

УДК 617.57.77

Danylo Kushnirenko¹, Valentyna Kuzmenko²

¹student of group CST-139 National University “Zaporizhzhia Polytechnic”

²senior teacher National University “Zaporizhzhia Polytechnic”

HISTORY OF SOFTWARE ENGINEERING

The first application of the term "software engineering" dates from August 1966, but the year of the emergence of software engineering (IPZ) as an industry is 1968, in which the first conference in Germany was called "Software engineering". The main topic of the conference was software design, production and maintenance. One of the keynote speakers at the conference, P. Naur, noted that

the work of software designers is similar to that of architects and civil engineers, especially those involved in the design of large heterogeneous structures such as cities and industrial enterprises. The term "software engineering" was deliberately chosen as provocative: "this term meant that software production should be based on the same type of theoretical foundations and practical applications as in traditional engineering fields." The origin of software engineering was associated with the software crisis - a term suggested by Program Committee Chairman FL Bauer. In his opinion, the crisis was the impossibility of applying "handicraft" (semi-intuitive) methods of development for the production of large scalable software systems. Existing software is developed by amateurs (no matter where - at universities, companies or in production) through the skill of singles (at universities) or the large number of workers ("one million monkeys") in production, is unreliable and needs constant "maintenance" (and the word "maintenance" is misused to refer to battles and failures that are expected from the manufacturer at the outset), is sloppy, transparent and imperfect (or at least too costly in to do this). And finally, there is the software arrives too late and costs more than expected, and does not live up to the expectations it relied on. As a result of the conference in 1971, FL Bauer gave a half-joking definition of IPE as a piece of computer science that is too heavy for computer scientists, and more serious as the creation and use of sound engineering principles to obtain economical, reliable and efficient work on real computers software.

According to such conference participants, A.J. Perlis and FL Bauer, mathematical preparation is not necessary for the design of software, but its presence adds to the software project elegance, since software systems are mathematical in nature and must be built in levels and modules that form a mathematical structure.

The general design criteria (common to different systems), user requirements, reliability, and logical completeness were chosen as the main design criteria. Design technologies discussed the sequence of design steps, the design of a software project, the provision of feedback through monitoring and modeling, the use of high-level programming languages, and more.

In December 1969, a third symposium on IPE was held in the United States, the first volume of materials being opened by a programmatic report by YT Tu "A New Profession: Software Engineering". Describing the role of engineering in the development of mankind, the author indicates that it has largely freed man from physical labor and routine mental activities, providing tools for both new scientific discoveries and new kinds of creativity. The rapid development of engineering, the emergence of new industries changed the content of engineering education - from utilitarian to scientific and theoretical. Just as the invention of the steam engine in the late eighteenth century made it possible to replace the muscular power of humans and animals with the driving force of machines, the invention of the digital

computer after World War II made it possible to replace many human mental tasks, such as arithmetic, data storage and accounting, computing operations.

We are now moving to a stage where it is advisable to replace some of the higher human mental tasks with machines. This includes the ability to recognize patterns, read images, process data in natural language, retrieve information, and make smart decisions. To do this, the author proposes to use the principles of IPE. According to YT Tu, IPE should cover computer architecture, system and application programming. With regard to IPE, the author emphasizes that in order to make significant progress in software development, we must have a much stronger scientific foundation.