

# Software programs

[Technology](#)



**ASSIGN  
BUSTER**

### III. THE SOFTWARE OF 2025

In 2025, current trends in programming demonstrate an increasing in use of codes and coding language. Extrapolated from Boehm [6] coding is relevant in 2025, although it is more developed. Ko et al. [2] further explain that about JavaScript, which currently is by 2025 more of an internet protocol. The systems' development and efficiency has helped it to gain popularity in 2025. The software language provides numerous advantages, especially its compatibility with another system. Nevertheless, Boehm [6] notes that to continue enjoying the benefits of the model: current software engineers must incorporate it to the web browsers. Consequently, it; s gaining the support of the different browsers will maximize its usefulness. In 2025, there are new software languages which were developed to meet the extensive technology inventions of earlier 2020s.

; The imperative was the development of artificial intelligence (; AI;), which had some impact on the 2025 world of software engineering. Lowry [11] explains the adoption of AI techniques as tools to enhance the efficiency of prototyping. Moreover, the AI techniques prove to be useful in establishing flexible and easy to modify systems, especially in the innovative environments. Moreover, it sets a stable foundation in the evolution of user-friendly information systems [11].; The use of AI programming models in 2025 provides an easier modification and maintaining of software programs. Consequently, the advantages of the AI models encourage their adoption in different conventional fields including graphic user interface, constraint-based and objected-orientated programming [11]. The automotive system will manage to interpret different codes without needing the human

presence. The use of AI explains to increased challenges in the world of programmers. As that stands in 2025, there is a need to develop more software models compatible with the future technologies.

#### IV. SOFTWARE ENGINEERING IN 2040

From 2025, ones looks to 2040. The world is in a constant development engineered by advancement in the fields of technology, science, mathematics, and engineering. By the year 2040, as described by Boehm [6], most of the population will be knowledgeable about software engineering. The extensive use of software products and applications in the society will have been on a steady rise. In particular, the need to solve problems persistent in the community through the application of technology is a leading factor contributing towards the adaptability of software programs in everyday life. Software engineering [13] point to a significantly advanced scale of software to manage the future challenges in humanity. Industries and business firms are realizing the need to adapt automation as a strategy to increase output and compete for market leadership. An example of the progress in the application of software includes the current over 500 million codes managing the health care insurance. It is important to note that the use of code will continue in the decades to come. In particular, Software Engineering [13] indicates the change in government policies and services to incorporate the application of software technologies. According to Boehm [6], the software provides a solution to competitive markets, in particular through differentiation of products. The public is more advanced in software programs like data managements and is already reaping its benefits. Nevertheless, with the growing population of software engineers, it

<https://assignbuster.com/software-programs/>

is highly likely that more challenges will develop, especially in designing the programs and integrating their operations.

In regards to Boehm [6], use of electronic equipment is fast growing. This is particularly the case when studying its growth in the developing countries. The electronic equipment provides the human with efficient tools to efficiently manage tasks and improve performance. It is in the author's opinion that by the year 2040, the performance of activities will have heavily relied on electronic machinery. The imperative is the ability of this equipment in handling the task, a result of carefully designed software programs of the time. Already, challenges in the economic sector are forcing companies to cut down the employment rate. Moreover, firms are opting to invest in automation as an alternative to increasing the output without incurring massive expenditure in wages and taxes.

Knowledge in software development is likely to integrate with other relevant disciplines. In particular, Boehm [6] notes the growing relationship between software engineering and other forms of engineering. As described in the article, the current trends point to interdependency in knowledge, especially in the scientific fields. Therefore by 2040 experience in software development will have been available in multiple areas both scientific and nonscientific fields. The nature of the course will be of great benefit towards its application in other vital areas. Moreover, the population will need such skills in developing even more excellent ideas to tackle the challenges of the times. As development in different fields progresses, so does the complexity of problems. Therefore, with an obsolete direction in software engineering,

most of the world population will be in a position to utilize software or understand its operation.;

A state-of-the-art software engineering in 2040 will include the use of network-centric systems of systems (; NCSOS;). The system will revolutionize the software architecture to include a spiral process model. An advantage of the systems of systems network will be more efficient means of addressing problems in the world of software. Nevertheless, there will be more user-based programming models. Much of these programs will be highly user-friendly allowing an easier integration with other software packages. Boehm [6] explains the production of new products using exploratory development processes. The process will be more active in fields such as biology, biotechnology, and nanotechnology. The application of the process will be a factor on the need to provide the population with robust, reliable, dependable, and highly cost-effective programs in meeting the increase in demand.

Software engineering [13] explains on a neural- embodied and augmented programming. The realization of brain-interface is a possible development of the software engineering of the 2040 period. The development of the technology will involve making accounts for mental capabilities in line with the stressors of the time. Moreover, other important factors that need consideration in developing software in 2040 include the age of the population and the memory capacities in the humans. The creation of software will incorporate passive technologies applying both biological and neurological sensors to produce functional prototypes [13]. The sensors will collect signals whose uses include the provision of feedback. The availability <https://assignbuster.com/software-programs/>

of feedback places developers at a position of advantage in designing the models of the programs.

