

Requirements Engineering Process Maturity Model Uni-REPM

Version: 0.9 CR

Owner/reference group: Dr. Tony Gorschek, Dr. Mikael Svahnberg
tony.gorschek@gmail.com

Project home: http://gorschek.com/doc/REPM_Project.html (updates, news)

Contributors:

Mai Nguyen

Nguyen Thi Thanh Loan

Andreas Pettersson

Andrigo Gomes

Kaarina Tejle

Blekinge Institute of Technology, Karlskrona, Sweden 2011-01-15.



Contents

PART I. Motivation.....	3
PART II. Uni-REPM overview	4
1. Introduction.....	4
2. Model Structure	4
2.1. Main Process Area (MPA).....	5
2.2. Sub-Process Area (SPA)	6
2.3. Action	6
3. Process maturity.....	8
4. Model usage	8
4.1. Who will directly use the model?	9
4.2. How to use the model?	9
4.3. How to read the result?.....	9
5. References.....	10
Part III. Model Description	13
1. Process Area View	13
2. Maturity Level View.....	47
Reference	50

PART I. Motivation

Requirements engineering (RE) has been acknowledged to be one of the most important processes since it is the critical determinant of the software quality and software development process effectiveness [1]. Currently, requirements engineering has been roughly divided into two main approaches: **Bespoke** and **Market-driven**. In software development, Bespoke approach, also known as traditional development approach, aims to produce software products for specific customers [2]. However, Market-driven development tends to provide software products for a massive market [2]. These two main characteristics bring many differences between them apart from the similarities in practices and technologies [3, 4]. For instance in MDRE, requirements engineering usually involves organizational and product strategies. The validation activities with the customers can hardly be performed directly due to the fact that they do not have specific customers, and the elicitation is mostly through market analysis or survey which is never made use of in Bespoke [2].

Despite those particular differences, both approaches are still facing many challenges in handling requirements in industry. As in the research of Juristo [5], Beecham [6] and Niazi [7], there still exist many problems. Personnel and tool support are listed as the most pressing problems. Besides, communication in the process, documentation, requirements estimation are also reported as high rank problems in the requirements engineering process in organizations. Specifically, in MDRE, the vast number of stakeholders makes it hard to elicit and manage the requirements since they usually provide a large and continuously growing number of requirements with many different levels of abstractions [3].

However, those problems tend to be reduced in higher maturity RE process [8]. Therefore, the organizations should improve their RE process in order to overcome those challenges. In order to do that, as the first step of the improvement process, the current state of the process needs to be evaluated [9,10]. There have been a number of models developed for assessment purpose, namely **CMM [11]**, **CMMI [12]**, **ISO 9000 [13]**. However, most of them did not pay an adequate attention on RE process. **Good Practice Guidelines (GPG) [14]** and **REPM [15]** models were introduced later to particularly focus on RE process. Nevertheless, these models are only limited to Bespoke development. This limits the model applicability in many cases such as in Market-driven or hybrid (in which both approaches could be used in development) organizations where RE practices usually start early at product-level. This fact motivated us to upgrade and broaden the original light-weight **REPM** model so that it could be applied intensively in all industrial organizations.

Based on this idea, we have performed research and construct the model named Universal Requirements Engineering Process Maturity model (Uni-REPM). The next sections of the document will provide an overview, structure and usage of the model (Part II) and the description of the model itself (Part III).

PART II. UNI-REPM overview

The purpose of this section is to give a brief view of the Uni-REPM model covering its structure, components and usage.

1. Introduction

Uni-REPM is a light-weight model presenting the maturity of RE process through sets of necessary activities. The assessment is basically an action to map those ideal activities to real work. The activities in the model are divided into 7 areas: Organizational support, Requirements Management Process, Elicitation, Requirements Analysis, Release Planning, Documentation and Requirements Specification, and Requirements Validation.

Uni-REPM provides industrial organizations a quick method to identify the strengths and weaknesses in their process. Besides the assessing purpose, this Uni-REPM model is also served as a guideline giving organizations a recommended improvement path toward a better requirements engineering process from basic to advanced level.

The model was constructed based on extensive literature reviews of REPM model [15], CMMI [12], ISO9000-TickIt [13] together with a systematic review on Market-driven requirements engineering researches.

