilitating its use in the domain of big data. To obtain said literature, *Scopus* was used, which is considered the most extensive abstract and citation database for scientific literature and also provides the user with a variety of tools to facilitate sophisticated searches [47, 48]. To get a first impression of the amount of general interest in TDD, an initial search was conducted, using the following search term:

TITLE("test driven development" OR "test-driven-development" OR "test-driven development" OR "tdd") AND (LIMIT-TO (DOCTYPE,"cp") OR LIMIT-TO (DOCTYPE,"ar")) AND (LIMIT-TO (SUBJAREA,"COMP")) AND (LIMIT-TO (LANGUAGE,"English")) AND (LIMIT-TO (SRCTYPE,"p") OR LIMIT-TO (SRCTYPE,"j"))

In doing so, all contributions whose title includes "*test driven development*" or "*test-driven-development*" or "*test-driven development*" where identified, if they were written in *English*, published in *conference proceedings* or a *journal* (therefore, assuring peer review) and are attributed to the domain of *computer science*. This led to 178 results. When also including the abbreviation "TDD" to the terms looked for in the title, the number grew to 1186. However, since the abbreviation is not unique, those results also included numerous publications from other domains as for example communications [49]. Therefore, in the following, it was dropped. When considering the 178 results, it comes to notice that the scientific interest appears to be mostly constant, as it can be seen in Fig. 1. While there have been some outliers, the number of yearly publications from 2006 to 2020 (therefore omitting the initial ramp up phase) has always been between eight and twelve, except for three years. Furthermore, after three months of 2021, there have been three publications, which also fits this range. This indicates a constant research interest and the ongoing relevance of TDD. However, to get a more condensed view, in the following considerations, a further stipulation was added to the search. Only literature review papers were considered, since those usually have a broader view than those that, for example, only consider one single case study, and therefore already represent a self-contained collection. Hence, it is to be assumed, that their findings are better generalizable for this paper's intended purpose. For this purpose, the initial search term was amended to also demand "*review*" or "*survey*" or "*overview*" or "*state of the art*" in the title. The new search term was:

(TITLE("test driven development" OR "test-driven-development" OR "test-driven development")) AND (TITLE("review" OR "survey" OR "overview" OR "state of the art")) AND (LIMIT-TO (SRCTYPE,"p") OR LIMIT-TO (SRCTYPE,"j")) AND (LIMIT-TO (DOCTYPE,"cp") OR LIMIT-TO (DOCTYPE,"ar")) AND (LIMIT-TO (SUBJAREA,"COMP")) AND (LIMIT-TO (LANGUAGE,"English"))