It is worthwhile to note the increasing shift of software towards cloud computing programs. The shift is a contribution of the benefits of the Internet-based computing that further achieves better and efficient data management. The system is highly versatile and cost profitable. By 2040, with increased human population, a majority of software will have operated under cloud computing systems. In regards to Jaworski et al. [7], the inevitable shift to include the application of cloud computing focuses on the scalability of the scheme.; Jaworski et al. further provide an example of the future of managing heavy traffic in the major cities [7]. The paper supports the application of cloud computing in developing an intelligent transportation system, AI, IOT, smart devices, automated data security, sensor technology and many other advanced interfaces for general public. Automated and self-driven cars will be available in 2040.

## V. MAJOR PROBLEMS AND CHALLENGES OF SOFTWARE ENGINEERING IN 2040

The major challenge will be to handle the bulks of data originating from different sources efficiently. Moreover, based on Lethbridge et al. [4], the increased use of the software will bring out the challenge of updating documentation. Much as it is a problem today, one can only imagine the magnitude of the problem in the decades to come. The adaptation of automation to solve problems in industrial production is not without challenges. Based on Vogel-Heuser et al. [9], a change in one sector of the

engineering affects other integrated areas. In particular, the development of new products to meet the need to automate requires putting together of technology and system engineering. The urgent challenge is how to integrate the different development phases of the various disciplines in creating the end product. Moreover, Vogel-Heuser et al. [9] highlight the difficulties in managing automation during the engineering stage.

Consequently, there will be a need to continually adjust the development of appropriate software to meet the engineering technology of the Inter-operate systems.

It is important to consider the end user as a necessary power in challenging the software engineering of the future. Vogel-Heuser et al. [9] note that, in automation, software developers will face the problem of designing programs suitable for the different customer needs. Nevertheless, there will be an issue in developing an integrated system for managing the various end-user applications. As mentioned, the trends in software engineering point to the development of interconnected systems to handle the increasing demand of software solution in everyday life.; Furthermore, as noted by Vyatkin [3], the growing demand and dependability in automation will reciprocate in an increase for developing more formal models in the software. As a result, engineers in the field will need new tools to meet the demand. About the challenge of managing the ultra-large systems, Northrop, et al. [12] describes three critical challenges including monitoring and assessment, design and evolution, and orchestration and control. The three sets of problems in the ULS systems highlight some of the key characteristics that developers, law and the society as a whole are prone to experience in

2040. In a greater extent, it will be challenging to efficiently develop a quality system that manages to integrate different end-users and maintains high efficiency.

One cannot possibly ignore the importance of power in running software program. Fuggetta and Di Nitto [10] highlight about energy consumption, which creates a significant challenge at present and will continue to do so in the future. The advent of mobile smartphone, artificial intelligence, tablets and other mobile devices influence the shift to portable devices. Moreover, with automation in the future, the population will turn into the online source of work, which indicates a high dependency on mobile equipment.

Consequently, software developers will be forced to develop programs that manage to concern power. Additionally, it will mean a redesigning of hardware resources to meet energy consumption, security, quality, testing and various dependencies between these processes.

The relevance of software developers will continue to decrease with the development of AI [14]. A significant factor contributing to their decline in performance is the availability of web designing tools available to the public. A disadvantage of the tool is that they do not require software. Moreover, AI replaces the human input in developing and managing software. Therefore, only a few developers are needed to achieve the different automation in various industries.

## VI. VISIONARY IDEAS OF OVERCOMING THE PROBLEMS AND CHALLENGES

As discussed in the paper, software engineering faces multiple challenges. In particular, with the advancement of its use in solving current and future

<https://assignbuster.com/software-programs/>



problems, it becomes necessary to identify probable ideas of tackling the problems to continue enjoying the benefits of the software. Therefore, I believe the development of a self-adaptive system will provide an efficient means of integrating the needs of different users. According to De Lemos et al. [1] a self-adaptive system will manage to self-configure allowing a smooth operation with the various requirements of the users. Moreover, a self-adaptive system comes with the advantage of being reliable, energy-efficient and dependable in the ever changing environment.

The system should be able to incorporate different languages to manage uncertainty. The availability of multiple languages makes the system flexible and much easy to understand, especially for the user. About the change in the run-time management of different systems, it is my opinion that a possible solution would be developing a universal program in run-time management. The program should work with a set of a command line language allowing different software to work simultaneously without producing an error.

Power conservation comes as a necessary field requiring new ideas to meet the ever increasing demand for power. The author believes the designing of an automatic power monitoring system will ensure that the software and hardware conserve energy. Suggestions from Gubbi et al. [8] confer that the need to manage energy as the usage in the future will surpass the production. Home use regulation of energy allows the people to be directly involved with IoT. Similarly, the use of the power monitoring system should enable the people to monitor their power consumption at any given place and time. I further hold the opinion that the surveillance system should

incorporate time factor whereby the supply of power to run the system is under time regulation. An advantage of the model will be the active management of energy since it is only available for software usage at the particular time of the day.

Gubbi et al. [8] further explain that the effective power consumption is attainable by the application of a continuous monitoring system. The information obtained will assist in designing energy-conserving hardware. Moreover, it will be quite relevant for future research on software engineering. Currently, power conservation measures address the regulation of its use in domestic appliances. Therefore, with the increased use of software and its application, it is quite essential also to design methods of conserving energy.;