Agile Machine Learning Processes

Julien Siebert[1], Anna Schmitt[1], Sven Theobald[1], Anna Maria Vollmer[1], and Adam Trendowicz[1]

[1] Fraunhofer-Institute for Experimental Software Engineering IESE, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
`first.lastname@iese.fraunhofer.de`

Abstract. Agile practices have proven to be one of the main success factors in the development of software systems. With the shift to data-driven products and services, machine learning methods and software development processes need to be (and are currently) increasingly integrated. Developing and operating data-driven software components raise new challenges and risks, in terms of communication, organization, as well as skills and technologies. We believe that agile practices have a lot to offer to address these aspects. In this article, we expose a research plan in order to investigate how agile practices can benefit machine learning development and operation processes.

Keywords: Machine Learning, agile software development, process

1 Scope and motivation

The successful transition to the "digital age" and the use of the potential of artificial intelligence (AI) and related data-driven methods, such as machine learning (ML) is not trivial. Technically, this often requires the acquisition of new skills (like data-science and data-engineering). Building data-driven software components entails a number of complexities both at the level of individual tasks (e.g., collection, storage and processing of data, as well as training and validation of models) and at the level of the overall process. The life cycle of data-driven components - including specification, development, deployment and continuous adaptation - is much more complex than for classical software and poses new challenges [1–6]. In addition, AI and data-driven methods have to be adapted to the use cases of the specific company. Even if progresses are made in the field of automated ML, not all steps can be automated [7].

The successful development of data-driven components still requires the knowledge of domain experts and a close cooperation between stakeholders [8]. The experiences of companies implementing projects involving ML show that a success factor is a deep understanding of the data-based products life cycle: 1) to understand what ML engineering changes (for the requirements, the development and operation processes) in comparison to “classical” software engineering, and 2) to understand how to deal with these changes [9, 10]. The challenges that companies face when developing and operating systems using ML are as follows: (1) frequently changing requirements, (2) data silos, (3) non-optimized data formats and data stores, (4) data errors, (5) manual processes, (6) communication problems and different organizations between different stakeholders (e.g., data-scientists and software engineers), (7) the lack of a transition path between the current and the new “ML-aware” processes, and (8) the lack of a starting point and initial analysis for this transition [11, 12].

2 State of the Art und State of the Practice

Most of the current development processes of data-driven products describe very well the different tasks (data preparation, features engineering, model training, etc. see Figure 1) involved in the development and operation of data-driven products. Examples of such process models include, for example, Cross Industry Standard Process for Data Mining (Crisp-DM) and its variations [13–15], Knowledge Discovery in Database (KDD) [16], Sample, Explore, Modify, Model, Assess (SEMMA) [17], IBM Analytics Solutions Unified Method for Data Mining/Predictive Analytics (ASUM) [18], Microsoft Team Data Science Process (TDSP) [19] und Domino Data Labs Domino Data Science Lifecycle (DDSL) [20]. Crisp-DM is so far the most known and most commonly used method for analysis, data mining or data science projects [21, 22]. Furthermore, recent applications of ML in the industry lead to new research works that empirically assess the challenges faced by developers, the corresponding processes used, and the design patterns that emerge when developing or operating systems based on ML [4, 6, 23–26].

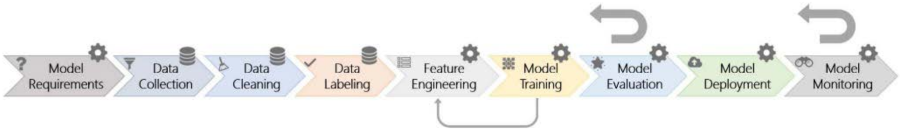

Figure 1: The nine stages of the machine learning workflow. [27]

These process models are usually described as "waterfall models" with long iteration cycles. Furthermore, their abstract description makes it difficult to implement them [15]. Due to the iterative and (potentially long) experimental nature of machine learning, agile processes for (data-science / knowledge engineering / machine learning engineering) are currently explored in order to avoid the “pull of waterfall” [28–34]. For instance, Jurney, in [33], proposes a method to transfer agile software development to data science in the field of web application and web development. One of the problems raised is the mutual lack of understanding between data science teams and engineering teams for each other’s work. In [11], Bergh, Benghiat and Strod found out that the cycle of data analysis and quality can be optimized by combining tools and methods – calling it "DataOps". Larson and Chang [35], describe current trends and the interaction between agile methods and data science. On the tooling side, it is worth noting that new tools are currently developed to tackle the challenges related to DevOps of data-driven software components (such as data version control, CI/CD for ML, or model and workflow management systems [36–40]).

However, to the best of our knowledge, the following questions have been so far seldom investigated: What has changed in processes and companies when ML was applied? Which state was reached after implementation? Which practices were applied and how? What was successful? What did not work? The only related work we found conducted a controlled experiment with students to compare four approaches in a data science project [13].

