UDC 004.41:004.5+656.7(082)

V.O. Talalaiev, PhD, Y.M. Bezkorovaina
(National Aviation University, Ukraine)

Organic modeling method in the tasks of developing specifications on the software of aviation automated control system

The development of software aviation automated control systems requires the use of techniques for the organic modeling of specifications of requirements. The aviation automation systems are quite complex systems that require distinct control, so their software must not be defective. Defects could be introduced at each stage of software development, but the very first ones are added at the stage of the requirements specifications.

The achievement of the required level of functionality and high-quality software of aviation automated control systems makes it necessary to adhere to the strict norms of defectless of the results of all stages of the life cycle of the software. The high cost from losses of defects requires a lot of effort to plan and implement the core process steps of developing and analyzing software product requirements. Aviation Automated Control Systems software (AACS) in most cases relates to critical uses, and therefore high software requirements for its defectless are advanced to software development and maintenance processes.

Much of the defects in the software specifications can be detected and eliminated at the modeling stage of requirements. A prerequisite for the high efficiency of modeling requirements, a perspective identifying possible defects in the specifications for the software, is the reproduction of process models that are maximally adequate to the actual conditions of application of future software AACS. In most known methodologies and methods of developing and analyzing requirements, the adequacy of requirements models is achieved through the use of effective techniques, for example, involving experts in the processes of development and analysis of requirements. Nowadays it has become one of the main reasons for the widespread use of agile software development methodologies. Agile methodologies involve the participation of customer representatives in the process of specification development is a prerequisite.

Considering well-known and successful software products and systems that were technically advanced and did not find their widespread use, one can conclude that the reasons for their "short life" are the discrepancy between the expectations of users and their real properties. As a result, it is possible to highlight some inconsistencies in modern development processes and may be due to:

1) the differences in views on the functionality of the future software product of stakeholders in the process of developing requirements: "customers" and "users";

2) the differences between the role behavior of users of the software product at the time of issue, compared with the adopted standard of their performance at the stage of development of requirements.

Both of these reasons for inconsistencies in the practice of developing and implementing software products are known as factors in their early aging, or even rejection in the early stages of implementation and use.

In such cases, attempts to eliminate the inconsistency of functional properties, as a rule, lead to significant costs for the refinement and maintenance of the software product, creates significant problems in its use, constraints timely commissioning, complicates the logic of the software product, increases the number of "hidden" defects, causing the appearance of additional costs of the customer-developer, etc.

Avoid systemic mismatches in software specifications may be due to the introduction of effective methods and models at the stage of analysis that provides the formation of requirements with a minimum level of defects. This state of the development process is achieved when the established requirements for software products on their functionality will be as close as possible to the real conditions for their future use and are organically "embedded" in the existing application processes subject processes. That is why the authors establish the indicated methods and models as organic.

Their application involves, firstly, the use of prognostic models of rational and rational behavior of users, which in combination with the capabilities of future software provides the maximum effect of the mutual organic interaction in the process of achieving the goals of the application system's business processes. Secondly, the proposed methods are intended to include in the specifications of the software adaptation mechanisms that provide customization of software properties to specific peculiarities of user behavior in real terms of the application of automated systems.

Figure 1 shows the general scheme of the proposed methodology in the requirements management processes.

As can be seen from the scheme in Fig. 1, the organicity of the generated requirements were achieved through the use of three types of models: models of the business processes, models of the requirements and models of rational behavior of users. Using the latter, as shown in the diagram, allows you to achieve the required level of organicity of the process of forming requirements, through the implementation of inverse correctional connections to adapt the requirements to changes in models of rational behavior of the user. At the same time, as a valuable artifact, the source data is generated to create user training programs (users+) to acquire the necessary competencies of rational behavior when using the software.

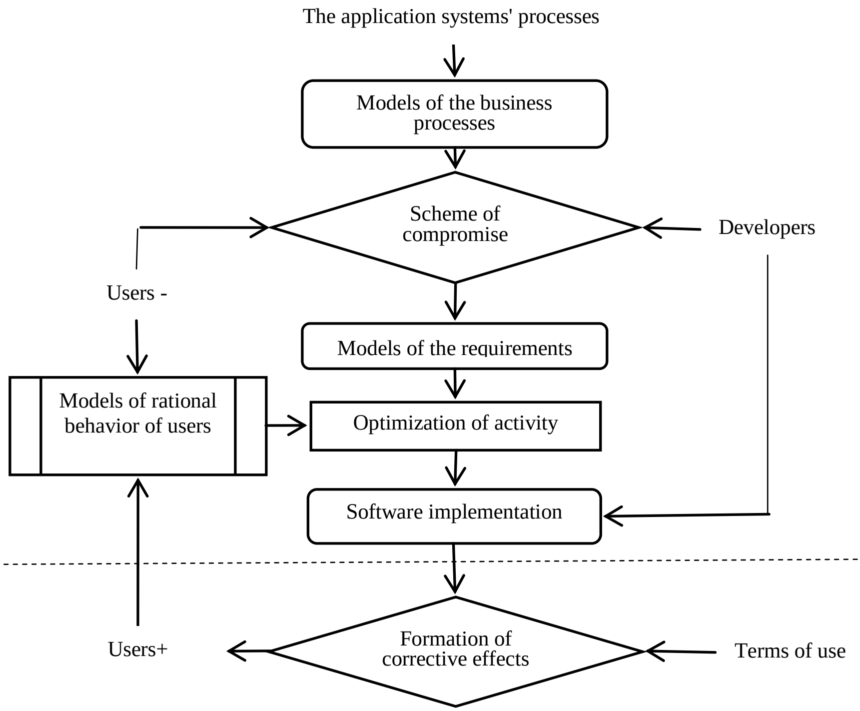

Fig. 1. The general scheme of the methodology in the requirements management processes

Conclusions

The introduction of such an approach allows at the time of the release of the software not only to ensure the maximum correspondence of its properties but also to achieve controllability of the process of adjusting the behavior of software to the real conditions of their application. This effect from the introduction of the proposed methods and models for software applications in the AACS in many cases becomes decisive because of the criticality of such systems to the effectiveness of human-computer interactions.

References

1. Karl Wiegers, Joy Beatty Software Requirements, Third Edition. – Microsoft Press. – 2013. – 673 p.
2. Kent Beck, Cynthia Andres. Extreme programming explained. Embrace change. Second edition – Addison-Wesley. – 2004. – 224 p.
3. M. Kamenova, A. Gromov, M. Ferapontov, A. Shmatalyuk. Business modeling. ARIS Methodology – Vest-MetaTechnology. – 2002. – 333 p.