العنوان:	Capability Maturity Model of Software Requirements Process and Integration (SRPcmmi)
المؤلف الرئيسي:	Najjar, Sireen Kamal
مؤلفين آخرين:	Al Sarayreh, Khalid Torki(Advisor)
التاريخ الميلادي:	2016
موقع:	الزرقاء
الصفحات:	1 - 59
رقم MD:	819762
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	البرامج، هندسة البرمجيات، هندسة المتطلبات
رابط:	https://search.mandumah.com/Record/819762

Chapter 1

Introduction

1.1 Overview

This chapter provides a general background on software requirement engineering process. It also states the problem of the study and the hypotheses to examine. In addition, the chapter outlines the significance, and the organization of the thesis.

1.2 Background of the Study

Requirements are the basis of any software development project, as they drive all activities that follow. It is widely acknowledged that software projects are critically vulnerable when the requirement-related activities are poorly performed [1-5]. As a result, it is very important to get requirements right – otherwise, the entire project will fail.

The term "requirement" has been defined in many research and standards. The IEEE standard glossary of software engineering technology [6] defines it as:

1) "A condition or capability needed by a user to solve a problem or achieve an objective.
2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.
3) A documented representation of a condition or capability as in 1 or 2".

Software requirement engineering process is the process of defining, documenting and maintaining requirements [7]. It is also defined as "the broad spectrum of tasks and techniques that lead to an understanding of requirements" [8]. Requirement engineering is one of most important software engineering phases that begins during the communication activity and continues into the modeling activity. The main goal of this process is to ensure the completeness, consistency, correctness, and relevance of the software requirements.

RE process goes through a set of activities that can be implemented simultaneously or parallel in order to deliver a clear, consistent, complete, modifiable and traceable set of requirements. They are classified into two types of activities namely, requirement development and requirement management. Requirement development contains activities related to discovering, analyzing, documenting, and validating requirements. Requirement management includes activities related to requirement and requirement change management.

Literature in software engineering area showed that system and software development projects have been plagued with problems since the 1960s [1], and that RE process remains the most problematic of all software engineering activities [2]. Therefore, RE process improvements area became a critical and a central research topic in the field of software engineering. Several guidelines, standards and specialized RE process models were constructed to help organization in assessing and improving their RE process and solving their technical and organizational problems.

The thesis focuses on improving RE process by proposing a new requirement engineering process capability maturity model. Maturity model is a type of models that is applied within the context of software process improvement (SPI). It provides indication of the quality of the software process and to which degree the specialist should understand and apply the process [8].

The construction of the proposed model is based on the CMMI-DEV model [4]. It is one of the most common used maturity models that were developed in the last decade order to help organizations to understand and implement their RE process. Moreover, it is one of the few process models that attempts to define maturity levels of IT-related processes [9].

1.3 Statement of the Problem

Once the software product has been deployed, it is typically straight-forward to observe whether or not a certain requirements have been met, as the areas of success or failure in their context can be strictly defined. The problem of lacking any early process of integration within the software requirements and CMMI models is likely to cause an increase in the effort and maintenance.

The importance of the imposed software requirements and CMMI phases with the integration process requires management of their scope, which brings up the importance of clearly defining, and tracing the complex and frequently ill-defined process and propose a solution for this case by a new standard based requirement engineering process improvement model.

This thesis identifies three major areas to investigate:

1) CMM and CMMI models
2) Software Requirements Knowledge area [5]
3) Integration process between 1 and 2 in order to propose the new model

Requirement engineering process is an effective phase of software development life cycle. It can help to reduce software errors at the early stage of the development of software, and produce high quality software, through delivering a clear, consistent, complete, modifiable and traceable set of requirements.

Organizations and software companies have realized that in order to improve their RE process and its related activities they need to follow a well-defined and best practice based model. The literature on the RE process showed that there are several standards, guidance and maturity models related to requirements improvements field. However, these models suffer from several problems that restrict their adoption such as they are too complex, developed based on an old or un-supported versions of the capability maturity model (CMM) , or support limited type of RE process.

1.4 Research Hypothesis

The capability maturity model that is introduced in this thesis aims to improve the requirement engineering process. More specifically, to help practitioners to better define, understand and apply of the requirement engineering process activities efficiently with the CMMI phases.

The proposed model is used to validate two hypotheses; First hypothesis is "To help organizations in implementing and improving their RE practices ". The second hypothesis is that "Provides solutions for some of the problems and limitations posed by the previous models including the CMMI".

1.5 Significance of the Study

Requirement engineering is considered important and critical for the success of software products [10, 11]. Several studies showed that improving RE process has a significant impact on improving productivity [12, 13], assuring quality [12, 14], and reducing project risk [15]. On the other side, if requirement engineering process is not improved then software development will face many problems such as, requirements conflict, lack of customer satisfaction, resource unavailability, and time delay [16]. The new proposed model, Capability Maturity Model of Software Requirements Process and Integration (SRP^{CMMI}), is a specialized RE process improvement and maturity model that provides organizations with detailed descriptions of how to interpret and implement their RE practices.

1.6 Thesis Organization

After explaining the reasons behind doing the research and the hypothesis it examines. The thesis proceeds as follows: Chapter 2 describes the methodology. Chapter 3 reviews the existing models related to the requirement engineering improvements area that appears in literature. It also provides an overview of the CMM/CMMI-based maturity models. An overview of the CMMI model is presented in chapter 4.

Chapter 5 describes the introduced capability maturity model of software requirements process, integration, and the components included. The maturity levels are described in details in chapter 6 and 7.

Finally, Chapter 8 discusses the results of the study, and presents the conclusions drawn from the study in addition to some future directions.

العنوان:	Capability Maturity Model of Software Requirements Process and Integration (SRPcmmi)
المؤلف الرئيسي:	Najjar, Sireen Kamal
مؤلفين آخرين:	Al Sarayreh, Khalid Torki(Advisor)
التاريخ الميلادي:	2016
موقع:	الزرقاء
الصفحات:	1 - 59
رقم MD:	819762
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	البرامج، هندسة البرمجيات، هندسة المتطلبات
رابط:	https://search.mandumah.com/Record/819762

Chapter 2

Research Methodology

A research methodology helps researchers to improve and characterize their research priorities and ensure that they use appropriate procedures. It is one of the main research success factors, since it helps researchers to validate their research and certify that they use suitable consistent methods. This chapter describes the methodology that is followed and it is composed of five main phases (See Figure 1) as follow:

Phase 1: Literature Review

Conducting a literature review is an essential step to understand to which extent the researchers have gone through the research topic, and to improve the understanding of the research problem. Hence, this step presents overview of the previous and published work by academia and best practices on the field of requirement engineering improvements, especially those developed based on the CMM/CMMI models**.**

Phase 2: Overview of the CMMI-DEV Model

This phase presents the structure of the CMMI-DEV model in terms of components and levels. It is also discusses the reasons behind referring to the CMMI –DEV as a base model for developing a RE maturity model. In addition, we show the problems that limit the adaption of the CMMI-DEV model.

Phase 3: Propose a New Capability Maturity Model of Software Requirements Process and Integration

Based on the CMMI-DEV model in the previous phase a new capability maturity model of software requirement engineering process and integration is proposed, this phase encompasses the following seven steps:

- **Step 1:** Defining the model structure.
- **Step 2:** Defining the requirement engineering process areas.
- **Step 3:** Defining RE goals and their related RE practises for each process area.

- **Step 4:** Defining sub-practices, techniques, and work products related to each RE practice.
- **Step 5:** Defining the model maturity levels.
- **Step 6:** Mapping each RE process area and its corresponding goals and practices to the appropriate maturity level.
- **Step 7:** Explain how to use the proposed model.

Phase 4: Results and Conclusion

In this phase we discuss the model characteristics and illustrate how the model deals with the problems of the previous models. Moreover, we show the limitations and the challenges that confronted the construction of the model.

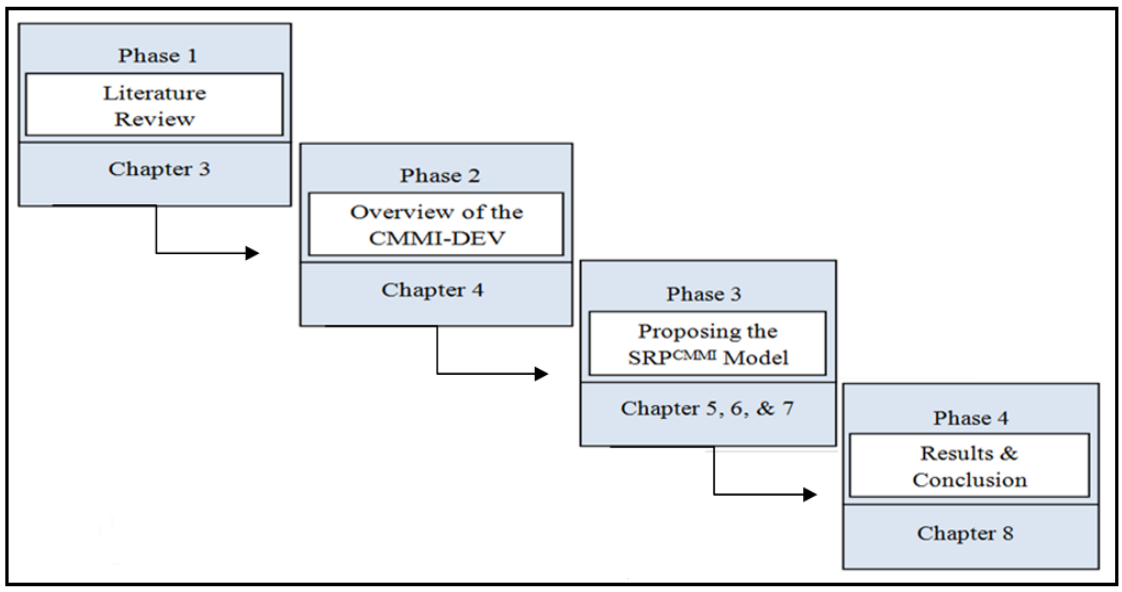

Figure 1: Thesis methodology

العنوان:	Capability Maturity Model of Software Requirements Process and Integration (SRPcmmi)
المؤلف الرئيسي:	Najjar, Sireen Kamal
مؤلفين آخرين:	Al Sarayreh, Khalid Torki(Advisor)
التاريخ الميلادي:	2016
موقع:	الزرقاء
الصفحات:	1 - 59
رقم MD:	819762
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	البرامج، هندسة البرمجيات، هندسة المتطلبات
رابط:	https://search.mandumah.com/Record/819762

Chapter 3

Literature Review

3.1 Overview

This chapter describes and summarizes the most relevant related work in the area of requirement engineering improvements (REI). Section 3.2 presents the most popular REI standards, guidelines, and models, while Section 3.3 reviews the CMM, CMMI models, and several other models that were developed based on them.

3.2 Requirement Engineering Improvement

Literature in software engineering area showed that RE process remains the most challenging of all software engineering activities [2]**.** As a result, and due to the increased consideration for the risks posed to software development projects by weak requirements engineering practices. RE process improvements became a critical and a central research topic in the field of software engineering. Several RE standards that provide general principles and detailed guidelines for performing the RE process were proposed such as, the IEEE Recommended Practice for Software Requirements Specifications [17], the IEEE Guide for Developing System Requirements Specifications [18], and the IEEE Guide for Information Technology [19]. However, these standards do not provide support to organization particularly in selecting appropriate methods or in designing an optimized RE process [20].

Several advices and recommendation for requirements engineering process improvement are identified clearly in [21, 22]. It provides sets of RE practices guidelines and recommendations. In addition to several practical advices on how organizations can improve their RE process. However, these recommendations and advices are not presented in a process maturity model [3].

Furthermore, many organizations tended to improve their RE process by adopting the Software Process Improvement (SPI) approach [23]. The SPI is "a systemic procedure for improving the performance of an existing process system by changing or updating the process" [24]. A European survey of organizations used the SPI programs during the 1980s showed that the then available SPI models do not help them in handling requirements problems [19].

The development of the capability maturity model was the result of the work done by (Humphrey, 1989) [25]. The idea of the model is based on the concept of the process maturity that represents the degree to which a process is defined, managed, measured, controlled, and effective [26]. The more mature a process, the more it is able to meet targets for cost, time of delivery and product quality accurately [20]. Five levels of maturity were defined by the CMM as shown in Figure 2.

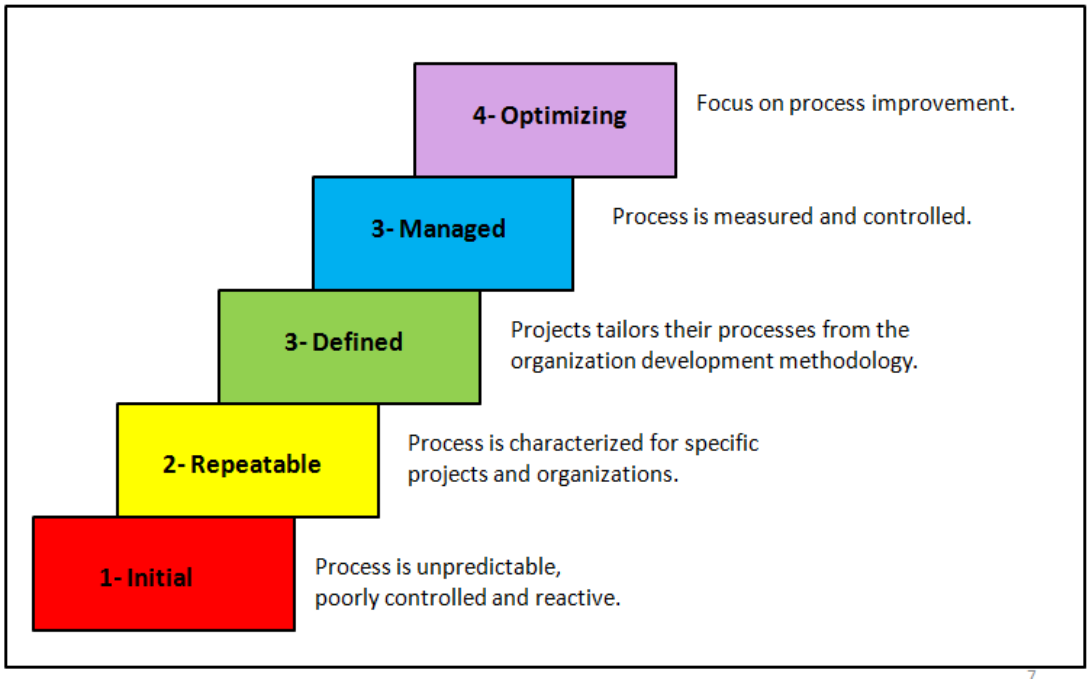

Figure 2: CMM maturity levels.

The success of the CMM directed to the development of several CMMs for a variety of software engineering subjects including the CMMI. The CMMI defines a six levels of maturity namely: incomplete, performed, managed, defined, quantitatively managed and optimizing as shown in Figure 3. Among 22 process areas that were defined in the CMMI, two process areas are related to the requirements engineering process improvements: Requirements Management (REQM) and Development (RD), in order to improve and assess these process areas a set of related goals and practices in each level should be implemented.

