



Research and practice of inertial measurement unit software productization

Yuzhao Tian^{1,a}, Li Zhou^{1,b*}, Qi Li^{1,c}, Yi Feng^{1,d}

¹Xi'an Aerospace Precision Electromechanical Institute, Xi'an, China

^a76434841@qq.com; ^{b*}lilizhou0124@163.com

^csissi_liqian@163.com; ^dfengyi721@163.com

Abstract. With the top-down promotion of the product design of inertial measurement unit, software, as an important component of inertial measurement unit products, plays an important supporting role in the product design. The design and development of software products are dominated by "generalization, serialization and modularization", the development of software organizational asset library is an important support for software productization, and the software reuse library is taken as the entry point to carry out the top-down software product design. With the help of information tools, the promotion and application of software productization can be realized.

Keywords: inertial measurement unit, software productization, software reuse library

1 Introduction

In the informationized space, software plays an important role in the weapon system.^[1] The proportion of functions realized by software gradually increases, "hardware software" has become a new development trend, and many important functions are realized by software. NASA believes that "software engineering is a core competency and a key enabling technology necessary to support NASA's business." The ESA lists software engineering as one of 13 critical technologies when discussing future space missions and technologies.^[2]

The comprehensive performance index of inertial measurement unit depends on the completion degree and quality of inertial measurement software. So the quality of software has also become the most concerned issue.^[3] The new normal and situation of short cycle and personnel shortage also pose greater challenges to software development and expose various problems.

Secondly, the traditional software development mode is difficult to achieve rapid iterative upgrade.^[4] Traditional software development mostly adopts the model as the traction development mode, each model software is developed by the model software

development team separately, usually exclusive custom development.^[5] This development mode can meet the customization needs of the project, but it does not take into account the sustainable development of the software in the development process, which directly leads to weak software expansibility and poor reusability, and it is difficult to adapt to the rapid development needs of similar projects. The efficiency of software development is greatly reduced, and the time cost and labor cost remain high.

Finally, software keeps up with the rapid development of hardware platform, which requires strong portability and versatility. As the inertial measurement unit hardware platform was updated, the software was ported and changed accordingly. In the case of little functional changes, the adaptability of the platform occupies a larger workload of the software.

Therefore, it is of great significance to promote software productization, promote software modularization, standardization and serialization design, use software component-based development technology^[6], establish software reuse standard specification system and infrastructure, and develop new software products on the basis of existing components to reduce software development costs, improve software efficiency and reuse rate, and enhance the core competitiveness of inertial group products.

2 Research on the software productization method of inertial measurement unit

2.1 Combination of "top-down" and "bottom-up" analysis

As the saying goes, "Everything is difficult at the beginning." To do a thing with a good beginning is half the battle, so is software development. In the initial planning stage of project development, fully identify and plan which requirements are mature, fully designed, developed and tested, as well as which are unique to the project or have a certain range of reuse, and can become mature reuse requirements in the future. In the requirements analysis process, a series of mature software products from requirements to design development to testing are identified and analyzed "top down" from the software organizational asset library.

2.2 Combination of software organization asset library and reusable software library construction

The list of authors should be indented 25 mm to match the abstract. The style for the names is initials then surname, with a comma after all but the last two names, which are separated by 'and'. Initials should not have full stops—for example A J Smith and not A. J. Smith. First names in full may be used if desired. If an author has additional information to appear as a footnote, such as a permanent address or to indicate that they are the corresponding author, the footnote should be entered after the surname.

The software organizational asset library covers all types of assets throughout the software life cycle. Through sustainable accumulation and use of these assets, the overall process performance is improved. The software organization asset library includes several sub-libraries of management class, support class and engineering class, among which the reusable software library is the focus of software productization. The reusable software library is divided into four levels from top to bottom: software spectrum, software architecture, software requirements list and software mature components, as shown in Figure 1.

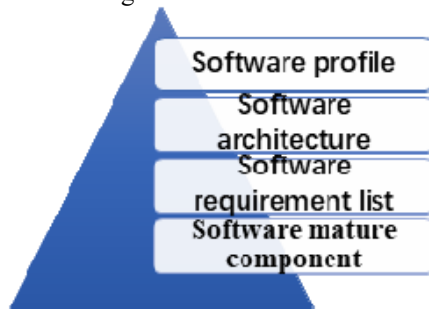


Fig. 1. Hierarchical division of reusable software libraries.