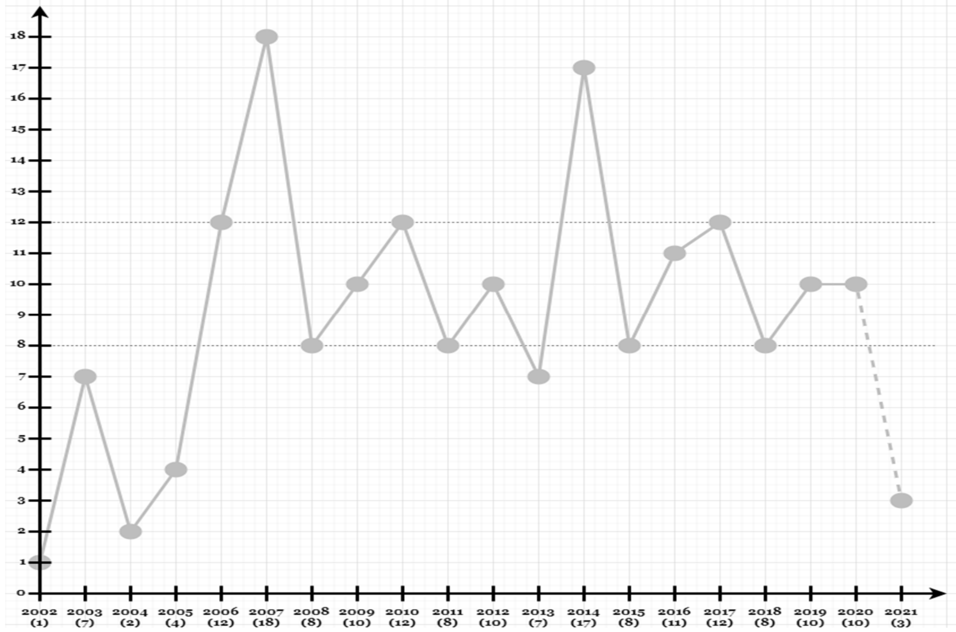

Fig. 1. Number of TDD Publications Per Year in Scopus

As a result, six publications, namely [50–55], were identified. However, despite being highly relevant to the domain of TDD in general, three of those did not fit the intended scope of this research, since they were not actually reviews. Instead, they dealt with a single experiment concerning peer code review in TDD [53], respectively an online survey on common mistakes in TDD [50] or a survey on TDD in scientific software [55]. Therefore, those three were excluded, resulting in a set of three review papers, which are further explored in the following section.

Furthermore, to increase the comprehensibility, the scope of the search was extended. Incorporating the contributions highlighted in the "related work" sections of the found papers yielded five additional studies, leading to a total of eight analyzed publications.

IV. Findings And discussion

As a result of the search and filter process, the final set of literature comprises eight publications, which each aim to provide an overview of (certain aspects of) the domain of TDD and will be discussed in the following.

A. Findings

The earliest article, "TDD: The Art of Fearless Programming" [56], which was published in 2007 as a guest editors introduction, highlighted a mixed notion on TDDs impact on quality and productivity, which is, inter alia, attributed to incomparable measurements. However, despite it being no silver bullet, a positive effect on task focus, test coverage and an increased attention on test design by the developers are stated.

In 2010, the "Seventh International Conference on the Quality of Information and Communications Technology" yielded two relevant contributions. The first one "Empirical Studies on Quality in Agile Practices: A Systematic Literature Review" [57] is not limited to TDD but focuses on agile practices in general. However, a part of this work is also designated to TDD in particular. There, a notable improvement in external quality was experienced in most studies. While the number of defects was in most cases reduced by 5% to 45%, some papers even reported higher numbers of up to 90%. The sentiment towards the internal quality was generally positive, while the results concerning productivity were indecisive.

Further, in "Test-Driven Development - Still a Promising Approach?" [58], the application of TDD was evaluated regarding its effect on quality and productivity. To further differentiate the quality, it was divided into two categories, internal software quality (e.g. test coverage) and external software quality (e.g. the results of black box testing). While for the former 38% of the studies reported an increase, 56% stated no difference and 6% reported a decrease, the sentiment concerning external quality was even more positive. An increase was determined in 73% of the studies, 23% found no difference and 4% noticed a reduction. With regard to TDDs influence on productivity, 22% of studies saw an increase, 30% were neutral and 48% detected a decrease. However, the authors also pointed out that the quality of the reviewed reports and the varying setups were a huge issue, which might skew the results.

The paper "Factors Limiting Industrial Adoption of Test Driven Development: A Systematic Review" [51] is from the year 2011 and focusses on obstacles that prevent the utilization of TDD in an industrial context. This was inspired by an industrial survey [59] of the same authors that showed that there was a high disparity between the desire to apply TDD and its actual implementation. Subsequently, they aspired to identify the underlying causes. By conducting a thorough literature review, the authors identified seven factors that are limiting the industrial adoption of TDD. The first limiting factor is an observed *increase in upfront development time*. While the use of TDD might pay off in the long run, since a lower amount of corrective re-work is to be expected and the quality of the product is probably higher, the initial implementation for a given set of requirements is usually slower compared to the traditional test last approach. However, which of the approaches is more beneficial for a project, respectively the conducting company, has to be evaluated on a case-by-case basis. Furthermore, a *lack of experience in implementing TDD* as well as *insufficient developer testing skills* can constitute significant barriers.

Some studies also reported that a system's envisioned *design* could constitute a problem. The fifth factor is an *inadequate adherence to the TDD methodology*. Additionally, *technical issues and limitations* have been highlighted as a very important factor. Finally, *legacy code* constitutes the seventh factor, since the TDD methodology assumes that the project is developed from scratch, which does not always hold true. To overcome the challenges, the study's authors proposed a set of measures. Those include the provision of training in TDD practice and test case design before its productive implementation as well as the recruitment of experienced developers who can serve as mentors. Furthermore, it is important to adjust the concrete planning of the applied TDD procedure to the specific situation of the respective project and organization, since sometimes trade-offs are necessary, which prohibit the use of only one general strategy.

