APPLYING SOFTWARE-MATHEMATICAL MODELS OF ONBOARD EQUIPMENT TO DEVELOP ONBOARD SOFTWARE

Yu. S. Lomaev, I. A. Ivanov, A. V. Tolstykh, E. V. Islent'ev

JSC "Academician M. F. Reshetnev "Information Satellite Systems"
52, Lenina St., Zheleznogorsk, Krasnoyarsk region, 662972, Russian Federation
E-mail: lomaif@rambler.ru

This paper deals with the testing of the functioning logic of spacecraft subsystems at the stage of developing system onboard software. The increasing complexity of the structure and operation logic of spacecraft due to the increased requirements in terms of providing consumers with information services (navigation, satellite monitoring of transport, geodesy, communications etc.) demands maintaining the reliability of uninterrupted operation, the implementation of automated parrying of emergency situations during the operation of spacecraft onboard equipment. In order to meet these requirements, it is necessary to test the interaction of onboard equipment and onboard integrated computing complex software that implements the target-oriented operation of spacecraft onboard systems. In such a case, meeting the requirements for reliability increase of onboard software should not lead to the increase of the manufacturing period of spacecraft.

In this work we propose the approach for testing information and logical interaction between onboard equipment and software of a spacecraft onboard integrated computing complex with the use of a laboratory testing sample unit and a software-mathematical model. We described the basic concepts of conducting two-stage testing of onboard software, involving autonomous and system testing on the ground testing complex. The proposed approach is applied as part of the onboard software development cycle in accordance with the standards of the JSC "Academician M.F. Reshetnev "Information Satellite Systems".

The approach proposed in this work helps reduce the number of errors during onboard software development and testing of information and logical interaction between onboard equipment and a spacecraft as a whole in every operation mode.

Keywords: spacecraft, onboard equipment, onboard software, software-mathematical model, laboratory testing, ground testing complex.

ПРИМЕНЕНИЕ ПРОГРАММНО-МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ БОРТОВОЙ АППАРАТУРЫ ПРИ РАЗРАБОТКЕ БОРТОВОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Ю. С. Ломаев, И. А. Иванов, А. В. Толстых, Е. В. Ислентьев

АО «Информационные спутниковые системы» имени академика М. Ф. Решетнева»
Российская Федерация, 662972, г. Железногорск Красноярского края, ул. Ленина, 52
E-mail: lomaif@rambler.ru

Работа посвящена отработке логики функционирования подсистем космического аппарата на этапе разработки бортового программного обеспечения системы. Усложнение структуры и логики функционирования космических аппаратов в связи с повышением требований в части обеспечения потребителей информационными услугами (навигации, спутникового мониторинга транспорта, геодезии, связи и т.д.) требует поддержания надежности и бесперебойного функционирования, реализации автоматизированного парирования нештатных ситуаций при работе бортовых аппаратур космических аппаратов. Для достижения поставленных требований необходима отладка взаимодействия бортовой аппаратуры и программного обеспечения бортового интегрированного вычислительного комплекса, реализующего целевое функционирование бортовых систем космического аппарата. При этом выполнение требований по увеличению надёжности бортового программного обеспечения не должно приводить к увеличению сроков изготовления космического аппарата.

В настоящей работе предложен подход для отработки информационно-логического взаимодействия бортовой аппаратуры и программного обеспечения бортового интегрированного вычислительного комплекса космического аппарата с применением лабораторно-отработочного изделия и программно-математической модели. Описаны основные идеи проведения двухуровневого тестирования бортового программного обеспечения, включающего в себя автономное и системное тестирование на наземном отладочном комплексе. Указанный подход применен в рамках реализации цикла разработки бортового программного обеспечения, проводимого согласно стандартам АО «Информационные спутниковые системы» имени академика М. Ф. Решетнева».

Предложенный в работе подход способствует сокращению ошибок при разработке бортового программного обеспечения и проверке информационно-логического взаимодействия бортовой аппаратуры и космического аппарата в целом во всех режимах функционирования.

Ключевые слова: космический аппарат, бортовая аппаратура, бортовое программное обеспечение, программно-математическая модель, лабораторно-отработочные испытания, наземный отладочный комплекс.

