The future of software engineering: Visions of 2025 and beyond

Firoz Khan[1], R. Lakshmana Kumar[2], Seifedine Kadry[3], Yunyoung Nam[4]
[1]Higher Colleges of Technology, United Arab Emirates
[2]Hindusthan College of Engineering and Technology, Coimbatore, India
[3]Department of Mathematics and Computer Science, Faculty of Science, Beirut Arab University, Lebanon
[4]Department of Computer Science and Engineering, Soonchunhyang University, South Korea

Article Info

Article history:

Received Aug 1, 2020
Revised Dec 21, 2020
Accepted Jan 13, 2021

Keywords:

Industry 4.0
Interaction-oriented software Engineering
Internet of things
Software engineering

ABSTRACT

In the current technological scenario of the industry and businesses, there has been increasing need of software within systems and also an increasing demand being put onto software-intensive systems. This in effect will lead to a significant evolution of software engineering processes over the next twenty years. This is due to the fact of emerging technological advancements like Industry 4.0 and Internet of Things in the IT field, among other new developments. This paper addresses and tries to analyses the key research challenges being faced by the software engineering field and articulates information that is derived from the key research specializations within software engineering. The paper analyses the past and current trends in software engineering. The future of software engineering is also looked with respect to Industry 4.0 which including emerging technological platforms like Internet of Things. The societal impact aspect of future trends in software engineering is also addressed in this paper.

Corresponding Author:

Yunyoung Nam
Department of Computer Science and Engineering
Soonchunhyang University
Asan 31538, South Korea
Email: ynam@sch.ac.kr

1. INTRODUCTION

The future of software engineering is only possible by understanding the past and the present status of software engineering. With the advent of many new technological advancements like internet of things (IoT) and Industry 4.0, the importance of a refined software engineering process in the future is much more paramount. Moreover, also observed is the fact that software engineering is an established process which has its own methods, procedures, and structure of the application. Software Engineering in the current world uses the established methods and tools across a multitude of domains. They are used in information processing applications like databases, commerce, and e-commerce and also are used in proprietary real-time industry for the control of energy, transportation systems and manufacturing automation in factories.

With more and more industries embracing Internet connected infrastructures, the software engineering domain is no longer the primary de-facto standards for just professional software developers. The focus is shifting in all these emerging fields from just coding to make the machines work to ensure that the software developed is of extremely high quality for the purpose for which the software was created. With relation to this background, this paper attempts to identify some key general-purpose issues related to software engineering. The relationship with IoT and Industry 4.0 is observed and discussed in detail. Furthermore, the societal aspects and impacts in relation to software engineering in the future are also discussed. In the final section of the paper, the future trends related to software engineering is predicted and speculated upon.

2. BACKGROUND

Society is increasingly relying on software in many areas including entertainment, education, politics, industrial and civil infrastructures, economic and business initiatives, as well as work and personal activities, among many others [1]. All these areas have become inextricably linked with software systems and applications. As such, software development is a field that requires support, improvement, and research. In the software engineering domain, the focus has been to tackle problems and issues. However, there are challenges that still remain to be overcome.

2.1. Challenges in software engineering

Firstly, researchers have pointed out that the internet has become an environment for development through which cooperation among developers occurs. As such, there are many changes in which software evolves from conception to deployment. One example of such change is crowd-sourcing; an area the researchers observe needs to be better evaluated and understood [1].

Another challenge is related to the fact that software continuously evolves based on customer requirements. The requirement not only creates a challenge with software deployment and configuration management, but also the required technologies that should operate reliably over the internet and the support needed in such an environment. Most importantly, the different types of applications that are deployed in the short term and then replaced with others, coupled with those which are deployed for the long term means that current models and standards should adapt to the various emerging contexts [1].

2.2. Research in software engineering

According to another group of researchers, the field of software engineering has over time shown that developers and engineers are critical to the success and outcomes of projects. Nevertheless, research on the human aspect has been scarce in comparison with a focus on process or technology. Increasing dissatisfaction with challenges related to developing software of high quality as well as growing interest in agile methods has however begun to place more emphasis on the human aspect of research in software engineering [2]. Results from many empirical research studies in software engineering illustrate the types of factors which have an impact on crucial outcomes such as quality and cost, but it has also been ascertained that programmers often hold prior opinions on some of these issues [3]. In a study among IBM developers, for example, it was found that they held prior beliefs on certain issues and their opinions were based on experience instead of findings from prior empirical research. While such prior opinions are important, especially because developers are professionals who are highly trained with beliefs that are critical to their work, dissemination of empirical evidence to the profession is nevertheless important.