In 2013, "The Effects of Test-Driven Development on External Quality and Productivity: A Meta-Analysis" [60] was another study that aimed to bring clarity to the effects of TDD on quality and productivity. For this purpose, also several subgroups within the complete set of observed studies were analyzed. However, unlike in other papers, only the external quality was regarded but not the internal one. Overall and similar to other contributions, this work also concluded that TDD leads to an increase in quality. Furthermore, the results obtained on the productivity were described as inconclusive.

For the year 2014, two relevant publications were found. In the first one, "Considering rigor and relevance when evaluating test driven development: A systematic review" [61], the authors deviated from the standard procedure by additionally characterizing the examined studies based on their observed rigor and relevance, therefore incorporating their quality into the final conclusion. For studies where either one or both of rigor and relevancy were deemed low, the different findings are quite conflicting. However, the results of the highest quality studies (high rigor and high relevancy) are consistent and indicate an improvement concerning code quality as well as a reduction in complexity, while the productivity remains the same or is decreasing. This disparity could also somewhat relativize some of the results of other studies whose findings show a bigger diversity in outcomes.

The other paper, "Test driven development contribution in universities in producing quality software: A systematic review" [52], has its focus in the educational sector. This review confirmed the findings of [51], that the developer's familiarity with the TDD process is highly influential on the success of its application. Furthermore, another crucial aspect has been identified, namely the willingness of the programmers to actually operate in a test driven manner instead of the sometimes more convenient test last approach.

Within the final paper, "The Effects of Test Driven Development on Internal Quality, External Quality and Productivity: A systematic review" [54], which was published in 2016, several quantitative insights concerning the effects of TDD on quality and productivity were obtained. For this purpose, studies that analyzed the respective effects of TDD were analyzed and the results subsequently consolidated. Therefore, this paper is quiet similar to [58] and could be somewhat considered an update, due to its later date of origin. The examined studies mainly used Java or C++ as the programming language and the participants were almost equally distributed between students and professionals (with a small proportion observing both groups). Further, it is notable, that the majority of the studies (about 85%) were published before 2010. In total, 76% of the analyzed studies stated a meaningful increase in internal quality. Regarded the external quality, the results are even more unambiguous, with 88% of the studies reporting a significant increase in external quality. Those results strongly indicate the positive effects of the application of TDD on the quality of created software. Concerning the influence on productivity, the sentiment is less clear. An increase of productivity was stated by 28% of the studies that regarded this aspect. The same proportion suggested that the productivity remained stable and 44% showed a decrease. However, when also taking the settings into account and differentiating between academic and industrial studies, the sentiment changes. In the academic setting, there were five studies reporting productivity increases, one decrease and three studies where it did not change. Yet, in the industrial setting, no study showed an increase in productivity, whereas two papers reported no change and seven noticed a decrease in productivity. Since the academic setting is likely to be stronger biased through a more heterogeneous distribution of skill and a generally lower ability of students compared to professional developers, it can be assumed that the validity and therefore the significance of the industrial findings is higher.

B. Discussion

By regarding the found publications in conjunction, it becomes apparent that the question how the application of TDD affects quality and productivity has been highly popular and was somewhat focused in seven out of eight papers. The only outlier was [51], which instead aimed to determine factors that hamper the utilization of TDD. Due to this, it is also arguably the most relevant one for addressing this paper's RQ, because it adds several unique aspects. However, the combined sentiment of the other studies, suggesting that TDD increases the software's quality but slows down the development process is also of great value, especially since it is based on numerous works, which reduces the risk of bias and therefore increases validity. As a consequence of those effects, the choice for or against TDD depends on the priorities of the respective project. While it can increase the overall quality and might therefore in the long run also help in reducing the effort needed for maintenance, sometimes speed or ease of implementation might be more important. Transferred to the big data domain this suggests that it might be beneficial to be less strict for (parts of) applications where quality is not mission critical as long as they are running without crashing and the results are somewhat accurate. This means that, instead of testing every single unit, just providing basic tests for (sub-)components could be an efficient compromise. That way it is possible to allow for regression tests and make sure that the general system is still working after implementing changes without increasing the workload too much. An example for this could be the use of highly proven databases, where the correct embedding into the system as a whole needs to be validated, but their general functionality can be taken for granted. At the same time, for parts that are prone to error, completely new or have to be of especially high quality, a rigorous TDD process should be applied. Moreover, some aspects, as for example the GUI [51], might be very hard to test in an automated fashion. Therefore, it appears reasonable to forego those aspect and resort to manual testing when the required effort is expected to be unreasonably high. Finally, the developer's skills, experience, and motivation as well as a proper tool support are vital. Therefore, when deciding to introduce TDD, a sufficiently long adaptation period should be factored in. Finally, since the TDD paradigm does not account for legacy code, its utilization works best for the implementation of from scratch applications. Retroactively applying it in an already working systems poses huge challenges and it should therefore be well considered if it is actually worth it.