Introduction. Currently, spacecraft have a crucial role in providing consumers with communications, television broadcasting and navigation services. Depending on the purpose, spacecraft contain a specific set of onboard equipment. The problems solved by individual instruments determine the functioning and logic of the operation of individual systems and a spacecraft as a whole. To ensure a reliable uninterrupted operation [1] and automated management of emergency situations arising from the operation of onboard equipment, it is necessary to adjust the interaction with the software of the onboard integrated computing system of spacecraft [2; 3].

The development of onboard software. A spacecraft includes a number of target and auxiliary equipment (spacecraft systems), which is determined according to the purpose of a spacecraft. For all onboard equipment, specialists develop software as part of the onboard software, as well as software as a part of equipment ensuring the interaction of this equipment with the onboard control complex of a spacecraft.

Onboard software is a combination of the software of onboard systems operating in single hardware and software environment of the onboard integrated computing system of a spacecraft [4]. Each component of the onboard software is a functional part of its onboard system and, together with its hardware, solves the problems assigned to this onboard system.

There are many approaches to software development related to its design. The development, autonomous and system testing of onboard software are key stages of the lifecycle of a spacecraft [5], which provide the interaction and logic of functioning data and instruments within its systems.

Let us consider the standard cycle of the development of onboard software:

1) defining the requirements to system software;

2) developing the architectural design of system software;

3) developing and autonomous testing of system software;

4) system testing of system software in system modes;

5) testing system software in spacecraft modes;

6) fit testing of software and onboard equipment as part of a spacecraft at the stage of electrical and radio tests and flight tests;

7) maintaining system software at the stage of normal operation.

If necessary, we return to the previous stages in the standard cycle of developing onboard software. In case of the successful completion of these stages we use the developed onboard software for work with spacecraft equipment and maintain the onboard software operation with the equipment in the spacecraft until the end of its active life. The onboard software architecture has the following form in general terms (fig. 1).

While developing the onboard software, it is necessary to conduct detailed testing of the interaction of software that is part of a spacecraft system and onboard equipment in order to detect errors before conducting integration tests of software and equipment as part of a spacecraft. Frequently, when errors are detected, changing the software of onboard equipment is no longer possible, since it may contain one-time programmable memories, or error correction may take much time, which leads to an increase in the duration of spacecraft testing. In this case, when abnormal situations occur, there is a possibility of correcting software failures of the onboard equipment from an onboard control complex by modifying its software.

The use of a mathematical software model. One of the ways to test software in the development of onboard software is to simulate the logic of the equipment by creating its software and mathematical model.

Software and mathematical model is a complex of the implemented logic of equipment operation in the form of a program code, through which we receive and transmit information when interacting with the onboard software.

When creating a software model of equipment, it is not necessary to have a high specification of the description of its internal structure and logic of functioning, but only to imitate those features of the equipment that are important from the point of view of its interaction with the external environment.

Fig. 2 shows the scheme of testing onboard software and software-mathematical model.

In the onboard software the logic of the interaction of equipment and the onboard control complex is implemented. The input information is the initial data, which contain the requirements for the development of onboard software.

The output information comprises the results of the stand-alone testing of the onboard software through debugging packages. The debug package is a pseudo-code for checking whether it is possible to achieve the

execution of a certain branch of the logic of onboard software.

As a result of package debugging, we determine the execution time, the size of the stack used, the coverage of the logic of the executable program code.

The essence of the debugging of information interaction between the software-mathematical model and the onboard control complex is to check the reception and transmission of various types of information messages, such as control commands, telemetry information, onboard time, software arrays, receipts, etc. Information exchange between the equipment and the BUD is performed via a multiplex exchange channel according to GOST R 52070–2003 (ГОСТ Р 52070–2003) [6] in accordance with the protocol of information and logic interaction with this equipment, developed by the equipment manufacturer and agreed with the developer of the onboard software.

In case of the successful normal operation of the interaction between the onboard control complex and the software-mathematical model, we check the occurrence of abnormal situations in information exchange: hardware malfunctions, failures of the multiplex exchange channel, errors in information messages, temperature failures, failures in signal power, current strength, voltage and other malfunctions.

