



Significance of Artificial Intelligence Techniques in Software Engineering to Reduce the Complexity of Software Development

Ms. Neelam Sunda¹, Ms. Nikita Gupta²

Research Scholar, Research Centre S. S. Jain Subodh College, RTU, Kota, India

Emails: ¹research.neelam@gmail.com, ²nikitagupta.ssm@gmail.com

ABSTRACT

Software Engineering (SE) is a complex process. It is not just writing codes, testing, and deployment. It comprises a set of procedures for creating and delivering software. The project management team frequently encounters problems during the software development process due to the rapid advancement of technology, ever-changing customer requirements, short time frames, and tight budgets. Artificial Intelligence (AI) techniques have recently attracted a lot of attention from the software development industry as it aids in minimizing the complexity of SE. This study presents a state-of-the-art literature review that exhibits the effort done in the past and present to automate the Software Development Life Cycle.

Key words: Software Engineering (SE), Artificial Intelligence (AI), Software Development Life Cycle (SDLC), Automated Software Engineering

1. INTRODUCTION

Software Engineering (SE) is a complex process. It is not just writing codes, testing them, and deploying them. It also comprises a set of procedures for creating and delivering software within a time frame. The project management team frequently encounters problems during the software development process due to the rapid advancement of technology, ever-changing customer requirements, short time frames, and tight budgets. This can be alleviated by imparting Artificial Intelligence Techniques.[1] Artificial Intelligence is the most popular buzzword nowadays as it is changing every business all over the globe. The software development sector is unquestionably the biggest beneficiary as there are numerous AI techniques available for the sector to automate the development process and improve almost every phase of SDLC [2].

The term "software engineering" was proposed at conferences by NATO in 1968 and 1969. The conference's objectives were to address the problems caused by the software crisis (the challenge of creating useful and effective computer programs in the scheduled amount of time) and to set standards and best practices for software development [3-4].

AI techniques have been striving to assist and automate the various activities of software engineering to develop software that exhibits some aspect of human intellect [5]. Automated software engineering is a research field that constantly develops new toolsets and frameworks based on mathematical models. With these AI-powered tools, software developers and testers are becoming more efficient and productive by developing high-quality software products. The goal of this review paper is to pinpoint the key contribution or effect that AI-powered tools have on easing the difficulty of software development.

2. SOFTWARE ENGINEERING AND ARTIFICIAL INTELLIGENCE

Software Engineering

SE is the application of engineering principles used for the designing, development, testing, implementation, and maintenance of software products. Each piece of software goes through various phases (Figure 1) of development during its complete life cycle known as SDLC, which defines these phases as requirement analysis, software design, coding/development, testing, deployment, and maintenance [4][5].

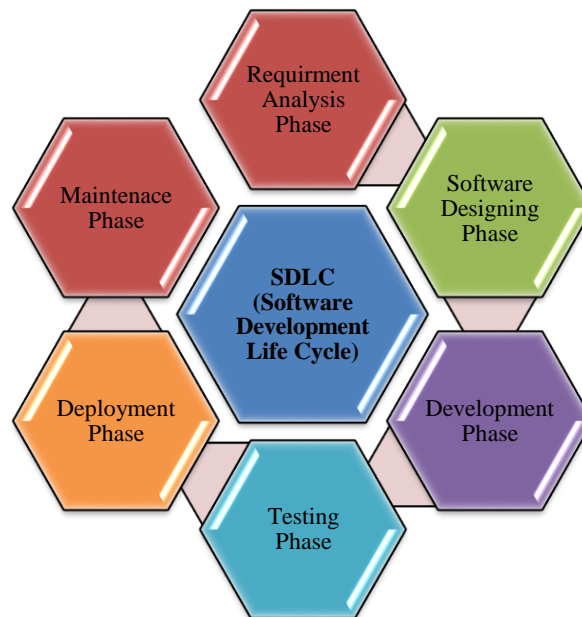


Fig. 1 Phases of Software Development Life Cycle

Requirement analysis is the most important phase, which directly deals with the client to identify and collect their requirements for analysis and documentation into software requirement specifications (SRS). The SRS served as a foundation for the software design phase. The requirements mentioned in the SRS document are used as input to derive the software architecture, which includes the user interface, hardware and software requirements, and database design. The design phase is an inevitable precursor to the development phase, where programmers write code and create the application according to the given specifications and prior design documents. To meet the quality standards, the modules are released for testing in the next phase. Once the individual modules were tested, they were integrated into their environment and eventually installed during the deployment phase. Furthermore, developers oversee any modifications that the software may require after it has been deployed in the maintenance phase [6].