2. Model Structure

The construction of Uni-REPM structure was based on the original model REPM. The model hierarchy has three levels, namely **Main process area** (MPA), **Sub-process area** (SPA) and **Action**. On the top level of the model, there are seven Main process areas corresponding to requirements engineering main activities. Each MPA is further broken down into several SPAs, which facilitates better understanding. On the bottom level, Action denotes a certain activity that should be done or a certain item that should be present.

A MPA may have action(s) attached directly to it. Closely related actions are grouped together and put under one SPA so as to improve the model structure. An Action must attach to one MPA or SPA and there is no MPA or SPA residing under an action. Figure 2 shows a snapshot of the model to illustrate its structure and components.

The model components are summarized and illustrated in Figure 1.

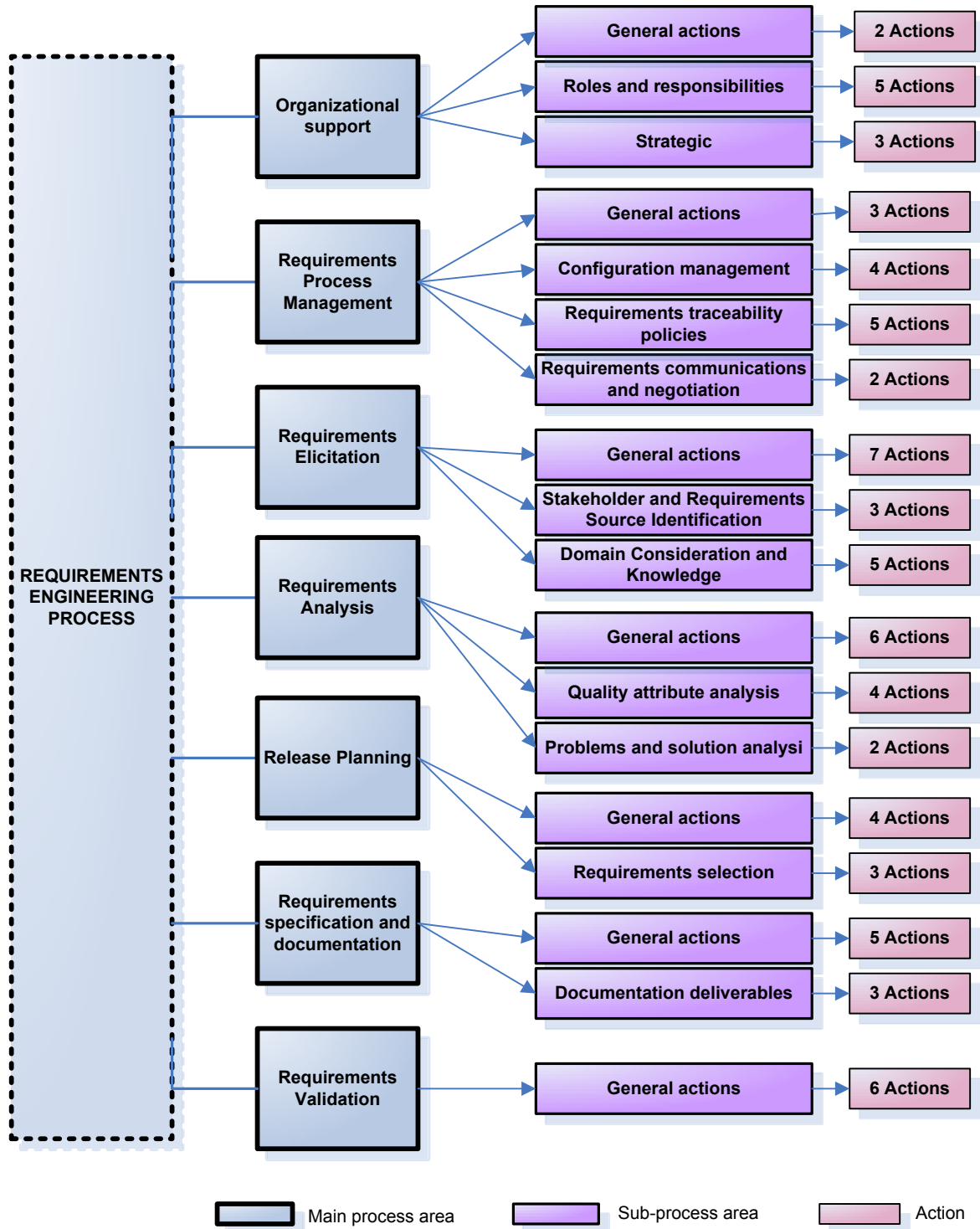


Figure 1. Uni-REPM Model structure inspired by REPM model [15]