3 Objectives

Our main objective is, on the one hand, to empirically assess whether existing agile methods (e.g., Scrum [41, 42]) or agile best practices (e.g., Sprint planning, Retrospectives) can help, or are currently helping, solving the current challenges faced in the development and the operation of software systems using ML. On the other hand, we want to analyze how the best practices and design patterns currently in use during the development and operation of ML systems can be integrated in an agile lifecycle and if and how those best practices need adaptation to fit in. We want to support companies both with a methodological approach (e.g., a review of agile ML process models, like in [28–34] and a set of appropriate and experimentally approved practices) and with a supporting interactive tool that can help in setting up or improving a company-specific agile process targeting ML. We expect the proposed methods and tools to help enterprises, regardless of their experience with ML, to introduce an efficient process or to improve existing processes when developing and/or operating systems including ML.

4 Methods and research directions

In order to satisfy our objectives, we plan to pursue the following research. First, we are currently doing a systematic literature review concerning the software engineering challenges related to the development and operation of systems using ML. The expected goals of this literature review are 1) to extract the current challenges faced during development and operation of systems based on ML, and 2) to classify the different challenges. We foresee a classification along two main axes: the types of challenges encountered (e.g., communication, skills, technology, organization), and in which phase these challenges appear (Requirement/Business understanding, Data preparation, Modeling & Evaluation, Deployment). In previous research, we identified different agile practices and the challenges (or goals) they are addressing (e.g., what agile practices address the goal “product quality” [43]). We plan to extend this with a literature review in order to extract specific agile practices related to ML (for example from [28–34]). Based on those inputs, we plan to construct a catalogue of agile and ML best practices.

Second, we compare the challenges when developing ML products with the challenges addressed by agile practices, in order to identify suitable agile practices to cope with the ML challenges. For gaps regarding this mapping, we plan either to adapt existing or to propose new practices (for example extending Code review to Data review and Model review).

In a third step, we plan to study existing agile processes dedicated to ML, and when needed extend them. We plan to experimentally assess the validity of the proposed agile practices and process models through case studies. We will reflect on which agile practices can be used for the development of ML, also considering necessary adaptations to existing agile practices. In parallel, we assess whether ML best practices can be aligned within an agile life cycle (e.g., the Scrum approach) and what adaptations need to be made in this case. In cases a close collaboration is not possible, we also want to think about the process interface between ML and software development, and towards other relevant stakeholders [44].

Last, we plan to concretize these results into a web-based supporting tool, that allows for interactively selecting and linking suitable components from the catalogue of both agile and ML practices (e.g. data version control, ML workflow management, ML model monitoring etc.), while taking into account the company-specific goals and process conditions. Similar to [45], an individual approach for an agile ML process can be configured, and single practices can be adopted in an iterative way to incrementally become more agile while developing ML products. In this way, we expect to efficiently derive a specific way to implement a new ML process or specific improvement measures for an existing process. Data scientists, software engineers, and product owners can jointly find suggestions for changing and improving their communication, the coordination and the organization of their work.

5 Conclusion

The recent popularity of machine learning has led to new challenges in the way software is built, and how organizations adapt themselves in order to integrate this technology in their current processes. Recent years have seen the publication of new empirical studies, and one can guess that more studies will be carried out in the years to come. It should be noted that the challenges raised by the application of artificial intelligence (AI) methods in software products is by itself not new. Already during the second wave of AI, similar questions were raised [46]. It is also interesting to note that in 1988 (when AI was mostly synonymous of expert systems), the authors of [47] noted that "Software Engineering (SE) is synonymous with the waterfall model, and the waterfall model is linear and hence not suitable for AI" (*sic*), and that, nowadays (almost 20 years after the publication of the Agile Manifesto [48]), most process models describing the development of data-driven software components are often seen as “waterfall” ones. We believe that agile practices can help solving some challenges faced when developing and operating software using ML, and we hope that, by specifically targeting data-driven methods such as ML, agile practices can be enriched.

This work highlighted the need to research on the combination of agile and ML, and proposed a research plan to investigate how agile practices can support efficient development of ML products.