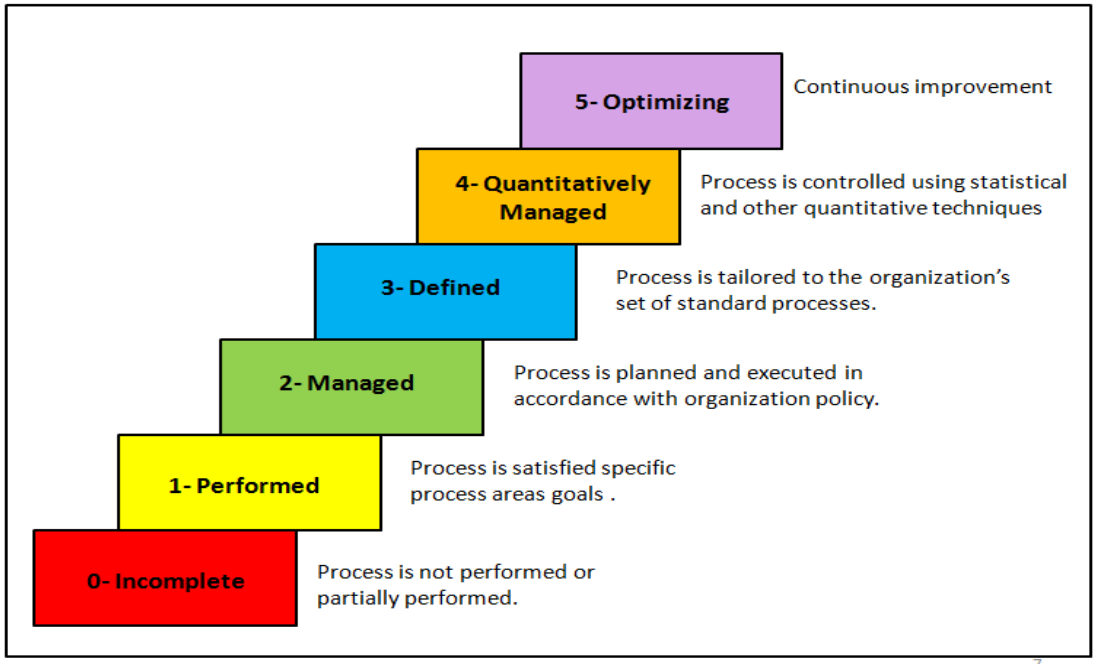

Figure 3: CMMI maturity levels.

Despite the great success of the CMM and CMMI, they did not pay detailed attention to the requirement engineering [27]. Consequently, a number of RE process improvement models were developed such as the Requirements Engineering Good Practice Guide (REGPG) [1]. The REGPG was the first public-domain process improvement and assessment model. It uses an improvement framework with three process maturity levels namely; initial, repeatable, and defined. The REGPG groups good practices for different requirements engineering activities in 66 guidelines, and classifies them as basic, intermediate, and advanced guidelines. For each guideline a score that indicates the rate of usage is assigned, and then the maturity level is determined based on the summation of the numerical scores and the practices' classifications. However, at the time of REGPG's design, the implementation of RE practices across the industry was inconsistent [28-29].

Inspired by the REGPG (Gorscheck *et.al*, 2002) [30] proposed a requirements engineering process maturity model called the REPM. The model uses six levels maturity model and defines 3 main process areas: elicitation, analysis and negotiation, and management. For each process area, a number of actions were assigned and used for the project evaluation process. For example, the REGPG assessment process uses checklist of 60 actions for an organization to reach a higher level of maturity. All actions of these levels must be completed and satisfied. A pilot study for evaluating the

REPM [31] showed that validation of the model's applicability was not an explicit goal of the evaluation.

Based on the REPM, a new maturity model called Uni-REPM [32] was proposed, it is a light-weight model that helps organizations in assessing and identifying the strengths and weaknesses of their process through a recommended improvement path toward a better requirements engineering process. Moreover, a model called a Market-Driven Requirements Engineering Process Model (MDREPM) [33] was developed. The MDREPM is not only a collection of good practices in market-driven requirements engineering area it is also an assessment tool that helps organizations in getting a snapshot of the current state of their MDRE practices.

3.3 Overview of the CMM/CMMI-Based Maturity Models

Several models related to the RE process improvements area developed in the past few years based on the CMM/CMMI models. Beecham *et.al* [2] proposed a specialized process improvement model called the R-CMM. It is a direct adaptation of the SW-CMM framework for assessing the capability of the sub processes that the RE process consists from. The model defines a set of 68 processes distributed over five maturity levels and classified according to the RE process activities, called phases. These phases are management, elicitation, analysis and negotiation, documentation and validation. Similar to the REGPG process assessment the model assess the degree to which a process is satisfied by an organization through allocating a score to each process against three assessment criteria:

- Approach: A measure of the organizational commitment and capability.
- Deployment: A measure of the degree to which a process is implemented across the organization.
- Results: A measure of the success of a process implementation

Each assessed process is then given a rating of: − outstanding (10) − qualified (8) − marginally qualified (6) − fair (4) − weak (2) − poor (score 0). The scores for all five phases are then summed and represented the overall score.

The whole R-CMM was then re-defined according to the characteristics of the capability maturity model for integration and development (CMMI-DEV) v1.2 by Solemon *et.al* [34]. A new Story Cards Based Requirements Engineering Maturity model based on the CMM levels is proposed in [35]. The model focused on how to

improve the agile RE practices and solve the problems related to the story cards like requirements conflicts, missing and ambiguous requirements.

The most recent work was a specialized RE process improvement and assessment model called the REPAIM [3]. The model was built based on the capability levels of the (CMMI-DEV) and composed of two main components: the PMM-RE which stands for process maturity model for RE and contains definitions for the RE process maturity levels and the FLA-RE which stands for Flexible Lightweight Assessment method for RE. It describes assessment requirements, stages and steps. The model defines four RE maturity levels: incomplete, performed, managed and defined. Each level consists of RE goal and related RE practices. For an organization to reach a particular RE maturity level all practices associated to this level must be satisfied.

In addition to the requirement improvement domain, the great success of using the CMM/CMMI in software process improvement has triggered the world wide to use them as a base for developing and adapting wide range of software process maturity models in different domains. For example, the Information Process Maturity Model (IPMM) [36] defines five maturity levels: ad-hoc, rudimentary, organized and repeatable, managed and sustainable, and optimizing. The IPMM process assessment based on eight key characteristics: organizational structure, quality assurance, planning, estimating and scheduling, hiring and training, publications design, cost control, and quality management. These characteristics are used to evaluate information-development organizations through describing the practices that make them successful.

Similar to the structure of the CMMI and based on practitioners' experience and international standards the software maintenance maturity model (SM^{mm}) was proposed in [37]. It defines six maturity levels namely: incomplete, performed, managed, established, predictable, and optimizing. The used the roadmap concept – a set of related practices that represents a significant capability for a software maintenance organization-, the main goal of this model is to assess and improve the quality of software maintenance function.

A maturity model for the implementation of software process improvement is proposed in [38]. The model was constructed based on CMMI and several resources from the SPI literature. It is composed of three components: SPI implementation plan, SPI implementation roadmap, and SPI implementation model. The aim of this model is

to help organizations in designing effective implementation strategies for software process improvement.

A framework for the validation and verification capability assessment in testing domain is proposed in [39]. The framework focused on the safety-criticality and it goes through five criticality-based V&V capability levels: none, low, mediate, high and rigorous. These levels and their associated V&V tasks were defined based on the CMMI process areas. The main goal of this framework is to provide the essential V&V practices that support the assessment of the "safety-criticality".

Based on the terms, concepts and maturity levels of the CMM/CMMI, IS12207, and IS15288 the Business Process Maturity Model (BP^{MM}) was proposed in [40]. The main goal of this model is to help companies in analyzing the strengths and the weaknesses of their business processes through comparing the maturity of their current practices against an industry standard. Hence, they can improve their process and achieve the organization's business objectives. Another model called Capability Maturity Model for Business Intelligence proposed in [41] to business process domain. The model consists of five levels namely: initiate, harmonize, integrate, optimize and perpetuate. It is based on the business intelligence maturity concepts. The model goal is to assess and evaluate capabilities of organizations in the field of BI.

E-Learning Maturity Model was developed in [42] based on the CMM and the SPICE models and consists from five process areas representing the whole e-learning life cycle from planning to delivery and evaluation. The model goal is to help organizations in measuring and improving process maturity from multiple aspects.

A configuration management capability model for medical device industry was developed [43]. The model aims to help medical device companies in improve the effectiveness and efficiency of configuration management capability. They track their evolution against five capability levels which were adopted based on the CMMI generic goals.

TMMi foundation developed test maturity model integration in [44] as complementary model to the CMMI Version 1.2. The model aimed to improve the test process through five levels of maturity namely; initial, managed, defined, measured and optimization. Each level has a set of process areas and its related goals. These goals must be implemented and satisfied to achieve the desired improvement at each level.

Applying TMMi has a positive impact on product quality, test engineering productivity, and cycle-time effort.

Based on the CMM 5- level structure a model called the Risk Management Capability Maturity Model for Complex Product Systems (CoPS) projects was developed in [45]. The model consists of five maturity levels: ad hoc, initial, defined, managed, and optimizing and 10 key capability areas grouped into three categories: organization context, PM/RM process, and technology content. The main goal of this model is to improve the predictability and controllability of CoPS on different types of risks such as process, organizational, and technical-related risks.

Moreover, Capability Maturity Model (CMM) for scientific data management practices was proposed in [46]. The model goes through five maturity levels namely: initial, managed, defined, quantitatively managed and optimizing. The model provides a well-defined set of goals, objectives and practices that covers all aspects of data management. The main goal is to help organizations in assessing their projects and organizational data management practices and plans.

3.4 Summary

This chapter summarized the most relevant related work in the area of requirement engineering improvements. There are several standards, guidelines, recommendations and specialized RE improvement models proposed in the literature in order to help organizations in enhancing the implementation of their RE process and its related practices. However, these works suffered from several problems that restrict their adoption, for example, they were developed based on an old or unsupported version of the CMM, they support limited type of RE process models, they did not pay a detailed attention to the RE process, they did not defined the RE process the way it should be defined regarding to the industry, they did not represented in a process maturity model, or they had not been validated.

العنوان:	Capability Maturity Model of Software Requirements Process and Integration (SRPcmmi)
المؤلف الرئيسي:	Najjar, Sireen Kamal
مؤلفين آخرين:	Al Sarayreh, Khalid Torki(Advisor)
التاريخ الميلادي:	2016
موقع:	الزرقاء
الصفحات:	1 - 59
رقم MD:	819762
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	البرامج، هندسة البرمجيات، هندسة المتطلبات
رابط:	https://search.mandumah.com/Record/819762

Chapter 4

Capability Maturity Model Integration (CMMI): An Overview

Capability maturity model integration (CMMI) is one of the most common used maturity models that is developed by the Software Engineering Institute (SEI). The CMMI is described on the official CMMI website [47] as "a collaborative effort to provide models for achieving product and process improvement. The primary focus of the project is to build tools to support improvement of processes used to develop and sustain systems and products. The output of the CMMI project is a suite of products, which provides an integrated approach across the enterprise for improving processes, while reducing the redundancy, complexity and cost resulting from the use of separate and multiple capability maturity models (CMMs)".

This chapter provides an overview about the CMMI model covering its evolution, structure, advantages and limitations. Moreover, the chapter illustrates how and when to use the CMMI in the requirement improvement area.

4.1 The CMMI Evolution

The Software Engineering Institute released the first version of Capability Maturity Model Integration for Development v1.02 [48] (CMMI-DEV) in 2000. It is a process improvement approach for product and service development organizations that provides the essential elements of effective processes. It is a combination of three source models the Capability Maturity Model for Software (SW-CMM) v2.0 draft C, the Systems Engineering Capability Model (SECM), and the Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98. Two years later version 1.1 was released and four years after that, version 1.2 was released.

Two other CMMI models are released after that for acquisition v1.2 [49] in 2007, and CMMI for services v1.2 [50] in 2009. In November 2010 version 1.3 [4] of these models were released. Figure 4 illustrates the history of the CMMI.

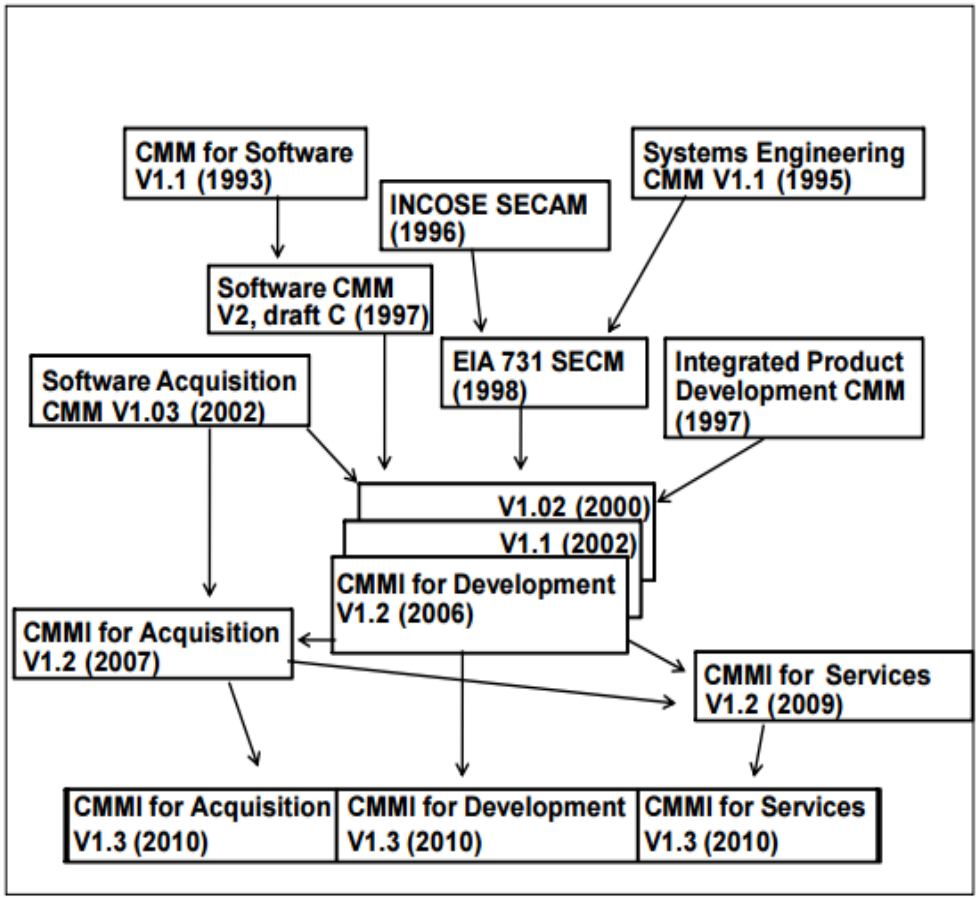

Figure 4: The history of the CMMI [4]

4.2 The CMMI Process Areas

A process area (PA) is a set of related practices that are implemented together in order to satisfy a set of goals that are important for making improvement in a specific area. The CMMI-DEV defines 22 process areas distributed over four categories: Project Management, Process Management, Engineering and Support as shown in Table 1. Each process area has the following component – See Figure 5:

- Purpose statements: A description of the purpose of the process area.
- Introductory notes: A description of the major concepts covered in the process area.
- Related process areas: A set of references to related process areas.
- Generic goals: A description of the characteristics that must be present to institutionalize processes that implement a process area.
- Generic practices: A description of the activities that are mandatory for achieving the generic goal and contribute in the institutionalization of the processes associated with a process area.

- Generic practice elaborations: A description of how to apply the generic practices uniquely to process areas.
- Specific goal: A unique characteristic that is mandatory for satisfying the process area.
- Specific practices: A description of an activity that is mandatory for achieving the associated specific goal.
- Example work products: A list of outputs produced from a specific practice.
- Subpractices: A detailed description of how to interpret and implement a specific or generic practice.