Interestingly, it has previously been proposed that evidence-based software engineering should be adopted, and that systematic reviews of literature in software engineering should be used to support evidence-based software engineering [4]. Since these proposals were made in 2004 and 2005, systematic reviews have gained relevance in empirical software engineering. As they became more popular among software engineers, the systematic review process also started receiving attention. Against this background, therefore, the researcher’s systematic review concerned with either technique to improve the process or experiences with the process found that systematic reviews face criticism because there are no appropriate digital libraries for software engineering and that the systematic review process itself is time consuming.

3. IOT AND SOFTWARE ENGINEERING

Wireless sensor network technologies have become a part of everyday life, allowing for the measurement and understanding of indicators in the environment that range from natural resources and delicate ecologies, to environments in the urban areas [5]. These devices have proliferated to form the internet of things (IoT) through which sensors and actuators blend into the environment and provide information across platforms that result in a common operating picture (COP).

The IoT has been taken out of its infancy with the introduction of enabling device technologies such as embedded actuator and sensor nodes, near field communication devices, and RFID tags and readers [5]. This evolution from the www, to web2, web3, and now IoT means that sophisticated, intuitive queries must continue to be developed to meet data-on-demand. Most significantly is the fact that the IoT creates huge amounts of data, an observation which means that sophisticated artificial intelligence algorithms should be developed so that the data can be understood and used intelligently.

3.1. Software defined systems (SDSys)

As the researcher Jararweh rightly observes; there are huge amounts of data generated through the IoT, data that requires innovative ideas in the management and design of the IoT network so that its performance can be enhanced and accelerated. To meet this challenge is the software defined systems (SDSys), a new model that simplifies underlying system architecture by removing management operations and controls from the devices in the IoT, and channeling such operations into a software layer and middleware layer [6].

Significantly, the researchers argue that several SDSys can be used to describe a software defined internet of things (SDIoT). This comprises a software defined network (SDN) that separates operations conducted at the data plane from those at the control plane, the software defined storage (STStore) which separates the data storage layer and data control layer operations, and the software defined security (SDSec) which separates the processing and forwarding plane from the security control plane. In this way, the IoT management operations are made simpler [6].

Another researcher suggests the view that developing applications for the IoT is difficult because there are many related issues that have to be overcome. The issue includes the fact that there is an absence of high level abstractions and no separation of concerns which makes it challenging to address heterogeneity and the large scale. In addition, applications have to be developed in a manner which addresses various life-cycle phases. The development requires an analysis of application logic so that tasks can be distributed over an underlying network, followed by implementation of tasks for distinct hardware [7]. Figure 1 illustrates the abstractions in an IoT.

Figure 1. The abstractions for IoT system

3.2. Appreciating IoT requirements

Despite the fact that IoT can enhance quality, efficiency, and innovation, and connect transportation systems, automotive, medical, and production with information technology (IT) systems in a manner that can provide vital information for businesses, many companies do not appreciate the combined IT, software needs, and required architectures that are needed to take advantage of IoT [8]. Corporate executives, for examples, understand the value chain but are least concerned with technology; IT departments seem removed from real products and people; while engineering departments view IT with suspicion, preferring to concentrate on embedded electronics and system development.

Figure 2 illustrates the complexity of the IoT for modern manufacturing [9]. For IoT to provide useful information to an enterprise, appreciation should be made of the various levels; that is, the machines and devices at the lowest level, a business application, and the ease with which all the information is transacted through cloud computing. An interesting perspective on the use of IoT is observed [10]. According to the researcher, the functionalities of IoT services and applications should not be merely related to the fundamental principle that data is generated from sensing and actuating, but smarter devices can be deployed to conduct some long term functions in autonomy and in the socio-physical space they reside. These services can range from autonomous personal assistants or even just basic cleaning robots [10].

