Research on Software Testing Technology and Methodology based on Revised and Modified Capability Maturity Model: A Novel Approach

LIN Qinhua[1, a]

[1]Sichuan Information Technology College, guangyuan 628017,China

[a] linqinhua319@sohu.com

Keywords: Software Testing; Capability Maturity Model; Small Software Project; Software Safety.

Abstract. Nowadays, the scale of software industry is getting larger and larger. Managing software development lifecycle is becoming complex and hard to control, many companies adopt capability maturity model to guide software development. However, companies usually prefer to choose detailed and operational single-layered process models for applying CMMI. If some activities in process have been changed, the process model must be totally redesigned; this reduces the flexibility and increases the cost of adopting CMMI. In this paper, we conduct research on software testing technology and methodology based on revised and modified capability maturity model. . As the final result, our method performs better compared with other standards. We plan to modify our method with more in-depth analysis in the future.

Introduction

Capability maturity model (CMM) since its launch, been adopted by more and more software organizations, in the national software industry has produced huge influence, become the industry standard for the international mainstream. But it is well known that the standard is geared to the needs of large and medium-sized software organization and project process management model, more suitable for large and medium-sized software organization and project implementation. For small software organization and project, the standard of "overload" phenomenon are obvious, the implementation of application problems, difficult is more, not applicable [1-4]. Therefore, re-evaluate the CMM standard practices for small software organizations and applicability of the project, studies the CMM standard is suitable for small software organization and project implementation of cutting method is of great necessity and realistic significance. In reality, the scale of the software organization and project size, the length of the software life cycle not neat, such a variety of factors determine the process we can't use a single management model for the same type of quality management, at least on the specific operating mode should have scalability and adaptability. Because the cut is based on the standard implementation of the CMM, it should keep the software capability maturity model of the original intent, requirements and structure, and that they can be more effectively used in small organizations and projects. So, despite the tailor to make many adjustments to model, but the introduction of change can't fundamentally change model. Software industry development in our country is currently in modernization, have a large number of small software organization and project, when using the CMM standard for process management and improvement, it should make reasonable cutting, make these small software organizations (hereinafter referred to as "small group") or project (hereinafter referred to as "project") to maintain standards at the same time, simplify the software implementation process, reduce the cost of the implementation of the software is more effective and efficient to the implementation of the CMM [5].

In order to deal with these problems, we added a third component to the strategy: the Reference Software Measurement Ontology (RSMO), which aims to capture the conceptualization involved in this domain, including traditional and high maturity aspects of software measurement. RSMO is a domain which is reference ontology. As domain ontology, it can be used for human communication, providing knowledge and promoting common understanding. As reference ontology, it is developed with the sole objective of making the best possible description of the domain in reality, with regard to a certain level of granularity and viewpoint. By adding RSMO strategy, we established a synergy between its components. On the one hand, RSMO provides a common vocabulary and knowledge in the other two components and to improve certain aspects of their definition. Using IESMR and real situation, on the other hand, as an extra BRSM RSMO evaluation means [6].

In this paper, we conduct research on software testing technology and methodology based on revised and modified capability maturity model [7]. Cut the goal is to generate a suitable for small organization and project implementation of the CMM cutting model. Process-centered software engineering environment is the most recent generation of environments supporting software development activities. It exploits an explicit representation of the process (called the process model) which specifies how to carry out software development activities, the roles and tasks of software developers, and how to use and control software development tools. It by adjusting the standard process of the original model, add, delete, modify or replace the standard content activities, in order to obtain the practical implementation of the operational requirements. Tailoring defines change standard guidelines, in essence, these guidelines stipulate the scope of allowed by the standard deviation or scaling, small groups and events purpose best "cost performance" of the implementation of the requirements. The detailed discussion will be finalized in the following sections. The general description of proposed method and the software development process is shown in the figure 1.