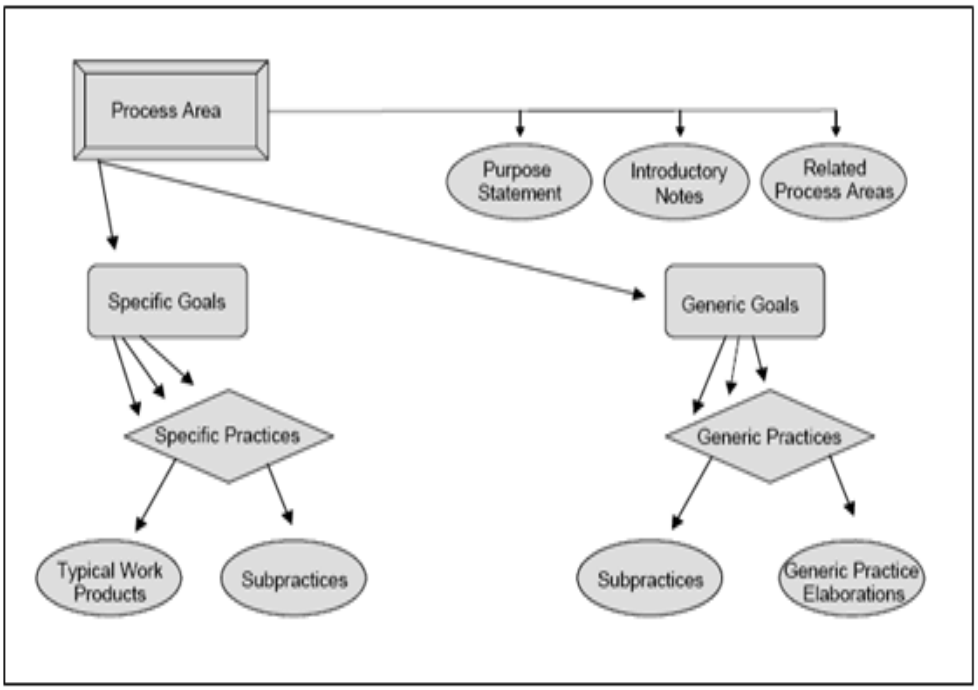

Figure 5: The CMMI model components [4]

Table 1: Process areas and their associated categories and maturity levels

Maturity level	Process Management	Project Management	Engineering	Support
Initial				
Managed		Project Planning (PP)	Requirements Management (REQM)	Configuration Management (CM)
		Project Monitoring and Control (PMC)		Process and Product Quality Assurance (PPQA)
		Supplier Agreement Management (SAM)		Measurement and Analysis (MA)
Defined	Organization Process Focus (OPF)		Requirements Development (RD)	Decision Analysis and Resolution (DAR)
	Organizational Process Focus (OPF)+ Integrated Product and Process Development (IPPD)	Integrated Project Management (IPM) + Integrated Product and Process Development (IPPD)	Technical Solutions (TS) Product Integration (PI)	
		Supplier Agreement Management (SAM)		Measurement and Analysis (MA)
	Organization Training (OT)	Risk Management (RSKM)	Verification (VER)	
Quantitatively Managed	Organization Process Performance (OPP)	Quantitative Project Management (QPM)		
Optimizing	Organization Innovation and Deployment (OID)			Causal Analysis and Resolution (CAR)

4.3 The CMMI Capability and Maturity Levels

The CMMI defines two levels - an improvements path that must be followed by organizations that want to improve their processes – capability and maturity levels. Capability levels are those that are related to the path that helps organizations to apply an incremental improvement to processes with corresponding to a given process area. It goes through 0 to 5 levels as follows:

0) Incomplete level: Process that either is not performed or partially performed.

1) Performed: A process that satisfies the specific goals of the process area.
2) Managed: A performed process that is planned and executed in harmony with organization policy.
3) Defined: A managed process that is tailored to the organization set of standard processes.
4) Quantitatively managed: A defined process that is controlled using statistical and other quantitative techniques.
5) Optimizing: A quantitatively managed process that focuses on continuous improvements.

Maturity levels are related to the path that helps organizations to apply improvements to a set of related processes by incrementally addressing successive sets of process areas. It goes through 1 to5 as follows:

1) Initial: There is no formal process.
2) Managed: There is a minimal process and the status of projects is visible to management at major milestones.
3) Defined: Processes are well characterized and understood. Processes are described in standards, procedures, tools, and methods.
4) Quantitatively managed: The organization and the projects establish quantitative objectives for quality and process performance. The quantitative objectives are then used as criteria in managing processes.
5) Optimizing: All processes are already defined and managed. Goals for all levels from 1 to 4 are all achieved successfully.

The difference between the capability and maturity levels is that the

4.4 Rationales of Using CMMI–DEV for Developing a RE Maturity Model

The CMMI-DEV model is one of the most common used maturity models that has been adopted worldwide. Building our model on a well-known software process improvement model has many advantages including but not limited to:

1. The easy accessibility of this model compared to other models.
2. It is a well-known framework that has been used widely and it is one of the few process models that attempts to define maturity levels for IT-related processes [9]. Implementing CMMI in an IT organizations has many benefits including:

- The integration of software engineering best practices in the organizations
- Cost saving in terms of effort due to less defects and less rework
- Shorter cycle time
- On-Time deliveries [51].
- Improved process effectiveness [51].
- Improved quality and productivity
- Improved estimation [52].
- Increased customer satisfaction
- Increased the return of investments
- Decreased costs

3. It is based on best practices and contains guidelines for RE practices. It defines two process areas relating to RE process: requirements management (REQM) and requirements development (RD). REQM: The purpose of this process area is to manage the products and product components requirements and to ensure alignment between those requirements and the project's work products. It defines the following specific goals and practices:

SG 1 Manage requirements

SP 1.1 Understand requirements

SP 1.2 Obtain commitment to requirements

SP 1.3 Manage requirements changes

SP 1.4 Maintain bidirectional traceability of requirements

SP 1.5 Ensure alignment between project work and requirements

Requirements development (RD): The purpose of this process area is to elicit, analyze, and establish customer, product, and product component requirements. It defines the following specific goals and practices:

SG 1 Develop customer requirements

SP 1.1 Elicit needs

SP 1.2 Transform stakeholder needs into customer requirements

SG 2 Develop product requirements

SP 2.1 Establish product and product component requirements

SP 2.2 Allocate product component requirements

SP 2.3 Identify interface requirements

SG 3 Analyze and validate requirements

SP 3.1 Establish operational concepts and scenarios

SP 3.2 Establish a definition of required functionality and quality attributes

SP 3.3 Analyze requirements

SP 3.4 Analyze requirements to achieve balance

SP 3.5 Validate requirements

4. The CMMI-DEV is designed to be tailored and adapted to focus on specific needs as it is a normative model [53].
5. The great success of the CMMI in the area of process improvement has triggered the world wide to use it as source model for developing and adapting wide range of software process maturity models in different domains such as requirements, maintenance, project management, security and testing as shown in the literature review section.

4.5 Limitations of CMMI-DEV

1. The CMMI-DEV doesn't define RE maturity the way it should be defined based on industry standards and practice [9]. The CMMI-DEV defines two PAs in two separate maturity levels. Requirements engineering process area (REQM) first at maturity level 2, followed by requirements development process area (RD) at maturity level 3.

   Requirements elicitation is supposed to be institutionalized in the maturity level 3, while requirements management is defined at maturity level 2. This order is not always logical and can create several issues [9], for example if an organizations does not have an institutionalized way of eliciting requirements until maturity level 3 how they can manage these requirements at maturity level 2 .
2. CMMI-DEV does not provide organizations with enough details of how to actually perform their REQM and RD work. Thus, they are forced to depend on

the CMMI training and/or experiences of their team members which increase the cost of using the model.

3. The number of organizations adopting the CMMI-DEV in most countries is still low with 58 as shown in the report published by SEI [54]. A study proposed by (Staples *et al*, 2007) [55] and another by (Khurshid *et al*, 2009) [56] showed that the main reasons of not adopting CMMI is the high cost associated with the model. Organizations were unsure of the SPI benefits, organizations were too small, or organizations had other priorities.

العنوان:	Capability Maturity Model of Software Requirements Process and Integration (SRPcmmi)
المؤلف الرئيسي:	Najjar, Sireen Kamal
مؤلفين آخرين:	Al Sarayreh, Khalid Torki(Advisor)
التاريخ الميلادي:	2016
موقع:	الزرقاء
الصفحات:	1 - 59
رقم MD:	819762
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	البرامج، هندسة البرمجيات، هندسة المتطلبات
رابط:	https://search.mandumah.com/Record/819762

Chapter 5

The Capability Maturity Model of Software Requirements Process and Integration (SRP^{CMMI})

5.1 Overview

This chapter introduces the capability maturity model of software requirements process and integration (SRP^{CMMI}), and provides a brief view of the model structure, components and resources.

5.2 Model Structure

The construction of SRP^{CMMI} structure was based on the latest version of the CMMI-DEV model. The model hierarchy has three levels: RE process area (REPA), RE goals (REGs) and RE practices (REPs). On the top level of the model, there are six RE process areas corresponding to requirements engineering main activities. Each REPA is further broken down into several REGs, which represents a unique characteristic that is mandatory for satisfying the process area. On the bottom level, REPs denotes a description of an activity that is mandatory for achieving the associated goal. The model components are summarized and illustrated in figure 6.

5.2.1 Requirements Engineering Process Area (REPA):

Like the CMMI-DEV the top level of the model consist of RE process areas which represents a set of related practices that are implemented together in order to satisfy a set of goals that are important for making improvement in a specific area .

Unlike CMMI-DEV that defines two process areas (requirement development and requirement management) related to the requirement engineering process. SRP^{CMMI} deals with the RE process as a knowledge area and define five REPAs in level 1. It is represented here according to the active order in the requirements engineering process that is defined by The Software Engineering Body of Knowledge (SWEBOK) [5].

Elicitation: In this RE process area -known as requirements gathering- stakeholder needs are collected in order to build an understanding of the problem the software is required to solve.

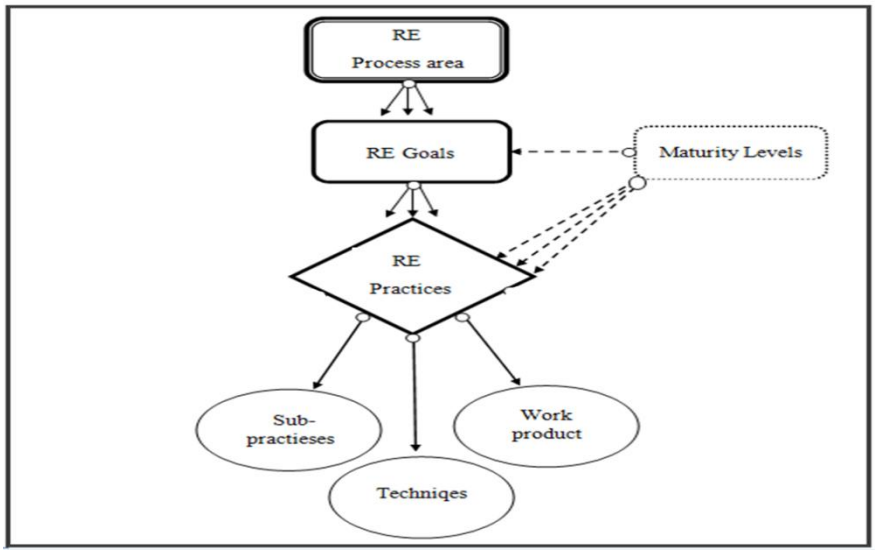

Figure 6: Components of the SRP^{CMMI}

Analysis: In this RE process area an analysis model that identifies data, function, features, constraints and behavioral requirements is created based on the information obtained during elicitation.

Specification: In this RE process area a description of the function, performance of a computer-based system and the constraints that will govern its development is produced in the form of requirements specification.

Validation: In this RE process area the quality of work products produced as a result of requirements engineering activities are assessed. The specification is examined to ensure that requirements are correct, complete, consistence, testable and satisfies customer needs.

Requirements Management: In this process area the project team executes a set of activities to identify, control, and track requirements and all changes that occur at any time as the project proceeds.

For level 2 the SRP^{CMMI} defines one REPA namely organizational support. It contains practices that support the implementation of the REPAs in level 1.

Organizational Support: In this process area organizational RE policy is established, relevant stakeholders are involved, resources are allocated, people are trained, responsibilities are assigned, and RE process adherence is evaluated.

5.2.2 Requirements Engineering Goal (REG):

Like the CMMI-DEV each process area has one or more RE goal which represents a unique characteristic that must be implemented for satisfying the process area. Unlike CMMI-DEV the model consider all goals as RE goals rather than separating them into generic and specific goals.

5.2.3 Requirements Engineering Practice (REP):

Like the CMMI-DEV each RE goal has one or more RE practice, which is a description of the activities that should be performed for achieving the associated goals and for improving organizations' RE processes, Unlike CMMI-DEV the model consider all practices as RE practices rather than separating them into generic and specific practices.RE practice in the SRP^{CMMI} consists of the following components:

- Sub-practices: A detailed description of how to interpret and implement a RE practice.
- Technique: A list of all techniques that can be used to perform a RE practices.
- Work product: A list of the outputs that resulted from an RE practice.

5.3 SRP^{CMMI} Maturity Levels

The SRP^{CMMI} model basically consists from three maturity levels which were adapted from the CMMI capability levels; the reason behind using the capability levels is that they enable the user to select the order of practices implementation, so they can use different RE process models. The SRP^{CMMI} levels are numbered 0 through 2: Incomplete, Performed, and Managed as shown in Figure 7.

Each RE maturity level of SRP^{CMMI} consist of a RE process area and its related RE goals and practices, reaching a particular RE maturity level depend on satisfying all of the targeted RE practices. The three RE maturity levels of our model provide a detailed description of how organization can measure and improve their RE process. The three maturity levels are as follow:

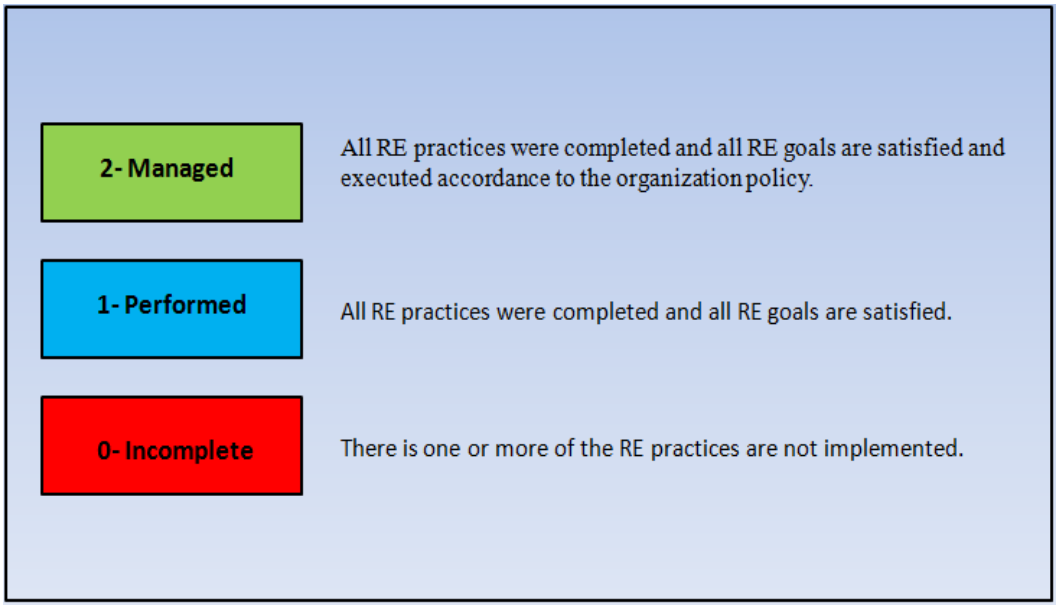

Figure 7: SRP^{CMMI} maturity levels

5.3.1 Level 0-Incomplete RE Process:

An incomplete RE process is similar to the capability level 0 of the CMMI-DEV where it either is not performed or partially performed, in other words there is one or more of the RE practices are not implemented. Since there is no reason to institutionalize a partially performed RE process this level doesn't have a RE goal.

