Requirements Engineering Maturity in the CMMI

Dennis Linscomb
Computer Sciences Corp.

Much has been written on requirements engineering (RE) but very little about RE maturity. Is there such a thing? If so, why and how do you measure it? This article discusses these topics and analyzes how the Capability Maturity Model® Integration addresses RE maturity.

The discipline relating to the systematic handling of requirements has typically been called requirements engineering (RE) [1]. One definition of RE that is regularly cited in RE literature is,

> … the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families. [2]

Many articles and books have been written on the components of RE and their interrelationship [3]. In the Institute of Electrical and Electronics Engineers' (IEEE) Software Engineering Body of Knowledge (SWEBOK) [4], the Software Requirements Knowledge Area consists of the following components: RE Process, Requirements Elicitation, Analysis, Specification, Validation, and Management. These components are common to the RE literature. While a lot has been written about RE scope, components, techniques, templates, and tools, there has not been a lot written about RE maturity[1]. Is there such a thing as progressing in RE from a basic to an advanced level? If so, how do you define it, and why should you measure it?

In my information technology (IT) experience in the software applications area at several companies, I have clearly seen levels of RE maturity. I think you will agree with me when you consider the following scenarios in Table 1.

Of course, many more scenarios could be listed, but I think I have made my point. The harder questions to answer are these: (1) Why do you need to define levels of RE maturity? (2) Assuming there is a good reason to define levels, how do you define them, i.e., what criteria do you use?

The question "why define RE maturity" is usually part of a larger question: Why define and measure the process maturity (usually software process maturity) of an organization? The main reason many organizations do it (at least from the executive management viewpoint) is economic, namely they want to get more business and retain existing business. One would hope that they also (primarily?) do it because it is the right thing to do, but that is not always the main motivator. The Software Engineering Institute's (SEI) Capability Maturity Model® (CMM®) Integration (CMMI®)[2] usually comes to mind when discussing a process maturity rating. Using this model, organizations are given a rating of Maturity Level (ML) 2-5[3] via formal assessments.

If you are not in an organization striving for a certain ML, or if the strategy of your organization is something other than operational excellence[4], then much of the rest of this article is not meant for you. If your organization fits this profile, I recommend pursuing RE in a way that makes sense for your organization's goals and strategy. However, assuming you can find a reason for rating the process maturity of your organization, then it is appropriate to analyze how best to fit RE maturity into your process model.

For my analysis of RE maturity, I chose the CMMI not only because it is widely used but also because it is one of the few[5] process models that attempts to define levels of maturity for IT-related processes. The CMMI defines two process areas (PAs) relating to RE: Requirements Management (REQM) and Requirements Development (RD). Although RE affects many more CMMI PAs due to its importance in the software development life cycle, these two PAs are the ones in which RE is specifically addressed. The REQM and RD PAs are measured for their maturity based on the type of CMMI representation of the model you are using.

In the CMMI Staged Representation, all PAs are defined at one of four MLs (2-5, with 5 being the most mature). This representation puts REQM at ML2 and RD at ML3. As you mature through the MLs, you must continue to perform at the previous MLs. Therefore, to implement RD means that you have institutionalized[6] REQM.

In the CMMI Continuous Representation, one of six capability levels (0-5, with 5 being the highest capability) is assigned to *each* PA. Theoretically (though not practically), an organization could be at a high capability level (e.g., 5) for REQM and a low capability level (e.g., 0) for RD.

Therefore, no matter which representation you use, the CMMI model describes a progression from less RE maturity to more RE maturity. At ML3 (in the Staged Representation) or capability level 3 (in the Continuous Representation), an organization is considered to be more mature[7] in RE than they would be at previous levels.

Although the CMMI is now being widely used and is at version 1.1, I think it still makes sense to ask the question, "Does the CMMI currently define RE maturity the way it should be defined based on industry standards and practice?" My answer is, "No," based on RE terminology and on the typical order of RE activity.

Table 1: *Requirements Engineering Maturity Levels*

Less Mature in RE	More Mature in RE
Requirements are taken verbally over the telephone from one stakeholder.	Requirements are documented after getting consensus from multiple stakeholders.
Only one requirements elicitation/gathering technique is used without regard to the nature of the stakeholders or the project.	Several requirements elicitation/gathering techniques are known and used based on the type of project and the mix of stakeholders.
The original requirements are documented in a repository but are never modified as individual requirements change over time.	A repository of up-to-date user-approved requirements is maintained throughout the life of the project.
There is no change control process defined for requirements or, if defined, it is never consistently used.	A requirements change control process is defined and consistently used.
There is no way of knowing whether or not every requirement was implemented.	A requirements traceability matrix is developed and maintained.

