Bing HAN, Yinglan FANG

North China University of Technology

Research of Software Model Method Oriented to CMMI System

Abstract. *The paper selects a number of important elements in CMMI text, adds software process areas and practices according to the advanced CMMI architecture concept. Via software development's requirements analysis, system design, coding, testing, system implementation in various stages, it applies CMMI to improve enterprise software development product quality.*

Streszczenie. *Artykuł przedstawia możliwości system umożliwiającego poprawę jakości opracowanego oprogramowania znanego jako CMMI (Capability Maturity Model Integration). Analizowano wymagania, projekt, kodowanie, testowania na różnych etapach projektu produktu. (**Badania modelowania oprogramowania dostosowanego do systemu CMMI**)*

Keywords: CMMI, software research, model method
Słowa kluczowe: oprogramowanie , system CMMI.

Introduction

In the 1970s and 1980s, the research focused on software engineering is the methods, techniques and tools of requirements analysis, software design, coding, testing, maintenance and other areas. We called it was the classic software engineering. But now most software projects have poor quality, delayed schedule, overrun cost and other problems. Improve software process is called as software process improvement. Software process improvement's intent it to enhance software quality, enhance software productivity and reduce development cost. From the 1970s to now, software process improvement had became a mainstream research to software engineering. Which CMMI was a significant results in the field, and became the standard to software process capability.

The full name of the CMMI is Capability Maturity Model Integration. CMMI is process improve method, it provides the basic elements of effective processes to organization. It can be used to guide cross-project, department or entire organization's process improvement. CMMI is mainly used for software development and software development capability assessment and improvement, focusing on the software development process management and engineering capabilities and evaluation. Since 1987, CMMI has come into effect authentication. It has turned into the most authoritative assessment certification system in the software industry.

CMMI includes five levels, 18 process areas, 52 goals and more than 300 key practices. CMMI consider: because the problem is caused by the method we manage software process, the new software technology can not automatically improve productivity and profitability. Improved process will produce better quality software, not only it makes many software projects avoid the overspend time and cost, but also it is conducive to realize parallel to software engineering and combine to multi-disciplinary. It can realize best benefit of process improvement. Software process improvement can not accomplish in an action, CMMI is gradually change in incremental mode. CMMI has clearly defined five different maturity layers. An organization can forward to a higher maturity level according to a series of small steps. CMMI has described five levels of software process. It reflected the transition from chaos software product to the discipline development process, and then standardized. Management and continuous ladder-type structure to improvement development process. Any project production of software institution can be included into it. In addition to the first-class level, each maturity is consisted some core process area. These core process areas respectively elaborated the level of software process in this regard should be to achieve the core group practice for one aspect of the software development process. The core process area is also called key process area(KPA). Each KPA is associated with a number of objectives, it behalf the requirements to the process. We can assess software process and find ways to improve the focus according to KPA. Implement each core process contains key practice, it is realize the aim established by the core area and achieve enhance software process capability. CMMI is to let everyone in the project team know when to do something in what way. In the project development process, it should be at what time do something based on what standard.

Requirement Analysis

The purpose of requirement development is to generate customer, demand for products and product components, and for the development and understanding of these needs for the required analysis. The relationship between requirement and design implementation activity is the product should be gradual achieved, with every progressive supports the latter progressive and improve requirements knowledge. For each progressive step, it freeze requirement before design. In the implementation process, if it fall across requirement change, it will delay the change to the next progressive step. If the change cannot be delayed, it will stop working, modify the requirement, amend the plan, and then began to redesign. Project need to take appropriate steps to ensure that management has been approved by the requirements set and support project planning and implementation needs. It gains understanding of the requirements, gathering requirements from the approved demand provider, and review with them. Before the requirement bring into the project plan, it can solve ambiguous problems and eliminate misunderstanding. Once the requirement provider has compass agreement to the requirement recipient, project participants gain promises from the requirement. As the project progress and the disagreement among identify plans, work products, the demand, the project manager needs to change requirement or plans.

