Cross fertilization in software engineering

Ricardo Colomo-Palacios [1[0000-0002-1555-9726]] [1]

[1] Østfold University College, Halden, Norway
ricardo.colomo-palacios@hiof.no

Abstract.
Software engineering is related to a set of disciplines both inside and outside computing. One of the aspects to consider in the development of a discipline is cross fertilization. In this paper, author reviews the cross fertilization produced by related disciplines in Software Engineering. The influences come from Computer Engineering and Computer Science inside computing outside this field quality and project management, naming just a few of them. More in particular by focusing on specific technologies, author will overview bidirectional relationships between two of the most promising technologies nowadays namely: blockchain and machine learning.

Keywords: Blockchain, Machine Learning, Software Engineering.

1 Introduction

According to the Merriam-Webster dictionary, cross-fertilization is the interchange or interaction between different ideas, cultures, or categories especially of a broadening or productive nature. Cambridge dictionary defines the term as the mixing of the ideas, customs, etc. of different places or groups of people to produce a better result. Finally, Collins dictionary defines the term as the interaction or interchange between two or more cultures, fields of activity or knowledge, or the like, that is mutually beneficial and productive. In the two latter definitions, we find the inspiration for this paper: exploring the benefits from the associations between software engineering and some other fields inside and outside computing.

Specialization is one of the current trends in scientific work [58]. However, there are also voices warning on the risk that high specialization will lead to knowledge silos and lack of understanding or concern about works conducted in other areas, while others underlining that specialization conducted in a non-isolated way is a foundation for cross-fertilization [32].

One of the main aspects of analyzing cross fertilization is the disciplines hybridization. Coming from social sciences, Dogan and Pahre presented a set of publications from their seminal work [15] to underline the fact that the fragmentation of disciplines leaves research gaps that could be filled by means of hybridization. Dogan, the first of the couple recently underlined that hybridization appears in all fields [14].

Literature reported tons of papers on similar concepts like intradisciplinary, multidisciplinary, interdisciplinary or transdisciplinary and their effects on science. There are works devoted to analyze and compare these terms e.g. [3]. In general, authors reported several benefits from collaboration among different disciplines and fields [16, 35].

Focusing on Software engineering, the discipline is itself highly influenced by several disciplines outside computing. Software engineering presents, according to the Software Engineering Body of Knowledge (SWEBOK), strong interfaces with Management, Economics, Mathematics and Engineering fields [1]. In the joint initiative of ACM-IEEE Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering [5, 33], Software Engineering also presents connections with computing, engineering, mathematics and statistics, psychology and social sciences and management. Apart from curricular efforts and bodies of knowledge, the cross-fertilization or cross pollination has been objective of study in professional magazines as well. Not in vain, twenty years ago, IEEE Software devoted a special issue to Benefits and Applications of Cross-Pollination in software engineering [39]. In this effort, authors reported several efforts and calls for a full-duplex exchange of results in a continuous way. More recently, in the same publication, a special issue devoted to Software Engineering in Society provides an overview of the connections, not necessarily in both directions, between software engineering and health, physical sciences, environmental sciences, social sciences, management, economics, computing and engineering, security, safety and privacy, policing, manufacturing, engineering emerging cyber physical systems or arts [26].

In any case, while the relationship with the rest of the computing fields is quite evident, the contact with engineering has been an arena for discussion, mostly coming from the licensing of software engineering [28, 46]. In any case, the similarities and differences with more traditional engineering are explained in the IEEE Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering [5, 33], as follows:

- Software is abstract and invisible.
- Software presents static and dynamic properties alike.
- Software is complex itself in terms of its organization.
- No universal measures of quality exist for assessing a software product.
- The manufacturing cycle for software products involves the needs of distribution mechanisms also.
- Software does not wear out (maintenance is a key activity).

Mathematics & Statistics is seen as one of the foundations of the discipline according to SWEBOK [1]. In this initiative, authors claim that this area "helps software engineers comprehend this logic, which in turn is translated into programming language code". Other authors pointed out that mathematics provides a scientific basis for the discipline, leading to a deeper understanding of the development process and backing up its methods by means of mathematical techniques [9]. While there are voices in the literature underlining the fact that some software engineers do their work without applying any mathematics, good or correct software engineering is a quite difficult task to accomplish and mathematical foundations are a key to face new scenarios in which statistics and mathematics will likely play the main role e.g. big data or machine learning projects [17].