With respect to terminology, it should be noted that CMMI treats the standard RE components (management, elicitation, analysis, specification, and validation) differently from that usually found in RE literature. For example, REQM is defined as a separate PA, but requirements elicitation, analysis, specification, and validation are all lumped into one RD PA. I have not found any SEI documentation describing the rationale of their taxonomy, as does the SWEBOK [5]. Part of the answer may lie in the fact that the RD PA in the CMMI was split out of the Software Product Engineering PA in the CMM. This difference in terminology is more than academic. By placing REQM and RD not only in separate PAs but also in separate MLs, there is an artificial dichotomy created between the components of RE. As I shall discuss later, REQM cannot be done in a vacuum.

At this point, you may object that I am mixing apples and oranges. Requirements management, elicitation, analysis, specification, and validation are categories or a taxonomy of RE activities, one may argue, whereas the CMMI is concerned with describing process areas relating to RE. However, these categories may also be viewed as activities in the RE process. According to Linda Macaulay,

> In general terms, the RE process can be thought of as a series of activities consisting of articulating the initial concept, problem analysis, feasibility and choice of options, analysis and modelling [sic], and requirements documentation. [6]

Requirements life cycles have been defined as consisting of three to five phases with the above RE categories, or equivalent terms, as phase names[8]. Although the CMMI does not require you to choose any specific RE life cycle, it should use standard RE terminology in describing PAs, goals, and practices relating to RE.

With respect to the typical order of RE activity, I believe there is room for improvement in the CMMI. While the CMMI does not dictate any specific RE life cycle, it does have something to say about the order of implementation and institutionalization of RE by its placement of a certain RE activity under a specific ML. I contend that this order is not always logical. Consider the following examples:

1. Requirements elicitation is supposed to be institutionalized in the ML3 RD PA under Specific Goal (SG) 1. However, under the ML2 REQM PA, you are supposed to be managing these requirements. How can you manage them at ML2 if you do not have an institutionalized way of eliciting requirements until ML3? The ML2 REQM Specific Practice (SP) 1.1 "Obtain an Understanding of Requirements" does not contain enough detail about the scope, source, and specificity of requirements to form a solid basis for managing those requirements at ML2. Requirements-related problems are closely tied to project failure[9]. Why wait until ML3 to institutionalize practices to ensure that you have complete and accurate requirements?
2. Requirements analysis and validation are defined under the ML3 RD PA (under SG 3). However, you need to do a certain amount of analysis and validation of requirements at ML2 in order to get them in a mature enough state to manage them.
3. Bidirectional requirements traceability is required under the ML2 REQM PA. While a certain amount of requirements traceability is necessary at ML2, should an organization concentrate on this full-blown bidirectional traceability before institutionalizing requirements elicitation and analysis (at ML3)? I think not. It is interesting to note that Rational Software puts traceability at Level 4 in their Five Levels of Requirements Management Maturity [7].

The CMMI recognizes that there is RE activity present even in ML1 organizations[10]. Also, the CMMI acknowledges the interrelationship of RE activities in the Introductory Notes to the REQM PA:

> … if the Requirements Development process area is implemented, its processes will generate product and product-component requirements that will also be managed by the requirements management processes. When the Requirements Management, Requirements Development, and Technical Solution process areas are all implemented, their associated processes may be closely tied and be performed concurrently. [8]

Therefore, the issue is not that the CMMI is opposed in principle to a normal progression and maturity of RE activity. The issue is whether the CMMI defines it the best way, i.e., using terms and maturity criteria that the industry can agree upon, and puts RE at the appropriate maturity levels.

The following is my proposal for the SEI CMMI Project Team:

1. Review the entire RE discipline (and not just the requirements-related goals and practices currently in the CMMI) with the goal of determining how RE should be presented in the CMMI. The review should include holding CMMI workshops to get consensus from a broad spectrum of RE practitioners about what they consider to be basic versus advanced requirements practices.
2. Work closely with the IEEE to ensure that their standards and work products, e.g., SWEBOK and the CMMI stay in sync with respect to terminology and processes.
3. Revise the CMMI model to reflect consensus from the above steps.

I think consensus from this effort will support the following concepts:

1. RE maturity should be represented at more than one ML. It is just not practical to assume that an organization can and should implement everything related to RE at one level.
2. A RE-related PA should, at minimum, exist at ML2 and ML3. Perhaps a case can be made for some advanced RE activity at ML4 and ML5. However, until that case is made, I believe the CMM and CMMI are correct in placing RE activity at ML2 and ML3.
3. The dichotomy between requirements management and other RE activities should be minimized.

Based on my IT experience, my recommendation (though I am willing to change it based on consensus from the above proposal) is that the CMMI Staged Representation should be changed to something like a Basic RE PA at ML2 and an Advanced RE PA at ML3. The concept of basic and advanced is not foreign to the CMMI. For example, there are basic and advanced project and process management PAs [9].