2.1. Main Process Area (MPA)

On the top level of the model, a main process area represent a cluster of related practices in one main requirements engineering activity such as Elicitation.

There are 7 MPAs in the model, represented here according to the active order in the requirements engineering process:

- **Organizational Support:** This main process area evaluates the amount of support given to requirements engineering practices from the surrounding organizations. This MPA supports both the Requirements Development and Requirements Management processes.
- **Requirements Process Management:** The requirements process management covers all the activities to manage, control requirements change as well as to ensure the organization of the process and coherence among team members. This MPA represents the Requirements Management process.
- **Requirements Elicitation:** Elicitation is the process of discovering, understanding, anticipating and forecasting the needs and wants of the potential stakeholders in order to convey this information to the system developers. This MPA is part of the Requirements Development process.
- **Requirements Analysis:** Requirements after being gathered from different sources need to be analyzed to detect errors, to create detailed view of requirements as well as to estimate necessary information for later activities (eg. risk, priorities...). This MPA is part of the Requirements Development process.
- **Release Planning:** Release planning covers crucial steps aiming to determine the optimal set of requirements for a certain release to be implemented at a defined/estimated time and cost to achieve some goals. This MPA is more applicable to Market-driven development.
- **Documentation and Requirements Specification:** Documentation and Requirement specification deals with how a company organizes requirements and other knowledge gathered during requirements engineering process into consistent, accessible and reviewable documents. This MPA is part of the Requirements Development process.
- **Requirements validation:** This process involves checking the requirements against defined quality standards and the real needs of various stakeholders. It ensures that the documented requirements are complete, correct, consistent, and unambiguous. This MPA is part of the Requirements Development process.

Each MPA has a unique identifier which enables traceability throughout the model. For example, “Organizational Support” MPA is referred to as “OS”.

2.2. Sub-Process Area (SPA)

Sub-process area contains closely related actions, which help to achieve a bigger goal. The unique identifier assigned to each SPA is composed of the MPA identifier to which the SPA attaches and its abbreviation. For example, “OS.RR” represents a sub-process area “Roles and Responsibilities” which resides under MPA “Organizational Support”.

2.3. Action

The smallest unit in the model is called “action” showing a specific good practice. By performing the action, the organization can improve their process and gain certain benefits. For example, an action

“Create a product-wide glossary of terms” once implemented will enable readers from different backgrounds to understand the technical jargons used in a requirements document.

Actions also follow the same format to form their unique identifiers. They are identified by the MPA/SPA under which they reside, followed by an “a” which stands for “action” and their position in the group. For example, “OS.a1” points to the first action which attaches directly to MPA “Organizational Support”. Another example is “OS.RR.a1”, which means the first action under MPA “Organizational Support” and SPA “Roles and Responsibilities”.

Each action is assigned a certain level depending on its difficulty to implement and essentiality for the requirements engineering process. The level structure will be discussed in detail in section 3.

Example(s) and Supporting Action(s)

Within the description of each Action, there can be **Example(s)** and **Supporting Action(s)**. The idea of **Example(s)** is to give practitioners suggestions on proven techniques or supporting tools. This information aims to help the practitioners with implementing the action. In addition, the **Supporting Action(s)** provided links to other Actions which will benefit the practitioners when implementing them together.

