International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 6, Issue 4 (JULY-AUGUST 2018), PP. 34-36

THE ROLE OF AI IN SOFTWARE ENGINEERING AND TESTING

Vaibhav Narayan
Department of Computer Science and Engineering,
MU, Mumbai, India

***Abstract*— Artificial Intelligence has major impact on the evolving technology of the world and now it is an option to transform the software engineering system into intelligence smart software. Software development through Artificial Intelligence is the hallmark of this era and it is going to shape the future of technology with more business opportunities and now more than 80% companies are working on Artificial Intelligence and those who already have it are redefining their strategies. There is more investment in this particular field because the cumulative role of both software engineering and artificial Intelligence would bring major and unexpected changes in the world within no time. Not only software development setup would be improved but it will also have positive effects on automated testing software and agile test automation. This paper also focuses on the approaches of software engineering process along with the application of Artificial Intelligence in improving the software quality systems and reduction of time in the market. This paper also drives some tasks and activities specified in standardizing the software engineering process. The paper will also discuss the AI techniques needed by software engineer to highlight the open research issues specifically for the research community.**

***Index Terms*—Software Engineering and Testing, Intelligence Smart Software, Artificial Intelligence.**

I. INTRODUCTION

Artificial intelligence (AI) is wide-ranging branch of computer science that deals with building smart machines which are capable of performing smart tasks with assistance of human intelligence [1]. AI is an interlinked science with multiple approaches, but advancements in machine learning and comprehensive deep learning are creating a model shift in virtually every sector of the software industry. Software engineering is a comprehensive and detailed study of engineering technology that deals with design, development and maintenance of software. Basically, Software engineering was introduced to solve the issues of all low-quality software projects. Most of the problems arise when a software typically exceeds budgets, timelines, and minimized levels of quality. It ensures that the application is built correctly, precisely, on time and on low budget and within all requirements. The demand of software engineering also come out to serve the immense rate of change in user environment and requirement on which application is expected to be working ideally.

Artificial Intelligence (AI) techniques have been favorably applied in different areas of software engineering same as like cloud services [5]. The intricacy of software systems has restricted the application of AI techniques in many real world implementations. There are a lot of applications of AI techniques in software engineering that changed the ways of thinking of new world. AI can aid in achieving ever hard and inflexible schedules for information technology and software engineering projects. Companies that established software applications do so in an environment attributed by increased pressure, limited resources to reduce cost and development schedules. It is a challenge to fulfill the demands of software applications [2]. One way to achieve this is to use software development tools operating automatically from the initial stage of software design up to the software installation after testing. Taking in account software testing as a template, automated software systems can aid in most software testing stages.

On the hand data availability, privacy, security and integrity are very important problems and emerging issues in the success of a business operation. Data privacy and security policies in business are controlled government regularities and business requirements. AI can also assist in software reliability, security and privacy. Implementing data privacy and security using data decoding solutions remain at the top priorities for data security. Many solutions to data translation at this level are intensive and disruptive. Artificial intelligences also used to classify data. It can help in identifying and decoding only the appropriate data thereby saving processing power and time. Without classification of data, organizations using decoding process would simply translate everything and accordingly impact users more than needs. Data classification is crucial and can help organizations with their data accessibility, security and privacy needs.

Software engineering is a comprehensive process, with a huge amount of variabilities and uncertainties. During the phase of requirements engineering, stakeholders with different points of view should come together and strive to distinctly define goals, often using terms that are alien to the other party. The technologies that will be used to realize the clarified requirements must be chosen. Change plans must be drawn out, often for a define time in the future. Deadlines and estimates for the delivery of landmark features will be set. Lost time due to unforeseen and obstructions delays must be taken into consideration. Continuous communication and feedbacks lines between multiple stakeholders will have to be invented. The absolute number of unknowns in all of the phases of software development makes it comprehensive and extremely challenging pledged. The rise of project management processes

such as nimble has partially assisted for these unknowns, and has assimilated them in the software development process using repeated approaches.

There are number of examples that describes that how AI is already being applied in SE, because of the similarities of the problem sets that artificial intelligence and software engineering are interacted in both AI and SE deal with complicated real world issues, where a great deal of secretes, ill-defined date and variables are often the initiating point. Another commonality is the practice of idealization over the course of time. In the field of AI, algorithms and techniques such as neural networks and probabilistic reasoning are aimed at learning and evolving so they can modify the output over the course of time. In a very similar manner, software architects and the experience of project managers and developer teams regarding the estimations of deadlines, the use of planning patterns, feasible sprint sizes are determining factors of the potential accuracy of those evaluate and the quality of the outcome.

