On the Challenges of Applying Test Driven Development to the Engineering of Big Data Applications

Daniel Staegemann[1][a], Matthias Volk[1][b], Mohammad Abdallah[2][c] and Klaus Turowski[1][d]
[1]*Magdeburg Research and Competence Cluster VLBA, Otto-von-Guericke University Magdeburg, Magdeburg, Germany*
[2]*Department of Software Engineering, Al-Zaytoonah University of Jordan, Amman, Jordan*

Keywords: Big Data, Big Data Engineering, Test Driven Development, TDD, Testing, Quality Assurance.

Abstract: Big data (BD) is one of the major technological trends of today and finds application in numerous domains and contexts. However, while there are huge potential benefits, there are also considerable challenges. One of these is the difficulty to make sure the respective applications have the necessary quality. For this purpose, the application of test driven development (TDD) to the domain was proposed. In general, the approach already has a rather long history and, thereby, the corresponding challenges are also known. However, since the BD domain has several demanding particularities, this also needs to be accounted for when applying TDD. Yet, to our knowledge, this specific aspect has not been discussed by now. The publication at hand bridges this gap by examining the challenges of applying TDD to the engineering of BD applications. In doing so, it facilitates the approach's use by practitioners and researchers while also constituting a foundation for further discourse regarding the quality assurance in the BD realm and the TDD approach in general.

1 INTRODUCTION

Big data (BD) is one of the major technological trends of today and finds application in numerous domains and contexts (Volk et al. 2020b). However, while there are huge benefits to gain (Al-Sai et al. 2022; Günther et al. 2017; Müller et al. 2018), there are also considerable challenges. One of these is the difficulty to make sure the respective systems have the necessary quality.

One rather recent proposition in that direction was the application of test driven development (TDD) in the BD domain by Staegemann et al. (2020b). While the potential benefits seem promising, it is also necessary to pay attention to the challenges that come with the intersection of both domains. And, although the challenges of applying TDD in general have already been discussed in the literature (Staegemann et al. 2022a), this has, to the best of our knowledge, not yet been done with an explicit focus on the application of TDD in the BD domain. However, due to the specificities of BD systems in comparison to non-BD applications, there might not only be similarities, but also some aspects that deviate regarding the challenges when applying TDD. To explore this aspect, the publication at hand aims to answer the following research question (RQ).

RQ: *Which are the major challenges when applying test driven development to the engineering of big data applications?*

In order to provide an answer to the RQ, the publication is structured as follows. Succeeding this introduction, the most important concepts for the understanding of this work are briefly outlined in the background section. Afterwards, the challenges of applying TDD specifically to the engineering of BD applications are discussed. Finally, a conclusion of the work is given.

2 BACKGROUND

To answer the RQ, it is at first necessary to have an understanding of the involved topic. Therefore, in the following, the necessary concepts are briefly outlined.

[a] https://orcid.org/0000-0001-9957-1003
[b] https://orcid.org/0000-0002-4835-919X
[c] https://orcid.org/0000-0002-3643-0104
[d] https://orcid.org/0000-0002-4388-8914

Staegemann, D., Volk, M., Abdallah, M. and Turowski, K.
On the Challenges of Applying Test Driven Development to the Engineering of Big Data Applications.
DOI: 10.5220/0012087600003552
In *Proceedings of the 20th International Conference on Smart Business Technologies (ICSBT 2023)*, pages 129-135
ISBN: 978-989-758-667-5; ISSN: 2184-772X

2.1 Big Data

Due to the continuous increase of society's data production (Dobre and Xhafa 2014; Yin and Kaynak 2015) and the coincident desire to make use of it, the concept of BD has become highly significant for the operations of many organizations (Al-Sai et al. 2022; Ghasemaghaei and Calic 2020; Oussous et al. 2018; Volk et al. 2020b). Hence, the corresponding discourse by scientists and practitioners is also very active (Staegemann et al. 2019; Yasmin et al. 2020). Yet, by now, a universally used definition for the term was still not found. Instead, there are many slightly varying explanations (Volk et al. 2022).

One of the most popular ones was provided by the National Institute of Standards and Technology (NIST), which states that BD "*consists of large datasets that primarily exhibit the characteristics of volume, velocity, variety, and/or variability and require a scalable architecture for efficient storage, processing, and analysis*" (Chang and Grady 2019).

