The research and practice of spacecraft software engineering

Chengxin Chen, Jinhua Wang, Xiaoguang Xu

Institute of Manned Space System Engineering, China Academy of Space Technology, Beijing 100094, China

Abstract. In order to ensure the safety and reliability of spacecraft software products, it is necessary to execute engineering management. Firstly, the paper introduces the problems of unsystematic planning, uncertain classified management and uncontinuous improved mechanism in domestic and foreign spacecraft software engineering management. Then, it proposes a solution for software engineering management based on system-integrated ideology in the perspective of spacecraft system. Finally, a application result of spacecraft is given as an example. The research can provides a reference for executing spacecraft software engineering management and improving software product quality.

1 Introduction

Treating the software as an engineering project, such as applying systematic, strictly constrained and quantifiable methods to the software development, operation and maintenance is software engineering. The feasibility analysis, technical process for development, quality audit and other engineering concepts are introduced into the software development in order to achieve the integrated optimization of planning, funding and quality\(^1\)[2].

The domestic and foreign scholars have carried out numerous and fruitful research on the software engineering management. Software Engineering Institute(SEI) of Carnegie Mellon University proposed Capability Maturity Model for Software (CMM), upgraded to Capability Maturity Model Integration(CMMI)\(^3\) in 2000.China proposed "GJB5000-2003 military software development capability maturity model" in 2003, upgraded to "GJB5000A-2008 military software development capability maturity model" in 2008\(^4\)-\(^5\).The scholars mainly focused on the software engineering theory, organizational management requirements and so on, paid little attention to the content that applicable to spacecraft, such as whether the system planning is comprehensive and whether the software is closely integrated with the technical process for spacecraft development\(^6\).With the development of large-scale, complex and mass production of spacecraft, the complexity and scale of the software are increasing, the development risk is more and more serious, the research on software engineering management applicable to spacecraft is urgent needed\(^7\).

Firstly, the paper introduces the research status and weakness of the spacecraft software engineering management. Then, it proposes a method for the spacecraft software engineering management in response to the weakness. Finally, it provides a systematic solution for spacecraft software engineering management by giving the application result of spacecraft as an example.

2 Research status and existing problems

China Aerospace Science and Technology Corporation released Q / QJA30A “Requirements for space product software engineering”, proposed specific requirements for program management, technical management and quality management\(^8\)-\(^9\). However, the following problems still exist in the current spacecraft engineering software.
2.1 system planning is not comprehensive
The comprehensiveness and effectiveness of system planning should be cared in software engineering. Firstly, the software development targets should be combined with the process for spacecraft development, such as completing software test and verification before the systematic spacecraft test. Then, the system planning should be proposed combining the analysis for spacecraft software development, such as software system design, software tool selection, ensuring that the process is effective and comprehensive.

The system planning method hasn't been formed in the field of spacecraft software engineering. Currently, software engineering is proposed mainly based on the analysis for spacecraft software development activities and process, such as evaluating the reliability and security for the new compiler. The software development targets are not combined with the technology process for spacecraft development and software development activities are depended on personal experience, it is likely to cause unreasonable arrangement or missing of the software development activities, resulting in delay of the progress for spacecraft development.

2.2 classified management is not uncertain
Classified management should be cared in software engineering. The management can be divided into system level, subsystem level and configuration item level. Such as software system design, subsystem interface protocol design belong to system level; software subsystem design, configuration item interface protocol design belong to subsystem level; software requirement inspection, software coding belong to configuration item level.

A classified management mechanism hasn't been formed in the field of spacecraft software engineering. The system and subsystem researchers mainly focus on software engineering of configuration item level, such as ensuring the quality of software product by software requirement review, configuration item test review, pay little attention to the system and subsystem work mode, software tool selection, the verification of subsystem interface protocol. It is likely to cause poor management results due to the uneven investment of resources.

2.3 Improved mechanism is not sustainable
Improved mechanism should be continuous. Some types of software development activities which have been completed in this phase may appear again in the later phase, so it is necessary to carry out periodic review, such as the confirmability of the software requirements which appear in the software requirements analysis phase may need to be carried out again in the software design phase due to the flight changes.