5.3.2 Level 1-Performed RE Process:

A performed RE process is a process that satisfies the RE goals and implements all the RE practices of the process area. At this maturity level, the SRP^{CMMI} defines five RE process areas: elicitation, analysis, specification, validation, and requirements management where the requirements are gathered, analyzed, prioritized, documented, validated, and requirements changes and traceability are managed, each process area has RE goals and related RE practices as follow:

A. Elicitation:

REG1: Establish an understanding about the problem, solutions and stakeholders**.**

REP1: Identify stakeholder and requirements resources**.**

REP 1.2: Elicit needs.

B. Analysis:

REG1: Create an analysis model that identifies data, function, features, constraints and behavioral requirements using the information obtained during elicitation.

REP 1.1: Establish operational concepts and scenarios.

REP 1.2: Model requirements.

REP1.3: Analyze requirements.

REP1.4: Analyze requirements to achieve balance.

C. Specification:

REG1: Formalizes the informational, functional, and behavioural requirements.

REP1.1: Produce a document that can be systematically reviewed, evaluated, and approved.

D. Validation:

REG1: Resolve conflicts, prioritize requirements, and identify risks, in order to gain a win-win result before proceeding to subsequent software engineering activities.

REP1.1: Resolve conflicts.

REP1.2: Prioritize requirements.

REP1.3: Obtain commitment to requirements.

REG2: Validate requirements to ensure the resulting product will perform as intended in the end user's environment.

REP1.1: Confirming that requirements are correct, complete, consistence, testable and satisfies customer needs.

E. Requirements Management:

REG1: Identifying, controlling and keeping track of all changes that occurs to the requirements.

REP1.1: Manage requirements changes.

REG2: Ensure that all source requirements are completely addressed.

REP2.1: Manage requirements traceability.

5.3.3 Level 2-Managed RE Process:

A managed process is a process that is planned and executed in accordance with organization policy, the SRP^{CMMI} define the organizational support process areas to this maturity level where RE Policy is established; relevant stakeholders are involved; resources are allocated; people are trained; responsibilities are assigned; and RE process adherence is evaluated, the RE practices at these process area are constructed by referring to four of the CMMI process areas namely project planning, organizational training, project management and control, and process and product quality assurance.

REG1: Evaluate the amount of organizational support given to requirements engineering practices.

REP1.1: Establish an organizational requirements engineering policy.

REP1.2: Monitor stakeholder involvement.

REP1.3: Identify project resources.

REP1.4: Assign responsibility.

REG2: Develop skills and knowledge, and train people so they can perform their roles effectively and efficiently.

REP2.1: Define a training program(s).

REP2.2: Deliver training.

REP2.3: Establish training records.

REP2.4: Assess training effectiveness.

REG3: Objectively evaluate adherence against applicable process descriptions, standards, and procedures.

REP3.1: Objectively evaluate processes and work products.

5.4 Model Sources

The construction of the SRP^{CMMI} model was based on the software engineering institute capability maturity model integration for development version 1.3, RE practices of level 1 are selected mostly by referring to the generic and specific goals and practices that are defined by the REQM and RD process areas of the CMMI-DEV.

RE practices of level 2 are selected by referring to the generic and specific goals and practices, that are defined by four CMMI-DEV process areas namely, project planning,

organizational training, project management and control and process and product quality assurance.

Several references like books and articles on software engineering are also used in order to add new practices or to provide more detailed guidelines such as:

1. The Software Engineering Body of Knowledge (SWEBOK) [5].
2. Software Engineering: A Practitioner's Approach book [8].
3. The Requirement Engineering Process Improvement and Assessment Model (REPAIM) [3].

5.5 Numbering scheme

- RE goals are numbered sequentially. Each RE goal begins with the prefix "REG" (e.g., REG 1).
- RE practices are also numbered sequentially. Each RE practice begins with the prefix "REP" followed by a number in the form "x.y" (e.g., REP 1.1). The x is the same number as the RE goal to which the RE practice maps. The y is the sequence number of the RE practice under the RE goal.
- All RE goals and practices in the model are numbered sequentially. However, the way the practices are listed in the sequence does not specify the order of implementation. The order of implementation depends on the used process model for example; Iterative or agile process model may require that the practices be implemented in parallel, while waterfall process model may require that the practices be implemented simultaneously.

5.6 Representations of the SRP^{CMMI}

Figure 8 illustrates a sample of the practices by process area representation. The figure also shows the main components of the SRP^{CMMI}, namely process areas, RE goal, RE practices, techniques, work product, and Subpractices.

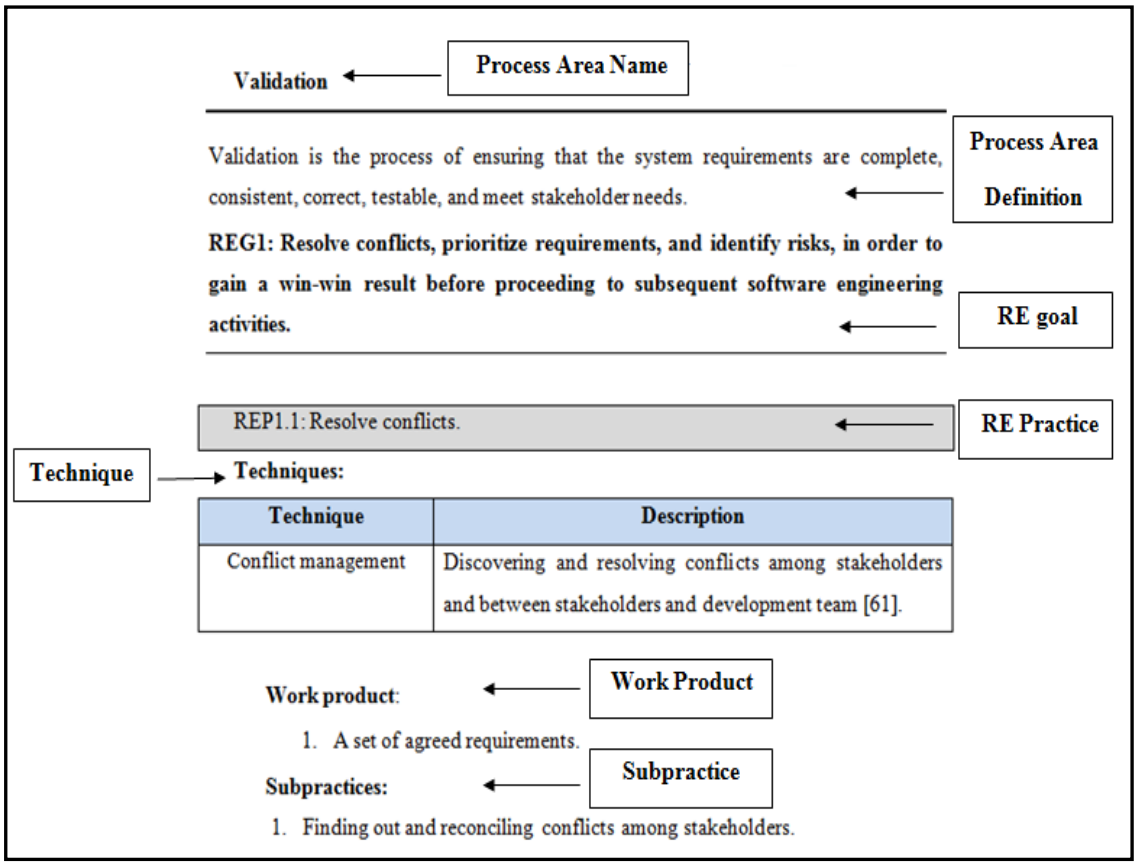

Figure 8: Practices by Process Area in the SRP[CMMI].

5.7 Model Usage

5.7.1 Who will use the model?

SRP[CMMI] intend to assess the RE process maturity; thus it can be used by people who are deeply understand the RE process, and involved in the process improvement in general such as software engineer, quality assurance engineer, project manager, and product manager.

5.7.2 How to use the model?

To assess the maturity of a RE process, the user should make a mapping from the RE practices presented in the model to the activities in his real RE process. He could find out one of the following situations:

1. There is one or more of the RE practices are not implemented. It should be marked as “Incomplete process”.
2. All RE practices were completed and all RE goals are satisfied in the RE process area. It should be marked as “performed process”.

3. All RE practices were completed and all RE goals are satisfied and executed accordance to the organization policy, it should be marked as "Managed process".

5.8 Summary

The capability maturity model of software requirements process and integration (SRP^{CMMI}) that is constructed based on the latest version of the CMMI-DEV model. The model includes six RE process areas namely; elicitation, analysis, specification, validation, requirement management, and organizational support, for each process area the model defines a number of RE goals and practices and provides a detailed description of how to implement these practices in order to satisfy the process area and obtain the desired improvement. Moreover, the model defines three maturity levels; incomplete, managed and performed RE process that will be discussed in details in the next two chapters.

العنوان:	Capability Maturity Model of Software Requirements Process and Integration (SRPcmmi)
المؤلف الرئيسي:	Najjar, Sireen Kamal
مؤلفين آخرين:	Al Sarayreh, Khalid Torki(Advisor)
التاريخ الميلادي:	2016
موقع:	الزرقاء
الصفحات:	1 - 59
رقم MD:	819762
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	البرامج، هندسة البرمجيات، هندسة المتطلبات
رابط:	https://search.mandumah.com/Record/819762

Chapter 6

Level 1: Performed Requirement Engineering Process

6.1 Overview

A performed RE process is a procedure that satisfies the RE goals and implements all the RE practices of the process area. At this maturity level, requirements are gathered, analyzed, prioritized, documented, validated, and requirements changes and traceability are managed.

There are thirteen RE practices implemented in this RE maturity level and distributed over five RE process areas: elicitation, analysis, specification, validation, and requirements management. The adaption of these RE practices are guided mostly by the generic and specific goals and practices of the REQM and RD process areas of the CMMI-DEV.

This chapter presents the RE goals related to this level and their associated RE practices, techniques, and work products.

6.2 Elicitation Process Area

Elicitation

Elicitation is the process of discovering, gathering, documenting, and understanding the user's needs and constraints for the system.

REG1: Establish an understanding about the problem, solutions and stakeholders

REP1.1: Identify stakeholder and requirements resources.

Techniques: Shown in Table 2.

Table 2: Elicitation techniques for identifying stakeholder and requirements resources

Technique	Definition
Interview	"Meeting between a requirements engineer and a stakeholder to discuss topics of relevance for the system" [57].
Scenarios	"A story of how users and systems interact to achieve a goal"[58].
Observation	"Study of system use, possibly in the target environment and by real users, to understand usage processes and strengths and weaknesses of a current system" [59].
User stories	A description of a software feature from an end-user perspective.

Work products: A list of stakeholders and requirements resources.

Subpractices:

1. Identify and involve stakeholders.
2. Identify overall goals, and benefits of the system
3. Identify the nature of the solution that is desired
4. Identify in general the operational and organizational environment in which the solution will be used.
5. Identify some of the constraints that may affect the solution

REP 1.2: Elicit needs.

Techniques: Shown in Table 3.

Table 3: Elicitation techniques for eliciting needs

Technique	Definition
Collaborative requirement gathering	Gathering requirements through communication between the stakeholders (business expert, developer, and customer).
Quality function deployment	"A quality management technique that translates the needs of the customer into technical requirements for software" [8]

Work products:

Several work products will be produced depending on the system, including one or more of the following items:

1. A statement of need and feasibility
2. A bounded statement of scope for the system or product
3. Technical and organizational environment description.
4. A list of requirements and the domain constraints.
5. A set of preliminary usage scenarios that describe the use of the system under different operating conditions.

Subpractices:

1. Elicit information about stakeholder's needs, expectation, constraint, and interfaces.
2. Elicit information about system's technical infrastructure, business process, operational domain, and boundaries.

6.3 Analysis Process Area

Analysis

Analysis is the process of refining the user's needs and constraints, checking requirements, and resolving stakeholder conflicts.

REG1: Create an analysis model that identifies data, function, features, constraints and behavioral requirements using the information obtained during elicitation.

REP1.1: Establish operational concepts and scenarios.

Techniques: Shown in Table 4.

Table 4: Analysis techniques for establishing operational concepts and scenarios

Technique	Definition
Use cases	Description of the behavior of a system from a user's point of view**.**

Work products:

1. Operational concept.
2. Use cases.
3. Timeline scenario.
4. New requirements.

Subpractices:

1. Define the environment in which the software will operate.
2. Define boundaries and constraints.
3. Develop operational concepts and scenarios that illustrate the interactions between the software, the end user and the environment.
4. Review operational concepts and scenarios to refine and discover new requirements.

REP1.2: Model requirements

Techniques: Shown in Table 5.

Table 5: Analysis techniques for modeling requirements

Technique	Description
Use cases	Description of the behaviour of a system from a user's point of view.
Activity diagrams	Graphical representations of the flow of control in a system.
Class diagram	Graphical representations for the system classes, operations, attributes, and interrelationships among them.
State diagram	Graphical representations for the event driven state change of a system.
Data flow	Graphical representation of the flow of data within a system.

Work products:

1. Analysis model
2. Scenario-based model
3. Flow model
4. Behavioural model
5. Class model

Subpractices:

1. Determine business objectives and drivers in order to identify desirable quality attributes.
2. Analysis different scenarios with relevant stakeholders in order to identify desirable functionality and quality attribute.
3. Analyze and quantify functionality required by end users.
4. "Partition requirements into groups, based on established criteria (e.g., similar functionality, similar quality attribute requirements, coupling)" [4], and allocate customer requirements to these partitions to facilitate the analysis of requirement.

REP1.3: Analyze requirements

Techniques: Shown in Table 6.

Table 6: Analysis techniques for analyze requirements

Technique	Description
Process decomposition diagram	Breaking down requirements into processes and analyzed each process to determine whether they are necessary and sufficient to meet the objectives of higher levels requirements.

Work products:

1. Requirements defects reports
2. Proposed requirements changes to resolve defects
3. Key requirements
4. Technical performance measures

Subpractices:

1. Analyze requirements to ensure that they satisfy the objectives of higher level requirements.
2. Analyze requirements to ensure their competence, feasibility, realizably, and verifiability.

REP1.4: Analyze requirements to achieve balance

Techniques: Shown in Table7.

Table 7: Analysis techniques for analyzing requirements to achieve balance

Technique	Description
Simulations	"Approximation and review of the behavior of the system with an appropriate tool to check correctness of the behavior" [60].
Prototyping	"Paper- or tool-based approximation of the end-systems to increase the tangibility and authenticity of the planned system" [61].

Work products: Assessment of risks related to requirements.

Subpractices:

1. Analyze the balance between stakeholder needs and constraints using proven models, simulations, and prototyping
2. Perform requirements risk assessment.

6.4 Specification Process Area

Specification

Specification is the process of documenting the requirements.

REG1: formalize the informational, functional, and behavioral requirements of the proposed software are in both a graphical and textual format.

REP1.1: Produce a document that can be systematically reviewed, evaluated, and approved.

Techniques: Shown in Table 8.

Table 8: Specification techniques for producing a document that can be systematically reviewed, evaluated, and approved

Technique	Description
Natural language	"Specifying requirement with words and sentences to achieve specification flexibility and understandability"[62].
UML diagrams	Using graphical notations to Specify functions, scenarios, processes, rules, relations, behavior, and deployment**.**
Tables	"Specifying concepts to achieve an understanding of the terminology"[63].

Work products:

Software requirements specifications (SRS) may be one of the following:

1. A written document.
2. A set of models.
3. A formal mathematical model.
4. A collection of user scenarios (use-cases).
5. A prototype.

Subpractices:

1. Establish standardized structure for SRS.
2. Define requirements attributes such as Id, title, description, and requirement source, etc…
3. Define requirements states (new, selected, implemented, or rejected).
4. Document requirements rationale.
5. Record rationale for rejected requirements.