Building and maintaining reusable software libraries is an effective way to support project development to the maximum extent. A reusable software library covers all assets with reusable properties, such as common requirements, reusable artifacts, common architectures, domain development platforms, and so on. The classification and management of reusable software libraries can effectively support and safeguard the development and implementation of software productization, and provide a platform for organizations and projects to accumulate and use.

2.3 Standardization of software development system

To ensure the security and reliability of reusable software, establish a reusable software development system that "combines multi-level review and special review". After the development of reusable software is completed, requirements analysts and testers of reusable software first carry out code review and testing. Secondly, other reusable software teams carry out cross-review by combining design documents and test documents, and finally submit to the software expert group for review. The review comments can be incorporated into the reusable software library after closed-loop.

2.4 Informationization tool support productization

In order to effectively combine the software productization work with the actual work of the project, a special information work is customized as a support to digitize the management and use of the reusable software library. In addition to the basic operation of storage, use and change, information tools also provide retrieval, extraction

and push services to ensure the benign operation of organizational assets and build a sharing platform for software knowledge.

3 Inertial measurement unit software productization practice

3.1 Establish a multi-dimensional software development team

Divide all software developers into multiple software transition development teams, each consisting of one requirements analyst, one designer developer, and one tester. The requirements analyst is a designer with more project experience and productization experience, who is responsible for software requirements analysis, architecture design and module review. The design developer is younger designer, responsible for code writing and implementation. The tester is an independent tester who is responsible for unit testing and assembly testing of the code.

The software product development team is not divided by model tasks, and transforms the model-based software development mode into the product-based product development mode, which strengthens the communication between different project teams, gives full play to the advantages of personnel at different levels, and avoids repeated development of similar functions. At present, 7 product development teams have been set up, covering software in various professional fields, and product development has been carried out in an orderly manner.

3.2 System and standard software development process

The software organizational asset library management measures were compiled, which explained the library structure, identification, content compilation and management requirements of the organizational asset library. By standardizing the construction of organizational asset library, building a software resource sharing platform, improving the efficiency of software development, reducing software design defects, realizing the sustainable accumulation of organizational assets, and improving the process performance of projects and organizations. The enterprise standard "Reusable Software Management System" is formulated, which sets requirements and constraints on reusable software design and management from the aspects of personnel responsibilities, reusable software design requirements, reusable software management methods and so on.

3.3 Form multi-level reusable software

In addition to the construction of reusable software modules, we should actively promote the transformation of software development types, and carry out software research and development by using and adapting the development types to improve the overall reuse rate and maturity of software. Through the sample data accumulation of a single inertial group product of a certain type, practice sampling was carried out in

more than 50 kinds of software of three types. Among them, the adaptive modification rate of control software was more than 60%, the usage rate of FPGA software was more than 40%, and the adaptive modification rate of embedded software was more than 40%. All the newly developed software realizes software architecture reuse. See Figure 2 for the new development, application and adaptation of the software.

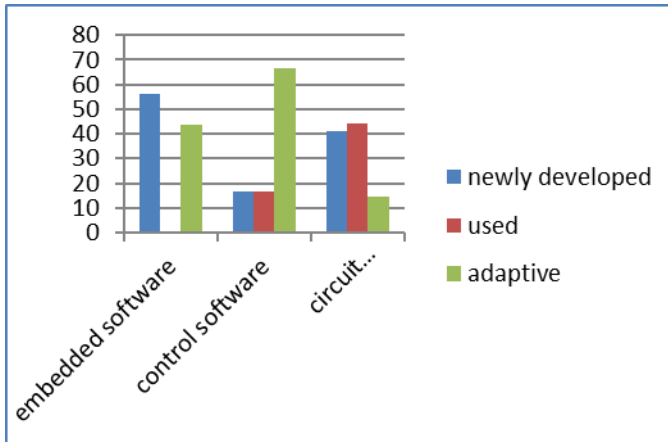


Fig. 2. Software new research, use, adaptability modification.