Here, the volume addresses how many files have to be processed and/or how big in size those are (Russom 2011). Velocity indicates the speed at which data is incoming as well as how fast the results have to be delivered (Gandomi and Haider 2015). The third characteristic, variety, refers to the changing types of structure, formatting, context, and content the data can have (Gani et al. 2016). Lastly, the alteration of the data's above discussed characteristics over time is expressed through the variability (Katal et al. 2013; Wu et al. 2014).

However, while these are arguably the most common characteristics, they are not the only ones and many more can be found in the literature, highlighting the concept's complexity (Volk et al. 2020b).

2.2 Test Driven Development

Commonly, when developing software, features that are supposed to be realized are first designed, then implemented, and afterwards tested. However, when applying TDD, this order is modified. Yet, it also starts with the design. Though, it is slightly altered because the corresponding functionality is fragmented into preferably small portions (Fucci et al. 2017). Then, the tests for these are written and subsequently executed with the expectation to fail, because the corresponding functionality was not yet implemented (Beck 2015). However, if they would pass, this would show that they do not cover any new functionality and have to be revisited. Once the tests are done, the actual code for fulfilling the functionality is written. At this stage, the goal is only to create working code that provides the desired capability and passes the tests, other factors like its adherence to certain conventions or its elegance are only an afterthought (Crispin 2006). Following that, the tests are run and if the code passes them, it can be refactored to improve its quality (Beck 2015). While doing so, the tests can be used to make sure that no errors are introduced during the refactoring. While this approach naturally leads to a high test coverage and short test cycles (Janzen and Saiedian 2005), it also influences the developed artifact's design through its emphasize on incremental changes to progress (Williams et al. 2003). As a result of this, unit tests are a highly important pillar of TDD. However, often times they are supplemented with other test types such as integration tests or system tests (Sangwan and Laplante 2006).

Overall, as a consequence of the strong modularization, the complexity of the distinct parts is reduced, the likelihood of implementation errors is minimized, and the system's maintainability is increased (Crispin 2006; Shull et al. 2010). Consequently, the application of the approach is associated with an increase in quality, which is, however, at the expense of a reduced implementation speed, as the corresponding literature highlights (Staegemann et al. 2021a).

2.3 Test Driven Development in Big Data Engineering

As discussed earlier, as an alternative to the common approach of engineering BD applications (Volk et al. 2020a), it has also been proposed to transfer the TDD method to the BD domain (Staegemann et al. 2020b). For this purpose, the use of microservices has been suggested (Staegemann et al. 2020b), since they are perfectly suited for breaking the envisioned application into small parts. Moreover, they are already widely used in the BD context (Staegemann et al. 2021b). By using microservices, each business functionality can be implemented as a separate service, allowing for independent scaling. Further, also the implementation process itself is affected because this approach allows to distribute the development of the services across different teams, which can each choose their preferred tools, environments, and languages for their respective task. Moreover, because the existing tests can be used to validate them, TDD also makes it easier and less risky to implement changes to the developed application. Consequently, developers get faster feedback, newly introduced errors can be minimized, and the users' confidence in the solution can be increased, potentially helping the application's acceptance.

3 THE CHALLENGES OF APPLYING TDD TO BD

While the utilization of TDD for the development of BD applications promises several benefits (Staegemann et al. 2020b), it also doesn't come without challenges. For the general application, these have already been discussed in multiple contributions, as an overview of the topic shows (Staegemann et al. 2022a). These are, regarding the involved people, mainly a lack of knowledge and experience by the developers (Buchan et al. 2011; Causevic et al. 2011; Causevic et al. 2013; Karac and Turhan 2018; Latorre 2014; Nanthaamornphong and Carver 2017), difficulties in shifting to the TDD mindset (Baldassarre et al. 2022; Causevic et al. 2013; Hammond and Umphress 2012; Kollanus 2011; Marchenko et al. 2009), and senior-level management's insufficient understanding of TDD (Buchan et al. 2011; Causevic et al. 2013). Further, it is tempting to create the tests in a way that they highlight what works, instead of actively looking for potential issues (Causevic et al. 2013). The envisioned application's initial design is often too poorly planned (Causevic et al. 2013; Hammond and Umphress 2012; Karac and Turhan 2018; Kollanus 2011), the existence of legacy code is not sufficiently considered in the TDD approach (Causevic et al. 2011), and there is obviously a necessity to create huge volumes of test code (Causevic et al. 2013). Moreover, a lack of suitable tools for test creation (Causevic et al. 2013; Kollanus 2011; Nanthaamornphong and Carver 2017) and the high technical complexity of TDD's application in certain scenarios (e.g., GUI development) are noteworthy (Causevic et al. 2011; Causevic et al. 2013; Marchenko et al. 2009).