Maintain two-way traceability requirements, requirement management needs to document requirement alteration and the reasons, and maintain the two-way traceability among source requirements and the demand for all then products and product components. Maintenance needs decompose two-way traceability to each level product of the requirement, including horizontal and vertical. From the original demand to its lower level of demand, then demand from the lower layer return to the original demand. When analysis the impact of requirement variation, it is special needs to track, so requirement alteration can reflect to project planning, activities and work products. If there is no

requirement traceability, it can not effectively manage requirement. Track-able information can be used find which requirement may be affected by the requirement alteration. Throughout the product life cycle, it need capture all the requirements and requirements change demand, and then place them under configuration management. When analysis impact of the project plan, activities and work products implementation to requests alteration demand, requirement should be traceable (each requirement has an unique identifier), generate a requirements traceability matrix. It can be used for tracking forward and backward. It analysis the impact to requirement alteration demand, including: the development progress, release date, the deal needed by system, staffing, arrangement, component, development and target device, risk, cost, project, the alteration needed by project to other system or interfaces. Demand management role: SQA Engineer review to the requirements management process activities and work products; project manager lead the project collect or define the requirements document, or re-plan projects; project engineers (developers, testers): review of the requirement distribution, identify requirement related to the distribution of potential demand, develop software work products, and consistent with the distribution of requirement.

System Design

System design is generally divided into two parts, essentials design and detailed design. The purpose of essentials design, first is analysis and design pre-function software system architecture (the module structure), determine the subsystem, function blocks and its inner and outer interfaces, determine data structure; the second is design the technical architecture used by all the system. The purpose of the detailed design is design the system with all modules of the main properties of the interface, data structures and algorithms, guidance module programming in a given technology architecture

Essentials Design: In the security design side, it considers the database login access restriction, user password encryption, access rights and other operating system security design. Analyze and optimize time-space efficiency of the database, as far as possible " improve processing speed" and "reduce the data occupy space." In interface design: the interface (including UI) design refers to exchange interface design requirements to the user, test person, definitude user interface, interface design rules, including the standard controls use planning, common interface (including the main interface and sub interface, etc.), interface design principles and so on. Interface design principles are as follows: It expands subsystems or modules and their relationships and constraints, implement interface design needed by the system, complete system data flow diagram after eliminate redundancy. If necessary, it comes into being explain and operation mode. If it is the object-oriented approach, it designs for the subsystem package, the properties between classes, methods and so on. Combine system error handling and data validation methods, it verify interface design result, and verify the reverse requirement. System architecture design of essentials design: with a selected set of tools and development plan delivery methods (such as a small version of the incremental delivery) and design methods, it combines with the design principles (such as functional modularity, etc.). It decomposes the system into several subsystems, function modules, and determine the subsystem, the functional modules and their relationship. It determines the subsystem, the constraints between the functional modules, assumptions and dependencies (such as system operating environment and development, testing environment, and consider the system concurrency and distribution requirements), set the subsystem and order the prioritization of functional simulation. Combine the above content, it designs the system module logic implementation and integration approach, reduces software difficult to implement, test, maintenance factors, forms high cohesion and low coupling system architecture. It defines error handling and recovery strategy, breaks down the possible failure, order priority and determine treatment strategies. It determines project database design rules for the unified system. Database design needs "Logical Design → Physical Design → Security Design → Optimization" and other steps. Usually this needs repeatedly step.

Coding process

Prepare coding work: The same as with other stages work, it need refinement and update the project schedule, develop a detailed implementation and test plan before at the beginning of this stage specific work. Project team together consult with implementation and test plan based on "project plan". Development team leader drafts "Implementation and test plan". The plan mainly includes the development environment configuration, programming planning and integration test plan describes. Review this plan, if it has not been passed, then it needs re-prepared; if it has been passed, the project team determines the module programming language according to the selected software development tools. If the agency has exist the appropriate programming standards, then use it; otherwise the development team together sets down new programming standards or modify existing programming specification.

