

CMMI-DEV Implementation Simplified

A Spiral Software Model

Maruthi Rohit Ayyagari¹

College of Business, University of Dallas
Irving, Texas, USA

Issa Atoum²

Department of Software Engineering
The World Islamic Sciences and Education, Jordan

Abstract—With the advance technology and increase in customer requirements, software organizations pursue to reduce cost and increase productivity by using standards and best practices. The Capability Maturity Model Integration (CMMI) is a software process improvement model that enhances productivity and reduces time and cost of running projects. As a reference model, CMMI does not specify systemic steps of how to implement the model practices, leaving a room for organization development approaches. Small organizations with low budgets and those who are not looking for CMMI appraisals cannot cope with the high price of CMMI implementation. However, they need to manage the risk of CMMI implementation under their administration. Therefore, this paper proposes a simplified plan using the spiral model to implement the CMMI to reach level 2. The objective is to make the implementation more straightforward to implement and fit CMMI specification without hiring external experts. Compared to related implementation frameworks, the proposed model is deemed competitive and applicable under organizations' conditions.

Keywords—CMMI; Spiral model; Capability Maturity Model Integration; process improvement; CMMI Implementation

I. INTRODUCTION

Software quality is considered the engine to productivity. Improving software quality will continue to be an open problem in this century [1], [2]. The Capability Maturity Model Integration for Development (CMMI-DEV) is a software process improvement model designed to leverage the level of software products' quality. The CMMI is usually used to develop higher software quality, increasing efficiency, improving customer satisfaction and achieving profit [3]. The CMMI defines the most critical elements that are required to build great products or deliver exceptional services. As a result of using CMMI, several studies have identified the advantages of adopting such model [3]–[6] and several organizations gained various benefits. A survey of 30 organizations reveal the following results from CMMI implementation [7]. The medians from the sample showed a 34% reduction in cost, a 50% reduction in the schedule, a 61% increase in productivity, a 48% increase in product quality, and a 14% increase in customer satisfaction.

Fig. 1 shows the conceptual diagram of the CMMI model which contains four interacting categories. An organization engages process improvement which empowers project management all over the project lifecycle. Consequently, project management employs engineering activities to develop new CMMI practices. As a result, decisions are taken based on

measurements and analysis of the current situation. However, implementing software process improvement models are hinged on achieving business results [8], [9].

Software organizations are highly exposed to an aggressive changing environment [10]; therefore, organizations need to have adequate risk management over time. Although CMMI could help to resolve the problem, the CMMI model is considered a complex model [11]. Therefore, proper implementation of CMMI should take care of the implementation time, cost and scope. The implementation process will affect the current business environment, and the current running products and services. Small organizations often have low budgets; nevertheless, they are willing to improve their practice towards better product delivery. Such organizations could improve their current software process by implementing the CMMI using their existing software's process models. However, often these organizations lack certified CMMI professionals, but they usually have teams of project managers, developers, quality assurance engineers.

The CMMI model is considered a long project that could take 6-8 months, and its implementations are risky on business. Therefore, to solve the abovementioned problems, this paper recommends the spiral model for implementing CMMI for the organizations that are willing to implement CMMI on their own as shown in Fig. 2.

Although some works try to implement CMMI, they are limited. The model proposed by [14] was applied to the KSA and was restricted to the Process and Product Quality Assurance (PPQA) process area. Moreover, commercial software is expensive [19]. Additionally, a small organization cannot cope with CMMI appraisals due to cost, expertise need, and time constraints.

The spiral model is an incremental risk-oriented life cycle model that has four main phases: determine objectives, identify and resolve risks, development, and test, and plan the next iteration. A software project will iteratively go through these four phases. In the context of this paper, the spiral model is employed for CMMI implementation which resolves the above mentioned problems of agility and risk management. In the first place, the requirement will be gathered which includes what business unit to focus on, what level of CMMI to focus on. In the second phase, the risks and the alternative solutions will be identified including resources and time constraints. The third step carries out the CMMI iterative plan. Then, an evaluation is carried out, and the next iteration is planned.

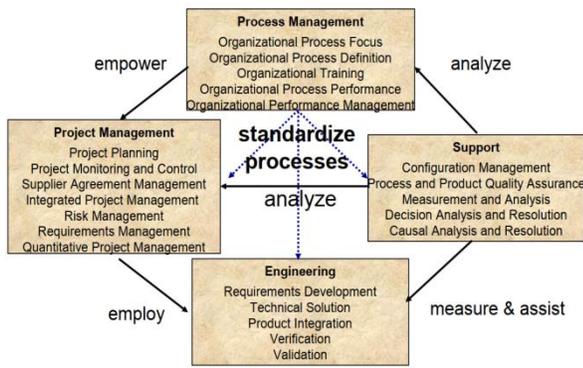


Fig. 1. CMMI Process Area Category.