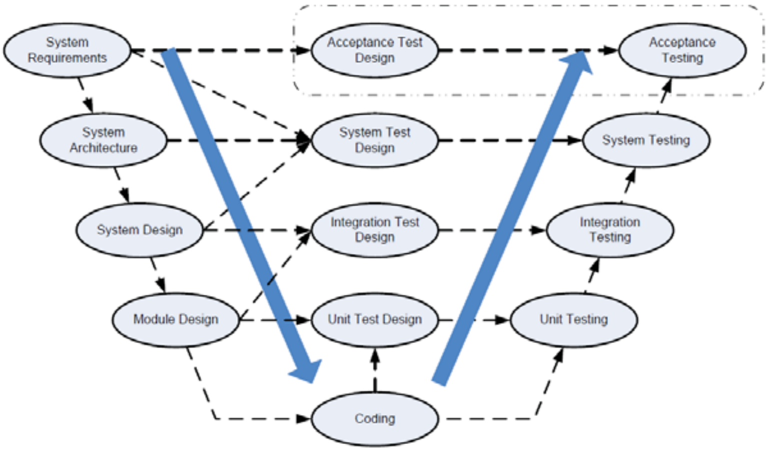

Fig.1The Software Development Life Circle and the Testing Procedure

The Designed Platform and System

The Instrument for Evaluating the Suitability. The IESMR is composed by four checklists that are used for evaluating the following items: measurement plan, measurement repository structure, measures and collected data. In each checklist, there is a set of requirements that must be satisfied in order to use the corresponding item in SPC. These requirements were identified from a study based on systematic review of the literature and they were refined considering the results of three practical experiences with previous versions of the IESMR. These practical experiences allowed us to evolve the IESMR until its current version. The key to the CMM standard practice includes all kinds of documents, such as policies, procedures, standards, plans, reports, etc. For a limited resource, for small projects with limited time, so many number of documents is not only a kind of management resources waste, is unrealistic. Documents related to the practice requirement has not been deleted, but be merged or restated, the form can be diversified, such as in a variety of informal documentation as a supplement to the formal documentation, replace or partly replace, document medium and its distribution, also includes a variety of forms such as electronic/written. For some documents related to the practice of, although not deleted, but can be considered not applicable, as long as the cause of the organization or project should be not apply documented, and approved. Software engineering facilities and tools, for instance, plan the project asked for additional facilities and tools or the use of existing facilities and tools influence project were formulated. The evaluation of an item against each requirement present on the checklists can produce one of the following results: (i) Satisfied (S), which means that the item satisfies totally the requirement and no corrective action is necessary; (ii) Largely Satisfied (LS), Reasonably Satisfied (RS) or Precariously Satisfied (PS), which means that the item does not completely satisfy the requirement, but it is possible to take actions to adapt it in order to satisfy the requirement and, consequently, to allow the use of the evaluated item in SPC. In the figure 2, we draw the flowchart of the proposed multi-layer analysis pattern.

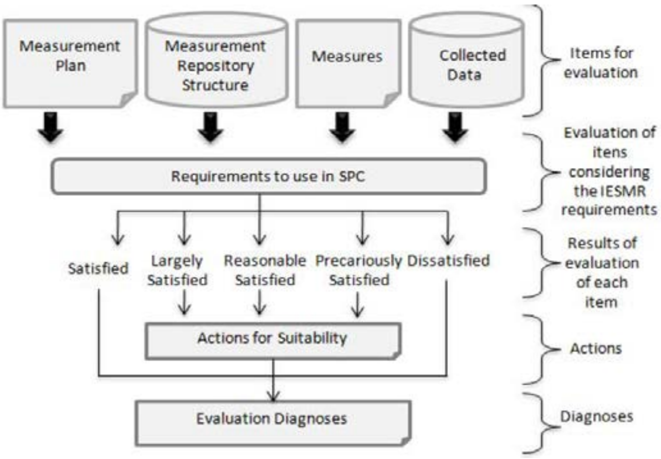

Fig.2The Overview of the IESMR

The Body of Recommendations for Software Measurement. Although there are models and standards devoted specifically to address measurement, they do not satisfactorily address how to carry out measurement for SPC. Thus, aiming to complement our strategy with a practical guide to organizations carry out software measurement suitable for SPC, we defined the BRSM. The BRSM groups are: (i) Software Measurement Preparation, which contains recommendations related to aspects that should be treated before starting the measurement; (ii) Alignment between Software Measurement and Organizational Goals, which contains recommendations for carrying out measurement aligned with organizational business goals and projects goals; (iii) Software Measures Definition, which contains recommendations for correctly elaborating operational definitions of measures; (iv) Software Measurement Execution, which contains recommendations for appropriately collecting and storing data for the measures defined; and (v) Software Measurement Analysis, which contains recommendations for analyzing the data collected, aiming to meet the information needs previously identified. The table 1 shows the structure.