References

1. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., Dennison, D.: Hidden Technical Debt in Machine Learning Systems. In:, pp. 2503–2511 (2015)
2. Wan, Z., Xia, X., Lo, D., Murphy, G.C.: How does Machine Learning Change Software Development Practices? IEEE Transactions on Software Engineering, 1 (2019)
3. Polyzotis, N., Roy, S., Whang, S.E., Zinkevich, M.: Data Management Challenges in Production Machine Learning. In: Chirkova, R., Data, A.S.I.G.o.M.o.M. (eds.) Proceedings of the 2017 ACM International Conference on Management of Data, pp. 1723–1726. ACM, [Place of publication not identified] (2017)
4. Bosch, J., Olsson, H.H., Crnkovic, I., Wang X., Munch J., Suominen A., Bosch J., Jud C., Hyrynsalmi S.: It takes three to tango: Requirement, outcome/data, and AI driven development. CEUR Workshop Proceedings 2305 (2018)
5. Belani, H., Vukovic, M., Car, Z.: Requirements Engineering Challenges in Building AI-Based Complex Systems. In: 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), pp. 252–255. IEEE (2019 - 2019)
6. Lwakatare, L.E., Raj, A., Bosch, J., Olsson, H.H., Crnkovic, I.: A taxonomy of software engineering challenges for machine learning systems: An empirical investigation. Lecture Notes in Business Information Processing 355, 227–243 (2019)
7. Tuggener, L., Amirian, M., Rombach, K., Lorwald, S., Varlet, A., Westermann, C., Stadelmann, T.: Automated Machine Learning in Practice: State of the Art and Recent Results, 31–36 (2019)
8. Heidrich, J., Trendowicz, A., Ebert, C.: Exploiting Big Data's Benefits. IEEE Softw. 33, 111–116 (2016)
9. Bernardi, L., Mavridis, T., Estevez, P.: 150 Successful Machine Learning Models. In: Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., Karypis, G. (eds.) KDD2019. Anchorage, Alaska, USA, pp. 1743–1751. Association for Computing Machinery, New York, NY (2019)
10. Saltz, J.S., Shamshurin, I.: Big data team process methodologies: A literature review and the identification of key factors for a project's success. In: Joshi, J.B.D. (ed.) Proceedings, 2016 IEEE International Conference on Big Data. Dec 05-Dec 08, 2015, Washington D.C., USA, pp. 2872–2879. IEEE, [Piscataway, New Jersey] (2016)
11. Christopher Bergh, Gil Benghiat, Eran Strod: The DataOps Cookbook. Methodologies and Tools that Reduce Analytics Cycle Time While Improving Quality
12. Jackson, S., Yaqub, M., Li, C.-X.: The Agile Deployment of Machine Learning Models in Healthcare. Front. Big Data 1 (2019)
13. Shearer, C.: The CRISP-DM Model: The New Blueprint for Data Mining. Journal of Data Warehousing 5, 14–22 (2000)
14. Studer, S., Bui, T.B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., Mueller, K.-R.: Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology (2020)

15. Saltz, J., Shamshurin, I., Crowston, K.: Comparing Data Science Project Management Methodologies via a Controlled Experiment. In: Proceedings of the 50th Hawaii International Conference on System Sciences (2017). Hawaii International Conference on System Sciences (2017)
16. Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth: From Data Mining to Knowledge Discovery in Databases
17. SAS Institute Inc.: Data Mining Using SAS® Enterprise Miner™: A Case Study Approach, Fourth Edition
18. Analytic Solutions Unified Method. Implementation with Agile Principles
19. Microsoft: Team Data Science Process Documentation, https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/
20. Domino Data Lab, I.: The Practical Guide to Managing Data Science at Scale. Lessons from the field on managing data science projects and portfolios. Whitepaper (2017)
21. Mariscal, G., Marbán, Ó., Fernández, C.: A survey of data mining and knowledge discovery process models and methodologies. The Knowledge Engineering Review 25, 137–166 (2010)
22. CRISP-DM, still the top methodology for analytics, data mining, or data science projects, https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html
23. Zhang, J.M., Harman, M., Ma, L., Liu, Y.: Machine Learning Testing: Survey, Landscapes and Horizons. IEEE Transactions on Software Engineering, 1 (2020)
24. Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., Zimmermann, T.: Software Engineering for Machine Learning: A Case Study. In:. IEEE Computer Society (2019)
25. Riungu-Kalliosaari, L., Kauppinen, M., Männistö, T.: What can be learnt from experienced data scientists? A case study (2017)
26. Washizaki, H., Uchida, H., Khomh, F., Gueheneuc, Y.-G.: Studying Software Engineering Patterns for Designing Machine Learning Systems. In: 2019 10th International Workshop on Empirical Software Engineering in Practice. IWESEP 2019 : proceedings : Tokyo, Japan, 13-14 December 2019, pp. 49–495. IEEE Computer Society, Conference Publishing Services, Los Alamitos, CA (2019)
27. Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., Zimmermann, T.: Software Engineering for Machine Learning: A Case Study. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), pp. 291–300 (2019)
28. Schmidt, C., Sun, W.N.: Synthesizing Agile and Knowledge Discovery: Case Study Results. Journal of Computer Information Systems 58, 142–150 (2018)
29. Alnoukari, M., Alzoabi, Z., Hanna, S.: Applying adaptive software development (ASD) agile modeling on predictive data mining applications: ASD-DM methodology. In: Zaman, H.B. (ed.) Cognitive informatics. Bridging natural and artificial knowledge proceedings, International Symposium of Information Technology 2008, Kuala Lumpur Convention Centre, Malaysia, August 26-29, 2008, pp. 1–6. IEEE, Piscataway NJ (2008)
30. Grady, N.W., Payne, J.A., Parker, H. (eds.): Agile big data analytics: AnalyticsOps for data science (2018)