The interface between Software Engineering and Psychology is also quite established. Psychologists have been studying the behavioral aspects of the discipline since the fifties [12]. Currently, software engineering literature is full of influences of different fields of study inside psychology like developmental psychology [21], cross-cultural studies [22, 43], personality studies [41, 54, 62], emotions [44, 49, 60] or motivation [20, 50], naming just some of the most important connections.

According to SWEBOK [1], Software engineering management is the application of management activities—planning, coordinating, measuring, monitoring, controlling, and reporting—to ensure that software products and software engineering services are delivered efficiently, effectively, and to the benefits of stakeholders. Influences in the software engineering field have been pervasive and constant from the very beginning of the discipline.

Although Management and psychology are part of social sciences, the interfaces between software engineering and this branch of science are quite important. Focusing on research methods, qualitative methods used in software engineering e.g. grounded theory, surveys or case studies are normally taken from social sciences [18, 42]. Other authors from the literature have also underlined the similarities of software engineering with social sciences, given the focus on subjects [25].

In this paper, the approach taken by the author is based on the analysis of the connections of Software Engineering with other computing disciplines and more specifically with research fields that are currently in different stages in the hype, namely, blockchain and machine learning.

The remaining of this document is structured as follows. Section 2 includes the intersections between machine learning and software engineering. In section 3 author reviews Blockchain and its implications with software engineering in both directions. Finally, section 4 wraps up the paper and presents main insights and future prospects.

2 Machine Learning

The popularity of Machine Learning is quite high. Not in vain, a query at Google Scholar including both terms produces more than 150,000 results by March 2020. Machine Learning alone as a search string is producing more than 3 million results in this academic search engine.

Machine learning is aimed to answer these two questions [24], firstly, how can we build computer-based systems that automatically improve through experience? And secondly, what are the fundamental statistical-computational-information-theoretic laws that govern all learning systems, including computers, humans, and organizations? As a field of study, Machine Learning combines several disciplines including statistics, mathematics, engineering, biology, neuroscience and computer science [38]. Machine Learning applications work and optimize their performance using example data from past experiences [2]. It starts with the definition of a predictive model based on some configuration parameters and from that, the learning comes in the execution of a computer program to optimize the model in a predictive (make predictions), descriptive (gain knowledge from data) or combined way. In other words, the machine learning focus is on making computers modify or adapt their actions in the search of better accuracy (measured by how well actions reflect the correct ones) [38].

Machine learning is based on algorithms and their capacity to learn a model. Learning techniques are often divided into supervised, unsupervised, reinforcement and evolutionary learning. In supervised learning, both examples of inputs and outputs are provided and the task of the algorithm is constructing a mapping from one to the other [53]. In unsupervised learning, there are not available sample correct responses and the task is processing the input data in the search of underlying patterns and categories. Reinforcement learning combines both the approaches [38], there is information whether an answer is wrong but not how to correct it, so it is needed to evaluate input–output pairs and hence discover, through trial and error, the optimal outputs for each input [53]. Finally, evolutionary learning is an approach inspired by biology and natural evolution. In this approach, the algorithm maintains a population of candidates, which are compared with the output. Then, through multiple generations of variation, selection, and reproduction, the population adapts to the selection criterion (the relative distance from the desired outcome) and produces fitter solutions [23].

To provide a structured review of the influence of the field, author used Google Scholar to search for the 10 knowledge areas defined in the 2004 edition of SWEBOK. Author decided to exclude the new five knowledge areas present in SWEBOK V3 (Software engineering professional practice, Software engineering economics, Computing foundations, Mathematical foundations and Engineering foundations), given their lack of specificity for tools.

Table 1. Number of hits of Machine Learning per SWEBOK 2004 Knowledge Areas

SWEBOK Knowledge Area	# Google Scholar Results
1 Software requirements	7760
2 Software design	18900
3 Software construction	1140
4 Software testing	15900
5 Software maintenance	12400
6 Software configuration management	833
7 Software engineering management	399
8 Software engineering process	1380
9 Software engineering models and methods	33
10 Software quality	16100

While it is true that the labeling of the knowledge areas could influence in deep the number of results (e.g. Software construction) and that a simple query is not able to give a full description of the influence, it is also unquestionable that some knowledge areas are more affected than others and that, overall, the penetration of machine learning in software engineering is not superficial. In order to see the evolution of this influence, Table 2 includes the results of the ten knowledge areas in the period 2011-2019.