The following are my recommendations for some of the goals and practices at Basic RE PA (for ML2):

1. Elicit/gather requirements. You do not have to have a trained staff of facilitators and many different ways of eliciting or gathering requirements at ML2. You just need at least one repeatable method of obtaining project requirements. Why wait until ML3 to institutionalize one method?
2. Analyze requirements. To ensure they meet the characteristics of good requirements, e.g., complete, clear, consistent, verifiable, traceable, feasi-

ble, and design independent. These characteristics are currently defined as examples in the ML2 REQM PA under SP 1.1. However, why use the ambiguous title "Obtain an Understanding of Requirements" when many ML 1 and 2 organizations know what you mean by requirements elicitation and analysis?

3. Document requirements. This is already in the ML2 REQM PA as a typical work product (an agreed-to set of requirements) under SP 1.1.
4. Get approval of requirements from appropriate stakeholders. This is already in the ML2 REQM PA as SP 1.2 (Obtain Commitment to Requirements).
5. Manage requirements changes. This is already in the ML2 REQM PA as SP 1.3.
6. Develop and maintain requirements traceability to the extent that you can demonstrate that all requirements have been implemented. See comment below on traceability at ML3.

The following are my recommendations for some of the goals and practices at Advanced RE PA (for ML3):

1. Develop different techniques of eliciting requirements, define criteria about when to use each based on project profiles, and institutionalize these techniques with formal training and mentoring.
2. Provide a staff (more than one – even if part-time) of trained requirements facilitators.
3. Develop and maintain a full-blown, bidirectional requirements traceability matrix showing that each requirement is satisfied in design, development, test, and implemented work products. I have never seen a ML2 organization do a good job at this type of full-blown traceability matrix. Yet it is required in SP 1.4 of the ML2 REQM PA.
4. Include all current ML3 RD goals and practices that involve showing interrelationship of requirements, requirements decomposition, assumed system requirements, and requirements change metrics. In other words, everything beyond the ML2 basics defined above.

Probably, some people may not want to tamper with REQM at ML2. They believe this PA simply follows the overall CMM process improvement road map to get management infrastructure in place at ML2 in order to support the engineering processes at higher levels[11]. They make the point that engineering processes are in effect at ML2, but they do not have to be documented and can be informal. While I agree with the CMM improvement strategy, it should not be interpreted in such a way as to exclude activities required to make work products mature enough to be managed at ML2.

In other words, you cannot manage in a vacuum. A certain level of formalization must be in place for some engineering practices in order for the management process areas to work properly. Consider the ML2 Project Planning PA. You need to perform a certain amount of technical (engineering?) activities for SP 1.4 (perhaps using some sophisticated tools) in order to get sound estimates of effort and cost so that you can put together the project plan in order to manage it. In like manner, the ML2 REQM PA assumes a certain amount of RE formalization and institutionalization in order to ensure that requirements are mature enough to be managed[12].

Also, it should be noted that ML3 has never been composed of pure engineering PAs. For example, management activities permeate the Integrated Software Management and Intergroup Coordination PAs in the CMM and several PAs in the CMMI, such as Integrated Project Management, Risk Management, Integrated Teaming, Integrated Supplier Management, and Decision Analysis and Resolution. That is the way it should be. Each ML should be composed of the correct mixture of technical and management activities so that management can be effective for that ML.

You may be asking, "If this RE maturity discrepancy is that obvious in the CMM/CMMI, why has it not been a problem for organizations that have attained ML2 or ML3?" My answer is twofold:

1. Some ML1 organizations fund their process improvement effort with the goal of achieving ML3. In other words, they are not first assessed at ML2 and then work toward ML3. Why? Because two separate efforts are more expensive than one. Also, they may be under management pressure to achieve ML3 by a certain date, and there is not enough time to do this in two independent efforts. Whatever the reason, by including both ML2 and ML3 in one process improvement effort, all of RE goals and practices are covered. Therefore, it never becomes an issue about how RE is split out between ML2 and ML3.
2. For those ML1 companies who work toward ML2 as their goal, they just know from past experiences and industry best practices that certain ML3 RE practices (e.g., elicitation and analysis) must be done as part of their life cycle. Therefore, they continue to do them because they make sense and are required to deliver quality work products.