However, while those factors mostly also apply to an application in the BD domain, several of its specificities must be considered for the proposed TDD approach (Staegemann et al. 2020b). One of the major points is oftentimes the sheer scale of the systems, respectively the scalability, that has to be reflected by the tests (Davoudian and Liu 2021; Qin and Zhou 2013). Another one is the fact that they comprise many different microservices that are somewhat independent of each other but still connected in potentially complicated ways. Consequently, just testing the parts is insufficient since the communication between the services is also a potential source of errors. However, only having an end-to-end testing is also not enough because it would not allow to exactly determine an error's cause. Hence, both aspects need to be incorporated (Davoudian and Liu 2021; Han et al. 2018; Qin and Zhou 2013). Therefore, in comparison to most TDD endeavours, many more tests are needed to achieve comprehensiveness. This, in turn, exacerbates the issue that was already identified for TDD in common software engineering and might also deter leadership from facilitating its application.

Further, the diversity of functionalities, who's combination often forms a BD application, also increases the complexity of the task. This includes, inter alia, the use of machine learning techniques, ETL processes, statistical computations, or complex visualizations, which all have to be tested and can require vastly different approaches for doing so (Braiek and Khomh 2020; Nwokeji et al. 2018; Zhang et al. 2017). Hence, the complexity of using TDD for BD development will often be higher in comparison to traditional software because a wider spectrum of capability types has to be covered.

Another source of heterogeneity emerges, when the flexibility regarding the choice of frameworks and programming languages, which is one of the advantages of the microservice based approach, is actually extensively exploited to pursue the best-in-class idea. Because in this scenario, it is not only necessary to know numerous solutions for realizing the envisioned application but also to have sufficient knowledge of the respective testing frameworks, which might greatly increase the associated demands on the developers (Staegemann et al. 2022b). Moreover, while TDD in general already has the issue of somewhat insufficient tool support, this is naturally further exacerbated in the BD domain, where the corresponding test tools also need to be capable to handle the BD characteristics, further reducing the number of available choices (Davoudian and Liu 2021). Additionally, taking the other perspective, some BD tools might also be excluded from use because they are less suitable for TDD due to their design or interfaces (Staegemann et al. 2022b).

Another considerable challenge that is more severe in the TDD of BD systems than for traditional ones is the oracle problem. Due to the nature of the applications and the inherent uncertainty of many tasks, it is often not known how exactly the perfect output looks like (Chen et al. 2019; Davoudian and Liu 2021). Consequently, it is also not possible to use it as a reference to compare the obtained results against it (Staegemann et al. 2019). Thus, it will often be necessary to settle for tests that assure the functionality only with a certain confidence (Chen et al. 2019). In turn, this also means that it is required to determine what exactly qualifies as a passed test and what is a failed one, in these scenarios. However, this is heavily depending on the respective use case and cannot be generalized.

As a consequence of the previously described complexity and uncertainty, having some degree of fault tolerance might be a valuable or even necessary property for certain applications. This can be assessed through tests that intentionally inject failures to see how the system reacts (Qin and Zhou 2013).

Moreover, besides just testing with respect to the pure functionality, it might also be beneficial to incorporate benchmarking into the test setup. This could, in addition to rather traditional metrics (e.g., processing speed), also account for other aspects such as, for instance, energy consumption (Qin and Zhou 2013).