According to the description required by the develop environment in "implement and test plan", development team leader designated personnel build the using development environment, and ensure the environment consistency throughout the development process. The team members should not install any software without the project manager or team leader agree in order to ensure development environment purity.

Developer code the module according to "programming plan" in "Implementation and test plan", "module design", "User Interface Design", "Database Design" and "Coding Standard". Developers must make the necessary self-examination and testing to its code after complete coding each module. After developers complete code checking and testing, he will prepare "module file". In the encoding implementation process, the project manager designates a person to conduct random checks on the system code, and will record review comments in the "module file".

System Testing

System testing includes unit testing, integration testing, interface testing, security test and so on.

Unit Testing: Project manager arrange testing person or system analysis person use relevant test tool write "unit testing management list" according to developers progress, Project manager examines and approves "unit testing management list", and specify the tester to unit test, and records it in the "unit testing management list". If it uses special test management tool, the results will be recorded into the tool. Tester tests corresponding unit according to the approved "unit testing management list ", produces "unit testing report" or registers into the test management tool, then the test management tool will generate reports related to unit test.

Integration Testing: When the development process arrives at "implementation and testing plan" expected integration point, and the involved unit modules passes unit testing in "integration test plan" of the "Implementation and test plan" , it begins integration testing activities. Systems

analysts and development team leader together establish the integrated "integration testing strategy", including the description of the integration scope, integration sequence, integration environment, integration methods. It prepares for product integration. Development team leader organize development and testing personnel establish product integration environment in accordance with "Integrated Testing Strategy". After this work is ready, the development team leader sign in basic information table of the "integration testing report", indicating the project has been integrated with the product and do integrated environment confirmed work. Under the project's "Integrated Testing Strategy" and "Outline Design", tester writes "integrated test case" or puts the integration test cases into test management tools, and review with peer. Development team head organizes tester do system integration testing according to the "integration testing plan" and "integration testing case" of "implementation testing plan ", testing results will be filled in the "integration testing report" or testing management tool.

User interface testing: In the case of ensure the user interface can pass test object control or access gain corresponding entry, testing user interface style whether meet user requirements, such as whether the interface is beautiful, the interface is intuitive, the operation is friendly or user-friendly, easy operability is good. Maintainability testing and maintainability is the convenience of system software, hardware implementation and maintenance functions. The aim is reduce system normal operation brought by maintenance function.

Safety testing: safety testing includes two parts: data security and operational security. Verify the provided data can only access the system, the other data which does not meet the requirements can not access the system; verify the operating privileges purview can only access the system. The other operating privileges which do not meet the requirements can not access the system.

Defect Management and Error Correction

Defect management and error correction guiding principles: In the process of unit testing and integration testing, when tester found the defects in the system, the defects must be recorded in the defect management lists or BUG management tools. Developers have in time eliminated the discovered defects. If it use BUG management tools, you can set the query condition and query the defects which yourself responsible for but not solve. After developers eliminate defects, tester should immediately do recursive testing to ensure no new defects introduced. After integration testers complete a integration test, it statistics fill out the "defect management statistical reports" or BUG management tools statistic analysis to defect based on "defect management list".

Defect management and tracking process: After tester found bug, he filled out the "defect management list" or defects information in BUG management tools, and made its status set to "on", submitted to the project manager. Project manager confirmed defect content, he made it convert to the relevant personnel resolve or assigned to the relevant personnel solve. When the defect resolve person think the defect has been fixed, he can fill out the "defect management list" or the corresponding list in BUG management tool, then it submit this "defect management list" and the restored program to the tester to back-tested. Testing personnel back to the test and fill out the "defect management list" or verify the information items of BUG management tool.