Groups	Aspects
Preparation	Measurement Repository Creation
Alignment	Identification of Measurement Goals
Measures Definition	Operational Definition of a Measure
Measurement Execution	Operational Definition of a Measure
Measurement Analysis	Recording of Measurement Context

Table 1.The General Groups and Aspects

The Hierarchical Management Problems. Although BRSM advice must be used for the implementation of SPC, we have recommended processing measurement in the initial maturity level. This is because, as discussed earlier results, and points out that the research based on the systematic review, most of the problems related to measuring high maturity level is usually at the origin of the original level. Frequently, these problems will not affect the measurement result in the initial maturity level, only when the group began to SPC practice, they realized these problems. Human resource limitations of small groups, events purpose, role responsibilities can be merged staff can be a part-time job as a variety of responsibilities. At this point, the related with the practice of merger, cut.

Task for small projects, for example, leadership, software manager, project manager and project manager software by a person as many characters such as many relevant practice can be combined, in the interaction between different management levels, roles, responsibilities, coordination and commitment to practice can be deleted. In the practice of the CMM standard describes many various types of review, such as management review, peer review, the SQA audit, formal reviews, technical reviews, etc. For small project resources, time is limited, can't allocate more review time which cannot bear the management costs. Let the small groups, small project implement all provisions of the review is unrealistic. The cutting should be implemented for the extensive, frequent review, mainly includes the review of mergers, review way/frequency of cutting schedule. The implementation of the CMM standard in many separate set of demand also lower the requirements for personnel or part-time personnel. Design activities to start the segment. The process segment will delegate on the activities in activity list. TNG is short for Technical Group, CMG stands for Configuration Management Group, TSG means Test Group and SQG represents Software Quality Assurance Group, activities derived by this process segment are listed in Derived Event. Such as the independence of the SQA in some flat never have been possible in small groups, then cut to allow the SQA independence not implemented in some of the organizational structure, but must ensure that it does not meet the problem put forward is objective notary, not blame or revenge.

Conclusion and Summary

Capability maturity model (CMM) since its launch, been adopted by more and more software organizations, in the national software industry has produced huge influence, become the industry standard for the international mainstream. The Maturity Level 3 stressed that the project level process must be tailored from the organization level process, that is to say, no matter how the project level process changes, it must satisfy the constraints of the organization level process. However, companies usually prefer to choose detailed and operational single-layered process models for applying CMMI. If some activities in process have been changed, the process model must be totally redesigned; this reduces the flexibility and increases the cost of adopting CMMI. In this paper, we conduct research on software testing technology and methodology based on revised and modified capability maturity model. As the final result, our method performs better compared with other standards. In the future, we plan to conduct more related methodology to level-up our current approach. We believe that through this steps, we will gain more satisfactory results.

References

[1] Daca P, Henzinger T A, Krenn W, et al. Compositional Specifications for ioco Testing[C]. //Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference on. IEEE, 2014:373 - 382.

[2] Papadakis M, Henard C, Traon Y L. Sampling Program Inputs with Mutation Analysis: Going Beyond Combinatorial Interaction Testing[C]. //Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference on. IEEE, 2014:1 - 10.

[3] Wojciak P, TzorefBrill R. System Level Combinatorial Testing in Practice -- The Concurrent Maintenance Case Study[C]. //Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference on. IEEE, 2014:103 - 112.

[4] Johansson W, Svensson M, Larson U E, et al. T-Fuzz: Model-Based Fuzzing for Robustness Testing of Telecommunication Protocols[J]. Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference on, 2014:323 - 332.

[5] Hunt C J, Brown G, Fraser G. Automatic Testing of Natural User Interfaces[C]. //Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference on. IEEE, 2014:123 - 132.

[6] Bardin S, Kosmatov N, Cheynier F. Efficient Leveraging of Symbolic Execution to Advanced Coverage Criteria[C]. //Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference on. IEEE, 2014:173 - 182.

[7] Aydin A, Alkhalaf M, Bultan T. Automated Test Generation from Vulnerability Signatures[J]. Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference on, 2014:193 - 202.