Another major point that differentiates the BD domain from more traditional software, when it comes to TDD, is the cost aspect. Usually in TDD, the old tests can be rerun without worrying about the incurring costs. However, for BD applications, the testing is potentially extremely resource-intensive (Davoudian and Liu 2021). Thereby, it might necessitate the provisioning of extra server capacities or the application might even entirely run in a chargeable cloud environment with usage-depending fees, leading to additional costs for each test run (Staegemann et al. 2022b). Thus, if possible, their number should be reduced without compromising the thoroughness of the process. Hence, sensible planning is necessary to find a feasible strategy that fits the respective use case. One proposition to reduce the extent of the required retesting is to determine logical entities that can be tested when a service inside of them is changed without the need to test the parts outside of the concerned components (Staegemann et al. 2020b). This seems feasible to make sure the system as a whole still works properly but the application of this strategy for individual cases is still challenging because it requires a good segmentation plan on top of the actual test creation.

One issue with regards to maintainability and clarity comes with the freedom provided by the MS architecture. Since the development teams can have more independence from each other, there can also be many different tools, languages, frameworks, coding styles, naming conventions, etc. in use, making it harder to understand what has been done when revisiting some parts in the future.

Finally, when using TDD in the BD domain, it might also be necessary to somewhat deviate from guidelines that are focused on TDD in general (Staegemann et al. 2022c). While it is usually advised to focus on the current requirements instead of trying to foresee potential future necessities (Guerra and Aniche 2016), in BD these can sometimes be seen as part of the requirements when it comes to, e.g., increasing user traffic over time, saisonal or event-driven usage peaks, or shifts regarding the relevant data. Hence, accounting for growth or changing circumstances seems reasonable (Staegemann et al. 2020a). This, however, also has to be reflected by the tests.

4 CONCLUSION

BD applications play an important role in today's society and are widely utilized. Therefore, the corresponding quality assurance is a relevant topic as well, which also shows in the scientific discourse. One rather recent proposition was the application of TDD in the BD domain. However, while this seems promising, there are of course also challenges associated with it. Some of these are linked to TDD in general, whereas others specifically result from its application in BD engineering. While the former ones are also mentioned, the publication at hand is primarily focused on the exploration of the second group. By highlighting them, prospective developers of BD applications are pointed towards aspects to consider when deciding if TDD should be pursued. Further, once the decision is made, being aware of the challenges is also a necessary step to be able to effectively deal with them. Moreover, the identified challenges can also be seen as a call for action and starting point for other researchers to base their future endeavors on.

REFERENCES

Al-Sai, Z. A., Husin, M. H., Syed-Mohamad, S. M., Abdin, R. M.'d. S., Damer, N., Abualigah, L., and Gandomi, A. H. (2022). "Explore Big Data Analytics Applications and Opportunities: A Review," Big Data and Cognitive Computing (6:4), p. 157 (doi: 10.3390/bdcc6040157).

Baldassarre, M. T., Caivano, D., Fucci, D., Romano, S., and Scanniello, G. (2022). "Affective reactions and test-driven development: Results from three experiments and a survey," *Journal of Systems and Software* (185) (doi: 10.1016/j.jss.2021.111154).

Beck, K. (2015). *Test-Driven Development*: *By Example*, Boston: Addison-Wesley.

Braiek, H. B., and Khomh, F. (2020). “On testing machine learning programs,” *Journal of Systems and Software* (164), p. 110542 (doi: 10.1016/j.jss.2020.110542).

Buchan, J., Li, L., and MacDonell, S. G. (2011). “Causal Factors, Benefits and Challenges of Test-Driven Development: Practitioner Perceptions,” in *Proceedings of the 2011 18th Asia-Pacific Software Engineering Conference,* Ho Chi Minh, Vietnam. 05.12.2011 - 08.12.2011, IEEE, pp. 405-413 (doi: 10.1109/APSEC.2011.44).

Causevic, A., Punnekkat, S., and Sundmark, D. (2013). “TDDHQ: Achieving Higher Quality Testing in Test Driven Development,” in *Proceedings of the 39th Euromicro Conference on Software Engineering and Advanced Applications,* Santander, Spain. 04.09. 2013 - 06.09.2013, IEEE, pp. 33-36 (doi: 10.1109/SEAA.2013.47).