System implementation

CMMI correspond practice. Technical solutions process areas implements the product design, its aim is implement product components and related support files in accordance with the design,. SG Prepare for Product Integration, its intent is complete preparations for product integration. The preparation of product integration includes establish and maintain integrated order, implements integrated environment and processes. SP Determine Integration Sequence forms the following work products: product integration sequence of instructions, select or reject the basic principles of the integrated sequence. Through the following steps, it completes the practice: The first is identifies the product components to be integrated. The second is identifies the valid work which need perform in the product integration process. The third is to identify the candidate solutions of product integration sequence. The forth is choose the most appropriate integrated sequence. The fifth is periodic review integration sequence, and revised as needed. The six is record the basic principles of established or delayed decision.

SP Establish Product integration environment, in order to support the integration of product components, it establish and maintain necessary environment. It can complete practice according to the following steps: The first is identify the product integration environment needs. The second is identifies the product integration environment verification process and criteria. The third is decide which requires self-control or which need to buy in the necessary integrated environment; The fourth is develop an integrated environment (such as automatic compiler development, etc.) if there is no appropriate integrated environment. The fifth maintains the integrated environment in the entire project process. The sixth is abolishes the department no longer used in the integrated environment. SP Establish Product integration procedures and criteria is establishes and maintains procedures and criteria for product component integration.

Product integration procedures and criteria are key considerations as follows: construct the component testing level, interface verification, the performance deviation threshold, the assembly derived demand and the external interface, allows replaced components, test environment parameters, the test cost constraints, integrated operation quality and cost trade-off, the proper functioning possibility, the delivery rate and its changes, the time from order to delivery. SG 2 Ensure Interface Compatibility, its purpose is to ensure the internal components and external interfaces are compatible. Statistics show that the problem of many products integration is caused by the unknown or uncontrolled internal and external interfaces. Effective product component interface requirements, specifications and design management can help determine implement the interface integrity and compatibility. SP Review Interface Descriptions for examination review the interface description coverage and integrity. In addition to contain product component interface, it also includes an interface with the product integration environment. Usually it produces the interface classification, the interface list under each category, the correspondence relation of interface, product components and product integrated environment. It can complete the practice through the following steps: The first is review the interface data integrity, and determine all interfaces of the complete coverage. The second is ensure product components and interfaces be standard training in order to ensure that the related products components easily interact and properly connected. The third is periodic review the adequacy of interface descriptions.

SP Manage Interface manages products and definition, design and change of product components inner and outer interface. Usually it forms the following work products: product components and the external environment relationship table, the relationship table between different product components, the application program interface, updated interface and protocol description. It can complete the practice through the following steps: The first is determines the compatibility of the interface in the entire product life cycle. The second is to solve conflict, un-compatible and change problem. The third is to maintain interface database which can access by a project reference.

SG Prepare for Verification, its intent is ensure measure has been implanted the product and product component requirements, design, development plans and progress. It defines support tools, test equipment and software, simulations, prototype systems and facilities. Authentication methods include checks, peer review, audit, and gradually review, analysis, simulation, testing and display.

SP Perform Verification validates the selected work products. It forms the following work products. It includes verification results, validation reports, demos program, running process log. It can complete practice according to the following steps. The first is to perform the selected work products verification according their needs. The second is to record the results of verification activities. The third is to identify the product verification activities.

SP Analyze Verification Result is analysis the results of verification activities. It can follow the following work. It includes analysis report, fault report, verification methods, criteria and environmental change request. It can complete practice according the following steps. The first is to compare actual results and expected results. The second is to identify verification products which can not meet demand based on established criteria, or identify problem using authentication methods, procedures, guidelines and verification environment. The third is to analysis the validation data on defects. The forth is to write all analytical results to report. The fifth is to compare the realization measurement and technical performance parameters using the validate result. The sixth, is to provide information of resolve deficiencies and take corrective measures.

Conclusion

In this paper, it applies CMMI in the various development stages through the software and improves its software process. Its core is to make the software development as a process. Accordance with the CMMI principles, it monitors and researches to software development and maintenance. So it can develop software to make more scientific, standardized, while enable enterprises to better achieve business goals.

Acknowledgment

The support of the North China University of Technology youth key research fund Project "software modelling method oriented CMMI system" in 2010-2012 is gratefully acknowledged.