Artificial Intelligence

The term AI was coined at the Dartmouth conference in 1956 [6]. Through AI, tasks carried out by humans can be performed better by intelligent machines, which simulate human intelligence with the use of powerful AI algorithms and techniques. These algorithms are used to program the machines to behave and think like humans [7]. Nearly every sector of the economy is being shaped by artificial intelligence be it economic, engineering, manufacturing, medicine, and law. The prevalent AI domains and AI techniques are shown in Figure 2.[6]

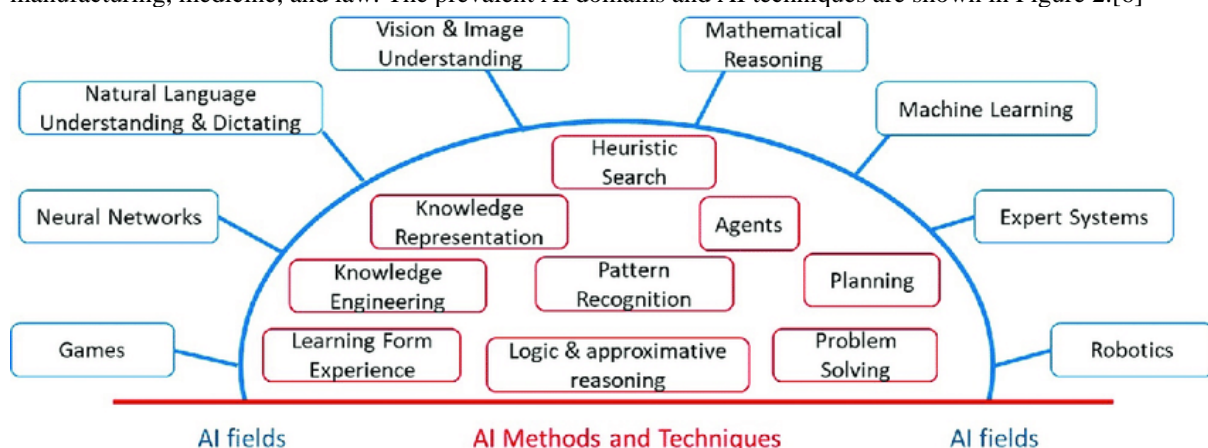


Fig. 2 AI fields, methods, and techniques [6]

3. USE OF AI IN SOFTWARE DEVELOPMENT

There is a great opportunity for the hybridization of AI and SE to manage the software crisis [4] and improve software development [7]. This section examines the techniques to automate major stages of the SDLC using artificial intelligence.

Requirement Analysis Phase: The foremost part of SDLC is requirement gathering and analysis. Redundancy and inconsistency in hand-written requirements frequently harm the success of the final software solution. Artificial intelligence techniques aid with requirement analysis, classification, and prioritizing as well as with avoiding incorrect requirements and human intervention [8]. It is surveyed that for requirement prioritizing, methods such as Genetic Algorithm, Artificial Neural Network, and K-Nearest Neighbour produce favorable outcomes and for requirement classification, traceability, and analysis Convolution Neural Network, Case-based reasoning outperforms the rest [10].

Software Design Phase: Creating high-quality architectural, function-oriented, and object-oriented designs from the requirements document is one of the biggest challenges faced by software engineers. Case-based reasoning, Genetic Algorithms, Fuzzy Systems, Neural Networks, Rule-Based Reasoning, Genetic Programming, and Artificial Neural Networks are the major techniques that may be used to circumvent the issues related to this phase [9].

Development Phase: Software engineers can apply AI techniques to automate or assist the development process and effort estimation. Case-based reasoning, Constraint Programming, and Genetic Programming are the possible AI technologies that could be applied in this phase to assist the code development [5][8]. Support Vector Machine (SVM) and Nearest Neighbour Algorithms have significant roles in accurate effort estimation [11].

Testing Phase: Testing is a labor-intensive, time-restricted, and costly investigation process to verify and validate the functionality of desired software. This phase may require almost 50% of the development recourses and due to this fact, automation of testing is required.[12] Machine Learning is a sub-field of AI and algorithms like Artificial Neural Networks, Decision Tree algorithm, [12] Imbalance Learning, Ensemble Learning, Multiple Classifiers, and Resampling are quite useful for testing automation.[13] Deep learning model such as Convolution Neural Network (CNN) is capable of detecting suspicious fault that may occur in the future [14].

4. BACKGROUND STUDY

Table -1 Literature Review.