Causevic, A., Sundmark, D., and Punnekkat, S. (2011). “Factors Limiting Industrial Adoption of Test Driven Development: A Systematic Review,” in *Proceedings of the 2011 Fourth IEEE International Conference on Software Testing, Verification and Validation,* Berlin, Germany. 21.03.2011 - 25.03.2011, IEEE, pp. 337-346 (doi: 10.1109/ICST.2011.19).

Chang, W. L., and Grady, N. (2019). “NIST Big Data Interoperability Framework: Volume 1, Definitions,” *Special Publication (NIST SP),* Gaithersburg, MD: National Institute of Standards and Technology.

Chen, T. Y., Kuo, F.-C., Liu, H., Poon, P.-L., Towey, D., Tse, T. H., and Zhou, Z. Q. (2019). “Metamorphic Testing,” *ACM Computing Surveys* (51:1), pp. 1-27 (doi: 10.1145/3143561).

Crispin, L. (2006). “Driving Software Quality: How Test-Driven Development Impacts Software Quality,” *IEEE Software* (23:6), pp. 70-71 (doi: 10.1109/MS.2006.157).

Davoudian, A., and Liu, M. (2021). “Big Data Systems,” *ACM Computing Surveys* (53:5), pp. 1-39 (doi: 10.1145/3408314).

Dobre, C., and Xhafa, F. (2014). “Intelligent services for Big Data science,” *Future Generation Computer Systems* (37), pp. 267-281 (doi: 10.1016/j.future.2013.07.014).

Fucci, D., Erdogmus, H., Turhan, B., Oivo, M., and Juristo, N. (2017). “A Dissection of the Test-Driven Development Process: Does It Really Matter to Test-First or to Test-Last?” *IEEE Transactions on Software Engineering* (43:7), pp. 597-614 (doi: 10.1109/tse.2016.2616877).

Gandomi, A., and Haider, M. (2015). “Beyond the hype: Big data concepts, methods, and analytics,” *International Journal of Information Management* (35:2), pp. 137-144 (doi: 10.1016/j.ijinfomgt. 2014.10.007).

Gani, A., Siddiqa, A., Shamshirband, S., and Hanum, F. (2016). “A survey on indexing techniques for big data: taxonomy and performance evaluation,” *Knowledge and Information Systems* (46:2), pp. 241-284 (doi: 10.1007/s10115-015-0830-y).

Ghasemaghaei, M., and Calic, G. (2020). “Assessing the impact of big data on firm innovation performance: Big data is not always better data,” *Journal of Business Research* (108:2), pp. 147-162 (doi: 10.1016/j. jbusres.2019.09.062).

Guerra, E., and Aniche, M. (2016). “Achieving quality on software design through test-driven development,” in *Software Quality Assurance*, Elsevier, pp. 201-220 (doi: 10.1016/B978-0-12-802301-3.00009-0).

Günther, W. A., Rezazade Mehrizi, M. H., Huysman, M., and Feldberg, F. (2017). “Debating big data: A literature review on realizing value from big data,” *The Journal of Strategic Information Systems* (26:3), pp. 191-209 (doi: 10.1016/j.jsis.2017.07.003).

Hammond, S., and Umphress, D. (2012). “Test driven development: the state of the practice,” in *Proceedings of the 50th Annual Southeast Regional Conference on - ACM-SE '12*, R. K. Smith and S. V. Vrbsky (eds.), Tuscaloosa, Alabama. 29.03.2012 - 31.03.2012, New York, New York, USA: ACM Press, p. 158 (doi: 10.1145/2184512.2184550).

Han, R., John, L. K., and Zhan, J. (2018). “Benchmarking Big Data Systems: A Review,” *IEEE Transactions on Services Computing* (11:3), pp. 580-597 (doi: 10.1109/TSC.2017.2730882).

Janzen, D., and Saiedian, H. (2005). “Test-driven development concepts, taxonomy, and future direction,” *Computer* (38:9), pp. 43-50 (doi: 10.1109/MC.2005. 314).

Karac, I., and Turhan, B. (2018). “What Do We (Really) Know about Test-Driven Development?” *IEEE Software* (35:4), pp. 81-85 (doi: 10.1109/MS.2018. 2801554).