In conclusion, I believe that RE maturity makes sense as a concept and reflects reality in IT organizations seeking operational excellence, whether or not they call it basic versus advanced RE. The attempt of the CMMI to define this RE maturity is admirable but deficient. However, this deficiency does not mean that we abandon the model. The CMMI is being widely used, and I have personally witnessed the success of CMM at several companies. I want the model to continue its success. However, for it to be durable for many years to come, I believe it needs an overhaul in the RE area.◆

References

1. Abran, Alain, and James W. Moore. Guide to the Software Engineering Body of Knowledge – SWEBOK. Eds. Pierre Bourque and Robert Dupuis. New York: Institute of Electrical and Electronics Engineers, May 2001: 9 <www.swebok.org>.
2. Zave, Pamela. "Classification of Research Efforts in Requirements Engineering." ACM Computing Surveys 29.4 (Dec. 1997): 315-321.
3. Davis, Alan M. "Requirements Bibliography." <http://web.uccs.edu/adavis/UCCS/reqbib.htm>.
4. Abran 15ff.
5. Abran 23f.
6. Macaulay, Linda. Requirements for Requirements Engineering Techniques. Proc. of the Second International Conference on Requirements Engineering. York, United Kingdom, 1995. New York: IEEE Computer Society Press, Apr. 1996: 158.
7. Heumann, Jim. "The Five Levels of Requirements Management Maturity." The Rational Edge Feb. 2003 <www.therationaledge.com/content/feb_03/f_managementMaturity_jh.jsp>.
8. CMMI Product Team. CMMI Ver. 1.1. Pittsburgh, PA: Software Engineering Institute, Mar. 2002: 82.
9. CMMI Product Team Chapter 5.

Notes

1. See [7] for Rational Software's Five Levels of Requirements Management Maturity. Some articles describe a Requirements Maturity Index (RMI), but this has to do with the readiness of

requirements for design and development and not the maturity of the RE *process*. For an article that discusses RMI, see Stuart Anderson and Massimo Felici, "Quantitative Aspects of Requirements Evolution," <www.dcs.ed.ac.uk/home/mas/doc/cameraready_compsac2002.pdf>.

2. The original version of this model, called the Capability Maturity Model (CMM), is still in use. However, since the CMMI will eventually replace the CMM, most of my references are to the CMMI [8].
3. There is a Level 1 but this is a starting point for all organizations and does not represent a level of assessed maturity. Also, Levels 2-5 are based on the Staged Representation of the CMMI.
4. Stan Rifkin has written several articles on applying the main thesis of the book "The Discipline of Market Leaders" by Michael Treacy and Fred Wiersema to using the CMM and other process improvement efforts <www.master-systems.com/Papers.ivnu>.
5. The only other models I know about that define maturity levels for IT-related processes are in the CMM family (e.g., the Capability Maturity Model for Software Acquisition and the FAA integrated Capability Maturity Model) and Electronic Industries Alliance 731. If you know of other models, please e-mail me.
6. The CMMI defines *institutionalization* as "… the ingrained way of doing business that an organization follows routinely as part of its corporate culture" (see [8], Glossary: 579).
7. Although some proponents of the CMMI Continuous Representation say that a capability level is not a ML, I contend that it is in a certain sense of the word *maturity*. The CMMI defines a capability level as applying to an organization's process-improvement achievement for a certain process area. Therefore, as you progress in capability levels for a certain process area, are you not becoming more mature in that process area?
8. For examples of three and five phases, see Jawed Siddiqi and M. Chandra Shekaran, "Requirements Engineering: The Emerging Wisdom." IEEE Software Mar. 1996: 15-19. For an example of four phases, see Ian Sommerville, Software Engineering. Harlow, England: Addison-Wesley, 1996: 67f.
9. Numerous studies show that requirements play a large role in the success or failure of projects. The following are only a few: Standish Group's "Chaos Report" for 1994, 1997, and 2000. Karl Wiegers, Software Requirements. Microsoft, 1999: 5, 24.
10. "Certainly, we would expect maturity level 1 organizations to perform requirements analysis, design, integration, and verification. However, these activities are not described until maturity level 3 …" (see [8], Chap. 2 Model Components: 16).
11. CMM for Software Ver. 1.1, Section 2.2.2, p.15f, "Understanding the Repeatable and Defined Levels" states: "Level 2 provides the foundation for Level 3 because the focus is on management acting to improve its processes before tackling technical and organizational issues at Level 3. … Level 3 builds on this project management foundation by defining, integrating, and documenting the entire software process."
12. For examples of what happens if you try to do requirements management without requirements engineering and vice versa, see Nancy R. Mead's article, "Requirements Management and Requirements Engineering: You Can't Have One Without the Other." Cutter IT Journal May 2000.

About the Author

Dennis Linscomb is an employee of Computer Sciences Corp. (CSC) through CSC's acquisition of DynCorp. At DynCorp, he served as the quality assurance manager for the corporate Information Technology department. He has been in information technology for 28 years and has worked in several areas of applications software, including programming, analysis, testing, quality assurance, production support, and management. He has been involved in software process improvement and the Capability Maturity Model®/Capability Maturity Model Integration for about 10 years. He has a master's degree in business administration from Pepperdine University.

Computer Sciences Corp.
11710 Plaza America Drive
Reston, VA 20190
E-mail: dennis_linscomb@msn.com