Title	Author	Year	Analysis
Software Engineering Using Artificial Intelligence Techniques: Current State and Open Problems	Hany H Ammar , Walid Abdelmoez , and Mohamed Salah Hamdi [8]	2012	The author of this paper surveyed numerous research works done on the use of AI techniques to solve the challenges related to software development phases such as requirement engineering, architecture design, coding, and testing. The paper highlights the prominent AI techniques used to mitigate the issues and also summarizes other open problems.
Integrating AI Techniques In SDLC: Design Phase Perspective	Shreta Sharma and S. K. Pandey [9]	2015	The paper describes the implementation of AI techniques in different stages of the design phase. The author properly mapped these issues with relevant AI techniques with the help of the Venn-Diagram to present a clear picture of the current and future status of the research
Effort Estimation of Web Based Applications Using ERD, Use Case Point Method and Machine Learning	Dhiraj Kumar Goswami, Soham Chakrabarti and Saurabh Bilgaiyan [11]	2020	This paper has focused on combining the complexity of entity relationship diagrams and Use Case Point Method methodologies. A comparison between the actual and the estimated effort from the methods has been compared. It can be derived that the methods trained with support vector regression are most suitable to estimate the effort of web-based applications.
Usage of Machine Learning in Software Testing	Sumit Mahapatra and Subhankar Mishra [13]	2020	This paper represents a review of the data-mining and machine-learning approaches towards the discovery and analysis of software vulnerabilities. An overview is given on software metrics and vulnerability prediction based on various approaches for anomaly detection. Pattern recognition of vulnerable code and a system that can be used for automated software testing is presented.
A Novel Approach	Debolina Ghosh and	2020	The author of this paper focuses on the use of the

to Software Fault Prediction Using Deep Learning Technique	Jagannath Singh [14]		Deep Learning Model for software fault prediction. The Convolution Neural Network model of deep learning was used to propose a multi-fault localization technique, which can identify the suspiciousness score of each statement.
A REVIEW OF ARTIFICIAL INTELLIGENCE TECHNIQUES FOR REQUIREMENT ENGINEERING	Kamaljit Kaur, Prabhsimran Singh, Parminder Kaur [10]	2021	In this study, the state-of-the-art artificial intelligence techniques applied in diverse requirements engineering techniques were demonstrated. It is surveyed that for requirement prioritizing, methods such as Genetic Algorithm, Artificial Neural Network, and K-Nearest Neighbour produce favorable outcomes and for requirement classification, traceability, and analysis Convolution Neural Network, Case-based reasoning outperforms the rest.

5. FINDINGS

In this study, we reviewed promising research on using AI approaches to address some of the most significant issues that software engineers currently face. The findings of this review are compiled in the table shown below.

Table -2 AI techniques for different phases of SDLC

S. No.	SDLC Phase	AI/ML Techniques	
1	Requirement Analysis Phase	For requirement prioritization	Genetic Algorithm, Artificial Neural Network, and K-Nearest Neighbour [10]
		For requirement classification, traceability and analysis	Convolution Neural Network, Case-based Reasoning [10]
2	Software Design Phase	Case-based reasoning, Genetic Algorithm, Fuzzy System, Neural Network, Rule Based Reasoning, Genetic Programming and Artificial Neural Networks [9]	
3	Development Phase	For automated code development	Case-based reasoning, Constraint Programming and Genetic Programming [5][8]
		For effort estimation	Support Vector Machine (SVM) and Nearest Neighbour Algorithms [11]
4	Testing Phase	For automated testing	Artificial Neural Network, Decision Tree algorithm [12] Imbalance Learning, Ensemble Learning, Multiple Classifiers and Resampling [13]
		For fault prediction	Convolution Neural Network (CNN) [14]

6. CONCLUSION

In today's digital world, computing devices play a major role in human life. The software has made humans dependent on it. Software will continue to be a part of human life as far as the future can be imagined. Although there are standard practices that are being followed in software development, however, they face many challenges during the different stages of development. Artificial intelligence has made some great leaps in the domains of software development. This paper highlights the challenges and issues confronted by the developers during SDLC and presents various artificial intelligence techniques used to resolve the challenges to reduce the complexity of the software development process. Merging AI/ML with software engineering can improve software development significantly. In the future, unresolved issues can be identified through quality research. Additionally, some more advanced AI techniques can be explored to make the software development process more robust and less complex.