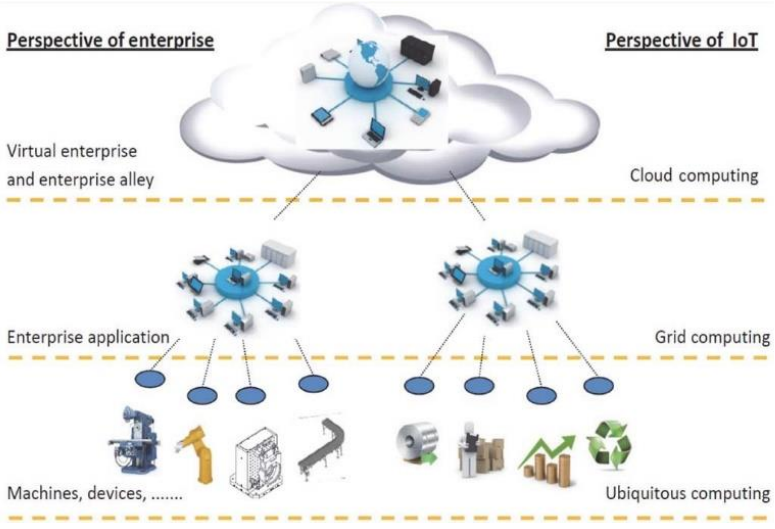

Figure 2. Internet of things (IoT) for modern manufacturing [9]

4. INDUSTRY 4.0 AND SOFTWARE ENGINEERING

Industry 4.0 is a term that has its origin from Germany as "Industrie 4.0" [11]. From a global perspective, comparable ideas are General Electric's *Industrial Internet*. Such an industrial internet, similar in description to Industry 4.0, is supported with a budget of 2 billion dollars by the U.S. government for purposes of researching and developing *Advanced Manufacturing*. Other terms that are used to describe Industry 4.0 include *Integrated Industry*, *Smart Manufacturing*, and *Smart Industry*.

4.1. Leveraging with industry 4.0

Industry 4.0 can be described as networking within an internet of things (IoT), people, data, and services. Significantly, it is characterized by a number of features. The technology enables vertical networking of smart products and smart factories, horizontal integration of customers and business partners, through-engineering that includes the production process and end product, and exponential technologies which allow for mass market application since their computing power has risen while their size and costs have gone down [12].

4.2. Integrating networks

The manufacturing industry in Germany, for example, is facing stiff global competition with regard to costs of production as well as quality. Since the cost of labor is high, competing countries have been able to gain an advantage in quality and productivity. As such, many companies in Germany are leveraging the ideas of Industry 4.0 through which companies participate in integrated networks, so that core competencies are shared.

In Industry 4.0, all businesses in the network are able to access real time product information in a virtualized space which obliterates boundaries and allows for mass customization and agile manufacturing. The presence of cyber-physical systems ensures that machines are able to communicate, and rapid product innovation becomes possible since modeling techniques and modular simulation is shared across the network. In essence, Industry 4.0 enables collaborative networks through its intelligent automation [13].

Most importantly, industrial automation is increasingly relying on software. The German Engineering Federation, for example, points out that the ratio of software development to cost of machinery has doubled from 20 to 40 percent within one decade [14]. In essence, this continuing trend will mean that automation systems developers and suppliers will in future be mainly concerned with software engineering. This is not curious since software engineering is applicable in many domains including applications related to information processing, web-commerce, trade, databases, manufacturing automation, and even real-time control of transportation and energy systems.

4.3. Quality management

According to another researcher, today's business climate is characterized by intense competition in which the quality of products, services, and processes determines if they will be successful. As such, quality management continues to be indispensable and integral in corporate management of every organization [15].

Since Industry 4.0 provides enormous social, ecological, and economic potential, it also stands to play a critical role in quality management. Against this background is the fact that ERP systems are software applications which track and control huge volumes of data that is essential for quality management. As such, they can leverage the advantages provided by Industry 4.0 and will in future be of value in quality management to address Industry 4.0 research challenges.

4.4. Self awareness

A prediction has been made that Industry 4.0 will in future create a fleet-wide information system that will enable machines to be self-aware, the ability to self-assess own degradation and health and also utilize smart decision making to keep problems at bay. This will only be attained through smart analytics. However, one reason why self-awareness in machines has not been fully achieved is because prognosis and health management systems (PHM) have a low level of adaptability [16]. At issue is the fact that the PHM algorithm is developed from experimental data in the lab and cannot change after implementation unless it is re-trained. This is a challenge for developers because in practice real-time data is from numerous machines which generate much more data than that which is generated from the lab. As such, algorithms that are capable of learning from real-time data will in future have to be developed.