Table 2. Evolution on the number of hits of Machine Learning per SWEBOK 2004 Knowledge Areas

KA	2011	2012	2013	2014	2015	2016	2017	2018	2019
1	258	304	362	497	510	745	805	1130	1190
2	709	812	960	1080	1250	1450	1750	2270	2780
3	43	54	47	67	66	71	81	103	141
4	521	631	788	964	1170	1400	1640	2130	2700
5	442	541	607	791	951	1100	1300	1650	2010
6	38	40	35	69	77	56	69	67	72
7	23	26	22	43	34	17	14	24	37
8	67	62	74	103	78	113	105	129	136
9	0	0	0	3	2	0	2	4	11
10	590	721	801	976	1190	1390	1680	2220	2630
	2691	3191	3696	4593	5328	6342	7446	9727	11707

A review of the last nine years shows a significant increase in the number of works published and available in Google Scholar with yearly increments of around 20% in almost all the knowledge areas.

The influence of Machine Learning applications in Software Engineering practice is quite apparent and the number of papers devoted to the topic is vast. The popularity of the so called ML4SE (Machine Learning for Software Engineering) has driven to a good set of courses, conferences and workshops devoted to the field. The popularity of the subject leads further to a set of tertiary studies on the topic devoted to aspects like effort estimation [57], software fault prediction [36], software optimization [40] or code smell detection [6], citing just some of the relevant examples.

If we switch direction towards SE4ML (Software engineering for Machine Learning), the road ahead was described back in 2018 [27] as a result of the First Symposium on Software Engineering for Machine Learning Applications. In their statement, authors claim that Machine Learning systems are difficult to test and verify, given that Machine Learning based applications are built on rules inferred from training data. Another relevant initiative is the International Workshop on Machine Learning and Software Engineering in Symbiosis. In this workshop, Software Engineering and Machine Learning communities were encouraged to work together to solve the critical aspects of assuring the quality of artificial intelligence and software systems. As a result of their discussions, they stated that the combined knowledge of Software engineering and Machine Learning is required to answer the key questions regarding the integration of Machine Learning pipelines into software development processes and identifying the desired new roles to address respective challenges.

3 Blockchain

Blockchain is a technology that because of its potentials could be ubiquitous [31]. Blockchain technology was first implemented in Bitcoin by Nakamoto back in 2008. From the technological standpoint, it consists of a sequence of blocks, each of which holds a complete list of transaction records like the conventional public ledger [61]. Thus, Blockchain is seen as a distributed ledger technology (DLT) that supports collaborative processes by means of a shared, distributed and trusted dataset implementing also point-to-point transmission, multi-node collective maintenance, consensus mechanisms and encryption algorithms [7]. Blockchain benefits include aspects like decentralization, persistency, anonymity and auditability [65]. However, blockchain is not a perfect technology and several challenges and limitations have been pointed out. Concerns on energy consumption, privacy, scalability and connectivity, among others, are reported in the literature. A recent and in deep review on challenges and issues is available at [29].

However, Blockchain has evolved in its features, and as a consequence, several generations of blockchain is developed. Blockchain 1.0 is the seminal Blockchain attached to crypto-currencies applications. Blockchain 2.0 started with Ethereum back in 2013 and provides a wider range of application scenarios by using the distributed ledger of blockchain to record, confirm and transfer various forms of contracts and properties [55]. This new generation includes smart contracts as one of its main features, a recent review on the topic can be found at [64]. Blockchain 3.0 includes a vast array of applications including art, health, and science, among others [19]. Blockchain 3.0 is aimed to enable interoperability [52] and increase network speed. It also incorporates features like immutability, transparency and no need for intermediaries, obtained by the blockchain trustless decentralization to other systems which are built on top of blockchain technology [13]. Finally, there is a new generation in sight [4], Blockchain 4.0 includes artificial intelligence as part of the platform, reducing the need of human management by enabling functions to make decisions and act on systems.