System testing on the ground debugging complex. The result of the procedures described above is the integration of the onboard software and the software-mathematical model with the aim of conducting system testing on the ground debugging complex [7; 8]. A ground testing complex is the integrated environment for testing the onboard software and the software-mathematical model instruments within a spacecraft. The testing of the interaction occurs by checking the execution of the program logic using sequence diagrams [9].
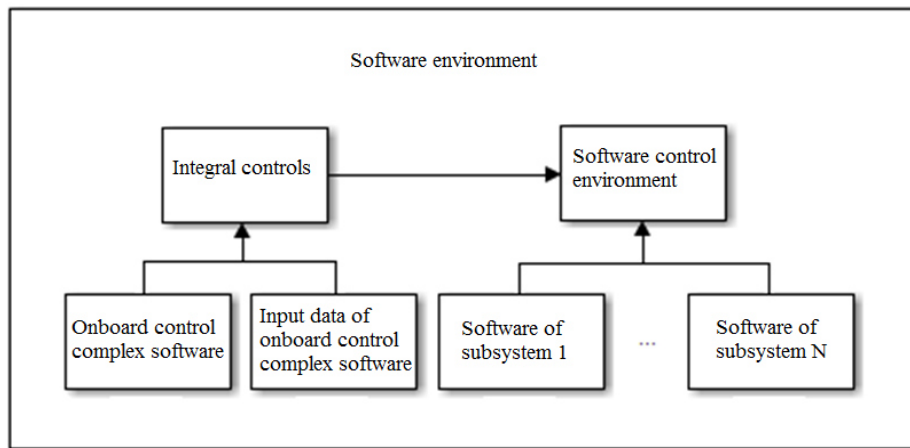

Fig. 1. Onboard software architecture

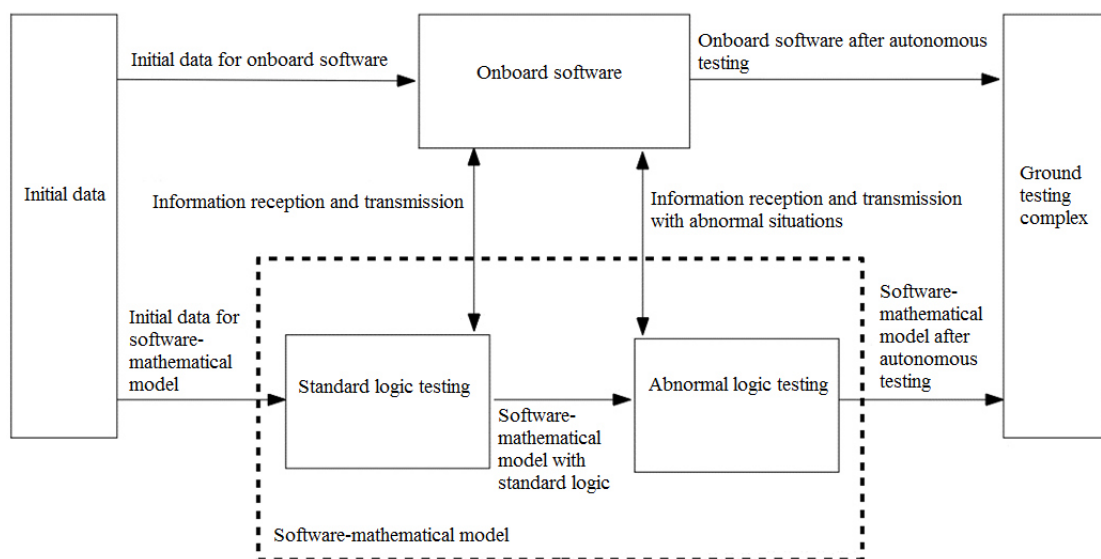Рис. 1. Архитектура бортового программного обеспечения

Fig. 2. Testing of onboard software and of a software-mathematical model

Рис. 2. Отработка бортового программного обеспечения
и программно-математической модели

A sequence diagram is a pseudo-code for system testing, based on a graphical formalized language describing the procedures and test options, allowing automating a testing process.

The result of this stage is the onboard software of the equipment, ready for testing in the spacecraft mode on the ground testing complex.

In addition, the ground testing complex has the ability to specify various initial data of the spacecraft operation, both in general and each spacecraft system in particular; that allows testing the onboard software for the whole set of its functionality. Fig. 3 shows the ground testing complex architecture.