4.5. The future of industry 4.0

In their analysis of German manufacturing in relation to the impact of Industry 4.0, it was found that the next ten years will experience a 6 percent growth in employment. During the short term, repetitive tasks will be replaced with more automation. However, the increased use of software, analytics, and connectivity will create more demand for people with skills in IT technologies and software development, including mechatronics professionals having expertise in software. In addition, suppliers will increasingly need to offer services driven by analytics [17].

There are many complex challenges that will confront Industry 4.0 practitioners in the future [18]. For example, there is system complexity in Industrial IoT devices because such devices may also have within them other IoT systems. Another challenge is management complexity related to software, networked systems, security, configuration, and compliance. Thames and Schaefer argue that software defined systems (SDNs) are best placed to overcome such challenges.

5. SOFTWARE ENGINEERING IN EVOLVING SOCIETY

5.1. Interaction-oriented software engineering (IOSE)

The aim of social machines is to enable social processes through the administrative support provided by computers. However, a social-technical machine as conceived by Chopra and Singh would be a social technical system (STS), going beyond the current emphasis of the technical parameters in social machines to give meaning to social processes. The researchers have in mind what they term interaction-oriented software engineering (IOSE) that can be able to connect with STS's social fundamentals [19].

The main objective of IOSE would be social protocols. In the IOSE, social expectations are given computational status and the progress of the interactions tracked by concerned principals. For example, a business contract can represent a social protocol that would then be given computational status. As the researchers observe, IOSE challenges the current principles of software engineering, and leads to new areas of research in information processing.

5.2. Big data

Big data, along with many disruptive technologies such as the internet of things (IoT), cloud computing, mobile computing, and social networking has created many opportunities for business through innovation, made project life cycles shorter, and made business transactions faster [20]. However, big data systems face the challenge of responding to mixed requests requiring either quick response or others requiring complex analytics before reply. Meeting these computational needs in a cost effective manner is a big challenge in software engineering. In addition, the high availability in an eco-system with thousands of nodes forming an application means that network and hardware failures will occur. As such, the data architectures and distributed software should be resilient.

5.3. Computational modeling

Computational modeling is increasingly important for scientists and engineers because events can be addressed in real-time. Meteorologists are able to make forecasts using current conditions, and also determine how changing conditions will have any impact. Using computational models thus hastens the process of forecasting instead of the time consuming work of extrapolating historical data. Computational modeling also makes it possible for natural phenomena that occur at a slow pace to be analyzed by scientists such as climatologists or geologists [21].

Additionally, computational modeling allows phenomena that are experimentally dangerous, such as nuclear reactions, to be studied safely [21]. However, in their study among software engineers and developers, Heaton and Carver found that software engineers need to contribute more to scientific software development so that it can become more accurate. In their view, scientific software developers are only now starting to accept the practices of software engineering to make their software more reliable.

5.4. Ethical concerns and sustainability

Current software systems are in an eco-system that is complex and hyper-connected [22]. As such, software engineers are designing systems which are more open, and which can absorb huge amounts of intricately linked data affecting people. Such software systems also contribute to fighting crime, energy demand, sustainable living, and also empowering groups in society who have been marginalized. This, therefore, means that software engineers encounter and must address ethical issues which are inevitable in software which has become so socially embedded. These ethical issues have usually been treated as requirements which are non-functional and also as issues to be considered against the better good [22].

In fact, another study on sustainable software engineering among academics found that there is very little in the software engineering curriculum about sustainability, the main focus being energy efficiency that is taught in a fact-based manner [23]. A researcher, however, argues that these approaches are insufficient since software changes constantly and as such ethics in software engineering must be relational and contextual [22].

There is another speculation which views that little research has been conducted on the place of software systems in the sustainability of society. According to the authors, this concern has only received attention in the early processes of the design of user interfaces during processes of software design [24]. The National Research Council of USA holds the view that information systems have a critical role to play in society, and "apps" deployed via smartphones is especially relevant in this debate because they have an impact on the welfare of society. Most importantly, research and analysis about socio-technical systems and their role in values, like privacy, while they are at the design stage, should be conducted.