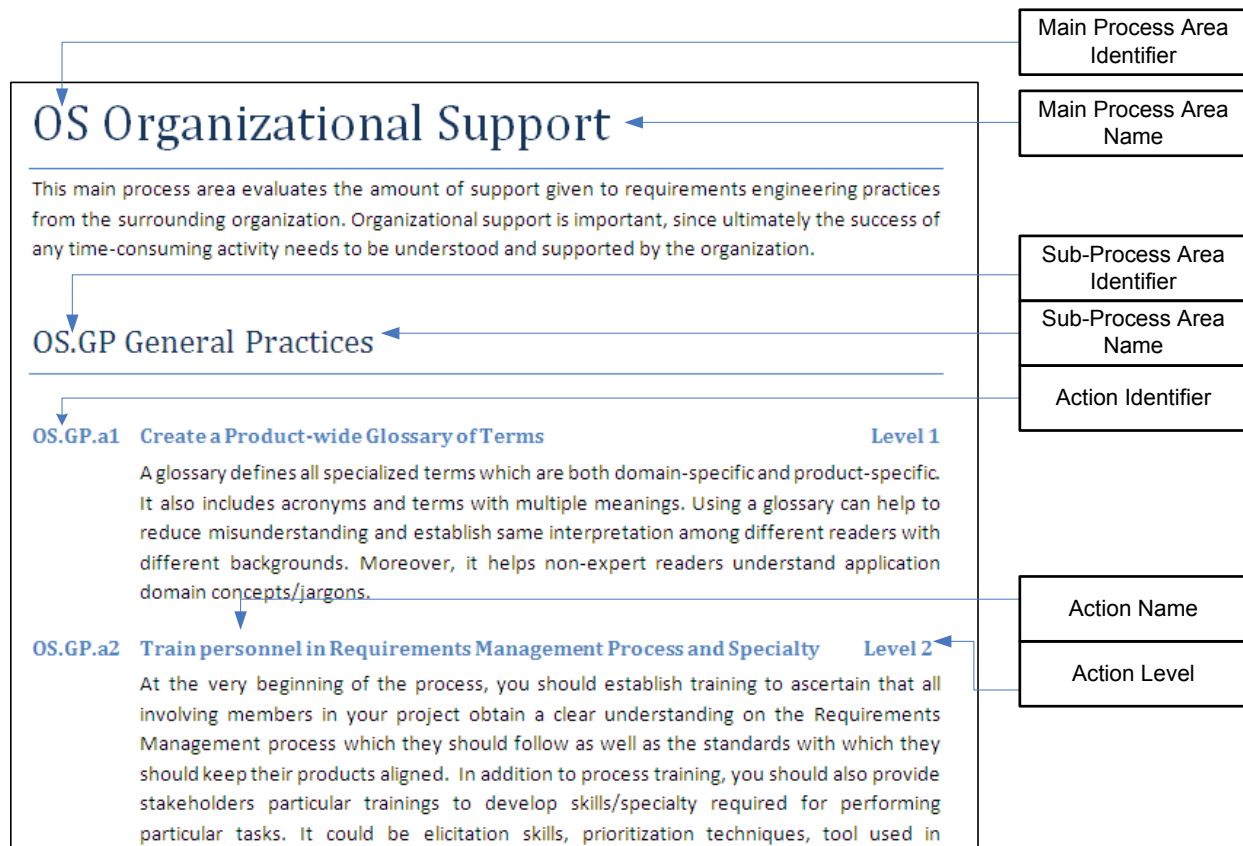


Figure 2. A snapshot of Uni-REPM model

3. Process maturity

Uni-REPM makes use of ordinal scale to assess the maturity of the process. The structure to assess process maturity is inspired from the REPM [15] and GPG [14] models; and the book “*Software requirements*” [16] presenting best practices from Microsoft. Concerning the fact that Requirements Engineering Process applied in industry is usually a small-sized and ad-hoc one (compare to the others) [5], we defined 3 levels of maturity, namely **Basic**, **Inter-mediate** and **Advanced**. The reason for changing from 5 levels as in REPM model [15] to 3 levels is to make the requirements engineering process significantly better after completing each level. Hence, the benefits gained from moving from one level to another level are more visible as well. The meaning of these levels is they will present how mature the evaluated process is. It is, however, not applicable to the whole organization maturity since the model scope only resides on Requirements Engineering Process. Nevertheless, it is possible to compare two processes in term of maturity using the evaluation results from the model.