6.5 Validation Process Area

Validation

Validation is the process of ensuring that the system requirements are complete, consistent, correct, testable, and meet stakeholder needs.

REG1: Resolve conflicts, prioritize requirements, and identify risks, in order to gain a win-win result before proceeding to subsequent software engineering activities.

REP1.1: Resolve conflicts.

Techniques: Shown in Table 9.

Table 9: Validation techniques for resolving conflicts

Technique	Description
Conflict management	Discovering and resolving conflicts among stakeholders and between stakeholders and development team [64].

Work product: A set of agreed requirements.

Subpractices:

1. Finding out and reconciling conflicts among stakeholders.

REP1.2: Prioritize requirements.

Techniques: Shown in Table 10.

Table 10: Validation techniques for prioritizing requirements

Technique	Description
Prioritizing	"Ranking the requirements to obtain an order of how they shall be addressed by the project work" [65].

Work product: prioritized requirements.

Subpractices:

1. Ranking the requirements to gain an order of how they shall be addressed by the project work.

REP1.3: Obtain commitment to requirements.

Techniques: Shown in Table 11.

Table 11: Validation techniques for obtaining commitment to requirements

Technique	Description
Win-Win negotiation	"Structured, possibly tool-supported approach to identification of options for agreement and selection of the appropriate option" [66].

Work products: Documented commitments to requirements.

Subpractices:

1. Identify the key stakeholders.
2. Determine the stakeholders' "wins conditions".
3. Negotiate all stakeholders until reach "win-win" result.
4. Record commitments.

REG2: Validate requirements to ensure the resulting product will perform as intended in the end user's environment.

REP2.1: Confirming that requirements are correct, complete, consistence, testable and satisfies customer needs.

Techniques: Shown in Table 12.

Table 12: Validation techniques for confirming that requirements are correct, complete, consistence, testable and satisfies customer needs

Technique	Description
Automated checking	"Testing a formal specification of the system to detect conflicting and missing requirements"[67].
Inspection	"Review of the requirements specification by all relevant stakeholders with a formal process that is effective at discovering problems and leads to in-depth understanding of the specification"[68].
Formal technical review	"A method involving a structured encounter in which a group of technical personnel analyzes or improves the quality of the original work product as well as the quality of the method" [69].
Prototype	"Paper- or tool-based approximation of the end-systems to increase the tangibility and authenticity of the planned system"[61].
Acceptance test	"formal testing with respect to user needs, requirements, and business processes conducted to determine whether a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system"[70].

Work product: A set of validated requirements.

Subpractices:

1. Ensure that each requirement is consistent with the overall project objective and goals.
2. Ensure that all requirements are specified at the appropriate level off abstraction.
3. Identify whether the requirement is essential to the project objective or it is an add-on features.
4. Ensure that each requirement bounded and unambiguous.
5. Identify the source for each requirement.
6. Ensure that there is no conflicting between the requirements.

6.6 Requirements Management Process Area

Requirements management

Requirements management is the process of managing requirement and requirement changes through a set of activities that are responsible for identifying, controlling and keeping track of all changes that occurs to software requirements.

REG1: Identifying, controlling and keeping track of all changes that occurs to the requirements.

REP1.1: Manage requirements changes.

Techniques: Shown in Table 13.

Table 13: Requirements management techniques for managing requirements changes

Technique	Description
Change management	"Controlled process of collecting change requests, analysing impact, and deciding about the change"[71].
Progress tracking	"Monitoring the lifecycle of requirements from discovery to selection, implementation, and release"[72].

Work products:

1. Requirements change requests.
2. Requirements change impact reports.
3. Requirements status.
4. Requirements database.

Subpractices:

1. Document all requirements and requirements changes that are produced during the project.
2. Maintain a requirements change history.
3. Evaluate the impact of requirement changes from stakeholder's point of view.
4. Make requirements and change data available to the project.

REG2: Ensure that all source requirements are completely addressed.

REP2.1: Manage requirements traceability

Techniques: Shown in Table 14.

Table 14: Requirements management techniques for managing requirements traceability

Technique	Description
Bidirectional requirements traceability matrix	A matrix that tracks the requirement forward by examining the output of the deliverables and backward by tracing each requirement back to its source(s).

Work products:

1. Requirements traceability matrix.
2. Requirements tracking system.

Subpractices:

1. Assign a unique identifier for each requirement.
2. Document and trace requirements source.
3. Document and trace Requirements' Relations.
4. Produce a requirements traceability matrix.

6.7 Summary

The chapter presented level 1 of the SRP^{CMMI} model**,** The level includes thirteen RE practices related to five RE process areas namely, elicitation, analysis, specification, validation, and requirements management, at this maturity level requirements are gathered, analyzed, prioritized, documented, validated, and requirements changes and traceability are managed. Next chapter will present level 2 the performed RE process and its related RE goals and practices.

العنوان:	Capability Maturity Model of Software Requirements Process and Integration (SRPcmmi)
المؤلف الرئيسي:	Najjar, Sireen Kamal
مؤلفين آخرين:	Al Sarayreh, Khalid Torki(Advisor)
التاريخ الميلادي:	2016
موقع:	الزرقاء
الصفحات:	1 - 59
رقم MD:	819762
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	البرامج، هندسة البرمجيات، هندسة المتطلبات
رابط:	https://search.mandumah.com/Record/819762

Chapter 7

Level 2: Managed Requirement Engineering Process

7.1 Overview

A managed process is a procedure that is planned and executed in accordance with the organization policy. At this maturity level RE policy is established. Relevant stakeholders are involved. Resources are allocated. People are trained. RE process adherence is evaluated. The RE practices at this level are related to one process area called organizational support. The RE practices constructed by referring to four of the CMMI-DEV process areas namely project planning, organizational training, project management, and product quality assurance.

This chapter presents the RE goals related to this level and their associated RE practices, techniques, and work products.

7.2 Organizational Support Process Area

Organizational Support

The organization support process is a set of activities that support the implementation of RE process practices.

REG1: Provide an organizational support to requirements engineering practices.

REP1.1: Establish an organizational requirements engineering policy.

Policy is a set of rules and guidelines adopted by an organization to perform their activities.

Techniques: Shown in Table 15.

Table 15: Organizational support techniques for establishing an organizational requirements engineering policy

Technique	Description
Legislation	The process of making the law.
Regulation	The process of monitoring and enforcing of rules.
Guidelines	A general rule, principle, or piece of advice.
Standards	A rule or principle that is used as a basis for judgment.
Procedures	Step-by-step sequence of activities to perform a task.

Work products:

1. Requirement development policy
2. Requirements management policy
3. Validation policy

Subpractices:

1. Outline rules
2. Outline the procedures
3. Provide principles that guide actions
4. Set roles and responsibilities
5. Reflect values and beliefs
6. State an intention to do something

REP1.2: Identify and involve stakeholders.

Techniques: Shown in Table 16.

Table 16: Organizational support techniques for identifying and involving stakeholders

Techniques	Description
Stakeholder involvement records	Records of stakeholders who involved in RE process**.**

Work Products:

Records of stakeholder involvement

Subpractices:

1. Periodically review and document the status of stakeholder involvement.
2. Identify and document significant issues and their impacts.

REP1.3: Identify project resources.

Technique: Shown in Table 17.

Table 17: Organizational support techniques for identifying project resources

Technique	Description
Work breakdown structure (WBS)	"Deliverable oriented hierarchical decomposition of the work to be executed by the project team." [73]

Work Products:

1. Work packages
2. WBS task dictionary
3. Staffing requirements based on project size and scope
4. Critical facilities and equipment list
5. Process and workflow definitions and diagrams
6. Project administration requirements list
7. Status reports

Subpractices:

1. Determine process requirements.
2. Determine communication requirements.
3. Determine staffing requirements.
4. Determine facility, equipment, and component requirements.

REG2: Develop skills and knowledge, and train people so they can perform their roles effectively and efficiently.

REP 2.1: Define a training program(s).

Techniques: Shown in Table 18.

Table 18: Organizational support techniques for defining training program(s)

Technique	Description
Training program	A program designed for training employees in specific skills, it contains a series of courses, and usually has a flexible time and cost and budget.

Work product: Training program

Subpractices:

1. Identify which training needs are the responsibilities of the organization.
2. Identify the skills and knowledge needed to perform the organization activities.
3. Define appropriate approaches to satisfy organizational training needs.
4. Develop or obtain training materials, qualified instructors, and instructional designers.
5. Develop training program to address the identified organizational needs.

REP 2.2: Deliver training.

Techniques: Shown in Table 19.

Table 19: Organizational support techniques for delivering training

Technique	Description
Training courses	A series of lessons or lectures teaching the skills in a particular field**.**

Work products: Training courses.

Subpractices:

1. Select those who will need the training to perform their roles effectively.
2. Define a training plan and schedule.
3. Deliver the training.
4. Track the delivery of training against the training plan and schedule.

REP 2.3: Establish training records.

Techniques: Shown in Table 20.

Table 20: Organizational support techniques for establishing training records

Technique	Description
Training records	Records of who was trained, when and what skills they have mastered**.**

Work products: Training records.

Subpractices:

1. Keep records of all staff who successfully complete their training courses and even those who are unsuccessful**.**
2. Keep records of all staff who are give-up training.
3. Make training records available to provide a summary of the experience and education of people.

REP 2.4: Assess training effectiveness

Techniques: Shown in Table 21.

Table 21: Organizational support techniques for assessing training effectiveness

Techniques	Description
Post-training surveys of training participants	A technique for measuring the effectiveness of employee training, and assessing the benefits of it against the projects and organizations.

Work products:

1. Training effectiveness surveys
2. Training program performance assessments
3. Instructor evaluation forms
4. Training examinations

Subpractices:

1. Evaluate in-progress or completed projects to determine whether trainees knowledge is adequate for performing project tasks.

2. Offer a mechanism for assessing the effectiveness of training courses with respect to the organization or project objectives.

REG3: Objectively evaluate Adherence against applicable process descriptions, standards, and procedures.

REP3.1: Objectively evaluate processes and products.

Techniques: Shown in Table 22.

Table 22: Organizational support techniques for evaluating processes and products

Techniques	Description
Process or product evaluation criteria	measures to evaluate the process or product adherence against applicable process descriptions, standards, and procedures

Work products:

1. Evaluation reports
2. Noncompliance reports
3. Corrective actions

Subpractices:

1. Establish and use evaluation criteria to evaluate the process or the product adherence against applicable process descriptions, standards, and procedures.
2. Select work products to be evaluated.
3. Record and resolve noncompliance cases found during the evaluation process.
4. Identify and record lessons learned in order to improve the process.

7.3 Summary

The chapter presented level 2 of the SRP^{CMMI} model**,** The level includes eight RE practices related to one process area called organizational support, the main goals of this maturity level is to establish a RE policy, involve stakeholders, allocate resources, training people, and evaluate the RE process adherence.

العنوان:	Capability Maturity Model of Software Requirements Process and Integration (SRPcmmi)
المؤلف الرئيسي:	Najjar, Sireen Kamal
مؤلفين آخرين:	Al Sarayreh, Khalid Torki(Advisor)
التاريخ الميلادي:	2016
موقع:	الزرقاء
الصفحات:	1 - 59
رقم MD:	819762
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	البرامج، هندسة البرمجيات، هندسة المتطلبات
رابط:	https://search.mandumah.com/Record/819762

Chapter 8

Results and Conclusion

8.1 Overview

SRP^{CMMI} is a new requirement engineering process capability maturity model that constructed based on the CMMI-DEV model. The main objective of the new proposed model is to help organizations in improving their RE process. This chapter discusses the proposed model in terms of contributions, properties, and limitations. It also illustrates in what manner the new model deals with the previous models problems.

8.2 Thesis Contribution

The contributions of this thesis are:

- Identifying, discussing and solving the problems posed by the previous requirements engineering improvements models.
- Support the requirement engineering improvements field.
- Proposing a new capability maturity model for software requirement engineering process and integration.

One of the outcomes of this thesis have been published in the International Conference on Intelligent Information Processing, Security and Advanced Communication:

Sireen Najjar and Khalid T. Al-Sarayreh, "Capability Maturity Model for Software Requirement Engineering Process and Integration (SRP^{CMMI})", *Proceeding of the International Conference on Intelligent Information Processing, Security and Advanced Communication (IPAC 2015)*, ISBN: 978-1-4503-3458-7, ACM, ICPS, Algeria, NOV. 2015.

8.3 SRP^{CMMI} Vs Previous Models

Literature in software engineering presented several standards, guidance and maturity models related to requirements improvements field, however these improvements model suffer from several problems that restricts their adoption. Table 23

discusses some of the previous models problems and illustrates how the new proposed model solves them.

Table 23: Previous models on requirements improvements problems and illustrates

Limitations of the previous models	**Proposed SRP^{CMMI}**
Some models were developed based on an old or unsupported version of the CMM	The new model is developed based on the latest version of the CMMI-DEV v1.3, which is a well-known framework that has been used widely to guide process improvement and assessment.
Several models support limited type of RE process models	The maturity levels of the new model have been developed based on the CMMI continuous representation. Hence, SRP^{CMMI} enable users to select the order for RE process practices based on the RE process model. For example, iterative or agile process model may require that the practices be implemented in parallel, while waterfall process model may require that the practices be implemented simultaneously.
CMM/CMMI did not pay detailed attention to the requirement engineering	The model is a specialized RE improvement model that goes deeply into RE process, since it deals with RE process as a knowledge area instead of a phase of SDLC.
The order of RE implementation and institutionalizations in the CMMI is not always logical and can create several issues	The CMMI defines two PAs in two separate maturity levels. REM first at maturity level 2, followed by requirements development process area RD at maturity level 3. Requirements elicitation is supposed to be institutionalized in the maturity level 3, while requirements management is defined at maturity level 2. This order is not always logical. The SRP^{CMMI} model splits the entire RE domain into five REPAs. Elicitation, analysis, specification, validation, and requirements management into one maturity level (level 1). SRP^{CMMI} model defines a new REPA called organizational support in maturity level 2 includes all practices that executed in accordance with organization policy and provides an organizational support to requirements engineering practices.

8.4 Conclusion

Software requirement engineering is the most significant phases of the software development and it affects the overall success of the software. The thesis presented a new capability maturity model for software requirement engineering process and integration (SRP^{CMMI}). SRP^{CMMI} was motivated by the many challenges that organizations suffer when implementing their software requirement engineering process. In addition, the new model is encouraged by the lack of research in the field of requirements engineering improvements.

The SRP^{CMMI} has three maturity levels, incomplete, performed, and managed. All the three levels are adopted based on the continuous representation of the CMMI-DEV model. It has been designed with the following properties:

1. The model is a specialized RE process that looks deeply into RE practices. It treats the RE process as a knowledge area.
2. The model provides a detailed guidance for implementing the RE practices.
3. The model reflects the best practices in RE domain, since it has been constructed based on different sources of references including CMMI.
4. The model support different type of RE process models.

Like other RE process improvement models, the proposed model moves from a high level view of the RE practices, to detailed descriptions in order to guide practitioners towards satisfying their RE process goals and achieving the desired improvements.

8.5 Threat to Validity

- Despite the detailed information provided by the model, the model implementation requires expertise in requirements engineering process. Practitioners should be familiar with applying RE process in order to use the model effectively.
- The validity of the model has not been assessed. The reason behind that is the lack of experienced institutions in RE improvement field. Moreover, it takes years for conducting an empirical study that assess the validity and the efficiency of the model.

8.6 Future Work

For future work we plan to conduct an empirical study in order to validate and prove the efficiency of our proposed model.