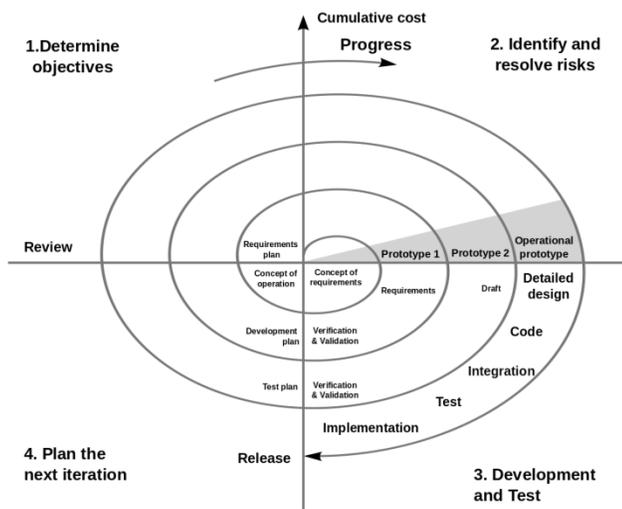


Fig. 2. Bohem Spiral Software Model.

The objective of this paper is to provide a systemic model to implement the CMMI in small organizations iteratively. Consequently, the proposed implementation model could be executed by the organization without hiring certified CMMI practitioners from outside the organization.

This paper is structured as follows. Section 2 provides more information about CMMI and software spiral model. Section 3 provides the related work. Section 4 explains the proposed model. Section 5 evaluates the proposed metric, while Section 6 provides conclusions, with implications and future research.

II. BACKGROUND

A. CMMI Simplified

CMMI came from the software engineering Institute, a US national lab with the mission of maturing the software engineering profession [6]. The CMMI was originated based development of its predecessor, the capability maturity model (CMM) as well as input from engineering practices of quality and management. The CMMI is a reference model of structure description of proven practices in product engineering and engineering management. Although the model is a trailing

edge, its practices are sound, and the model has faced concurrent and construct validity as measured over time. One of its main advantages is that it describes what, not how, thus it leaves much room for innovation exploration and new approaches to product development.

The CMMI has three categories: CMMI for services, CMMI for Acquisition and CMMI-DEV for development. The later, CMMI-DEV is concerned with developing software projects, which is the concern of this paper. There are two ways to represent or view the CMMI-DEV model. The first and the most common way describes how to improve the organization as a series of stages of maturity (shown in Fig. 3), while the second is concerned with the process areas and implement the ones that must address the company needs at the time (Fig. 4). There are five levels of maturity, each of which represents an organizational plateau of the overall capability of the organization. Each level has a predefined set of processes that are assigned to it for cohesive implementation and results. The latest version CMMI-DEV 1.3 model has five maturity levels, each of which has specific process areas. The expectation is that as these process areas are implemented, product productivity and quality increase. At level one (initial) there are no process areas, the processes are usually ad hoc and chaotic. Level two (managed), ensures requirements are systematically converted to quality-accepted products. This level has essential project management and product support practices. At level three (defined), processes are well characterized and understood as described in standards and procedures. At level four (quantitatively managed), processes are continually improving process performance through incremental and state-of-the-art technology. The last level demonstrates the tuning and optimization of organizational processes and practices.

CMMI-DEV is divided into categories of practice called process areas. There are 22 Process Areas or (PAs) that are described in a pure form in a manner that resembles a project lifecycle. PAs are windows through which a developer looks at organization current practices. Requirements Development (RD) and Requirements Management (REQM) process areas handle the elicitation, development, and validation of requirements and validation of demands as well as managing the changes that naturally occur during the project. Project Planning (PP) and Project Management and Control (PMC) deal with essential project management estimating, planning and then managing a project. These two process areas are augmented by Supplier Agreement Management (SAM) when using outsources and by Risk Management (RSKM) for mitigating risk before they occur or managing them when they become issues. Technical Solution (TS) and Product Integration (PI) take care of the design, development and build up a product along with ensuring that the parts of the product fit together when delivered to the customer. Verification (VER) and Validation (VAL) deal with making sure that the product is free of defects. Verification refers to building the product right so that it should work as intended in the customer's environment while validation refers to making the right product.

tools that could be used to manage and implement CMMI are available [20]; however, without human guidance the software is useless.

Although many organizations have implemented CMMI-DEV successfully, the implementation plans were not revealed, and certified practitioners executed the implementation most of the time.