Sample application for laboratory testing. Another way to test the onboard software is to use a laboratory testing sample [10; 11]. A laboratory testing sample is a special sample of equipment for testing, which is a prototype of standard equipment in terms of functionality and basic technical characteristics, on which we develop software and technical solutions. Such a product is cheaper than standard equipment and it is manufactured faster, therefore there is a possibility of the onboard software testing before the completion of the production of a standard sample. Fig. 4 shows the scheme for testing the interaction of onboard software with the laboratory testing sample.
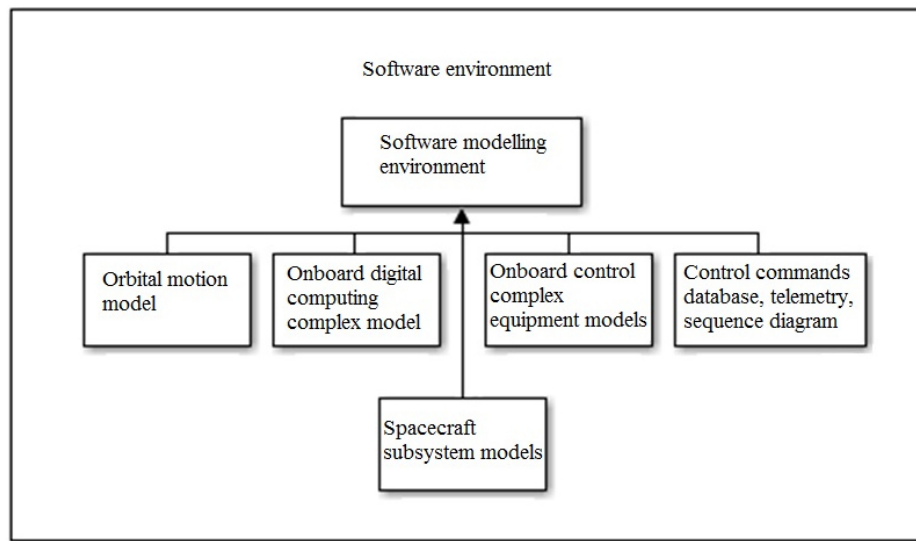

Fig. 3. Ground testing complex architecture

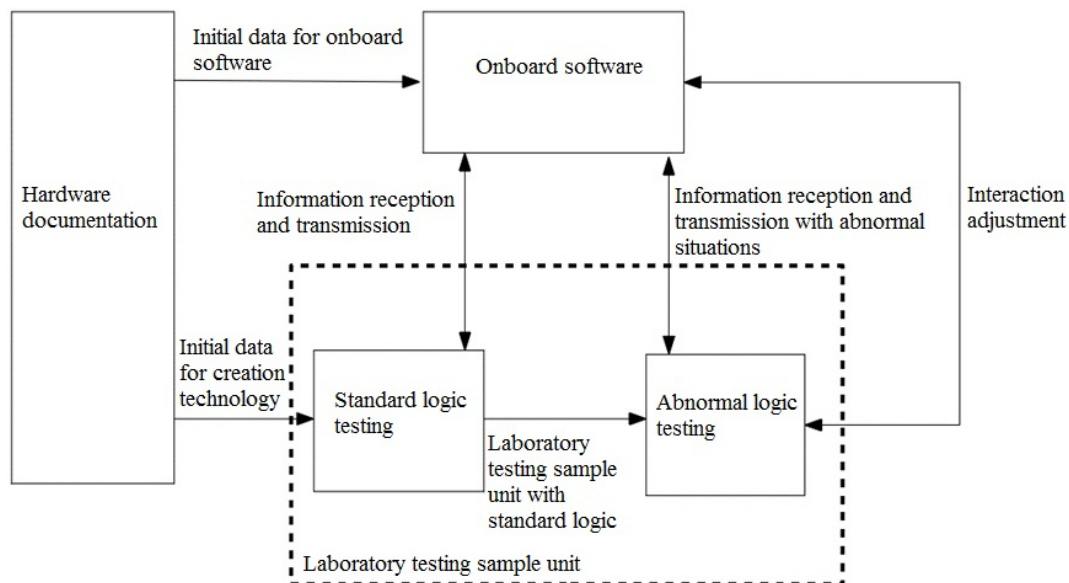Рис. 3. Архитектура наземного отладочного комплекса

Fig. 4. Testing the interaction between onboard software and a laboratory testing sample unit