Katal, A., Wazid, M., and Goudar, R. H. (2013). “Big data: Issues, challenges, tools and Good practices,” in *Sixth International Conference on Contemporary Computing*, Parashar (ed.), Noida, India. 08.08.2013 - 10.08.2013, IEEE, pp. 404-409 (doi: 10.1109/IC3.2013.6612229).

Kollanus, S. (2011). “Critical Issues on Test-Driven Development,” in *Product-Focused Software Process Improvement*, D. Caivano, M. Oivo, M. T. Baldassarre and G. Visaggio (eds.), Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 322-336 (doi: 10.1007/978-3-642-21843-9_25).

Latorre, R. (2014). “A successful application of a Test-Driven Development strategy in the industrial environment,” *Empirical Software Engineering* (19:3), pp. 753-773 (doi: 10.1007/s10664-013-9281-9).

Marchenko, A., Abrahamsson, P., and Ihme, T. (2009). “Long-Term Effects of Test-Driven Development A Case Study,” in *Agile Processes in Software Engineering and Extreme Programming*, P. Abrahamsson, M. Marchesi and F. Maurer (eds.), Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 13-22 (doi: 10.1007/978-3-642-01853-4_4).

Müller, O., Fay, M., and Vom Brocke, J. (2018). “The Effect of Big Data and Analytics on Firm Performance: An Econometric Analysis Considering Industry Characteristics,” *Journal of Management Information Systems* (35:2), pp. 488-509 (doi: 10.1080/07421222. 2018.1451955).

Nanthaamornphong, A., and Carver, J. C. (2017). "Test-Driven Development in scientific software: a survey," *Software Quality Journal* (25:2), pp. 343-372 (doi: 10.1007/s11219-015-9292-4).

Nwokeji, J., Aqlan, F., Anugu, A., and Olagunju, A. (2018). "Big Data ETL Implementation Approaches: A Systematic Literature Review (P)," in *Proceedings of the 30th International Conference on Software Engineering and Knowledge Engineering.* 01.07.2018 - 03.07.2018, KSI Research Inc. and Knowledge Systems Institute Graduate School, pp. 714-721 (doi: 10.18293/SEKE2018-152).

Oussous, A., Benjelloun, F.-Z., Ait Lahcen, A., and Belfkih, S. (2018). "Big Data technologies: A survey," *Journal of King Saud University - Computer and Information Sciences* (30:4), pp. 431-448 (doi: 10.1016/j.jksuci.2017.06.001).

Qin, X., and Zhou, X. (2013). "A Survey on Benchmarks for Big Data and Some More Considerations," in *Intelligent Data Engineering and Automated Learning – IDEAL 2013*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, H. Yin, K. Tang, Y. Gao, F. Klawonn, M. Lee, T. Weise, B. Li and X. Yao (eds.), Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 619-627 (doi: 10.1007/978-3-642-41278-3_75).

Russom, P. (2011). "Big Data Analytics: TDWI Best Practices Report Fourth Quarter 2011,"

Sangwan, R. S., and Laplante, P. A. (2006). "Test-Driven Development in Large Projects," *IT Professional* (8:5), pp. 25-29 (doi: 10.1109/MITP.2006.122).

Shull, F., Melnik, G., Turhan, B., Layman, L., Diep, M., and Erdogmus, H. (2010). "What Do We Know about Test-Driven Development?" *IEEE Software* (27:6), pp. 16-19 (doi: 10.1109/MS.2010.152).

Staegemann, D., Volk, M., Daase, C., and Turowski, K. (2020a). "Discussing Relations Between Dynamic Business Environments and Big Data Analytics," *Complex Systems Informatics and Modeling Quarterly* (23), pp. 58-82 (doi: 10.7250/csimq.2020-23.05).

Staegemann, D., Volk, M., Jamous, N., and Turowski, K. (2020b). "Exploring the Applicability of Test Driven Development in the Big Data Domain," in *Proceedings of the 31st Australasian Conference on Information Systems (ACIS),* Wellington, New Zealand. 01.12.2020 - 04.12.2020.

Staegemann, D., Volk, M., Lautenschlager, E., Pohl, M., Abdallah, M., and Turowski, K. (2021a). "Applying Test Driven Development in the Big Data Domain – Lessons From the Literature," in *2021 International Conference on Information Technology (ICIT),* Amman, Jordan. 14.07.2021 - 15.07.2021, IEEE, pp. 511-516 (doi: 10.1109/ICIT52682.2021.9491728).