IV. PROPOSED SPIRAL METHOD

The research methodology depends on Spiral software model. First, the spiral model is run as a project; then the outcomes are evaluated against a set of criteria extracted from literature and proposed by this research. The criteria include implementation cost, fitness for purpose and use, satisfaction, and usability of the proposed model.

The rationale for the CMMI project improvement program is to improve competitive posture and adapt to organizational changes to achieve and maintain the CMMI level 2. The first step towards implementation is to assign a project manager. Once the project manager is appointed, he first identifies the primary stakeholders. The CMMI implementation is a critical project; therefore, gaining executive support is crucial primarily if the implementation affects more than one business unit simultaneously. Then the project manager will develop the steering committee and a project organization. Ideally, the implementation for the in-house project should include teams of quality, technical, and the engineering process group. Fig. 5 shows a sample CMMI implementation project organization. Table I shows their roles and responsibility in CMMI-DEV 1.3 implementation for Process Improvement (PI).

The set of activities that could be performed to implement the CMMI are shown in Fig. 6. The steps are at high-level, and further detailing will be carried out by the project manager. The activities in Fig. 6 are executed in phases as shown in the CMMI spiral model shown in Table II. If these activities are not performed iteratively, a project failure may occur due to high risk and change management.

Therefore, the proposed setup of the experiment integrates the list of CMMI development activities (shown in Fig. 6) and the list of stories executed by the proposed spiral model presented in Table II.

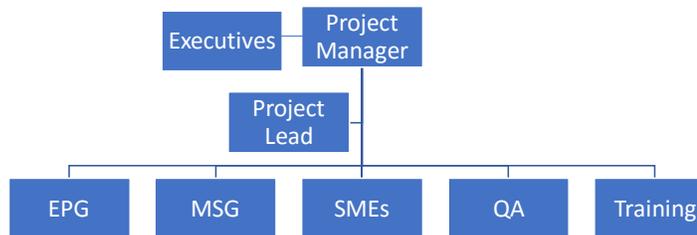


Fig. 5. CMMI Development Project Organization.

TABLE I. ROLES AND RESPONSIBILITIES OF CMMI IMPLEMENTATION TEAMS

Role	Responsibility
EPG	<ul style="list-style-type: none"> • Build sponsorship of PI, • Nurture/sustain improvement activities, • Manage PI effort, and • Ensures PI effort coordination.
Management Steering Group (MSG)	<ul style="list-style-type: none"> • Link organization’s vision/mission to the implementation, • Signifying sponsorship, • Assigning resources, • Monitoring progress, and • Providing guidance and correction.
Subject Matter Experts (SMEs)	<ul style="list-style-type: none"> • Knowledge of organization processes implemented within their business unit, • Work with the process (users), • Affected by the new business unit processes and stakeholders
QA	<ul style="list-style-type: none"> • Testing and peer review • Quality metrics modeling and analysis
Training	<ul style="list-style-type: none"> • Train relative stakeholders to new technology • Training for CMMI core activities • Train users for new practices

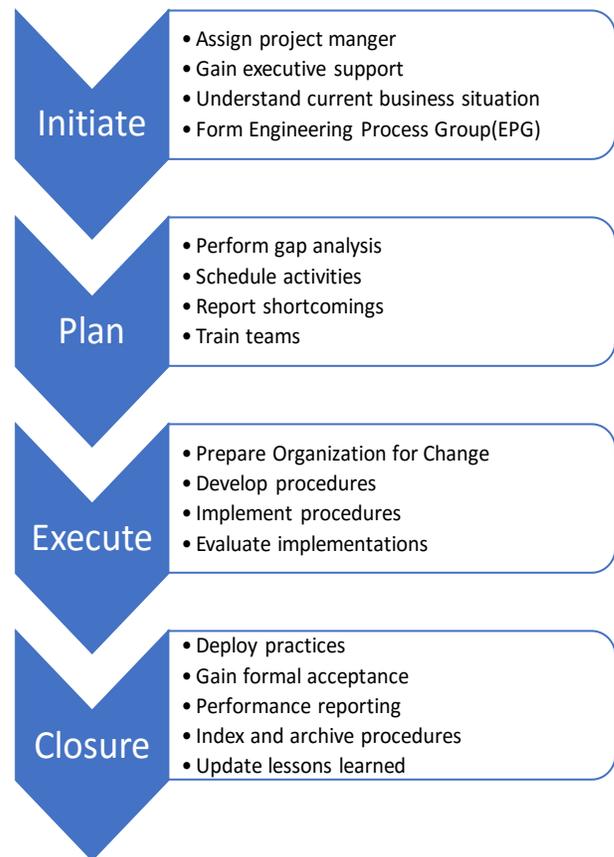


Fig. 6. CMMI Development Project Activities.