العنوان:	Capability Maturity Model of Software Requirements Process and Integration (SRPcmmi)
المؤلف الرئيسي:	Najjar, Sireen Kamal
مؤلفين آخرين:	Al Sarayreh, Khalid Torki(Advisor)
التاريخ الميلادي:	2016
موقع:	الزرقاء
الصفحات:	1 - 59
رقم MD:	819762
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	البرامج، هندسة البرمجيات، هندسة المتطلبات
رابط:	https://search.mandumah.com/Record/819762

Abstract

Capability Maturity Model of Software Requirements Process and Integration (SRP^{CMMI})

By

Sireen Kamal Najjar

Supervisor

Khalid Al-Sarayreh

Software requirement engineering (RE) process is one of the most important phases of the software development life cycle (SDLC) that affect its overall success. RE has a significant role in determining the software quality and software development process effectiveness. Due to the increased consideration for software requirement engineering process and process improvements at the SDLC, several standards and specialized RE improvements model were constructed in order to improve the RE process such as the Requirements Engineering Good Practice Guide (REGPG), Requirements Capability Maturity Model (R-CMM), and the RE Process Improvement and Assessment Model (REPAIM). However, the above models suffer from several problems that limits their acceptance by the organization that are interested in the RE process improvement.

Capability maturity model (CMM) and capability maturity model integration (CMMI) are the most common used maturity models that were developed by the Software Engineering Institute (SEI) in aim to help organizations in assessing and improving their process. The main idea is to focus on the capability of software organizations to produce high-quality products consistently and predictably.

The great success of using the CMM/CMMI in software process improvement has triggered the world wide to use them as a base for developing and adapting wide range of software process maturity models in different domains including but not limited to requirements, maintenance, project management, business process, security and testing.

The research presented in this thesis proposes a new requirement engineering process capability maturity model based on the capability maturity model integration for development (CMMI-DEV) version 1.3. The intention is to provide a generic maturity model that is based on international standards and literature on software requirement engineering.

ملخص

النموذج الكامل النضوج لمتطلبات البرامج و عملياتها

إعداد

سرين كمال النجار

المشرف

د.خالد تركي الصرايرة

(أستاذ مساعد)

عملية هندسة متطلبات البرامج هي احدى اهم مراحل دورة حياة تطوير البرامج و التي تؤثر بشكل كبير على نجاحها,كما ان لها دورا مهما في تحديد جودة البرامج وفعالية تطويرها. و نتيجة لادراك المؤسسات المتزايد لاهمية هذه العملية فقد تم تطوير العديد من المعايير والنماذج المتخصصة من اجل تحسينها,مثل دليل الممارسات الجيدة في هنسة المتطلبات , و نموذج تحسن وتقيين هندسة المتطلبات. ومع ذلك، فإن النماذج المذكورة أعلاه تعاني من العديد من المشاكل التي تحد من تبينها من قبل المؤسسات المهتمة في تحسين عملية هندسة المتطلبات.

يعد نموذج نضوج القدرات المتكامل من اكثر النماذج شيوعا واستخداما في مجال تحسين عملية تطوير البرامج والتي تم تطويرها من قبل مؤسسة هندسة البرمجيات من اجل مساعدة المؤسسات في تقييم وتحسين عملياتها وتركيز قدرات هذه المؤسسات على انتاج برامج ذات جودة عالية على الدوام وكما هو متوقع.

أدى النجاح الكبير الذي حققه استخدام هذه النماذج في تحسين عملية تطوير البرامج الى استخدامها كقاعدة لتطوير وتكييف مجموعة واسعة وجديدة من النماذج كامله النضوج في مجالات مختلفة كعمليه هندسة المتطلبات, والصيانة، وإدارة المشاريع، العمليات التجارية والأمن,وفحص البرامج.

البحث المقدم من خلال هذه الرساله يطرح نموذج هندسة متطلبات جديد بالاعتماد على الاصدار ١.٣ من نموذج نضوج القدرات المتكامل. والهدف من ذلك هو تقديم نموذج عام و كامل النضوج بناءا على المعايير الدولية والدراسات والابحاث السابقة في مجال هندسة متطلبات البرامج.

العنوان:	Capability Maturity Model of Software Requirements Process and Integration (SRPcmmi)
المؤلف الرئيسي:	Najjar, Sireen Kamal
مؤلفين آخرين:	Al Sarayreh, Khalid Torki(Advisor)
التاريخ الميلادي:	2016
موقع:	الزرقاء
الصفحات:	1 - 59
رقم MD:	819762
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	البرامج، هندسة البرمجيات، هندسة المتطلبات
رابط:	https://search.mandumah.com/Record/819762

List of Symbols and Abbreviations

Abbreviation	Meaning	Page number
SRP^{CMMI}	Capability maturity model of software requirements process and integration	1
RE	Requirement engineering	1
SDLC	Software development life cycle	1
CMM	Capability maturity model	1
CMMI	Capability maturity model integration	1
CMMI-DEV	Capability maturity model integration for development	1
SPI	Software process improvement	12
PA	Process area	24
REQM	Requirements management	32
RD	Requirements development	32
REPA	Requirement engineering process area	34
REG	Requirement engineering goals	34
REP	Requirement engineering practice	34

Table of Contents

Table of Contents

List of Tables

List of Figures

العنوان:	Capability Maturity Model of Software Requirements Process and Integration (SRPcmmi)
المؤلف الرئيسي:	Najjar, Sireen Kamal
مؤلفين آخرين:	Al Sarayreh, Khalid Torki(Advisor)
التاريخ الميلادي:	2016
موقع:	الزرقاء
الصفحات:	1 - 59
رقم MD:	819762
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	البرامج، هندسة البرمجيات، هندسة المتطلبات
رابط:	https://search.mandumah.com/Record/819762

Capability Maturity Model of Software Requirements Process and Integration (SRP^{CMMI})

By

Sireen Kamal Najjar

Supervisor

Khalid Al-Sarayreh

Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Software Engineering

Faculty of Graduate Studies at the Hashemite University

Zarqa - Jordan

April/4/2016

This thesis was defended on April/04/2016

Examination committee	**Signature**
Dr. Khalid Al-Sarayreh, Supervisor Assistant Professor, Department of Software Engineering	…….
Dr. Khaled Almakadmeh, Member Assistant Professor, Department of Software Engineering	…….
Dr. Ahmed Fawzi Ali Otoom, Member Assistant Professor, Department of Computer Science	……
Dr. Sufyan Almajali, Member Assistant Professor, Department of Software Engineering. Princess Sumaya University for Technology	…….

Dedication

I dedicate my dissertation work to my family, who has been a source of encouragement and inspiration to me throughout my life. A special feeling of gratitude to my dear husband, Emad and my lovely daughters, Zaina and Asal, thank you for the support and for giving me the time and the space to be able to accomplish my thesis. I am truly thankful for having you in my life.

Acknowledgment

It is my great pleasure to express my sincere gratitude and appreciation to my supervisor Dr. Khalid Al-Sarayreh for his support, patience, encouragement, guidance, and motivation throughout my MSc, from him I have learned how an ideal teacher and monitor should be.

My love, gratitude and special thanks goes to my wonderful parents for their love and support. My sisters and brothers deserve my wholehearted thanks as well.

List of Symbols and Abbreviations

Abbreviation	Meaning	Page number
SRP^{CMMI}	Capability maturity model of software requirements process and integration	1
RE	Requirement engineering	1
SDLC	Software development life cycle	1
CMM	Capability maturity model	1
CMMI	Capability maturity model integration	1
CMMI-DEV	Capability maturity model integration for development	1
SPI	Software process improvement	12
PA	Process area	24
REQM	Requirements management	32
RD	Requirements development	32
REPA	Requirement engineering process area	34
REG	Requirement engineering goals	34
REP	Requirement engineering practice	34

Table of Contents

Table of Contents

List of Tables

List of Figures

Abstract

Capability Maturity Model of Software Requirements Process and Integration (SRP^{CMMI})

By

Sireen Kamal Najjar

Supervisor

Khalid Al-Sarayreh

Software requirement engineering (RE) process is one of the most important phases of the software development life cycle (SDLC) that affect its overall success. RE has a significant role in determining the software quality and software development process effectiveness. Due to the increased consideration for software requirement engineering process and process improvements at the SDLC, several standards and specialized RE improvements model were constructed in order to improve the RE process such as the Requirements Engineering Good Practice Guide (REGPG), Requirements Capability Maturity Model (R-CMM), and the RE Process Improvement and Assessment Model (REPAIM). However, the above models suffer from several problems that limits their acceptance by the organization that are interested in the RE process improvement.

Capability maturity model (CMM) and capability maturity model integration (CMMI) are the most common used maturity models that were developed by the Software Engineering Institute (SEI) in aim to help organizations in assessing and improving their process. The main idea is to focus on the capability of software organizations to produce high-quality products consistently and predictably.

The great success of using the CMM/CMMI in software process improvement has triggered the world wide to use them as a base for developing and adapting wide range of software process maturity models in different domains including but not limited to requirements, maintenance, project management, business process, security and testing.

The research presented in this thesis proposes a new requirement engineering process capability maturity model based on the capability maturity model integration for development (CMMI-DEV) version 1.3. The intention is to provide a generic maturity model that is based on international standards and literature on software requirement engineering.

Chapter 1

Introduction

1.1 Overview

This chapter provides a general background on software requirement engineering process. It also states the problem of the study and the hypotheses to examine. In addition, the chapter outlines the significance, and the organization of the thesis.

1.2 Background of the Study

Requirements are the basis of any software development project, as they drive all activities that follow. It is widely acknowledged that software projects are critically vulnerable when the requirement-related activities are poorly performed [1-5]. As a result, it is very important to get requirements right – otherwise, the entire project will fail.

The term "requirement" has been defined in many research and standards. The IEEE standard glossary of software engineering technology [6] defines it as:

1) "A condition or capability needed by a user to solve a problem or achieve an objective.
2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.
3) A documented representation of a condition or capability as in 1 or 2".

Software requirement engineering process is the process of defining, documenting and maintaining requirements [7]. It is also defined as "the broad spectrum of tasks and techniques that lead to an understanding of requirements" [8]. Requirement engineering is one of most important software engineering phases that begins during the communication activity and continues into the modeling activity. The main goal of this process is to ensure the completeness, consistency, correctness, and relevance of the software requirements.

RE process goes through a set of activities that can be implemented simultaneously or parallel in order to deliver a clear, consistent, complete, modifiable and traceable set of requirements. They are classified into two types of activities namely, requirement development and requirement management. Requirement development contains activities related to discovering, analyzing, documenting, and validating requirements. Requirement management includes activities related to requirement and requirement change management.

Literature in software engineering area showed that system and software development projects have been plagued with problems since the 1960s [1], and that RE process remains the most problematic of all software engineering activities [2]. Therefore, RE process improvements area became a critical and a central research topic in the field of software engineering. Several guidelines, standards and specialized RE process models were constructed to help organization in assessing and improving their RE process and solving their technical and organizational problems.

The thesis focuses on improving RE process by proposing a new requirement engineering process capability maturity model. Maturity model is a type of models that is applied within the context of software process improvement (SPI). It provides indication of the quality of the software process and to which degree the specialist should understand and apply the process [8].

The construction of the proposed model is based on the CMMI-DEV model [4]. It is one of the most common used maturity models that were developed in the last decade order to help organizations to understand and implement their RE process. Moreover, it is one of the few process models that attempts to define maturity levels of IT-related processes [9].

1.3 Statement of the Problem

Once the software product has been deployed, it is typically straight-forward to observe whether or not a certain requirements have been met, as the areas of success or failure in their context can be strictly defined. The problem of lacking any early process of integration within the software requirements and CMMI models is likely to cause an increase in the effort and maintenance.

The importance of the imposed software requirements and CMMI phases with the integration process requires management of their scope, which brings up the importance of clearly defining, and tracing the complex and frequently ill-defined process and propose a solution for this case by a new standard based requirement engineering process improvement model.

This thesis identifies three major areas to investigate:

1) CMM and CMMI models
2) Software Requirements Knowledge area [5]
3) Integration process between 1 and 2 in order to propose the new model

Requirement engineering process is an effective phase of software development life cycle. It can help to reduce software errors at the early stage of the development of software, and produce high quality software, through delivering a clear, consistent, complete, modifiable and traceable set of requirements.

Organizations and software companies have realized that in order to improve their RE process and its related activities they need to follow a well-defined and best practice based model. The literature on the RE process showed that there are several standards, guidance and maturity models related to requirements improvements field. However, these models suffer from several problems that restrict their adoption such as they are too complex, developed based on an old or un-supported versions of the capability maturity model (CMM) , or support limited type of RE process.

1.4 Research Hypothesis

The capability maturity model that is introduced in this thesis aims to improve the requirement engineering process. More specifically, to help practitioners to better define, understand and apply of the requirement engineering process activities efficiently with the CMMI phases.

The proposed model is used to validate two hypotheses; First hypothesis is "To help organizations in implementing and improving their RE practices ". The second hypothesis is that "Provides solutions for some of the problems and limitations posed by the previous models including the CMMI".

1.5 Significance of the Study

Requirement engineering is considered important and critical for the success of software products [10, 11]. Several studies showed that improving RE process has a significant impact on improving productivity [12, 13], assuring quality [12, 14], and reducing project risk [15]. On the other side, if requirement engineering process is not improved then software development will face many problems such as, requirements conflict, lack of customer satisfaction, resource unavailability, and time delay [16]. The new proposed model, Capability Maturity Model of Software Requirements Process and Integration (SRP^{CMMI}), is a specialized RE process improvement and maturity model that provides organizations with detailed descriptions of how to interpret and implement their RE practices.

1.6 Thesis Organization

After explaining the reasons behind doing the research and the hypothesis it examines. The thesis proceeds as follows: Chapter 2 describes the methodology. Chapter 3 reviews the existing models related to the requirement engineering improvements area that appears in literature. It also provides an overview of the CMM/CMMI-based maturity models. An overview of the CMMI model is presented in chapter 4.

Chapter 5 describes the introduced capability maturity model of software requirements process, integration, and the components included. The maturity levels are described in details in chapter 6 and 7.

Finally, Chapter 8 discusses the results of the study, and presents the conclusions drawn from the study in addition to some future directions.

Chapter 2

Research Methodology

A research methodology helps researchers to improve and characterize their research priorities and ensure that they use appropriate procedures. It is one of the main research success factors, since it helps researchers to validate their research and certify that they use suitable consistent methods. This chapter describes the methodology that is followed and it is composed of five main phases (See Figure 1) as follow:

Phase 1: Literature Review

Conducting a literature review is an essential step to understand to which extent the researchers have gone through the research topic, and to improve the understanding of the research problem. Hence, this step presents overview of the previous and published work by academia and best practices on the field of requirement engineering improvements, especially those developed based on the CMM/CMMI models**.**

Phase 2: Overview of the CMMI-DEV Model

This phase presents the structure of the CMMI-DEV model in terms of components and levels. It is also discusses the reasons behind referring to the CMMI –DEV as a base model for developing a RE maturity model. In addition, we show the problems that limit the adaption of the CMMI-DEV model.

Phase 3: Propose a New Capability Maturity Model of Software Requirements Process and Integration

Based on the CMMI-DEV model in the previous phase a new capability maturity model of software requirement engineering process and integration is proposed, this phase encompasses the following seven steps:

- **Step 1:** Defining the model structure.
- **Step 2:** Defining the requirement engineering process areas.
- **Step 3:** Defining RE goals and their related RE practises for each process area.

- **Step 4:** Defining sub-practices, techniques, and work products related to each RE practice.
- **Step 5:** Defining the model maturity levels.
- **Step 6:** Mapping each RE process area and its corresponding goals and practices to the appropriate maturity level.
- **Step 7:** Explain how to use the proposed model.

Phase 4: Results and Conclusion

In this phase we discuss the model characteristics and illustrate how the model deals with the problems of the previous models. Moreover, we show the limitations and the challenges that confronted the construction of the model.