II. Influence on The Development Process

Several parts of the software development process have been benefiting significantly from automation of software intelligence. The particular areas that are most affected have been designed, deployment, quality assurance and testing. Numerous developments and studies have shown the probabilities of AI being successfully utilized in earlier phases of the software engineering process as well, such as actual developments of software and requirements engineering. By considering a lot of work off the developer's hands using AI techniques, the process of self-automation could be automated to some extent. Research shows that implementing AI to the software development procedure can also sustainably minimized work as well as threats in various parts of the process. In that preview, AI presents itself as a vital tool for increasing quality and efficiency in many stages.

For example, requirements elicitation, architectural design and code refactoring can be facilitated using several AI techniques such as Search Based Software Engineering (SBSE) [4]. In SBSE, search techniques such as genetic algorithms are applied by redefining SE problems as optimization problems. An example of the application of SBSE is the use of genetic algorithms for code generation. The obtained code samples can then be considered as the population, which are then evolved and tested for survivability and optimization.

Using natural-language requirements (NLR) documents such as user stories and use cases, several attempts have been made translate these documents into formal specifications[3]. However, much work is still to be done in this area, and the produced result should still only be seen as a guideline, yet to be approved by stakeholders with the relevant domain knowledge.

Estimation in SE is performed to plan deadlines, anticipate obstacles, and prepare for changes in architecture, requirements, technologies and the very business domain itself. These estimates are often made using incomplete and blurry data, gathered from sources that may not always be as trustworthy as we would like. In this context, statistical methods such as Bayesian models have been used in order to make more accurate estimates in a speedier fashion, instead of solely depending on the experience of project managers. Another application of these models in the field of requirements engineering is by using them to test the quality of requirements specifications [6].

III. Testing And Deployment

A quiet time-consuming portion of the development process goes to testing and verification of the produced software. Although techniques such as Test Driven Design (TDD) have facilitated this by taking up a test-first approach, integration testing and testing scenarios where certain resources become unavailable at unpredictable times are still time-consuming and difficult to define and generate, especially for modern cloud-based distributed applications [7]. Similar to using AI for code generation, SBSE can be applied to the generation of increasingly optimal test cases. AI algorithms can be used to take down certain services during certain test scenarios, or introduce high loads to several parts of the infrastructure to test the integrity of the architecture. An additional contribution of AI in this regard is the possibility of continuously making small adjustments to configurations in order to optimize performance.

IV. Use of AI to Assist the Process of Programming

An expert system must be created in order for software development in order to assist software engineers and this project is called as Programmer's Apprentice Project and it should be capable of replacing human assistants as human programmers and thus it would increase the human productivity. This Apprentice would help in most of programming routine parts and with the passage of time as like cloud computing [11]; the Apprentice would also be able to deal the complicated tasks.

Areas Where Software Engineering - AI Transform:

A. Software Design

Designers need experience and specialized learning when they have to find alternative solutions in designing and project planning before presenting any definite solutions for it. For reaching the desired solution, a designer must have visionary solution and should forward the investigating plan. The stage of correct plan choices is mistake prone and tedious for them. So for that purpose, some of the AI developments would enhance the traditional methods. Just like in AIDA (platform for website building) which creates website through AI which help them in understanding the desires and needs of users through knowledge [7]. They use combinations in millions to create website image, style and focus for the customized user just in 120 seconds. It design the first copy and then from that point, they drop and drag operation.

B. Software Testing

There are countless APIs which interact with applications and then they grow in complexity and leverage the legacy systems every day. The challenges are minimized by machine-based intelligence while the tools of AI are used for examining

the scope and test management, advancement, exploring the information authenticity and creating the test information. Artificial Intelligence ensure that the tests performed are error free and those are testing are free from repeating the manual tests and thus there is more availability to work on new automated software testing. Repeating the tests is not only a time consuming process but it is also costly. With the automated testing, we can increase the overall tests scope so as to improve the software quality [8]. There is also AI enabled cloud testing facility to test and release faster. The user's only need to type their plan in English and it will automatically convert to functional test case. There are also other tools like SapFix and Self-healing tests to fix the specific bugs and then it is approved by engineers.