Given the current hype, the relationship between Blockchain and software engineering is quite broad in both directions. Although by early March 2020 a simple query in Google Scholar with both terms just retrieves 564 results, the cross fertilization present in both fields is a growing research area. Moreover, it is also true that there is an increasing interest in the development of a dedicated field inside software engineering for blockchain-oriented applications, so called Blockchain-based software engineering [8, 47]. Examples of this influence can be found in aspects like designing [37], architecting [59], modelling [48], programming [10, 11] or testing [45].

Conversely, there are also influences of blockchain in the software engineering research field. Blockchain has been used to improve the integrity of the software development process [63], to complement agile practices [34], service composition [56], to enable distributed teams [51] or as a support for collaborative software teams [30].

As presented in the case of Machine Learning, Table 3 presents the total number of hits per knowledge area in Blockchain and Table 4 presents results of the ten knowledge areas in the period 2011-2019

Table 3. Number of hits of Blockchain per SWEBOK 2004 Knowledge Areas

SWEBOK Knowledge Area	# Google Scholar Results
1 Software requirements	346
2 Software design	886
3 Software construction	52

4 Software testing	598
5 Software maintenance	379
6 Software configuration management	53
7 Software engineering management	8
8 Software engineering process	65
9 Software engineering models and methods	2
10 Software quality	694

As underlined in the case of Table1, there are knowledge areas that initially could receive less attention because of its labelling or its nature. In any case, champions are again, requirements, testing, maintenance quality and overall design.

Table 4. Evolution on the number of hits of blockchain per SWEBOK 2004 Knowledge Areas

KA	2011	2012	2013	2014	2015	2016	2017	2018	2019
1	0	2	1	0	5	7	33	104	144
2	3	6	14	15	11	28	99	214	375
3	0	0	0	0	1	1	3	8	23
4	3	1	5	6	9	12	57	142	266
5	2	0	1	5	7	12	42	85	167
6	1	0	1	4	3	1	4	4	16
7	0	0	0	0	0	0	2	1	5
8	0	0	2	0	2	2	9	14	20
9	0	0	0	1	0	0	0	0	1
10	2	2	3	6	5	12	56	168	345
	11	11	27	37	43	75	305	740	1362

Regarding the evolution over time, in the case of Blockchain, the progression has accelerated from 2017 with an extensive increment of around 200% yearly. This could be rooted in the novelty of the technology that in the early 2010s was in the Blockchain 1.0 phase and expanded its applications just some years ago.

To sum up the interaction of blockchain and machine learning, in the near future, both the generalization of blockchain-based solutions and the advance on the software engineering practices will lead to a more mature hybrid field and a more intense cross-fertilization.

4 Conclusions

In this paper, author presented the cross-fertilization between software engineering and two different knowledge areas: Machine Learning and Blockchain. Being both the topics among the technologies in the hype, it is unquestionable the higher influence and repercussion of Machine Learning in Software Engineering and vice versa. However, Blockchain technologies are beginning to be more mature and their interchange with the Software Engineering field is also increasing in deep, specially from 2018.

Both Machine learning and Blockchain fields are advancing and new prospects will impact Software Engineering in the next coming years. For instance, in the Machine Learning field, Adaptive machine learning, as underlined by Gartner, will provide a plus to machine learning-based systems. Adaptive machine learning is about retraining ML models when they are in their runtime environment frequently. This will impact, in deep, not only Software Engineering tools but also will need new software engineering approach to govern this continuous training. Regarding Blockchain, the new wave 4.0 that includes artificial intelligence as part of the platform will be an opportunity to expand the influence of the technology on Software engineering practices and also to connect blockchain-based methods with Software engineering processes.