V. Conclusion

With the importance of big data continuously rising and the corresponding implementation and testing still being very challenging tasks, approaches that facilitate those activities are in high demand. One of those concepts is the application of TDD to the big data domain. Therefore, in this publication, existing literature reviews on TDD have been compiled and analyzed to deduct lessons that can be transferred to this new application area and thereby help in creating new BDA systems. While the total number of eight studies seems rather low, their mostly unanimity still indicates a high validity. However, with the latest study being from 2016, an updated analysis of the effects of TDD on software quality and productivity appears to be a worthwhile future endeavor. Furthermore, additional qualitative studies that go beyond those aspects and shed light on the adoption and implementation of TDD in general as well as specifically in the big data domain could also lead to valuable new insights. Overall, the contribution of this paper is threefold, it aggregates the findings of the implications when applying TDD, it provides an overview of relevant studies to allow further investigations and it outlines avenues for future research that might help and inspire other scientists to address the underlying topics.

References

[1] S. Deb, "Information Technology, Its Impact on Society and Its Future," Advances in Computing, vol. 4, no. 1, pp. 25–29, 2014.

[2] T. N. Theis and H.-S. P. Wong, "The End of Moore's Law: A New Beginning for Information Technology," Comput. Sci. Eng., vol. 19, no. 2, pp. 41–50, 2017, doi: 10.1109/MCSE.2017.29.

[3] I. Sommerville, Software engineering. Boston, Columbus, Indianapolis, New York, San Francisco, Hoboken, Amsterdam, Cape Town, Dubai, London, Madrid, Milan, Munich, Paris, Montreal, Toronto, Delhi, Mexico City, São Paulo, Sydney, Hong Kong, Seoul, Singapore, Taipei, Tokyo: Pearson, 2016.

[4] K. Bronson and I. Knezevic, "Big Data in food and agriculture," Big Data & Society, vol. 3, no. 1, 2016, doi: 10.1177/2053951716648174.

[5] R. Häusler, D. Staegemann, M. Volk, S. Bosse, C. Bekel, and K. Turowski, "Generating Content-Compliant Training Data in Big Data Education," in Proceedings of the 12th CSEdu, Prague, Czech Republic, 2020, pp. 104–110.

[6] G.-H. Kim, S. Trimi, and J.-H. Chung, "Big-data applications in the government sector," Commun. ACM, vol. 57, no. 3, pp. 78–85, 2014, doi: 10.1145/2500873.

[7] K. Nagorny, P. Lima-Monteiro, J. Barata, and A. W. Colombo, "Big Data Analysis in Smart Manufacturing: A Review," IJCNS, vol. 10, no. 03, pp. 31–58, 2017, doi: 10.4236/ijcns.2017.103003.

[8] R. Rein and D. Memmert, "Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science," SpringerPlus, vol. 5, no. 1, p. 1410, 2016, doi: 10.1186/s40064-016-3108-2.

[9] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big Data Analytics in Intelligent Transportation Systems: A Survey," IEEE Trans. Intell. Transport. Syst., vol. 20, no. 1, pp. 383–398, 2019, doi: 10.1109/TITS.2018.2815678.

[10] V. Damasiotis, P. Fitsilis, and J. F. O'Kane, "Modeling Software Development Process Complexity," International Journal of Information Technology Project Management, vol. 9, no. 4, pp. 17–40, 2018, doi: 10.4018/IJITPM.2018100102.

[11] M. Volk, D. Staegemann, M. Pohl, and K. Turowski, "Challenging Big Data Engineering: Positioning of Current and Future Development," in Proceedings of the 4th International Conference on Internet of Things, Big Data and Security, Heraklion, Crete, Greece, 2019, pp. 351–358.