C. GUI Testing

This is also called as a Graphical User Interface and it is very important to use when interacting with software of today. Their use is increased critical systems and they are necessary in testing systems to avert failures and with very few techniques and tools, it is difficult to utilize. One of the most important testing methods of GUI is ad hoc and they require test designers for performing the humongous tasks like identification of test execution conditions and to evaluate the test adequately [9]. AI is used in one of GUI tester, application tools which are used to test the functioning of visual code. It helps the user to take the expected look of their application and how it looks like in multiple screen layouts if they are fitting in the design or not.

D. Strategic Decision making

The featuring is very important when developing a product and it is a very long process for developers but now AI solution based on machine learning can analyze the existing performance applications to help both business stakeholders and engineers to find out the solution and cut the maximized risks. Significant timeline for planning is required when transforming the business requirements into technology specifics [10]. Software Development companies thus use machine learning to increase revenue fast, delivering the product quickly and speeding up the whole process. One of the examples of AI influenced tool is AL Canvas and it helps in strategic decision making to identify the feasibility and key questions associated with deploying and building the machine learning models of enterprise. It is a very simple tool used in seven different categories e.g. feedback, training, input, outcome, action, judgment and prediction. All of these seven categorizes are helpful in identifying the opportunities to enhance the performance or to reduce the costs.

E. Intelligent Programming Assistants.

Coding is a time consuming and labor intensive project so AI has also assisted in this field by enabling the AI programming assistant to reduce the workload. Here we can take an example of AI based application called as Bayou which is intelligent programming assistant and it aims at extraction of knowledge from source codes [12]. This application follows the method called neural sketch learning to train artificial neural network in order to recognize the high-level patterns of Java programs and it does this just be creating each program sketch and then associating it with intent lying behind the program.

V. CONCLUSION

Software Engineering with the assistance of Artificial Intelligence is now considered as a massive transformation and both have derived such tools which make the software development more reliable and easier. Still the cumulative effects are not as much exposed but in near future, it will dominate in every field so engineers are advised to adopt this skill as early as possible because without it, they would be of no use in coming years.

REFERENCES

[1] Harman, M. (2012, June). The role of artificial intelligence in software engineering. In 2012 First International Workshop on Realizing AI Synergies in Software Engineering (RAISE) (pp. 1-6). IEEE.

[2] Raza, F. N. (2009, March). Artificial intelligence techniques in software engineering (AITSE). In International MultiConference of Engineers and Computer Scientists (IMECS 2009) (Vol. 1).

[3] Kaiser, G. E., & Feller, P. H. (1986). Intelligent assistance without artificial intelligence (No. CMU/SEI-86-TM-14). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

[4] Barstow, D. (1988). Artificial intelligence and software engineering. In Exploring artificial intelligence (pp. 641-670). Morgan Kaufmann.

[5] Tadapaneni, N. R. (2017). Different Types of Cloud Service Models. Available at SSRN 3614630.

[6] Jackson, P. C. (2019). Introduction to artificial intelligence. Courier Dover Publications.

[7] Rich, C., & Waters, R. C. (Eds.). (2014). Readings in artificial intelligence and software engineering. Morgan Kaufmann.

[8] Tadapaneni, N. R. (2017). Artificial Intelligence In Software Engineering. Available at SSRN: https://ssrn.com/abstract=3591807 or http://dx.doi.org/10.2139/ssrn.3591807

[9] Jain, P. (2011). Interaction between software engineering and artificial intelligence-a review. International Journal on Computer Science and Engineering, 3(12), 3774.

[10] Kaiser, G. E., & Feiler, P. H. (1987, March). An architecture for intelligent assistance in software development. In Proceedings of the 9th international conference on Software Engineering (pp. 180-188). IEEE Computer Society Press.

[11] Tadapaneni, N. R. (2018). Cloud Computing: Opportunities and Challenges. SSRN Electronic Journal.10.2139/ssrn.3563342.

[12] Li, B. H., Hou, B. C., Yu, W. T., Lu, X. B., & Yang, C. W. (2017). Applications of artificial intelligence in intelligent manufacturing: a review. Frontiers of Information Technology & Electronic Engineering, 18(1), 86-96.