References

1. Abran, A., Fairley, D.: SWEBOK: Guide to the software engineering Body of Knowledge Version 3. IEEE Computer Society (2014).
2. Alpaydin, E.: Introduction to Machine Learning. MIT Press (2020).
3. Alvargonzález, D.: Multidisciplinarity, Interdisciplinarity, Transdisciplinarity, and the Sciences. International Studies in the Philosophy of Science. 25, 4, 387–403 (2011). https://doi.org/10.1080/02698595.2011.623366.
4. Angelis, J., Ribeiro da Silva, E.: Blockchain adoption: A value driver perspective. Business Horizons. 62, 3, 307–314 (2019). https://doi.org/10.1016/j.bushor.2018.12.001.
5. Ardis, M. et al.: SE 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Computer. 48, 11, 106–109 (2015). https://doi.org/10.1109/MC.2015.345.
6. Azeem, M.I. et al.: Machine learning techniques for code smell detection: A systematic literature review and meta-analysis. Information and Software Technology. 108, 115–138 (2019). https://doi.org/10.1016/j.infsof.2018.12.009.
7. Bai, C., Sarkis, J.: A supply chain transparency and sustainability technology appraisal model for blockchain technology. International Journal of Production Research. 0, 0, 1–21 (2020). https://doi.org/10.1080/00207543.2019.1708989.
8. Beller, M., Hejderup, J.: Blockchain-based software engineering. In: Proceedings of the 41st International Conference on Software Engineering: New Ideas and Emerging Results. pp. 53–56 IEEE Press, Montreal, Quebec, Canada (2019). https://doi.org/10.1109/ICSE-NIER.2019.00022.
9. Broy, M.: Mathematics of software engineering. In: Möller, B. (ed.) Mathematics of Program Construction. pp. 18–48 Springer, Berlin, Heidelberg (1995). https://doi.org/10.1007/3-540-60117-1_3.
10. Chakraborty, P. et al.: Understanding the software development practices of blockchain projects: a survey. In: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 1–10 Association for Computing Machinery, Oulu, Finland (2018). https://doi.org/10.1145/3239235.3240298.

11. Coblenz, M.: Obsidian: A Safer Blockchain Programming Language. In: 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C). pp. 97–99 (2017). https://doi.org/10.1109/ICSE-C.2017.150.
12. Curtis, B.: Fifteen years of psychology in software engineering: Individual differences and cognitive science. In: Proceedings of the 7th international conference on Software engineering. pp. 97–106 IEEE Press, Orlando, Florida, USA (1984).
13. Di Francesco Maesa, D., Mori, P.: Blockchain 3.0 applications survey. Journal of Parallel and Distributed Computing. 138, 99–114 (2020). https://doi.org/10.1016/j.jpdc.2019.12.019.
14. Dogan, M.: Creative Marginality: Innovation At The Intersections Of Social Sciences. Routledge (2019).
15. Dogan, M., Pahre, R.: Fragmentation and recombination of the social sciences. Studies in Comparative International Development. 24, 2, 56–72 (1989). https://doi.org/10.1007/BF02687172.
16. Domik, G., Fischer, G.: Coping with Complex Real-World Problems: Strategies for Developing the Competency of Transdisciplinary Collaboration. In: Reynolds, N. and Turcsányi-Szabó, M. (eds.) Key Competencies in the Knowledge Society. pp. 90–101 Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15378-5_9.
17. Dougherty, J.P.: MATH COUNTS: Where mathematics meets software engineering. ACM Inroads. 8, 3, 13–15 (2017). https://doi.org/10.1145/3123734.
18. Dybå, T. et al.: Qualitative research in software engineering. Empir Software Eng. 16, 4, 425–429 (2011). https://doi.org/10.1007/s10664-011-9163-y.
19. Efanov, D., Roschin, P.: The All-Pervasiveness of the Blockchain Technology. Procedia Computer Science. 123, 116–121 (2018). https://doi.org/10.1016/j.procs.2018.01.019.
20. França, A.C.C. et al.: Motivation in software engineering industrial practice: A cross-case analysis of two software organisations. Information and Software Technology. 56, 1, 79–101 (2014). https://doi.org/10.1016/j.infsof.2013.06.006.
21. Gren, L. et al.: The Perceived Effects of Group Developmental Psychology Training on Agile Software Development Teams. IEEE Software. 0–0 (2019). https://doi.org/10.1109/MS.2019.2955675.
22. Hoda, R. et al.: Socio-Cultural Challenges in Global Software Engineering Education. IEEE Transactions on Education. 60, 3, 173–182 (2017). https://doi.org/10.1109/TE.2016.2624742.
23. Hu, T. et al.: An evolutionary learning and network approach to identifying key metabolites for osteoarthritis. PLOS Computational Biology. 14, 3, e1005986 (2018). https://doi.org/10.1371/journal.pcbi.1005986.
24. Jordan, M.I., Mitchell, T.M.: Machine learning: Trends, perspectives, and prospects. Science. 349, 6245, 255–260 (2015). https://doi.org/10.1126/science.aaa8415.
25. Juristo, N., Moreno, A.M.: Basics of Software Engineering Experimentation. Springer Science & Business Media (2013).
26. Kazman, R., Pasquale, L.: Software Engineering in Society. IEEE Software. 37, 1, 7–9 (2020). https://doi.org/10.1109/MS.2019.2949322.