A continuous improved mechanism hasn't been formed in the field of spacecraft software engineering. The software engineering management planning is proposed in the initial development of spacecraft, the corresponding software development activities are carried out in the process for spacecraft development. For external reasons, the work which has been completed may needed to be recarried out regularly, or may missed.

3 Solving the problems based on system integration management method
In order to solve the existing problems of spacecraft software engineering, the paper puts forward the system integration management method in the aspect of spacecraft system. The core idea of this method is to fully identify the software development activities ensuring that the planning is comprehensive; separate the software management activities ensuring that the management is effective at all levels; review software development activities regularly ensuring that the improvement is continuous. The detailed content is as follows.

3.1 The target-oriented identification method of the software development activities ensuring that the system planning is comprehensive
The method is proposed based on the combination of the software development target and the process for spacecraft development, combining the various aspects of software development activities. Firstly,
the elements of software development activities that affect the project are derived in the aspect of the software project development target. Secondly, the relationship between software development activities is analyzed. Finally, the software development activities and the relationship between them of the entire project is distinguished. The detailed identification process is shown in Figure 1 [10].

Figure 1  Target - oriented method for identifying software development activities

Firstly, the software development target are divided into sub-targets in each phase according to the technological process for spacecraft development and whether the sub-targets can be measured are judged, if the sub-targets can’t be measured, they are decomposed until all targets can be measured. Secondly, all the elements of software development activities that affect the measured target are analyzed and whether the sub-elements can be evaluated are judged, if the sub-elements can’t be evaluated, they are decomposed until all elements can be evaluated. The software engineering planning management can be comprehensive by using the method.

3.2 The classified analysis methods of the software development activities ensuring that the management is effective at all levels

After the completion of software engineering planning, the software development activities are classified into system level, subsystem level and configuration item level in the perspective of spacecraft system. Such as the range of compiler selection and the margin design of subsystem interface protocol belong to system level; the subsystem software working mode design and the margin design of configuration item interface protocol belong to subsystem level; software coding and test verification belong to configuration item level. Software engineering management is carried out at all levels after the classification of software development activities, the detailed process is shown in Figure 2.

Firstly, the software development activities are divided into the system level, the subsystem level and the configuration item level. Secondly, the software is classified into the core software, the important software and the general software according to the criticality, inheritance, scale and so on. The development activities of core software such as requirements review, configuration item test should be submitted to the higher level for reviewing, the review-conclusion of the development activities for the important software should be submitted to the higher level.
3.3 Establish periodic analytic and continuous improved mechanism
Software engineering has a gradual iterative process. It is necessary to summarize the management activities in the whole life cycle of software development, classify the new identified work, carry out the corresponding control measures. By regular analysis and continuous improvement, the feature and performance of the software product can meet the mission requirements, ensuring the safety and reliability of spacecraft software products.

4 The practice of a spacecraft software engineering
The spacecraft which carried out the rendezvous and docking task has the features of high difficulty, high quality and short mission period requirement. The spacecraft achieved the system integration of the software engineering and the spacecraft development process by using the integrated management method.

The risk of technology, quality and schedule in the process of software development were avoided or reduced ensuring the reliable operation of the software. This section takes the software development activities of the spacecraft project design stage as an example and gives the practical results.

4.1 The planning of software development activities
The software development target in spacecraft project design phase is to complete the spacecraft system software design and subsystem software design. 18 evaluated software development activities were identified by using the integrated management method, the detailed steps is showed in Table 1.