Рис. 4. Отработка взаимодействия бортового программного обеспечения с образцом
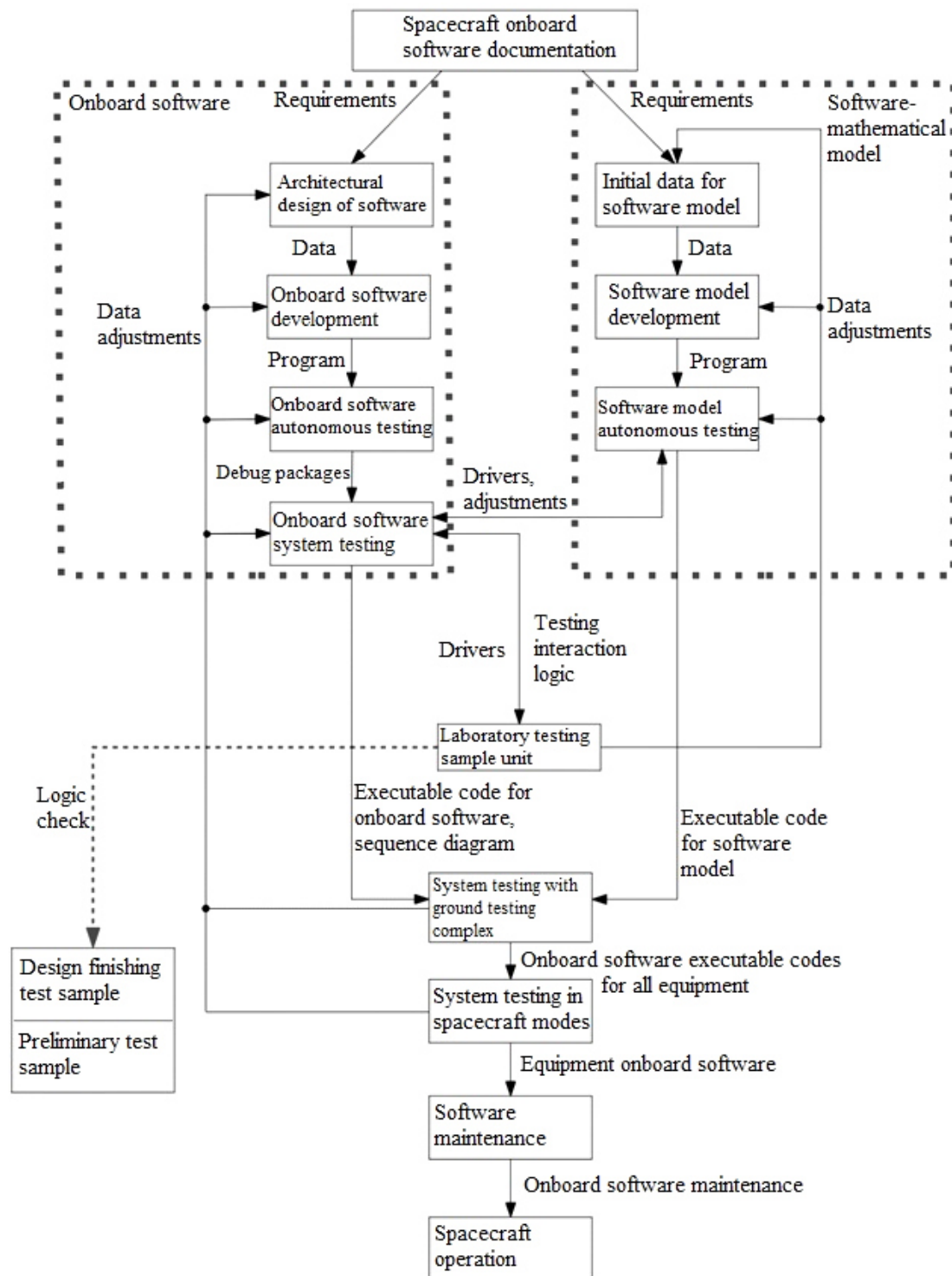для лабораторно-отработочных испытаний

Fig. 5. The general scheme of the development of onboard software with the application
of a software-mathematical model, and laboratory testing sample unit

Рис. 5. Общая схема разработки БПО с использованием ПММ, образца ЛОИ

The input information is the logic of the functioning of the sample, which is governed by the requirements for the creation of an experimental product. The output information is the executable code of the onboard software for testing with the ground testing complex.