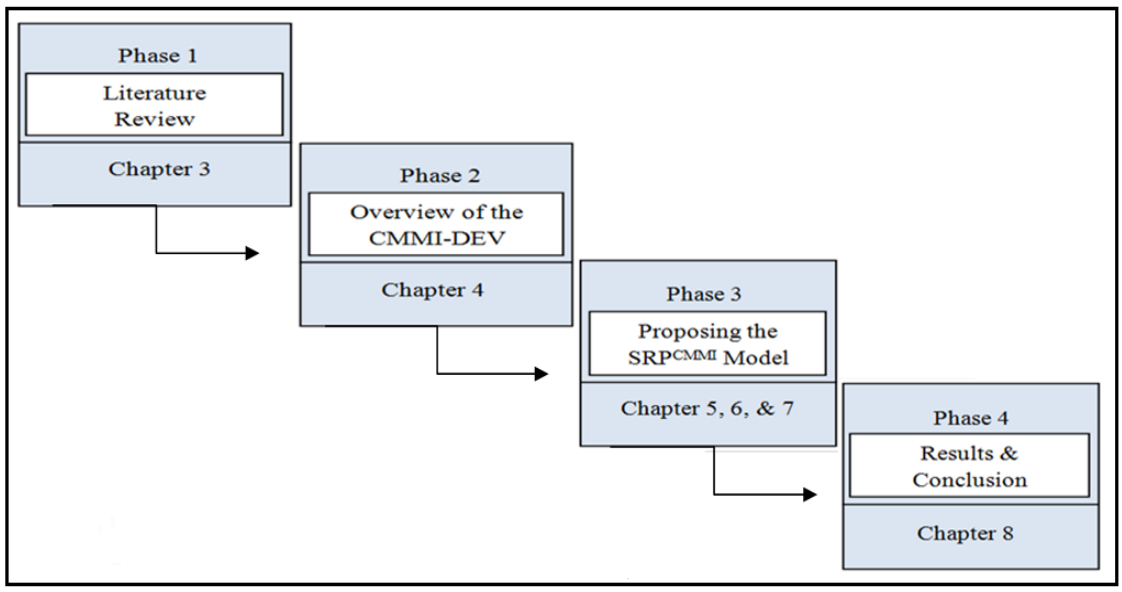

Figure 1: Thesis methodology

Chapter 3

Literature Review

3.1 Overview

This chapter describes and summarizes the most relevant related work in the area of requirement engineering improvements (REI). Section 3.2 presents the most popular REI standards, guidelines, and models, while Section 3.3 reviews the CMM, CMMI models, and several other models that were developed based on them.

3.2 Requirement Engineering Improvement

Literature in software engineering area showed that RE process remains the most challenging of all software engineering activities [2]**.** As a result, and due to the increased consideration for the risks posed to software development projects by weak requirements engineering practices. RE process improvements became a critical and a central research topic in the field of software engineering. Several RE standards that provide general principles and detailed guidelines for performing the RE process were proposed such as, the IEEE Recommended Practice for Software Requirements Specifications [17], the IEEE Guide for Developing System Requirements Specifications [18], and the IEEE Guide for Information Technology [19]. However, these standards do not provide support to organization particularly in selecting appropriate methods or in designing an optimized RE process [20].

Several advices and recommendation for requirements engineering process improvement are identified clearly in [21, 22]. It provides sets of RE practices guidelines and recommendations. In addition to several practical advices on how organizations can improve their RE process. However, these recommendations and advices are not presented in a process maturity model [3].

Furthermore, many organizations tended to improve their RE process by adopting the Software Process Improvement (SPI) approach [23]. The SPI is "a systemic procedure for improving the performance of an existing process system by changing or updating the process" [24]. A European survey of organizations used the SPI programs during the 1980s showed that the then available SPI models do not help them in handling requirements problems [19].

The development of the capability maturity model was the result of the work done by (Humphrey, 1989) [25]. The idea of the model is based on the concept of the process maturity that represents the degree to which a process is defined, managed, measured, controlled, and effective [26]. The more mature a process, the more it is able to meet targets for cost, time of delivery and product quality accurately [20]. Five levels of maturity were defined by the CMM as shown in Figure 2.

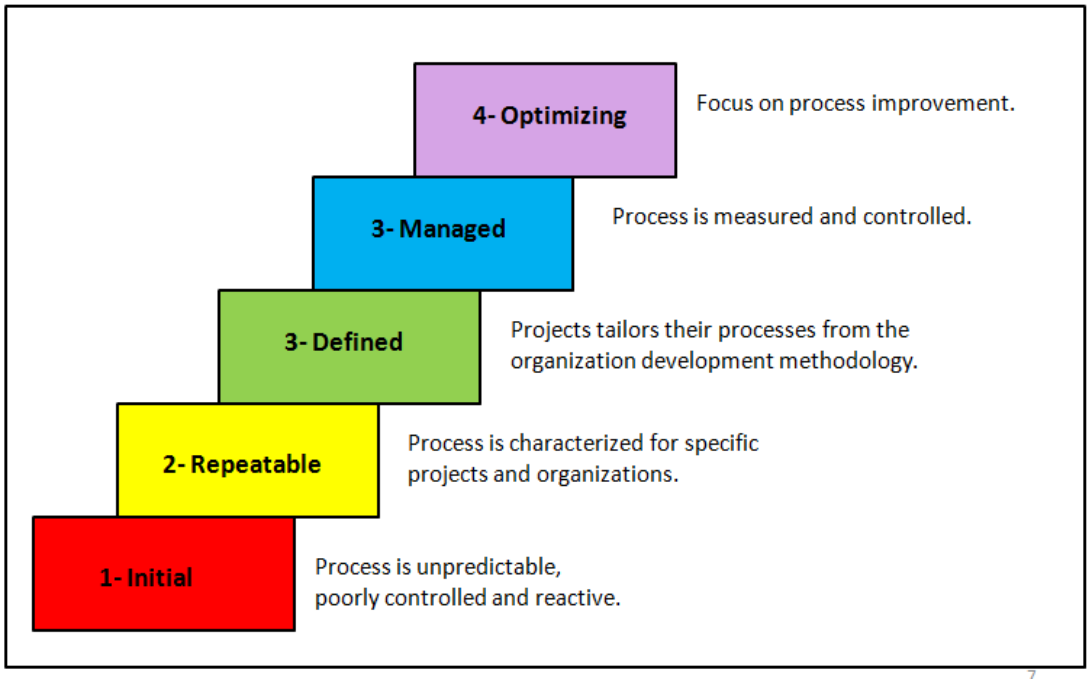

Figure 2: CMM maturity levels.

The success of the CMM directed to the development of several CMMs for a variety of software engineering subjects including the CMMI. The CMMI defines a six levels of maturity namely: incomplete, performed, managed, defined, quantitatively managed and optimizing as shown in Figure 3. Among 22 process areas that were defined in the CMMI, two process areas are related to the requirements engineering process improvements: Requirements Management (REQM) and Development (RD), in order to improve and assess these process areas a set of related goals and practices in each level should be implemented.

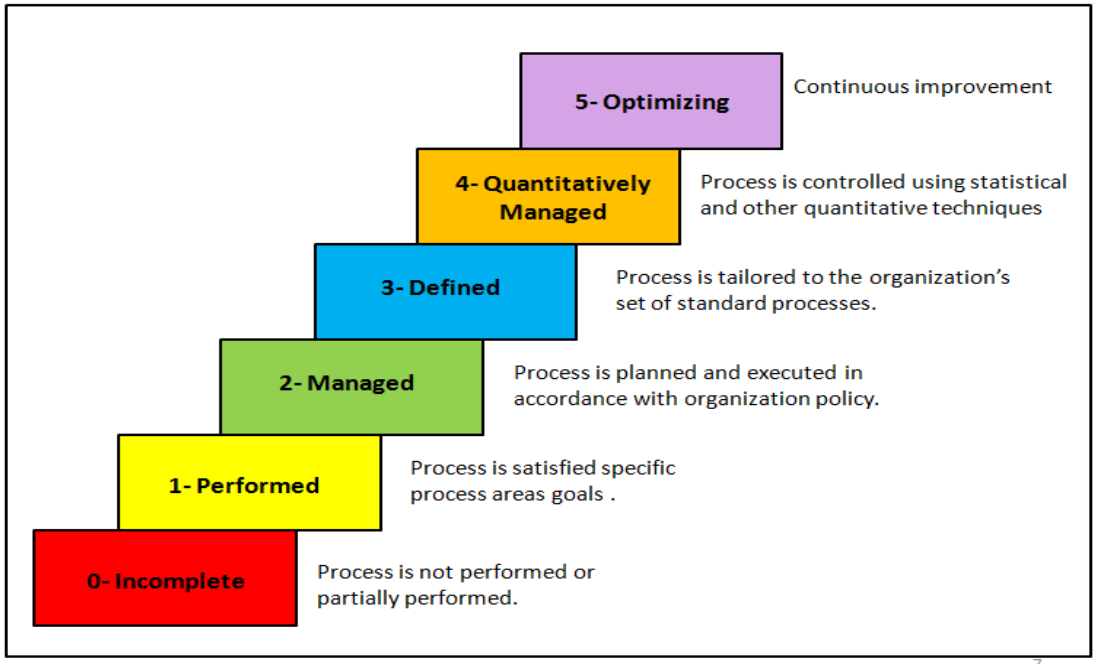

Figure 3: CMMI maturity levels.

Despite the great success of the CMM and CMMI, they did not pay detailed attention to the requirement engineering [27]. Consequently, a number of RE process improvement models were developed such as the Requirements Engineering Good Practice Guide (REGPG) [1]. The REGPG was the first public-domain process improvement and assessment model. It uses an improvement framework with three process maturity levels namely; initial, repeatable, and defined. The REGPG groups good practices for different requirements engineering activities in 66 guidelines, and classifies them as basic, intermediate, and advanced guidelines. For each guideline a score that indicates the rate of usage is assigned, and then the maturity level is determined based on the summation of the numerical scores and the practices' classifications. However, at the time of REGPG's design, the implementation of RE practices across the industry was inconsistent [28-29].

Inspired by the REGPG (Gorscheck *et.al*, 2002) [30] proposed a requirements engineering process maturity model called the REPM. The model uses six levels maturity model and defines 3 main process areas: elicitation, analysis and negotiation, and management. For each process area, a number of actions were assigned and used for the project evaluation process. For example, the REGPG assessment process uses checklist of 60 actions for an organization to reach a higher level of maturity. All actions of these levels must be completed and satisfied. A pilot study for evaluating the

REPM [31] showed that validation of the model's applicability was not an explicit goal of the evaluation.

Based on the REPM, a new maturity model called Uni-REPM [32] was proposed, it is a light-weight model that helps organizations in assessing and identifying the strengths and weaknesses of their process through a recommended improvement path toward a better requirements engineering process. Moreover, a model called a Market-Driven Requirements Engineering Process Model (MDREPM) [33] was developed. The MDREPM is not only a collection of good practices in market-driven requirements engineering area it is also an assessment tool that helps organizations in getting a snapshot of the current state of their MDRE practices.

3.3 Overview of the CMM/CMMI-Based Maturity Models

Several models related to the RE process improvements area developed in the past few years based on the CMM/CMMI models. Beecham *et.al* [2] proposed a specialized process improvement model called the R-CMM. It is a direct adaptation of the SW-CMM framework for assessing the capability of the sub processes that the RE process consists from. The model defines a set of 68 processes distributed over five maturity levels and classified according to the RE process activities, called phases. These phases are management, elicitation, analysis and negotiation, documentation and validation. Similar to the REGPG process assessment the model assess the degree to which a process is satisfied by an organization through allocating a score to each process against three assessment criteria:

- Approach: A measure of the organizational commitment and capability.
- Deployment: A measure of the degree to which a process is implemented across the organization.
- Results: A measure of the success of a process implementation

Each assessed process is then given a rating of: − outstanding (10) − qualified (8) − marginally qualified (6) − fair (4) − weak (2) − poor (score 0). The scores for all five phases are then summed and represented the overall score.

The whole R-CMM was then re-defined according to the characteristics of the capability maturity model for integration and development (CMMI-DEV) v1.2 by Solemon *et.al* [34]. A new Story Cards Based Requirements Engineering Maturity model based on the CMM levels is proposed in [35]. The model focused on how to

improve the agile RE practices and solve the problems related to the story cards like requirements conflicts, missing and ambiguous requirements.

The most recent work was a specialized RE process improvement and assessment model called the REPAIM [3]. The model was built based on the capability levels of the (CMMI-DEV) and composed of two main components: the PMM-RE which stands for process maturity model for RE and contains definitions for the RE process maturity levels and the FLA-RE which stands for Flexible Lightweight Assessment method for RE. It describes assessment requirements, stages and steps. The model defines four RE maturity levels: incomplete, performed, managed and defined. Each level consists of RE goal and related RE practices. For an organization to reach a particular RE maturity level all practices associated to this level must be satisfied.

In addition to the requirement improvement domain, the great success of using the CMM/CMMI in software process improvement has triggered the world wide to use them as a base for developing and adapting wide range of software process maturity models in different domains. For example, the Information Process Maturity Model (IPMM) [36] defines five maturity levels: ad-hoc, rudimentary, organized and repeatable, managed and sustainable, and optimizing. The IPMM process assessment based on eight key characteristics: organizational structure, quality assurance, planning, estimating and scheduling, hiring and training, publications design, cost control, and quality management. These characteristics are used to evaluate information-development organizations through describing the practices that make them successful.

Similar to the structure of the CMMI and based on practitioners' experience and international standards the software maintenance maturity model (SM^{mm}) was proposed in [37]. It defines six maturity levels namely: incomplete, performed, managed, established, predictable, and optimizing. The used the roadmap concept – a set of related practices that represents a significant capability for a software maintenance organization-, the main goal of this model is to assess and improve the quality of software maintenance function.

A maturity model for the implementation of software process improvement is proposed in [38]. The model was constructed based on CMMI and several resources from the SPI literature. It is composed of three components: SPI implementation plan, SPI implementation roadmap, and SPI implementation model. The aim of this model is

to help organizations in designing effective implementation strategies for software process improvement.

A framework for the validation and verification capability assessment in testing domain is proposed in [39]. The framework focused on the safety-criticality and it goes through five criticality-based V&V capability levels: none, low, mediate, high and rigorous. These levels and their associated V&V tasks were defined based on the CMMI process areas. The main goal of this framework is to provide the essential V&V practices that support the assessment of the "safety-criticality".

Based on the terms, concepts and maturity levels of the CMM/CMMI, IS12207, and IS15288 the Business Process Maturity Model (BP^{MM}) was proposed in [40]. The main goal of this model is to help companies in analyzing the strengths and the weaknesses of their business processes through comparing the maturity of their current practices against an industry standard. Hence, they can improve their process and achieve the organization's business objectives. Another model called Capability Maturity Model for Business Intelligence proposed in [41] to business process domain. The model consists of five levels namely: initiate, harmonize, integrate, optimize and perpetuate. It is based on the business intelligence maturity concepts. The model goal is to assess and evaluate capabilities of organizations in the field of BI.

E-Learning Maturity Model was developed in [42] based on the CMM and the SPICE models and consists from five process areas representing the whole e-learning life cycle from planning to delivery and evaluation. The model goal is to help organizations in measuring and improving process maturity from multiple aspects.

A configuration management capability model for medical device industry was developed [43]. The model aims to help medical device companies in improve the effectiveness and efficiency of configuration management capability. They track their evolution against five capability levels which were adopted based on the CMMI generic goals.

TMMi foundation developed test maturity model integration in [44] as complementary model to the CMMI Version 1.2. The model aimed to improve the test process through five levels of maturity namely; initial, managed, defined, measured and optimization. Each level has a set of process areas and its related goals. These goals must be implemented and satisfied to achieve the desired improvement at each level.

Applying TMMi has a positive impact on product quality, test engineering productivity, and cycle-time effort.