[12] A. A. Khan, J. W. Keung, Fazal-E-Amin, and M. Abdullah-Al-Wadud, "SPIIMM: Toward a Model for Software Process Improvement Implementation and Management in Global Software Development," IEEE Access, vol. 5, pp. 13720–13741, 2017, doi: 10.1109/ACCESS.2017.2728603.

[13] M. Hill, "Sharpening the Axe for Test Driven Development," in Lecture Notes in Computer Science, Extreme Programming and Agile Methods — XP/Agile Universe 2002, G. Goos, J. Hartmanis, J. van Leeuwen, D. Wells, and L. Williams, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 269–270.

[14] D. Staegemann, M. Volk, N. Jamous, and K. Turowski, "Exploring the Applicability of Test Driven Development in the Big Data Domain," in Proceedings of the ACIS 2020, Wellington, New Zealand, 2020.

[15] M. Volk, D. Staegemann, and K. Turowski, "Big Data," in Springer Reference Wirtschaft, Handbuch Digitale Wirtschaft, T. Kollmann, Ed., Wiesbaden: Springer Fachmedien Wiesbaden, 2020, pp. 1–18.

[16] NIST, "NIST Big Data Interoperability Framework: Volume 1, Definitions, Version 3," Gaithersburg, MD, 2019.

[17] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," International Journal of Information Management, vol. 35, no. 2, pp. 137–144, 2015, doi: 10.1016/j.ijinfomgt.2014.10.007.

[18] S. Kaisler, F. Armour, J. A. Espinosa, and W. Money, "Big Data: Issues and Challenges Moving Forward," in Proceedings of the 46th Hawaii International Conference on System Sciences, Wailea, HI, USA, 2013, pp. 995–1004.

[19] A. Katal, M. Wazid, and R. H. Goudar, "Big data: Issues, challenges, tools and Good practices," in Parashar (Hg.) – 2013 sixth International Conference, 2013, pp. 404–409.

[20] D. Staegemann, M. Volk, C. Daase, and K. Turowski, "Discussing Relations Between Dynamic Business Environments and Big Data Analytics," CSIMQ, no. 23, pp. 58–82, 2020, doi: 10.7250/csimq.2020-23.05.

[21] I. Emmanuel and C. Stanier, "Defining Big Data," in Proceedings of the International Conference on Big Data and Advanced Wireless Technologies - BDAW '16, Blagoevgrad, Bulgaria, 2016, pp. 1–6.

[22] M. Younas, "Research challenges of big data," SOCA, vol. 13, no. 2, pp. 105–107, 2019, doi: 10.1007/s11761-019-00265-x.

[23] M. A.-u.-d. Khan, M. F. Uddin, and N. Gupta, "Seven V's of Big Data understanding Big Data to extract value," in Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education, Bridgeport, CT, USA, 2014, pp. 1–5.

[24] I. E. Alaoui, Y. Gahi, and R. Messoussi, "Full Consideration of Big Data Characteristics in Sentiment Analysis Context," in Proceedings of the IEEE 4th International Conference on Cloud Computing and Big Data Analysis, Chengdu, China, 2019, pp. 126–130.

[25] M. Abdallah, M. Muhairat, A. Althunibat, and A. Abdalla, "Big Data Quality: Factors, Frameworks, and Challenges," Compusoft: An International Journal of Advanced Computer Technology, vol. 9, no. 8, pp. 3785–3790, 2020.

[26] D. Staegemann, J. Hintsch, and K. Turowski, "Testing in Big Data: An Architecture Pattern for a Development Environment for Innovative, Integrated and Robust Applications," in Proceedings of the WI2019, 2019, pp. 279–284.

[27] D. Staegemann, M. Volk, A. Nahhas, M. Abdallah, and K. Turowski, "Exploring the Specificities and Challenges of Testing Big Data Systems," in Proceedings of the 15th International Conference on Signal Image Technology & Internet based Systems, Sorrento, 2019.

[28] M. Volk, D. Staegemann, S. Bosse, R. Häusler, and K. Turowski, "Approaching the (Big) Data Science Engineering Process," in Proceedings of the 5th International Conference on Internet of Things, Big Data and Security, Prague, Czech Republic, 2020, pp. 428–435.