As part of the development of software interaction with the equipment, we conduct testing to check the standard and abnormal logic of the laboratory testing sample.

The result is a corrected logic of the equipment, which we take into account at further stages of the development of onboard software. Fig. 5 shows the proposed software development scheme for onboard spacecraft systems [12; 13].

The particularities of using a software-mathematical model and a laboratory testing sample. We develop the onboard software and the software-mathematical model in parallel, and we pair them with the help of software drivers.

The peculiarity of using the software-mathematical model is to simulate the logic of the equipment in both regular and abnormal situations in which there are predictable failures in information exchange between it and the onboard control complex [14; 15]. We perform testing of the docking of the onboard software and the software-mathematical model during the delivery of the software to the ground testing complex, which allows us to prevent errors in the logic of the functioning of the onboard software at subsequent stages.

The use of the laboratory testing sample contributes to the accurate reflection of the procedures for recording and reading information from the point of view of the logic of operation of the device.

Testing on the laboratory testing sample gives a complete picture of the logic of the equipment, the physical component of the equipment, the processor, the runtime and the number of operations for a given time, which allows us to work out the hardware software and onboard software. In this case, there is information about real-time execution of operations, making it possible to adjust the system resources in advance (allocated memory, execution time, pauses during operation, etc.).

The use of the laboratory testing sample allows to reduce the time for making changes to the code of the onboard software, since the introduction of changes at the system testing stage in spacecraft modes requires consistency in working with the developers of other systems. The laboratory testing sample serves as a prototype for design finishing test sample and preliminary test sample, we take into account the adjustments for the laboratory testing sample on the indicated samples.

The advantages of the described methods in the development of onboard software. The use of the proposed methodology for the development of system software, including direct interaction with the laboratory testing sample and the use of the software-mathematical model, contributes to the reduction of time and parity of errors in the development cycle, autonomous and system testing of software.

The benefits of using a software-mathematical model is to organize the development of regular and emergency situations in the flight control center, to provide training for operation personnel, which contributes to rapid response when such situations occur during the operation of products.

Conclusion. The need to develop and use a software-mathematical model to test the onboard software of spacecraft is determined by several factors. Firstly, with the help of the software-mathematical model at the NOC, we sreated and tested the sequence diagrams, which are later used in testing and final adjusting actual onboard equipment and spacecraft systems, which can significantly reduce the time required for software testing and final adjustment. Secondly, the development stages of software and hardware run in parallel, therefore the possibility of developing software with real hardware is not available. Consequently, there is a need for its software simulation. The use of a software-mathematical model when testing onboard software allows us to produce better and more complete development of the logic of the operation of the equipment and various emergency situations. Thirdly, it is necessary to envisage the possibility of making changes in the onboard software after launching the spacecraft, changes can be preliminarily worked out on software models of spacecraft systems before they are installed in the onboard integrated computing complex; that helps prevent emergency situations and increases the overall reliability of spacecraft.

The modified methodology for the development of onboard software is used in the development of software of systems in the process of the creation of the "Glonass-K" spacecraft.

References

1. Tyugashev A. A., Iljin I. A., Ermakov I. E. [Ways of increasing software reliability and quality in space industry]. *Upravlenie bolshimi sistemami.* 2012, No. 39, P. 288–299 (In Russ.).

2. Tzapko G. P., Martynov Ya. A. [Unified information environment for creating and supporting onboard software of navigation and communication satellites]. *Doklady TUSURa.* 2015, No. 3(37), P. 97–102 (In Russ.).

3. Koltashev A. A. [Main principles of system testing and confirmation of satellite onboard software]. *Vestnik SibGAU.* 2010, No. 1(27), P. 4–7 (In Russ.).

4. Antamoshkin A. A., Koltashev A. A. [Technological aspects of creating communication satellite onboard software]. *Vestnik SibGAU.* 2005, No. 6, P. 93–95 (In Russ.).

5. Koltashev A. A. [Effective technology for controlling onboard software lifecycle of communication and navigation satellites]. *Aviakosmicheskoe priborostroenie.* 2006, No. 12, P. 20–25 (In Russ.).