REFERENCES

- [1]. <https://www.computer.org/publications/tech-news/trends/ai-is-changing-software-development>
- [2]. Sorte, Bhagyashree & Joshi, Pooja & Jagtap, Vandana. (2015). Use of Artificial Intelligence in Software Development Life Cycle: A state of the Art Review. *International Journal of Technology Management*. 03. 2309-4893.
- [3]. <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/NATOREports/>
- [4]. Rech, Jörg & Althoff, Klaus-Dieter. (2004). *Artificial Intelligence and Software Engineering: Status and Future Trends*. KI. 18. 5-11.
- [5]. Raza, Engr.Farah. (2009). *Artificial Intelligence Techniques in Software Engineering (AITSE)*. Lecture Notes in Engineering and Computer Science. 2174.
- [6]. Mohammad Shehab, Laith Abualigah, Muath Ibrahim Jarrah, Osama Ahmad Alomari & Mohammad Sh. Daoud (2020) (AIAM2019) Artificial Intelligence in Software Engineering and inverse: Review, *International Journal of Computer Integrated Manufacturing*, 33:10-11, 1129-1144, DOI: 10.1080/0951192X.2020.1780320
- [7]. R. Feldt, F. G. de Oliveira Neto and R. Torkar, "Ways of Applying Artificial Intelligence in Software Engineering," 2018 IEEE/ACM 6th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE), Gothenburg, Sweden, 2018, pp. 35-41.
- [8]. Ammar, Hany & Abdelmoez, Walid & Hamdi, Mohamed. (2012). *Software Engineering Using Artificial Intelligence Techniques: Current State and Open Problems*.
- [9]. Shreta Sharma and S. K. Pandey. 2015. Integrating AI Techniques In SDLC: Design Phase Perspective. In *Proceedings of the Third International Symposium on Women in Computing and Informatics (WCI '15)*. Association for Computing Machinery, New York, NY, USA, 383–387. <https://doi.org/10.1145/2791405.2791418>
- [10]. Kaur, Kamaljit & Singh, Prabhsimran & Kaur, Parminder. (2021). A Review of Artificial Intelligence Techniques for Requirement Engineering. 10.1007/978-981-15-7907-3_20.
- [11]. Goswami, Dhiraj & Chakrabarti, Soham & Bilgaiyan, Saurabh. (2020). Effort Estimation of Web Based Applications Using ERD, Use Case Point Method and Machine Learning. 10.1007/978-3-030-38006-9_2.
- [12]. Noorian, Mahdi & Bagheri, Ebrahim & Du, Wheichang. (2011). Machine Learning-based Software Testing: Towards a Classification Framework. *SEKE 2011 - Proceedings of the 23rd International Conference on Software Engineering and Knowledge Engineering*. 225-229.
- [13]. Mahapatra, Sumit & Mishra, Subhankar. (2020). Usage of Machine Learning in Software Testing. 10.1007/978-3-030-38006-9_3.
- [14]. Ghosh, Debolina & Singh, Jagannath. (2020). A Novel Approach of Software Fault Prediction Using Deep Learning Technique. 10.1007/978-3-030-38006-9_5.
- [15]. Nicholas R. Jennings, On agent-based software engineering, *Artificial Intelligence*, Volume 117, Issue 2, 2000, Pages 277-296, ISSN 0004-3702, 10.1016/S0004-3702(99)00107-1.
- [16]. B. Yalçınır and M. Özdeş, "Software Defect Estimation Using Machine Learning Algorithms," 2019 4th International Conference on Computer Science and Engineering (UBMK), Samsun, Turkey, 2019, pp. 487-491, doi: 10.1109/UBMK.2019.8907149.
- [17]. Anthony, A & Prasad, G & Randunuge, S & Alahakoon, S & Wijendra, Dinuka & Krishara, Jenny. (2020). *Software Development Automation: An Approach to Automate the Processes of SDLC*.
- [18]. Z. Wan, X. Xia, D. Lo and G. C. Murphy, "How does Machine Learning Change Software Development Practices?," in *IEEE Transactions on Software Engineering*, vol. 47, no. 9, pp. 1857-1871, 1 Sept. 2021, doi: 10.1109/TSE.2019.2937083.
- [19]. H. Sofian, N. A. M. Yunus and R. Ahmad, "Systematic Mapping: Artificial Intelligence Techniques in Software Engineering," in *IEEE Access*, vol. 10, pp. 51021-51040, 2022, doi: 10.1109/ACCESS.2022.3174115.
- [20]. D. P. Wangoo, "Artificial Intelligence Techniques in Software Engineering for Automated Software Reuse and Design," 2018 4th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 2018, pp. 1-4, doi: 10.1109/CCAA.2018.8777584.
- [21]. M. Harman, "The role of Artificial Intelligence in Software Engineering," 2012 First International Workshop on Realizing AI Synergies in Software Engineering (RAISE), Zurich, Switzerland, 2012, pp. 1-6, doi: 10.1109/RAISE.2012.6227961.