[29] W. A. Günther, M. H. Rezazade Mehrizi, M. Huysman, and F. Feldberg, "Debating Big Data: A Literature Review on Realizing Value From Big Data," The Journal of Strategic Information Systems, vol. 26, no. 3, pp. 191–209, 2017, doi: 10.1016/j.jsis.2017.07.003.

[30] Z. A. Al-Sai, R. Abdullah, and M. H. Husin, "Critical Success Factors for Big Data: A Systematic Literature Review," IEEE Access, vol. 8, pp. 118940–118956, 2020, doi: 10.1109/ACCESS.2020.3005461.

[31] P. Maroufkhani, M.-L. Tseng, M. Iranmanesh, W. K. W. Ismail, and H. Khalid, "Big data analytics adoption: Determinants and performances among small to medium-sized enterprises," International Journal of Information Management, vol. 54, p. 102190, 2020, doi: 10.1016/j.ijinfomgt.2020.102190.

[32] D. Staegemann, M. Volk, N. Jamous, and K. Turowski, "Understanding Issues in Big Data Applications - A Multidimensional Endeavor," in Proceedings of the Twenty-fifth Americas Conference on Information Systems, Cancun, Mexico, 2019.

[33] O. Müller, M. Fay, and J. Vom Brocke, "The Effect of Big Data and Analytics on Firm Performance: An Econometric Analysis Considering Industry Characteristics," Journal of Management Information Systems, vol. 35, no. 2, pp. 488–509, 2018, doi: 10.1080/07421222.2018.1451955.

[34] M. Ghasemaghaei, "Improving Organizational Performance Through the Use of Big Data," Journal of Computer Information Systems, vol. 60, no. 5, pp. 395–408, 2020, doi: 10.1080/08874417.2018.1496805.

[35] F. Shull, G. Melnik, B. Turhan, L. Layman, M. Diep, and H. Erdogmus, "What Do We Know about Test-Driven Development?," IEEE Softw., vol. 27, no. 6, pp. 16–19, 2010, doi: 10.1109/MS.2010.152.

[36] L. Crispin, "Driving Software Quality: How Test-Driven Development Impacts Software Quality," IEEE Softw., vol. 23, no. 6, pp. 70–71, 2006, doi: 10.1109/MS.2006.157.

[37] C. M. Keet and A. Ławrynowicz, "Test-Driven Development of Ontologies," in Lecture Notes in Computer Science, The Semantic Web. Latest Advances and New Domains, H. Sack, E. Blomqvist, M. d'Aquin, C. Ghidini, S. P. Ponzetto, and C. Lange, Eds., Cham: Springer International Publishing, 2016, pp. 642–657.

[38] K. Davies, C. M. Keet, and A. Lawrynowicz, "More Effective Ontology Authoring with Test-Driven Development and the TDDonto2 Tool," Int. J. Artif. Intell. Tools, vol. 28, no. 7, 2019, doi: 10.1142/S0218213019500234.

[39] T. Slaats, S. Debois, and T. Hildebrandt, "Open to Change: A Theory for Iterative Test-Driven Modelling," in Lecture Notes in Computer Science, Business Process Management, M. Weske, M. Montali, I. Weber, and J. Vom Brocke, Eds., Cham: Springer International Publishing, 2018, pp. 31–47.

[40] K. Beck, Test-Driven Development: By Example, 20th ed. Boston: Addison-Wesley, 2015.

[41] D. Fucci, H. Erdogmus, B. Turhan, M. Oivo, and N. Juristo, "A Dissection of the Test-Driven Development Process: Does It Really Matter to Test-First or to Test-Last?," IIEEE Trans. Software Eng., vol. 43, no. 7, pp. 597–614, 2017, doi: 10.1109/tse.2016.2616877.

[42] L. Williams, E. M. Maximilien, and M. Vouk, "Test-driven development as a defect-reduction practice," in Proceedings of the 14th ISSRE, Denver, Colorado, USA, 2003, pp. 34–45.

[43] D. Janzen and H. Saiedian, "Test-driven development concepts, taxonomy, and future direction," Computer, vol. 38, no. 9, pp. 43–50, 2005, doi: 10.1109/MC.2005.314.

[44] R. S. Sangwan and P. A. Laplante, "Test-Driven Development in Large Projects," IT Prof., vol. 8, no. 5, pp. 25–29, 2006, doi: 10.1109/MITP.2006.122.