6. *GOST R 52070–2003. Interfeis magistralniy posledovatelniy sistemy electronnykh moduley* [State standard R 52070-2003. Trunk serial interface for electronic modules system]. Moscow, Standartinform Publ., 2003. 24 c.

7. Mostovoi Ya. A. [Simulation mathematical model of the external environment in the life cycle of the onboard sowtware for managing the space platform]. *Komp'yuternaya optika.* 2012, No. 3(36), P. 412–418 (In Russ.).

8. Lomaev Yu. S. [Application of onboard software system testing]. *Aktualnye problemy aviatsii i kosmonavtiki.* 2016, No. 1, P. 531–532 (In Russ.).

9. Shoshmina I. V. [Design of software onboard control systems with verification support]. *Modelirovanie i analiz informatsionnyh system.* 2010, No. 4, P. 125–136 (In Russ.).

10. Lomaev Yu. S., Ivanov I. A. [Usage of laboratory testing sample unit for reducing time costs for onboard software development and testing]. *Sbornik tezisov IV nauchno-technicheskoy konferentsii molodyh spetsialistov "Razrabotka, proizvodstvo, ispytaniya i ekspluatatsiya*

kosmicheskih apparatov i sistem" [Theses compilation of IV science-technical conference of young specialists "Development, manufacturing, testing and operation of spacecrafts and systems"]. JSC "ISS", Zheleznogorsk, 2017, P. 99–101 (In Russ.).

11. Lomaev Yu. S., Ivanov I. A. [Usage of laboratory testing sample unit for optimization of onboard software development]. *Sbornik tezisov IV Vserossiyskoy nauchno-technicheskoy konferentsii "Sistemy svyazi i radionavi-gatsii"* [Theses compilation of IV All-Russian science-technical conference "Communication and radionavigation systems"]. JSC "NPP Radiosvyaz", Krasnoyarsk, 2017, P. 339–342 (In Russ.).

12. Lomaev Yu. S., Tolstykh A. V., Zvonar V. D. [Modification of onboard software development stages]. *VIII ezhegodnaya nauchno-technicheskaya konferentsiya molodyh uchenyh i spetsialistov FGUP TSNIIMASH* [VIII annual science-technical conference of FGUP TSNII-MASH young scientists and specialists]. Moscow, 3–6 April 2018, P. 41–43 (In Russ.).

13. Lomaev Yu. S., Tolstykh A. V. [Software testing approach modification for onboard software development]. *Tezisy dokladov X Obscherossiyskoy molodezhnoy nauchno-tekhnicheskoy konferentsii "Molodezh. Tekhnika. Kosmos"* [Report theses of X All-Russian youth science-technical conference "Youth. Technics. Space."]. Saint-Petersburg, 18–20 April 2018, P. 30–31 (In Russ.).

14. Lomaev Yu. S., Tolstykh A. V. [Usage of software-mathematical model for onboard software complex development]. *Aktualnye problemy aviatsii i kosmonavtiki"*. 2018, Vol. 1, P. 119–120 (In Russ.).

15. Ivanov I. A., Antropov N. R., Deryshev R. A. [Usage of program models for testing spacecraft onboard software]. *Trudy X Obsherossiyskoy molodezhnoy nauchno-tekhnicheskoy konferentsii "Molodezh. Tekhnika. Kosmos"* [Report theses of X All-Russian youth science-technical conference "Youth. Technics. Space."]. Saint-Petersburg, 18–20 April 2018, Vol. 2, P. 69–73 (In Russ.).

Библиографические ссылки

1. Тюгашев А. А., Ильин И. А., Ермаков И. Е. Пути повышения надежности и качества программного обеспечения в космической отрасли // Управление большими системами. 2012. № 39. С. 288–299.

2. Цапко Г. П., Мартынов Я. А. Единая информационная среда создания и сопровождения бортового программного обеспечения спутников навигации и связи // Доклады ТУСУРа. 2015. № 3(37). С. 97–102.

3. Колташев А. А. Основные принципы системного тестирования и подтверждения бортового программного обеспечения спутников // Вестник СибГАУ. 2010. № 1(27). С. 4–7.

4. Антамошкин А. А., Колташев А. А. Технологические аспекты создания бортового программного обеспечения спутников связи // Вестник СибГАУ. 2005. № 6. С. 93–95.

5. Колташев А. А. Эффективная технология управления циклом жизни бортового программного обеспечения спутников связи и навигации // Авиакосмическое приборостроение. 2006. № 12. С. 20–25.