REFERENCES

[1] Choetkiertikul, Morakot, A risk assessment model for offshoring using CMMI quantitative approach, 5th International Conference on Software Engineering Advances, 2010, 331-336

[2] Yahya, Maged A. Effects of software process maturity on COCOMO II's effort estimation from CMMI perspective, RIVF 2008 - 2008 IEEE International Conference on Research, Innovation and Vision for the Future in Computing and Communication Technologies, 2008, 255-262

[3] Vanamali, Bhaskar, From CMMI to SPICE - Experiences on how to survive a SPICE assessment having already implemented CMMI, 32nd Annual IEEE International Computer Software and Applications Conference, COMPSAC 2008, 1045-1052

[4] Xiaosong, Zhao, Zhen, He; Min, Zhang; Jing, Wang; Dainuan, Yu Process integration of six sigma and CMMI IEEE INDIN 2008: 6th IEEE International Conference on Industrial Informatics, 2008, 1650-1653

[5] Wang, Mei-Hui, Lee, Chang-Shing An intelligent PPQA web services for CMMI assessment , 8th International Conference on Intelligent Systems Design and Applications, ISDA 2008, 229-234

[6] Romeu, Leonardo Audy, Jorge; Covatti, Andressa, Integration method among BSC, CMMI and Six Sigma using GQM to support measurement definition (MIBCIS), ICEIS 2009 - 11th International Conference on Enterprise Information Systemsn 2009, 171-176

[7] Lester, N.G. Wilkie, F.G.; McFall, D.; Ware, M.P , Investigating the role of CMMI® with expanding company size for small- To medium-sized enterprises, Journal of Software Maintenance and Evolution , 2010, v 22, n 1, 17-31

[8] Jakobsen, Carsten Ruseng, Scrum and CMMI - Going from good to great: Are you ready-ready to be done-done, gile Conference, AGILE 2009, 333-337

[9] Ehsan, N. Malik, O.A.; Shabbir, F.; Mirza, E.; Wasim Bhatti, M. Comparative study for PMBOK & CMMI frameworks and identifying possibilities for integrating ITIL for addressing needs of IT service industry , 5th IEEE International Conference on Management of Innovation and Technology, ICMIT2010, 113-116

[10] Garcia, I. Andrea, I , Using the software process improvement approach for defining a methodology for embedded systems development using the CMMI-DEV v1.2, 7th IEEE International Conference on Embedded Software and Systems, ICESS-2010, 233-240

[11] Kovacheva, Tsvetelina , Optimizing software development process: A case study for integrated Agile-CMMI process model, International Conference on Computer as a Tool - Joint with Conftele 2011,

[12] Pino, Francisco J. Baldassarre, Maria Teresa; Piattini, Mario; Visaggio, Giuseppe; Caivano, Mapping software acquisition practices from ISO 12207 and CMMI, ommunications in Computer and Information Science, 2010, v 69 CCIS, p 234-247,

[13] Zhu, Kedi , Yang, Hongping, Application of Earned Value analysis in project monitoring and control of CMMI, 2010 3rd International Conference on Advanced Computer Theory and Engineering, 2010, 4164-4168

[14] Hu, Min , Zhou, Peien; Yu, Zhuguo; Pan, Dingding, Application and research of process improvement based on SPP Model of CMMI level 3, 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010, 598-601

[15] Abdul Latif, Aliza Md Din, Marina; Ismail, Challenges in adopting and integrating ITIL and CMMi in ICT division of a public utility company, 2010 2nd International Conference on Computer Engineering and Applications, 2010, 81-86

***Authors**: Engineer, Bing Han, Computer Application Laboratory of Information Engineering, North China University of Technology, Beijing 100144, China, E-mail: jluhan_bin@163.com; Associate prof Yinglan Fang, Robot Laboratory of Information Engineering, North China University of Technology, Beijing 100144, China, E-mail: jlufangyl@163.com.*