6. FUTURE OF SOFTWARE ENGINEERING

In an article written in 2001 for the American society for quality by John Cusick, a software expert, he predicted that software engineering would remain useful in the future if those in the field were able to engage with the design of large-scale systems, were able to tackle complexity, enable communication and organization, and manage software evolution. In addition, software engineering was to be able to assist engineers and teams obtain objectives efficiently, and also arrive at solutions which were effective [25].

6.1. Scaling out

According to another researcher, Poppendieck, software engineering in the next decade will continue to experience scaling out, a phenomenon where scaling up is no longer possible. Google's sheer growth in the 2000's meant that it could not continue scaling up, but resorted to multiple servers to meet increased traffic. In layman's terms, it is more practical to get two people to lift a heavier load than trying to get a large person to lift the same load. In addition, the researcher noted that platforms would continue to be a disruptive power. Companies such as YouTube, AirBnB, and Spotify, create an environment in which people are able to connect together and have transactions [26].

6.2. Engaging with critical issues

Mathematician Alan Perlis, the first recipient of the Turin Award, described the relationship between computers and humans as one of a youth in endless puberty but constantly growing out of his clothes [27]. Our lives have been heavily influenced and interfered by software, and though the discipline of software engineering is quite new, society has been seen to be maturing. In one sense, the time has come to engage in a discussion about critical issues of security, privacy, dependability, and affordability.

All these are crucial subjects because they are with society every day. For example, software flaws are causing taxpayer money to be wasted, emergency call centers to be unreachable, malfunctions in planes and cars, theft of credit card data, and the closure of airspaces. In addition, smart objects can be used for nefarious purposes to disrupt normal day to day activity of people [27]. However, as the researcher argue, there is a lot of reason to be optimistic about the future of software engineering.

In relation to software analytics, it is speculated that the past focused on reporting and modeling, the present focuses on alerting and recommending, while the future will extrapolate and predict [28]. In essence, the future of software engineering will allow decision making which is data driven, better software artifacts will be developed using logic based tools, and the future will witness challenges posed by new platforms.

7. CONCLUSION

The future of computing and industries that use computing based equipment, in general, is largely based on software written to make them work. The software-intensive systems, in general, are an integral part of human civilization and will continue to do so in the future. Computing products are an essential part of human lives, and humans are more and more interacting and using these devices in their daily routine. The success of these devices is based on the software on which the devices operates and in effect contribute to the general acceptability and success of these devices. There have been several major and distinctive methodologies used in the past and also currently to build such software systems. The future with technologies like Industry 4.0 and IoT still needs software for its successful operation. With newer technologies that may come up in the future and the nature of such systems that is being built continues to change, it is important to attend to the not just the technical elements of software development, but also the general effect of such technologies on society as a whole. The future of software engineering is to be controlled and manipulated by upcoming generations, and some certain pro-action and determination are needed to adapt towards the successful implementation.

ACKNOWLEDGEMENTS

This work was supported by the Soonchunhyang University Research Fund.

REFERENCES

[1] Fuggetta, A., and Di Nitto, E., "Software process," *Paper presented at FOSE 2014*, Hyderabad, [Online], Available: http://www.alfonsofuggetta.org/wordpress/wp-content/uploads/2014/06/ICSE-2014-2014-Fuggetta.pdf

[2] Lenberg, P., Feldt, R. and Wallgren, L., "Towards a behavioral software engineering," *Proceedings of the 7th international workshop on cooperative and human aspects of software engineering*, Hyderabad, 2014, pp. 48-55.

[3] Devanbu, P., Zimmermann, T. and Bird, C., "Belief and evidence in empirical software engineering," *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*, USA, Austin, 2016, pp. 108-119.

[4] Kitchenham, B., and Brereton, P., "A systematic review of systematic review process research in software engineering," *Information and Software Technology*, vol. 55, no. 12, pp. 2049-2075, 2013.

[5] Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M., "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645-1660, 2013.

[6] Jararweh, Y., Al-Ayyoub, M., Darabseh, A., Benkhelifa, E., Vouk, M. and Rindos, A., "SDIoT: A software defined based Internet of Things framework," *Journal of Ambient Intelligence and Human Computing*, vol. 6, no. 4, pp. 453-461, 2015, doi:10.1007/s12652-015-0290-y

[7] Patel, P., and Cassou, D., "Enabling high-level application development for the Internet of Things," *Journal of Systems and Software*, vol. 103, pp. 62-84, 2015.