4 Software Productization effect of inertial measurement unit

4.1 Software development to reduce cost and increase efficiency

Software productization can meet the requirements of different models and meet different software requirements through the adaptation of common software modules and interfaces. The mature requirements can be directly selected, and the functions that need to modify interface parameters can be quickly completed. The development work only needs to be carried out according to the specific requirements of the project. It greatly reduces the labor cost and time of software development, and provides the basis for the software development of generalization, standardization and serialization.

4.2 Software product quality is steadily improved

Since the construction of software reuse library, mature modules that have been examined, tested and verified for several rounds are used by project software, with higher reliability and security. It effectively strengthens the robustness and reliability of the design of young software coders, and the product quality problems caused by software are greatly reduced.

4.3 Optimize the staff structure of the software development team

Software development is no longer dependent on a small number of team members, high-level software personnel are responsible for the design of unified architecture, and other software developers are responsible for the configuration of parameters between different projects, which reduces the requirements for developers, applies to the current personnel structure, and improves the technical level and corresponding ability of team software design and development. Strive for greater growth space for young designers.

5 Thinking about the inertial measurement unit software

5.1 Dynamic management of software organization assets

Software organization assets are not a stagnant pool, through effective management mechanism to make the organization assets efficient use and operation. In the process of project development, common technologies are extracted to form organizational assets. In the process of software development of new projects, remind software designers to use mature modules in real time by means of push and link, so as to improve the reuse rate and quality of software product development. After the completion of project development, the organizational assets in the organizational assets database should be improved and updated according to the application of organizational assets in the project, new generic technologies should be extracted synchronously, and the organizational assets database should be continuously expanded. On the basis of project application and trial, constantly improve the maturity of software organizational assets. Organizational assets that have not been used for a long time shall be removed from the shelves.

5.2 Quality assurance of software organization assets

Since the construction of software reuse library, mature modules that have been examined, tested and verified for several rounds are used by project software, with higher reliability and security. It effectively strengthens the robustness and reliability of the design of young software coders, and the product quality problems caused by software are greatly reduced.

5.3 Optimize the staff structure of the software development team

In the process of project software development, the mature modules are clearly used in the software requirements analysis stage, and the software code is directly sent out for use. When the code is checked, compare and analyze the reused code according to the reuse situation of functional modules decomposed in the Requirement Specification. The function modules used by more than two mature models do not need to be checked. During unit testing and configuration item testing, mature modules used are

not tested, and the test results of mature modules in the organizational asset library are directly used.

At the same time, in order to ensure the quality of the software, during the assembly test, the call to the mature module and the software interface related to the mature module must be checked and the interface test must be completed. The use of modules and the impact on software function items need to be paid more attention when software is stored and reviewed.

By building a software knowledge sharing platform, software technology exchange can be realized, and knowledge reserve can be accumulated for software development. With the support of information tools, the whole field, life cycle and practice of software development can be covered. To realize the construction of efficient and versatile platform, build an open architecture, and provide a tailoring environment for embedded software design. To achieve the improvement of personnel coding efficiency, to achieve the purpose of reducing software development costs, to ensure the quality of product design.

6 Conclusions

The promotion of software productization, and the implementation of software modular, standardized, serialized design, can reduce the cost of software development and improve the efficiency of software. It is of great significance. It is of great significance to enhance the core competitiveness of inertial measurement units.

References

1. Dong Qianli, Ma Xianchao, Du Ke. Research and practice of radar seeker software productization. China Aerospace, 2022, s1.
2. Yang Haicheng. Aerospace Model Software Engineering [M]. Beijing: China Aerospace Press, 2011.
3. Jiang Jian, Jiang Zhigang. Research and implementation of external quality measurement for embedded software. Control and Information Technology, 2021,01.
4. Wang Baolu, Yu Xihong, Xue Kai. Research and practice of software Productization system construction for aerospace field. Aerospace Industry Management, 2022,01.
5. Guo Huitao. Research on Productization of space Enterprises [M]. Harbin Institute of Technology, 2019.
6. Liu Changyong, WANG Yihuai, Peng Tao. Research and implementation of component-based mbedOS engineering framework. Computer Applications in Software, 2021,05.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