[45] M. Shahin, M. Ali Babar, and L. Zhu, "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices," IEEE Access, vol. 5, pp. 3909–3943, 2017, doi: 10.1109/ACCESS.2017.2685629.

[46] M. Karlesky, G. Williams, W. Bereza, and M. Fletcher, "Mocking the Embedded World: Test-Driven Development, Continuous Integration, and Design Patterns," in Embedded Systems Conference, San Jose, California, USA, 2007.

[47] S. Fosso Wamba, M. N. C. Angéla, and E. E. J. Bosco, "Big Data, the Internet of Things, and Smart City Research: A Literature Review and Research Agenda," in Lecture Notes in Electrical Engineering, EAI International Conference on Technology, Innovation, Entrepreneurship and Education, A. Reyes-Munoz, P. Zheng, D. Crawford, and V. Callaghan, Eds., Cham: Springer International Publishing, 2019, pp. 263–276.

[48] Elsevier, Scopus® Expertly curated abstract & citation database. [Online]. Available: https://www.elsevier.com/solutions/scopus (accessed: Mar. 31 2021).

[49] R. Shrivastava, K. Samdanis, and V. Sciancalepore, "Towards service-oriented soft spectrum slicing for 5G TDD networks," Journal of Network and Computer Applications, vol. 137, pp. 78–90, 2019, doi: 10.1016/j.jnca.2019.01.009.

[50] M. F. Aniche and M. A. Gerosa, "Most Common Mistakes in Test-Driven Development Practice: Results from an Online Survey with Developers," in 2010 Third International Conference on Software Testing, Verification, and Validation Workshops, Paris, France, 2010, pp. 469–478.

[51] A. Causevic, D. Sundmark, and S. Punnekkat, "Factors Limiting Industrial Adoption of Test Driven Development: A Systematic Review," in 2011 Fourth IEEE International Conference on Software Testing, Verification and Validation, Berlin, Germany, 2011, pp. 337–346.

[52] N. Yahya and N. S. Awang Abu Bakar, "Test driven development contribution in universities in producing quality software: A systematic review," in The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M), Kuching, Malaysia, 2014, pp. 1–6.

[53] R. Swamidurai, B. Dennis, and U. Kannan, "Investigating the impact of peer code review and pair programming on test-driven development," in IEEE SOUTHEASTCON 2014, Lexington, KY, USA, 2014, pp. 1–5.

[54] W. Bissi, A. G. Serra Seca Neto, and M. C. F. P. Emer, "The effects of test driven development on internal quality, external quality and productivity: A systematic review," Information and Software Technology, vol. 74, pp. 45–54, 2016, doi: 10.1016/j.infsof.2016.02.004.

[55] A. Nanthaamornphong and J. C. Carver, "Test-Driven Development in scientific software: a survey," Software Qual J, vol. 25, no. 2, pp. 343–372, 2017, doi: 10.1007/s11219-015-9292-4.

[56] R. Jeffries and G. Melnik, "Guest Editors' Introduction: TDD--The Art of Fearless Programming," IEEE Softw., vol. 24, no. 3, pp. 24–30, 2007, doi: 10.1109/MS.2007.75.

[57] P. Sfetsos and I. Stamelos, "Empirical Studies on Quality in Agile Practices: A Systematic Literature Review," in 2010 Seventh International Conference on the Quality of Information and Communications Technology, Porto, Portugal, 2010, pp. 44–53.

[58] S. Kollanus, "Test-Driven Development - Still a Promising Approach?," in 2010 Seventh International Conference on the Quality of Information and Communications Technology, Porto, Portugal, 2010, pp. 403–408.

[59] A. Causevic, D. Sundmark, and S. Punnekkat, "An Industrial Survey on Contemporary Aspects of Software Testing," in 2010 Third International Conference on Software Testing, Verification, and Validation Workshops, Paris, France, 2010, pp. 393–401.

[60] Y. Rafique and V. B. Misic, "The Effects of Test-Driven Development on External Quality and Productivity: A Meta-Analysis," IIEEE Trans. Software Eng., vol. 39, no. 6, pp. 835–856, 2013, doi: 10.1109/TSE.2012.28.

[61] H. Munir, M. Moayyed, and K. Petersen, "Considering rigor and relevance when evaluating test driven development: A systematic review," Information and Software Technology, vol. 56, no. 4, pp. 375–394, 2014, doi: 10.1016/j.infsof.2014.01.002.