6. ГОСТ Р 52070–2003. Интерфейс магистральный последовательный системы электронных модулей. М. : ИПК Издательство стандартов, 2003. 24 с.

7. Мостовой Я. А. Имитационная математическая модель внешней среды в жизненном цикле бортового программного обеспечения управления космической платформой // Компьютерная оптика. 2012. № 3(36). С. 412–418.

8. Ломаев Ю. С. Применение системного тестирования бортового программного обеспечения // Актуальные проблемы авиации и космонавтики : сб. тез. Всеросс. науч.-практич. конф. 2016. № 1. С. 531–532.

9. Шошмина И. В. Проектирование программных бортовых систем управления с поддержкой верификации // Моделирование и анализ информационных систем. 2010. № 4. С. 125–136.

10. Ломаев Ю. С., Иванов И. А. Применение лабораторно-отработочного изделия для сокращения временных затрат на разработку и тестирование бортового программного обеспечения // Разработка, производство, испытания и эксплуатация космических аппаратов и систем : сб. тез. IV науч.-техн. конф. молодых специалистов / «АО «ИСС». Железногорск, 2017. С. 99–101.

11. Ломаев Ю. С., Иванов И. А. Применение лабораторно-отработочного образца для оптимизации разработки бортового программного обеспечения // Системы связи и радионавигации : сб. тез. IV Всеросс. науч.-техн. конф. / АО «НПП «Радиосвязь». Красноярск, 2017. С. 339–342.

12. Ломаев Ю. С., Толстых А. В., Звонарь В. Д. Модификация этапов разработки бортового программного обеспечения // Тез. докладов VIII ежегодной науч.-техн. конф. молодых ученых и специалистов ФГУП ЦНИИМАШ. Москва, 3–6 апреля 2018 г. С. 41–43.

13. Ломаев Ю. С., Толстых А. В. Модификация подходов проведения тестирования программного обеспечения при разработке бортового программного обеспечения // Молодежь. Техника. Космос : тез. докладов X Общеросс. молодежной науч.-техн. конф. СПб, 18-20 апреля 2018 г. С. 30–31.

14. Ломаев Ю. С., Толстых А. В. Применение программно-математической модели при разработке бортового программного комплекса // Актуальные проблемы авиации и космонавтики : материалы III междунар. науч.-практич. конф. творческой молодежи. Красноярск, 9–13 апреля 2018 г. Т. 1. С. 119–120.

15. Иванов И. А., Антропов Н. Р., Дерышев Р. А. Применение программных моделей для отработки бортового программного обеспечения космических аппаратов // Молодежь. Техника. Космос : тр. X Общеросс. молодежной науч.-техн. конф. СПб, 2018. Т. 2. С. 69–73.

Ломаев Юрий Сергеевич – инженер второй категории; АО «Информационные спутниковые системы» имени академика М. Ф. Решетнева». E-mail: lomaif@rambler.ru.

Иванов Илья Андреевич – кандидат технических наук, инженер второй категории; АО «Информационные спутниковые системы» имени академика М. Ф. Решетнева». E-mail: ilyaiv92@gmail.com.

Толстых Анастасия Владимировна – техник; АО «Информационные спутниковые системы» имени академика М. Ф. Решетнева». E-mail: avtolstykh1@gmail.com.

Ислентьев Евгений Владимирович – начальник сектора; АО «Информационные спутниковые системы» имени академика М. Ф. Решетнева». E-mail: islentev@iss-reshetnev.ru.

Lomaev Yuri Sergeevich – an engineer of the second category; JSC "Academician M. F. Reshetnev "Information Satellite Systems". E-mail: lomaif@rambler.ru.

Ivanov Ilia Andreievich – Cand. Sc., an engineer of the second category; JSC "Academician M. F. Reshetnev "Information Satellite Systems". E-mail: ilyaiv92@gmail.com.

Tolstykh Anastasia Vladimirovna – technician; JSC "Academician M. F. Reshetnev "Information Satellite Systems". E-mail: avtolstykh1@gmail.com.

Islentev Eugene Vladimirovich – Cand. Sc., a head of sector; JSC "Academician M. F. Reshetnev "Information Satellite Systems". E-mail: islentev@iss-reshetnev.ru.