Based on the CMM 5- level structure a model called the Risk Management Capability Maturity Model for Complex Product Systems (CoPS) projects was developed in [45]. The model consists of five maturity levels: ad hoc, initial, defined, managed, and optimizing and 10 key capability areas grouped into three categories: organization context, PM/RM process, and technology content. The main goal of this model is to improve the predictability and controllability of CoPS on different types of risks such as process, organizational, and technical-related risks.

Moreover, Capability Maturity Model (CMM) for scientific data management practices was proposed in [46]. The model goes through five maturity levels namely: initial, managed, defined, quantitatively managed and optimizing. The model provides a well-defined set of goals, objectives and practices that covers all aspects of data management. The main goal is to help organizations in assessing their projects and organizational data management practices and plans.

3.4 Summary

This chapter summarized the most relevant related work in the area of requirement engineering improvements. There are several standards, guidelines, recommendations and specialized RE improvement models proposed in the literature in order to help organizations in enhancing the implementation of their RE process and its related practices. However, these works suffered from several problems that restrict their adoption, for example, they were developed based on an old or unsupported version of the CMM, they support limited type of RE process models, they did not pay a detailed attention to the RE process, they did not defined the RE process the way it should be defined regarding to the industry, they did not represented in a process maturity model, or they had not been validated.

العنوان:	Capability Maturity Model of Software Requirements Process and Integration (SRPcmmi)
المؤلف الرئيسي:	Najjar, Sireen Kamal
مؤلفين آخرين:	Al Sarayreh, Khalid Torki(Advisor)
التاريخ الميلادي:	2016
موقع:	الزرقاء
الصفحات:	1 - 59
رقم MD:	819762
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	البرامج، هندسة البرمجيات، هندسة المتطلبات
رابط:	https://search.mandumah.com/Record/819762

References

[1] Sommerville. I, Sawyer. P. Requirements engineering: a good practice guide 1st edition. 1997, John Willey & Sons, New York, NY, USA.

[2] Beecham.S, Hall.T and Rainer.A, "Defining a Requirements Process Improvement Model," Software Quality Journal, 13, 247-279 (2005).

[3] Solemon.B, Sahibuddin.S, and Abdul Ghani .A. A, "A New Maturity Model for Requirements Engineering Process: An Overview ", Journal of Software Engineering and Applications, 5.5, 340 (2012).

[4] Team, C. P. "CMMI® for Development, Version 1.3, Improving processes for developing better products and services. no. CMU/SEI-2010-TR-033". Software Engineering Institute, (2010).

[5] Bourque, P., and Fairley, R. E. Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0 3rd edition. 2014, IEEE Computer Society Press, Los Alamitos, CA, USA.

[6] IEEE Std 610.121990, IEEE Standard Glossary of Software Engineering Terminology, The Institute of Electrical and Electronics Engineers IEEE, USA, 1990.

[7] Sommerville, I., and Kotonya, G. Requirements engineering: processes and techniques 1st edition. 1998, John Wiley & Sons, Inc.

[8] Pressman, R.S. Software engineering: a practitioner's approach 7th edition. 2005, McGraw-Hill.

[9] Linscomb. D. "Requirements Engineering Maturity in the CMMI", the Journal of Defense Software Engineering, 16, 12 (2003).

[10] Hofmann H. F and Lehner. F, "Requirements Engineering as a Success Factor in Software Projects," IEEE Software, 18(4), 58-66 (2001).

[11] Martin.S , Aurum.A , Jeffery.R and Paech.B, Requirements Engineering Process Models in Pactice, Proceedings of *the Seventh Australian Workshop in Requirements Engineering (AWRE'2002)*, 2002, 141-155, Melbourne.

[12] Wohlwend. H and Rosenbaum. S, Software Improvements in an International Company, Proceedings of *the 15th International Conference on Software Engineering (ICSE 93)*, 1993, 212- 220, Baltimore, Maryland.

[13] Lauesen.S and Vinter.O , "Preventing Requirement Defects: An Experiment in Process Improvement," Requirements Engineering, 6 ,37-50 (2001)

[14] Herbsleb J. D. and Goldenson D. R., A Systematic Survey of CMM Experience and Results, Proceedings of *the 18th International Conference on Software Engineering—ICSE*, 1996, 323-330, Berlin.

[15] Brodman, J. G. and Johnson, D. L. "Return on Investment (ROI) from Software Process Improvement as Measured by US Industry", Software Process: Improvement and Practice, 1, 35-47 (1995).

[16] Jan, S. and Majeed, M. N. "Role of Requirement Management as a CMMI Process Area in Requirement Engineering". International Journal of Computer Applications, 51(19), 0975-8887 (2012).

[17] IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements. New York: IEEE

[18] IEEE Std 1233-1998. IEEE Guide for Developing System Requirements Specifications. New York: IEEE. IEEE Std 1362-1998.

[19] IEEE Guide for Information Technology - System Definition - Concept of Operations (ConOps) Document. New York: IEEE.

[20] Sawyer, P. "Maturing requirements engineering process maturity models" Requirements engineering form sociotechnical systems, 84-99 (2005).

[21] Young, R. Effective Requirements Practices 1st edition. 2001, Addison-Wesley Longman, Boston, MA, USA.

[22] Wiegers, K. Software Requirements 1st edition. 1999, Microsoft Press, Redmond.

[23] Napier, N. P., Mathiassen, L., and Johnson, R. D. "Combining perceptions and prescriptions in requirements engineering process assessment: an industrial case study" Software Engineering, IEEE Transactions on, 35(5), 593-606 (2009).

[24] Wang, Y., & King, G. Software engineering processes: principles and applications 1st edition. 2000, CRC press, Boca Raton, FL, USA.

[25] Humphrey, W. Managing the Software Process. 1st Edition, 1989, Addison-Wesley Longman, Boston.

[26] Paulk, M., Curtis, W., Chrissis, M. and Weber, C. "Capability Maturity Model for Software, Version 1.1. (CMU/SEI-93-TR-24, ADA 263 403)" Software Engineering Institute, (1993).

[27] Ahern, D. M., Clouse, A., & Turner, R. , CMMI Distilled – A Practical Introduction to Integrated Process Improvement 2ed edition, 2004, Addison Wesley, Boston.

[28] El Eman, K. and Madhavji, N. A Field Study of Requirements Engineering Practices in Information Systems Development. Proceeding of *2nd IEEE International Symposium on Requirements Engineering (RE95)*, 1995, 68-80, York, UK.

[29] Forsgren, P. and Rahkonen, T. Specification of Customer and User Requirements in Industrial Control System Procurement Projects. Proceeding of *2nd IEEE International Symposium on Requirements Engineering (RE95)*, 1995, 81-88, York, UK.

[30] Gorschek, T. and Tejle, k. A Method for Assessing Requirements Engineering Process Maturity in Software Projects, Master Thesis. 2002, Blekinge Institute of Technology.

[31] Gorscheck, T., Svahnberg, M. & Kaarina, T. Introduction and Application of a Lightweight Requirements Engineering Process Evaluation Method. Proceeding of *Requirements Engineering Foundations for Software Quality '03 (REFSQ'03)*, 2003, 83-92, Klagenfurt/Velden, Austria

[32] Svahnberg, M., Gorschek, T., Nguyen, T. T. L., & Nguyen, M. "Uni-REPM: a framework for requirements engineering process assessment" Requirements Engineering, 20(1), 91-118 (2015).

[33] Gomes, A., & Pettersson, A. "Market-driven requirements engineering process model–MDREPM" Karlskrona: Blekinge Institute of Technology, (2007).

[34] Solemon, B., Shahibuddin, S., & Ghani, A. A. A. (2009, December). Re-defining the Requirements Engineering Process Improvement Model. Proceeding of *Software Engineering Conference, 2009. APSEC'09. Asia-Pacific*, 2009, 87-92, Penang.

[35] Patel, C., & Ramachandran, M. " Story card maturity model (smm): A process improvement framework for agile requirements engineering practices" Journal of Software, 4(5), 422-435 (2009).

[36] Southard, S. "The Information Process Maturity Model: A 2004 Update" Technical Communication, 52(1), 105-106 (2005).

[37] April, A., Huffman Hayes, J., Abran, A., and Dumke, R. "Software Maintenance Maturity Model (SMmm): the software maintenance process model" Journal of Software Maintenance and Evolution: Research and Practice, 17(3), 197-223 (2005).

[38] Niazi, M., Wilson, D., and Zowghi, D. "A maturity model for the implementation of software process improvement: an empirical study" Journal of Systems and Software, 74(2), 155-172 (2005).

[39] Yoon, K. A., Park, S. H., Bae, D. H., Chang, H. S., and Jung, J. C. A Framework for the Vamp; V Capability Assessment Focused on the Safety-Criticality. Proceeding of *13th IEEE International Workshop on Software Technology and Engineering Practice* 2005, 17-24, Budapest.

[40] Lee, J., Lee, D., & Kang, S. "An overview of the business process maturity model (BPMM)". In: Advances in web and network technologies, and information management, 2007, Springer Berlin Heidelberg, 384-395.

[41] Raber, D., Winter, R., & Wortmann, F. Using quantitative analyses to construct a capability maturity model for business intelligence. Proceeding of *45th Hawaii International Conference on System Science (HICSS) 2012*, 4219-4228, Maui, HI.

[42] Petch, J., Calverley, G., Dexter, H., & Cappelli, T. Piloting a Process Maturity Model as an e-Learning Benchmarking Method. In *ECEL2006-5th European Conference on elearning: ECEL2006* 2007, 273, Academic Conferences Limited

[43] Mc Caffery, F., & Coleman, G. "Developing a configuration management capability model for the medical device industry" International Journal of Information Systems and Change Management, 2(2), 139-154 (2007).

[44] Van Veenendaal, E., & Cannegieter, J. J. "Test maturity model integration (TMMi)". TMMi Foundation, (2008).

[45] Yeo, K. T., & Ren, Y. "Risk management capability maturity model for complex product systems (CoPS) projects" Systems Engineering, 12(4), 275-294 (2009).

[46] Crowston, K., & Qin, J. A. Capability Maturity Model for Scientific Data Management: Evidence from the literature. Proceedings of *the American Society for Information Science and Technology* 2011, 48(1), 1-9.

[47] CMMI official website Available: http://www.sei.cmu.edu/cmmi/

[48] Team, C. P. "CMMI for Development, version 1.2.", Software Engineering Institute, (2006).

[49] Team, C. P. "CMMI for Acquisition, Version 1.2, *CMMI-ACQ v1. 2*. CMU/SEI-2007-TR-017, Technical Report", Software Engineering Institute. (2007).

[50] Team, C. P. "CMMI for Services (CMMI-SVC) v1. 2. Software Engineering Institute Technical Report CMU/SEI-2009-TR-001". Software Engineering Institute, (2009).

[51] Oh, D. S. and Rhew, S. Y. "An Empirical Study on Requirement Management Process for Implementation Project of Information System". Lecture Notes on Software Engineering, 2(1), 31 (2014).

[52] de Vasconcelos, A. M. L., de la Vara, J. L., Sanchez, J., & Pastor, Ó. (2012, September). Towards cmmi-compliant business process-driven requirements engineering. Proceedings of *Eighth International Conference On the Quality of Information and Communications Technology (QUATIC)* 2012, 193-198, Lisbon.

[53] M. C. Paulk, C. Weber, B. Curtis and M. B. Chrisses. The Capability Maturity Model: Guidelines for Improving the Software Process 1st edition. 1994, Reading, MA: Addison-wesley.

[54] SEI, "Process Maturity Profile, September 2011," 7 October 2011. http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2011SeptCMMI-2.pdf

[55] Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P., & Murphy, R. "An exploratory study of why organizations do not adopt CMMI" Journal of systems and software, 80(6), 883-895, (2007).

[56] Khurshid. N, Bannerman, P. L. And Staples, M. Overcoming the First Hurdle: Why Organizations Do not Adopt CMMI. Proceedings of *the ICSP 2009-International Conference on Software Process (Co-Located with ICSE 2009)* 2009, 38-49, Vancouver.

[57] Alvarez, R. and J. Urla. "Tell me a Good Story: Using Narrative Analysis to Exmine Informaton Requirements Interviews during an ERP Implementation" Data Base for Advances in Information Systems, 33(1), 38-52, (2002).

[58] Alexander, I. and N. Maiden. Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle 1st edition, 2005, John Wiley & Sons.

[59] Beyer, H. and K. Holtzblatt. "Apprenticing with the Customer." Communications of the ACM, 38(5), 45-52, (1995).

[60] Seybold, M. G. C., & Meier, S. Simulation-driven creation, validation and evolution of behavioral requirements models. In *Dagstuhl-Workshop MBEES: Modellbasierte Entwicklung eingebetteter Systeme III* 2007, 103, Germany

[61] Rettig, M. "Prototyping for Tiny Fingers" Communications of the ACM, 37(4), 21-27, (1994).

[62] Denger, C., D. Berry and E. Kamsties. Higher Quality Requirements Specifications through Natural Language Patterns. Proceeding of *IEEE International Conference on Software: Science, Technology and Engineering (SwSTE'03)* 2003, 80-90, Herzelia.

[63] Bajec, M. and M. Krisper. "A Methodology and Tool Support for Managing Business Rules in Organisations" Information Systems, 30(6), 423-443, (2005).

[64] Pohl, K. and C. Rupp. Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB Compliant 1st edition, 2001, Rocky Nook Computing.

[65] Achimugu, P., A. Selamat, R. Ibrahim and M. N. r. Mahrin. "A Systematic Literature Review of Software Requirements Prioritization Research" Information and Software Technology, 56(6), 568-585, (2014).

[66] Boehm, B., P. Grünbacher and R. Briggs. "Developing Groupware for Requirements Negotiation: Lessons Learned." IEEE Software, 18(3), 46-55, (2001).

[67] Easterbrook, S., R. Lutz, R. Covington, J. Kelly, Y. Ampo and D. Hamilton. "Experiences Using Lightweight Formal Methods for Requirements Modeling." IEEE Transactions on Software Engineering, 24(1), 1-11, (1998).

[68] Porter, A., L. Votta and V. Basili. "Comparing Detection Methods for Software Requirements Inspections - Replicated Experiment." IEEE Transactions on Software Engineering, 21(6), 563-575, (1995).

[69] Johnson, P. M. "Reeering inspection" Communication ACM, 41(2), 49-52, (1998).

[70] Standard glossary of terms used in Software Testing, Version 2.1. ISTQB. 2010.

[71] Kobayashi, A. and M. Maekawa. Need-based Requirements Change Management. Proceeding of *the 8th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2001) 2001*, 171-178, Washington, DC, USA.

[72] Kniberg, H. and M. Skarin (2010). Kanban and Scrum - Making the Most of Both 1st edition,2010, lulu.com

[73] Guide, A. "Project Management Body of Knowledge (PMBOK® GUIDE)", Project Management Institute, Pennsylvania USA, (2001).

العنوان:	Capability Maturity Model of Software Requirements Process and Integration (SRPcmmi)
المؤلف الرئيسي:	Najjar, Sireen Kamal
مؤلفين آخرين:	Al Sarayreh, Khalid Torki(Advisor)
التاريخ الميلادي:	2016
موقع:	الزرقاء
الصفحات:	1 - 59
رقم MD:	819762
نوع المحتوى:	رسائل جامعية
اللغة:	English
الدرجة العلمية:	رسالة ماجستير
الجامعة:	الجامعة الهاشمية
الكلية:	كلية الدراسات العليا
الدولة:	الاردن
قواعد المعلومات:	Dissertations
مواضيع:	البرامج، هندسة البرمجيات، هندسة المتطلبات
رابط:	https://search.mandumah.com/Record/819762

Capability Maturity Model of Software Requirements Process and Integration (SRP^{CMMI})

By

Sireen Kamal Najjar

Supervisor

Khalid Al-Sarayreh

Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Software Engineering

Faculty of Graduate Studies at the Hashemite University

Zarqa - Jordan

April/4/2016