

D. Zavadsky, student
V. Kovalchuk, PhD, Ass. Prof., research advisor
S. Kobzar, Senior lecturer, language advisor
Zhytomyr State Technological University

THE KEY FEATURES AND PROBLEMS OF MODERN SOFTWARE PROJECTS

Experience in creating software systems shows that this is a complex and time-consuming work that requires highly skilled professionals involved in it. However, up for now the establishment of such systems is often performed on an intuitive level, using formalized methods based on the art of practical experience, expert opinions and costly experimental verification of the quality of the software operation. The problems of creating software depend on its properties. As early as 1975, Frederick Brooks, having analyzed his unique (at that time) experience in management of the largest projects of developing the operating system OS /360, made a list of the inherent properties of software: complexity, consistency, changeability and invisibility [1].

At the end of the 60s of the last century in the United States a phenomenon called "software crisis" was noted. This was expressed in the fact that large projects were performed behind schedule or within budget, designed products did not possess the required functionality, performance was low, the quality of the software did not suit the consumers [2]. One reason for the prevalence of "the chaotic" process of software development was the desire to save money on the stage of development without spending time and money on training and implementation process of software development. The same reason was for the "heaviness" of technological processes. "Heavy" process has the following features:

- the necessity to document every action of developers;
- a lot of work products (primarily – documents) produced in the bureaucratic atmosphere;
- the lack of flexibility;
- determinism (long-term detailed planning and predictability of all activities, as well as the distribution of human resources for the long term, covering a large part of the project).

The alternative to "heavy" process is adaptive (flexible) process based on the principles of "rapid software development", intensively developed in the last decade. It should be clearly understood that for all the merits of rapid software development, this approach is not universal and is applicable only to projects of a certain class. To characterize such projects Alistair Cockburn has introduced two parameters – the criticality and the scale. Criticality is determined by the consequences caused by the defects in the software, its level can be one of four values:

- C – defects cause the loss of a menity;
- D – defects cause the loss of recoverable resources (material or financial);

- E – defects cause the loss of irreplaceable resources;
- L – defects pose a threat to human life [3].

The determining factor for the development of software engineering is the desire of software companies to increase their productivity, cut time to market, control costs and quickly obtain benefit from the investments [4]. These goals promotes the use of development environments that can reduce the complexity of the processes of software development, increase their efficiency, reduce development costs and maximize the potential of new technologies. Systematic, reasoned approach to the selection and use of Software Technology can reduce the time and improve the quality of software development, provide a high degree of independence from specific developers, as well as reduce the cost of software development and maintenance .

REFERENCES

1. F.Brooks-The Mythical Man-Month: Essays on Software Engineering: Trans. from English.-SPb .:Symbol-Plus 1999
2. ru.wikipedia.org/wiki/Разработка_программного_обеспечения
3. <http://citforum.ru/programming/application/program/>
- a. <http://dailyold.sec.ru/dailypblshow.cfm?rid=45&pid=11800&pos=55&stp=25>