27. Khomh, F. et al.: Software Engineering for Machine-Learning Applications: The Road Ahead. IEEE Software. 35, 5, 81–84 (2018). https://doi.org/10.1109/MS.2018.3571224.
28. Knight, J.C., Leveson, N.G.: Should software engineers be licensed? Communications of the ACM. 45, 11, 87–90 (2002).
29. Kolb, J. et al.: Core Concepts, Challenges, and Future Directions in Blockchain: A Centralized Tutorial. ACM Comput. Surv. 53, 1, 9:1–9:39 (2020). https://doi.org/10.1145/3366370.
30. Król, M. et al.: ChainSoft: Collaborative Software Development Using Smart Contracts. In: Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems. pp. 1–6 ACM, New York, NY, USA (2018). https://doi.org/10.1145/3211933.3211934.
31. Lao, L. et al.: A Survey of IoT Applications in Blockchain Systems: Architecture, Consensus, and Traffic Modeling. ACM Comput. Surv. 53, 1, 18:1–18:32 (2020). https://doi.org/10.1145/3372136.
32. Leahey, E., Reikowsky, R.C.: Research Specialization and Collaboration Patterns in Sociology. Soc Stud Sci. 38, 3, 425–440 (2008). https://doi.org/10.1177/0306312707086190.
33. LeBlanc, R.J., Sobel, A.: Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. IEEE Computer Society (2014).
34. Lenarduzzi, V. et al.: Blockchain applications for agile methodologies. In: Proceedings of the 19th International Conference on Agile Software Development: Companion. pp. 1–3 Association for Computing Machinery, Porto, Portugal (2018). https://doi.org/10.1145/3234152.3234155.
35. Madni, A.M.: Transdisciplinarity: reaching beyond disciplines to find connections. Journal of Integrated Design and Process Science. 11, 1, 1–11 (2007).
36. Malhotra, R.: A systematic review of machine learning techniques for software fault prediction. Applied Soft Computing. 27, 504–518 (2015). https://doi.org/10.1016/j.asoc.2014.11.023.
37. Marchesi, M. et al.: An Agile Software Engineering Method to Design Blockchain Applications. In: Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia. pp. 1–8 Association for Computing Machinery, Moscow, Russian Federation (2018). https://doi.org/10.1145/3290621.3290627.
38. Marsland, S.: Machine Learning: An Algorithmic Perspective, Second Edition. CRC Press (2015).
39. Matsubara, T., Ebert, C.: Guest Editor's Introduction: Benefits and Applications of Cross-Pollination. IEEE Software. 17, 1, 24 (2000).
40. Memeti, S. et al.: Using meta-heuristics and machine learning for software optimization of parallel computing systems: a systematic literature review. Computing. 101, 8, 893–936 (2019). https://doi.org/10.1007/s00607-018-0614-9.
41. Mendes, F.F. et al.: The relationship between personality and decision-making: A Systematic literature review. Information and Software Technology. 111, 50–71 (2019). https://doi.org/10.1016/j.infsof.2019.03.010.