| target | Measured target | Software development activities based on measured goal | Evaluated software development activities |
|--------|-----------------|-----------------------------------------------------|------------------------------------------|
| Complete the software system design | (1) Determine system requirements (2) Determine software tools (3) Complete | (1) The design of software system architecture (2) The design of subsystem interface protocol (3) Determine the range of software tool (4) Determine the baseline of software | (1) Software system requirements analysis based on the flight program and failure mode (2) The design of subsystem interface protocol (3) The simulation of subsystem interface protocol (4) Determine the range of software development tools, process management tools (5) Reliability and safety verification of the |
| target | Measured target | Software development activities based on measured goal | Evaluated software development activities |
|--------|----------------|------------------------------------------------------|------------------------------------------|
| system software risk analysis | development (5) Determine the risk events and causes of system software | new software tools (6) Determine the baseline of software development (7) The impact of system failure mode on software (8) The analysis of software-related system failure mode (9) System verification environment constructing |
| Complete the subsystem software design | (1) Determine subsystem requirements (2) Determine software set (3) Complete subsystem software risk analysis | (1) The design of subsystem architecture (2) The design of configuration item interface protocol (3) Determine the type and criticality of software development (4) Determine the risk events and causes of subsystem software | (1) Subsystem software requirements analysis based on the flight program and failure mode (2) The design of configuration item interface protocol (3) The simulation of configuration item interface protocol (4) Reliability and safety verification of commercial software (5) The review of development qualification for new company (6) The software engineering train for new person (7) The impact of subsystem failure mode on software (8) The analysis of software-related subsystem failure mode (9) Subsystem verification environment constructing |

Table 1 A list of software development activities in the spacecraft design phase

4.2 Software development activities classification control
The 18 software development activities included 9 system-level activities, 9 subsystem-level activities. The development activities of core software were submitted to the higher level for reviewing, the review conclusion of important software were submitted to the higher level. The management was effective at all levels by the coordination of the system and subsystem.

4.3 Regular analysis of software engineering and continuous improvement
The spacecraft software system and subsystem design were divided into two stages: the initial stage of the development, the last stage of the development. Firstly, the system and subsystem software management planning was executed by identifying and completing the elements of management activities. Then, the system and subsystem detailed design reports and other aspects of information were summarized in order to re-identify the weakness of software system design. Finally, the verification of the software system design was summarized by combining system test and flight plan.

4.4 Practical results
The spacecraft that using the integrated management method added the requirements review, development qualification examination, simulation of subsystem protocol comparing the previous
spacecraft. 43 problems such as lack of time between the subsystem protocol were identified, flight program don't match with software work mode and so on. All of the problems are carried out influence domain analysis and implemented in the documents such as software requirements ensuring the success of the rendezvous and docking mission.

5 Conclusion
The software development target should be fully integrated into the whole spacecraft development and management process in order to ensure the realization of software technology, quality and progress targets in each phase. The management activities should cover comprehensively, classify certain by paying attention to classification and coordination. Software development activities should be re-identified and re-analyzed continually in order to reduce the risk of development.

References
[1] BAGERT D J. “A model for the software engineering component of computer science curriculum”, Information and Software Technology, 40(4), pp. 195-201, (1998).
[2] SUN Zhenming, SONG Zhengji, JIANG Xingwei. “Application of software engineering technique in satellite fault diagnosis system”, JOURNAL OF HARBIN INSTITUTE OF TECHNOLOGY, 35(7), pp. 784-786, (2003).
[3] XIE Chao, MA Shilong, LI Mengxiang. “A process improvement mode of next generation: CMMI”, Computer Science, 8, pp. 81-84, (2001).
[4] NI Guangnan. “The importance of the development of embedded software industry in China”, Electronic Engineering &Product World, 3, pp. 11-14, (2012).
[5] NI Ting. “Promote GJB5000A in software development and testing”, Computer Engineering&Software, 2, pp. 31-35, (2013).
[6] GUO Jian, LI Lin, Wang Junna. “Research on CMM Application in Spacecraft Software Development”, SPACECRAFT&ENGINEERING, 23(6), pp. 75-79, (2014).
[7] CHEN Chenxing, LIANG Ke, WANG Jinghua, GUO Juan. “Integration Management Method of Spacecraft Software Risk”, SPACECRAFT&ENGINEERING, 24(3), pp. 114-119, (2015).
[8] China Aerospace Science and Technology Corporation standard. “QJ/QJ A 30A Requirements for space product software engineering”, Beijing: China aerospace institute of standardization and product guarantee, pp. 3-9, (2013).
[9] SUN Xu, YANG Shunkun, LIU Bin. “Software engineering management framework for complex avionics software”, Moaem Electronics Technique, 35(24), pp. 110-114, (2012).
[10] Liang Tao. “Software risk management based the goal-risk”, Xi’an: Northwestern Polytechnical University, (2006).