TABLE II. SAMPLE ITERATIONS IN CMMI SPIRAL MODEL

Iteration#	Phase1	Phase2	Phase3	Phase4
1	Study current business	Identify the best business unit to cover	Document current development processes	Plan for EPG team
2	Form EPG	Check alternative resources and project constraints (cost, time)	Announce the group and authorize the group to CMMI-DEV	Plan for Gap analysis
3	Run gap analysis	Manage resistance also, gain executive support	Document discrepancies from CMMI practice	Plan for training
4	Train EPG team	Allocate sponsorship and timeframes	Certify members or evaluate their knowledge gain	Plan for development
5	Develop practices	Manage resources, scope and cost	Document new procedures	Plan for practices' implementation
6	Implement procedures (can be more than one iteration)	Identify and control change	Verify new changes on the organization	Plan for deployment
7	Deploy new practices (can be more than one iteration)	Resolve issues related to current practices adopted by the organization	Evaluate and validate the practices	Plan for formal acceptance
8	Gain formal acceptance	Plan for possible rejection (plan B)	Communicate change and formalize the implementations	Plan for closure
9	Run closure activities	Identify alternatives to future change and document ownership.	Communicate closure	Plan for continuous improvement activities
10	Go to activities in iteration 1			

V. EVALUATION AND DISCUSSION

Although many implementations have been carried out to implement CMMI-DEV, most of them are executed by certified CMMI partners. To the authors' knowledge, no implementation methods were reported. The proposed method is an effective tactic envisioned to implement CMMI-DEV using currently available resources on small organizations that target the CMMI level 2.

The proposed method is evaluated against the following evaluation criteria proposed by this paper:

- A. Easy to implement: In small organizations with limited resources, the implementation should be carried out systematically and with ease.
- B. Fit for specification: The goal of the implementation is to transfer the organization to the required level of the CMMI maturity model.
- C. Fit for purpose: The purpose of the implementation is to increase the productivity of the organization and reduce the risk of project failures without adding extra unneeded efforts. Generally, small organizations do not look for CMMI appraisals.
- C. Satisfaction: All related stakeholders should be satisfied with the improvement made in the organization.
- D. Cost: Small organizations with low budgets harvest the opportunity of currently available resources.

E. Table III shows the comparison of the spiral method and a list of carefully chosen works from the literature — the ☑ signposts that the criterion is an obtainable while ☒ means unavailable criterion.

According to Table III, all methods are fit for specification (Criterion B), that is, their goal is to reach the CMMI best practice at the intended level. Moreover, all compared models are easy to implement (Criterion A), except for the general CMMI-DEV guide as it leaves the implementation details to the developers. Also, compared models are fit for purpose (Criterion C), except the KSA model [14] which is limited to the PPQA process area. The proposed spiral model is competitive to the compared models concerning satisfaction (Criterion D) and cost (Criterion E) perspectives.

Furthermore, the proposed framework has been evaluated by expert judgment of two project managers, three senior managers from three different organizations who are willing to implement the CMMI. Project managers approved and appreciated the proposed spiral model; however, they were concerned with what is going to happen during project progress. The senior managers advocate the approach but were worried about how to manage resistance during the implementation process.

This paper extends previous work on CMMI implementation guidelines and provides operational implementations activities as a project. The proposed approach provides several immediate implications for research and practice; therefore, it should help in the evaluation and CMMI-DEV, and the adoption of the proposed model in small organizations.

TABLE III. COMPARISON OF THE PROPOSED METHOD AND RELATED CMMI IMPLEMENTATION APPROACHES

Criterion	Proposed Spiral method	CMMI Guide [13] [17]	[14] [15]	Certified CMMI Partners	Implementation tools
A	☑	☒	☑	☑	☑
B	☑	☑	☑	☑	☑
C	☑	☑	☒	☑	☑
D	☑	☑	☒	☑	☒
E	☑	☒	☑	☒	☒

Despite the vital contributions of this paper, the spiral approach encompasses limitations that should be well-thought-out when construing the results. First, the model was verified with a set of criteria for small organizations. This limitation may restrain the generalizability of the results on large organizations looking for appraisal certifications. Second, the proposed approach has not been implemented in any organization due to organizations privacy concerns. Therefore, results possibly only imitate the general concept of iterative CMMI implementation. Future work should consider conducting an extensive survey on various small organizations, before proceeding to approve the implementation approach.