Staegemann, D., Volk, M., Nahhas, A., Abdallah, M., and Turowski, K. (2019). "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. 26.11.2019 - 29.11.2019.

Staegemann, D., Volk, M., Perera, M., Haertel, C., Pohl, M., Daase, C., and Turowski, K. (2022a). "A Literature Review on the Challenges of Applying Test-Driven Development in Software Engineering," *Complex Systems Informatics and Modeling Quarterly* (31), pp. 18-28 (doi: 10.7250/csimq.2022-31.02).

Staegemann, D., Volk, M., Perera, M., and Turowski, K. (2022b). "Exploring the Test Driven Development of a Fraud Detection Application using the Google Cloud Platform," in *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management,* Valletta, Malta. 24.10.2022 - 26.10.2022, SCITEPRESS - Science and Technology Publications, pp. 83-94 (doi: 10.5220/0011559000003335).

Staegemann, D., Volk, M., Pohl, M., Haertel, C., Hintsch, J., and Turowski, K. (2022c). "Identifying Guidelines for Test-Driven Development in Software Engineering—A Literature Review," in *Proceedings of Seventh International Congress on Information and Communication Technology*, X.-S. Yang, S. Sherratt, N. Dey and A. Joshi (eds.), Springer Nature Singapore, pp. 327-336 (doi: 10.1007/978-981-19-2397-5_30).

Staegemann, D., Volk, M., Shakir, A., Lautenschläger, E., and Turowski, K. (2021b). "Examining the Interplay Between Big Data and Microservices – A Bibliometric Review," *Complex Systems Informatics and Modeling Quarterly* (27), pp. 87-118 (doi: 10.7250/csimq.2021-27.04).

Volk, M., Staegemann, D., Bosse, S., Häusler, R., and Turowski, K. (2020a). "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. 07.05.2020 - 09.05.2020, SCITEPRESS - Science and Technology Publications, pp. 428-435 (doi: 10.5220/0009569804280435).

Volk, M., Staegemann, D., and Turowski, K. (2020b). "Big Data," in *Handbuch Digitale Wirtschaft*, T. Kollmann (ed.), Wiesbaden: Springer Fachmedien Wiesbaden, pp. 1-18 (doi: 10.1007/978-3-658-17345-6_71-1).

Volk, M., Staegemann, D., and Turowski, K. (2022). "Providing Clarity on Big Data: Discussing Its Definition and the Most Relevant Data Characteristics," in *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management,* Valletta, Malta. 24.10.2022 - 26.10.2022, SCITEPRESS - Science and Technology Publications, pp. 141-148 (doi: 10.5220/0011537500003335).

Williams, L., Maximilien, E. M., and Vouk, M. (2003). "Test-driven development as a defect-reduction practice," in *Proceedings of the 14th ISSRE,* Denver, Colorado, USA. 17.11.2003 - 20.11.2003, IEEE, pp. 34-45 (doi: 10.1109/ISSRE.2003.1251029).

Wu, X., Zhu, X., Wu, G.-Q., and Ding, W. (2014). "Data mining with big data," *IEEE Transactions on*

Knowledge and Data Engineering (26:1), pp. 97-107 (doi: 10.1109/TKDE.2013.109).

Yasmin, M., Tatoglu, E., Kilic, H. S., Zaim, S., and Delen, D. (2020). "Big data analytics capabilities and firm performance: An integrated MCDM approach," *Journal of Business Research* (114), pp. 1-15 (doi: 10.1016/j.jbusres.2020.03.028).

Yin, S., and Kaynak, O. (2015). "Big Data for Modern Industry: Challenges and Trends [Point of View]," *Proceedings of the IEEE* (103:2), pp. 143-146 (doi: 10.1109/JPROC.2015.2388958).

Zhang, P., Zhou, X., Li, W., and Gao, J. (2017). "A Survey on Quality Assurance Techniques for Big Data Applications," in *2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService),* Redwood City, CA, USA. 06.04.2017 - 09.04.2017, IEEE, pp. 313-319 (doi: 10.1109/BigDataService.2017.42).