[8] Weyrich, M. and Ebert, C., "Reference architectures for the Internet of Things," *IEEE Software*, vol. 33, no. 1, pp. 112-116, 2016.

[9] Bi, Z., Xu, L. D. and Wang, C., "Internet of Things for enterprise systems of modern manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1537-1546, 2014, doi: 10.1109/TII.2014.2300338.

[10] Zambonelli, F., "Towards a general software engineering methodology for the Internet of Things," *arXiv preprint arXiv:1601.05569*, 2016.

[11] Mario, H., Tobias, P. and Boris, O., "Design principles for Industrie 4.0 scenarios: A literature review," *2016 49th Hawaii international conference on system sciences (HICSS)*, 2015, pp. 3928-3937.

[12] Deloitte, "Industry 4.0: Challenges and solutions for the digital transformation and use of exponential technologies," *Finance, Audit Tax Consulting Corporate: Zurich*, Swiss, pp. 1-12, 2015.

[13] Brettel, M., Friederichsen, N., Keller, M. and Rosenberg, M., "How virtualization, decentralization and network building change the manufacturing landscape: An Industry 4.0 perspective," *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, vol. 8, no. 1, pp. 37-44, 2014.

[14] Vyatkin,V., "Software engineering in industrial automation: State-of-the-art review," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1234-1249, 2013. doi: 10.1109/TII.2013.2258165.

[15] Foidl, H. and Felderer, M., "Research challenges of Industry 4.0 for quality management," *International conference on enterprise resource planning systems*, 2015, pp. 121-137.

[16] Lee, J., Kao, H. and Yang, S., "Service innovation and smart analytics for Industry 4.0 and big data environment," *Procedia CIRP*, vol. 16, pp. 3-8, 2014, doi: 10.1016/j.procir.2014.02.001.

[17] Rubmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P. *et al*., "Industry 4.0: The future of productivity and growth in manufacturing industries," *Boston Consulting Group*, vol. 9, no. 1, pp. 54-89, 2015.

[18] Thames, L. and Schaefer, D., "Software-defined cloud manufacturing for Industry 4.0.," *Procedia CIRP*, vol. 52, pp. 12-17, 2016.

[19] Chopra, A. K. and Singh, M. P., "From social machines to social protocols: Software engineering foundations for socio-technical systems," *Proceedings of the 25th International Conference on World Wide Web,* 2016, pp. 903-914, doi: 10.1145/2872427.2883018.
[20] Gorton, I., Bener, A. B. and Mockus, A., "Software engineering for Big Data systems," *IEEE Software,* vol. 33, no. 2, p. 33-35, 2016.
[21] Heaton, D. and Carver, J. C., "Claims about the use of software engineering practices in science: A systematic literature review," *Information and Software Technology,* vol. 67, pp. 207-219, 2015.
[22] Rashid, A., Moore, K., May-Chahal, C. and Chitchyan, R., "Managing emergent ethical concerns for software engineering in society," *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering,* vol. 2, pp. 523-526, 2015.
[23] Torre, D., Procaccianti, G., Fucci, D., Lutovac, S. and Scanniello, G., "On the presence of green and sustainable software engineering in higher education curricula," *IEEE/ACM 1st International Workshop on Software Engineering Curricula for Millennials (SECM),* 2017, pp. 54-60.
[24] Barn, B., Barn, R. and Raimondi, F., "On the role of value sensitive concerns in software engineering practice," *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering,* vol. 2, pp. 497-500, 2015.
[25] Cusick, J., "Software engineering: Future or oxymoron? American Society for Quality," *Software Quality Professional,* vol. 3, no. 3, pp. 40-48, 2001.
[26] Poppendieck, M., "The future of software engineering," Government Technology Agency of Singapore, [Online] Available: https://www.tech.gov.sg/en/TechNews/Innovation/2016/11/The-Future-of-Software-Engineering
[27] Shull, F., Carleton, A., Carriere, J., Prikladnicki, R. and Zhang, D., "The future of software engineering," *IEEE Software,* vol. 33, no. 1, pp. 32-35, 2016.
[28] Schulte, W., "Future trends in software engineering," Faculty Summit 2011, Cartagena, Colombia, 2011.