31. do Nascimento, G.S., Oliveira, A.A. de: An Agile Knowledge Discovery in Databases Software Process. In: Xiang, Y., Pathan, M., Tao, X., Wang, H. (eds.) Data and knowledge engineering. Third international conference, ICDKE 2012, Wuyishan, China, November 21-23, 2012 : proceedings / Yang Xiang, Mukaddim Pathan, Xiaohui Tao, Hua Wang (eds.), 7696, pp. 56–64. Springer, Heidelberg (2012)
32. Sharma, V., Stranieri, A., Ugon, J., Vamplew, P., Martin, L.: An Agile Group Aware Process beyond CRISP-DM. In: Unknown (ed.) Proceedings of the International Conference on Compute and Data Analysis, pp. 109–113. ACM, New York, NY (2017)
33. Jurney, R.: Agile Data Science 2.0, 1st Edition. O'Reilly Media, Inc, [S.l.] (2017)
34. Saltz, J., Hotz, N., Wild, D., Stirling, K.: Exploring Project Management Methodologies Used Within Data Science Teams. AMCIS 2018 Proceedings (2018)
35. Larson, D., Chang, V.: A review and future direction of agile, business intelligence, analytics and data science. International Journal of Information Management 36, 700–710 (2016)
36. Kira, A.: Managing Uber's Data Workflows at Scale | Uber Engineering Blog, https://eng.uber.com/managing-data-workflows-at-scale/
37. Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, et al.: Accelerating the Machine Learning Lifecycle with MLflow. IEEE Data Eng. Bull. 41, 39–45 (2018)
38. Apache Airflow Documentation — Airflow Documentation, https://airflow.apache.org/
39. Data Science Version Control System, https://dvc.org/
40. CI/CD for Machine Learning & AI, https://blog.paperspace.com/ci-cd-for-machine-learning-ai/
41. Sutherland, J., Schwaber Ken: The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game (2017)
42. Diebold, P., Theobald, S., Schmitt, A., Schmidt, C.: Weg vom unvollständigen Scrum! Hin zum vollständigeren „Scrum++. In: Volland, A., Engstler, M., Fazal-Baqaie, M., Hanser, E., Linssen, O., Mikusz, M. (eds.) Projektmanagement und Vorgehensmodelle 2017 - Die Spannung zwischen dem Prozess und den Mensch im Projekt, pp. 25–34. Gesellschaft für Informatik, Bonn (2017)
43. Theobald, S., Diebold, P.: Beneficial and Harmful Agile Practices for Product Quality. In: Felderer, M., Méndez Fernández, D., Turhan, B., Kalinowski, M., Sarro, F., Winkler, D. (eds.) Product-focused software process improvement. 18th International Conference, PROFES 2017, Innsbruck, Austria, November 29-December 1, 2017, Proceedings / Michael Felderer, Daniel Méndez Fernández, Burak Turhan, Marcos Kalinowski, Federica Sarro, Dietmar Winkler (eds.), pp. 586–593. Springer, Cham (2017)
44. Theobald, S., Diebold, P.: Interface Problems of Agile in a Non-agile Environment. In: Garbajosa, J., Wang, X., Aguiar, A. (eds.) Agile Processes in Software Engineering and Extreme Programming. 19th International Conference, XP 2018, Porto, Portugal, May 21–25, 2018, Proceedings / Juan Garbajosa, Xiaofeng Wang, Ademar Aguiar, 314, pp. 123–130. Springer, Cham (2018)

45. Diebold, P., Theobald, S., Wahl, J., Rausch, Y.: An Agile transition starting with user stories, DoD & DoR. In: Kuhrmann, M. (ed.) Proceedings of the 2018 International Conference on Software and System Process, pp. 147–156. ACM, [Place of publication not identified] (2018)
46. Partridge, D., Wilks, Y.: Does AI have a methodology which is different from software engineering? Artif Intell Rev 1, 111–120 (1987)
47. Tsai, W.T., Heisler, K.G., Volovik, D., Zualkernan, I.A.: A critical look at the relationship between AI and software engineering. In: Symbiotic and intelligent robotics. Annual workshop on languages for automation : Technical papers and discussions, pp. 2–18. IEEE Comput. Soc. Press (1988)
48. Manifesto for Agile Software Development, https://agilemanifesto.org/