VI. CONCLUSION

CMMI is a software process improvement model that improves efficiency, productivity and reduced the implementation cost of organization projects. The CMMI as a reference model does not specify systemic steps of how to implement the model, leaving room for organization development methods. The paper proposed a spiral software model to implement CMM-DEV 1.3 in small software development organizations. The model powers the use of small iteration based on available resources and organization constraints, and risk management to reduce failure. The proposed implementation approach was compared with a set of CMMI implementation approaches. The proposed model was deemed applicable and viable on cost and satisfaction criteria.

REFERENCES

- [1] Atoum, C. H. Bong, and N. Kulathuramaiyer, "Towards Resolving Software Quality-in-Use Measurement Challenges," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 5, no. 11, pp. 877–885, 2014.
- [2] I. Atoum, "A Novel Framework for Measuring Software Quality-in-use based on Semantic Similarity and Sentiment Analysis of Software Reviews," *J. King Saud Univ. - Comput. Inf. Sci.*, p. , 2018.
- [3] J. Samalikova, R. J. Kusters, J. J. M. Trienekens, and A. J. M. M. Weijters, "Process mining support for Capability Maturity Model Integration-based software process assessment, in principle and in practice," *J. Softw. Evol. Process*, vol. 26, no. 7, 2014.
- [4] A. G. A. Saeed, R. S. Afgun Usmani, H. Akram, S. M. Saqlain, "The Impact of Capability Maturity Model Integration on Return on Investment in IT Industry: An Exploratory Case Study," *Eng. Technol. Appl. Sci. Res.*, vol. 7, no. 6, 2017.
- [5] D. Chevers, "Software Process Improvement: Awareness, Use, and Benefits in Canadian Software Development Firms," *Rev. Adm. Empres.*, vol. 57, no. 2, pp. 170–177, Apr. 2017.
- [6] Software Engineering Institute, "CMMI for Development, Version 1.3," Carnegie Mellon University, 2010.
- [7] R. W. Reitzig, D. R. Goldenson, D. Gibson, and M. R. Cavanaugh, "Calculating CMMI-Based ROI: Why, When, What, and How?," 2007.
- [8] C. Ebert, "Technical controlling and software process improvement," *J. Syst. Softw.*, vol. 46, no. 1, pp. 25–39, Apr. 1999.
- [9] S. Bayona, J. A. Calvo-Manzano, and T. San Feliu, "Critical Success Factors in Software Process Improvement: A Systematic Review," in *Software Process Improvement and Capability Determination*, 2012, pp. 1–12.
- [10] K. Lee, Y. Park, and D. Lee, "Measuring efficiency and ICT ecosystem impact: Hardware vs. software industry," *Telecomm. Policy*, 2017.
- [11] L. Valverde, M. M. da Silva, and M. R. Gonçalves, "CMMI-DEV v1.3 Reference Model in ArchiMate," in *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, 2018, pp. 191–208.
- [12] B. W. Boehm, "A spiral model of software development and enhancement," *Computer (Long Beach, Calif.)*, vol. 21, no. 5, pp. 61–72, 1988.
- [13] SEI CMMI Production Team, *CMMI for Development v1.3*. Lulu.com, 2010.
- [14] I. Keshta, M. Niazi, and M. Alshayeb, "Towards Implementation of Process and Product Quality Assurance Process Area for Saudi Arabian Small and Medium Sized Software Development Organizations," *IEEE Access*, vol. 6, pp. 41643–41675, 2018.
- [15] J. Persse, *Project management success with CMMI: seven CMMI process areas*. Upper Saddle River, NJ: Prentice Hall, 2007.
- [16] J. Aguiar, R. Pereira, J. B. Vasconcelos, and I. Bianchi, "An overlapless incident management maturity model for multi-framework assessment (ITIL, COBIT, CMMI-SVC)," *Interdiscip. J. Information, Knowledge, Manag.*, pp. 137–163, 2018.
- [17] R. Nandyal, *CMMI: a framework for building world class software and systems enterprises*. New Delhi: Tata McGraw-Hill, 2004.
- [18] CMMI-Live, "CMMI® Tool That Can Reduce Implementation Time and Cost." [Online]. Available: www.cmmilive.com/cmmi-tool-that-can-reduce-implementation-time-and-cost/. [Accessed: 01-Mar-2019].
- [19] DQS India, "CMMI Process Implementation Toolkit." [Online]. Available: <https://dqsindia.com/cmmi/toolkit/>. [Accessed: 01-Mar-2019].
- [20] SourceForge.net, "CMMI Management and Monitoring Tool download." [Online]. Available: <https://sourceforge.net/projects/cmmi-toolmanager/>. [Accessed: 01-Mar-2019].