42. Méndez Fernández, D., Passoth, J.-H.: Empirical software engineering: From discipline to interdiscipline. Journal of Systems and Software. 148, 170–179 (2019). https://doi.org/10.1016/j.jss.2018.11.019.
43. Niazi, M. et al.: Software Process Improvement barriers: A cross-cultural comparison. Information and Software Technology. 52, 11, 1204–1216 (2010). https://doi.org/10.1016/j.infsof.2010.06.005.
44. Novielli, N., Serebrenik, A.: Sentiment and Emotion in Software Engineering. IEEE Software. 36, 5, 6–23 (2019). https://doi.org/10.1109/MS.2019.2924013.
45. Parizi, R.M. et al.: Empirical vulnerability analysis of automated smart contracts security testing on blockchains. In: Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering. pp. 103–113 IBM Corp., Markham, Ontario, Canada (2018).
46. Parnas, D.: Software Engineering - Missing in Action: A Personal Perspective. Computer. 44, 10, 54–58 (2011). https://doi.org/10.1109/MC.2011.268.
47. Porru, S. et al.: Blockchain-Oriented Software Engineering: Challenges and New Directions. In: 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C). pp. 169–171 (2017). https://doi.org/10.1109/ICSE-C.2017.142.
48. Rocha, H., Ducasse, S.: Preliminary Steps Towards Modeling Blockchain Oriented Software. In: Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain. pp. 52–57 ACM, New York, NY, USA (2018). https://doi.org/10.1145/3194113.3194123.
49. Sánchez-Gordón, M., Colomo-Palacios, R.: Taking the emotional pulse of software engineering — A systematic literature review of empirical studies. Information and Software Technology. 115, 23–43 (2019). https://doi.org/10.1016/j.infsof.2019.08.002.
50. Sharp, H. et al.: Models of motivation in software engineering. Information and Software Technology. 51, 1, 219–233 (2009). https://doi.org/10.1016/j.infsof.2008.05.009.
51. Singi, K. et al.: Compliance Adherence in Distributed Software Delivery: A Blockchain Approach. In: 2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE). pp. 126–127 (2018).
52. Siris, V.A. et al.: Interledger Approaches. IEEE Access. 7, 89948–89966 (2019). https://doi.org/10.1109/ACCESS.2019.2926880.
53. Smith, A.J.: Applications of the self-organising map to reinforcement learning. Neural Networks. 15, 8, 1107–1124 (2002). https://doi.org/10.1016/S0893-6080(02)00083-7.
54. Soomro, A.B. et al.: The effect of software engineers' personality traits on team climate and performance: A Systematic Literature Review. Information and Software Technology. 73, 52–65 (2016). https://doi.org/10.1016/j.infsof.2016.01.006.
55. Wang, N. et al.: When Energy Trading Meets Blockchain in Electrical Power System: The State of the Art. Applied Sciences. 9, 8, 1561 (2019). https://doi.org/10.3390/app9081561.

56. Wang, P. et al.: QoS-aware Service Composition Using Blockchain-based Smart Contracts. In: Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings. pp. 296–297 ACM, New York, NY, USA (2018). https://doi.org/10.1145/3183440.3194978.
57. Wen, J. et al.: Systematic literature review of machine learning based software development effort estimation models. Information and Software Technology. 54, 1, 41–59 (2012). https://doi.org/10.1016/j.infsof.2011.09.002.
58. Wenger, E.: Communities of Practice: Learning, Meaning, and Identity. Cambridge University Press (1999).
59. Wessling, F., Gruhn, V.: Engineering Software Architectures of Blockchain-Oriented Applications. In: 2018 IEEE International Conference on Software Architecture Companion (ICSA-C). pp. 45–46 (2018). https://doi.org/10.1109/ICSA-C.2018.00019.
60. Wrobel, M.R.: Applicability of Emotion Recognition and Induction Methods to Study the Behavior of Programmers. Applied Sciences. 8, 3, 323 (2018). https://doi.org/10.3390/app8030323.
61. Xie, S. et al.: Blockchain for cloud exchange: A survey. Computers & Electrical Engineering. 81, 106526 (2020). https://doi.org/10.1016/j.compeleceng.2019.106526.
62. Yilmaz, M. et al.: An examination of personality traits and how they impact on software development teams. Information and Software Technology. 86, 101–122 (2017). https://doi.org/10.1016/j.infsof.2017.01.005.
63. Yilmaz, M. et al.: Applying Blockchain to Improve the Integrity of the Software Development Process. In: Walker, A. et al. (eds.) Systems, Software and Services Process Improvement. pp. 260–271 Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-28005-5_20.
64. Zheng, Z. et al.: An overview on smart contracts: Challenges, advances and platforms. Future Generation Computer Systems. 105, 475–491 (2020). https://doi.org/10.1016/j.future.2019.12.019.
65. Zheng, Z. et al.: Blockchain challenges and opportunities: a survey. International Journal of Web and Grid Services. 14, 4, 352–375 (2018). https://doi.org/10.1504/IJWGS.2018.095647.