The resulting level of a process is constructed from levels of actions performed within such process. In Uni-REPM, each action is placed under a certain level concerning its essentiality and required skills/cost to carry out. We also considered the dependencies among actions when assigning levels to them, e.g. if action A requires another pre-requisite action to be performed, it must be placed at least at the same or higher level than the pre-requisite action.

Level 1 – Basic

The aim of this level is to achieve a rudimentary repeatable requirements engineering process. The process in this level is defined and followed. Quality of requirements is managed because of relevant stakeholder involvement in elicitation, in-depth requirements analysis and pre-defined document standards.

However, the process does not maintain any kind of communications among stakeholders and within the organization in term of strategies.

Level 2 – Intermediate

In this level, the process is more rigorous because it involves various perspectives and is led by product strategies/goals. Roles and responsibilities for particular tasks are clearly defined and documented. Change requests are handled in the consistent manner throughout the project. Well-informed decisions about requirement selection can be made by analyzing and prioritizing the requirements systematically.

This process still stays in “**present-state**”; meaning that there is no activity performed to collect and analyze data/feedback for future improvement of the process.

Level 3 – Advanced

This level denotes the most mature process. The improvements in the process are shown in the advanced way of capturing requirements, ensuring their high quality, maintaining communications and common understanding among different stakeholders and pro-actively assessing the decision making process.

The process takes into account the “*future-state*” since it not only covers pre-defined and structured procedures but also adequately pay attention on future works (e.g. reusable materials, port-term evaluation, etc.).

4. Model usage

4.1. Who will directly use the model?

Uni-REPM aims to assess the RE process maturity; hence it can be used by people who are involved in the process, deeply understand it and be in charge of process improvement in general. Example users can be:

- Software Engineer
- Quality assurance engineer
- Project manager
- Product manager

4.2. How to use the model?

To assess the maturity of a RE process, the user basically perform a mapping from the actions present in the model to the activities in his real process. He could find out one of the following situations:

- The action was deemed vital but was performed partially or not at all in this RE process. It should be marked as “**Incomplete**” (IC)
- The action was completed in this RE process. It should be marked as “**Complete**” (C)
- The action was not necessary or possible to be performed in this process. It should be marked as “**Inapplicable**” (IA)

More about “Inapplicable”

In reality, as organizations and processes vary in their characteristics and environments, they may not benefit from implementing all the actions in the model. Some of the actions are deemed unnecessary to be performed in particular situations of organizations. For example, a company has an internal glossary of terms but it is not released to the users. They find it hard and useless to give the user the whole document and ask them to read it as they might not do so. Instead, the terms and examples are explained directly to the users in the meeting and it was more effective this way. In this case, the action “OS.GA.a1 Create a Product-wide Glossary of Terms (Basic Level)” is not useful for them. If we consider it as “Incomplete”, the process may not reach the Basic level as not all actions in this level are fulfilled. This is even more unfair if all other actions in higher maturity level are completed. Therefore, companies should not be “punished” if they do not perform a certain nonessential action (in their point of view). In order to take into account this factor, the option “Inapplicable” is devised. In this way, the model is more fitting to the real process and the evaluation result is less distorted. Besides, as described at the beginning, the differences between two types of development settings (Bespoke and Market-driven) do exist. Therefore, in some cases, the organization may find some actions only applicable in one of the settings.

Whether an action is “Inapplicable” or not is solely based on the judgment of the project evaluator. Reasons for deeming an action “Inapplicable” should be considered carefully to avoid accidentally skipping an important action. Lack of time, resource or unawareness cannot be accounted for an “Inapplicable” action.

4.3. How to read the result?

After mapping all the actions present in the model, the user can collect the results for each MPA and consider the following rules.

- For each MPA, all actions at a certain level must be **Completed** (or **Inapplicable**) in order for the MPA to achieve such level.
- For the whole process, all actions at a certain level must be **Completed** (or **Inapplicable**) in order for the process to achieve such level.

An example

An example

The result of MPA “Organizational Support” after evaluating may look like in Table 2.

Table 1. Assessment result in MPA "Organizational Support"

Level	Actions in real process		Total actions in OS in Uni-REPM
	Completed	Inapplicable	
Basic	0	0	2
Intermediate	3	1	6
Advanced	1	1	2

To have a better view, the result can be presented in graph as follows.

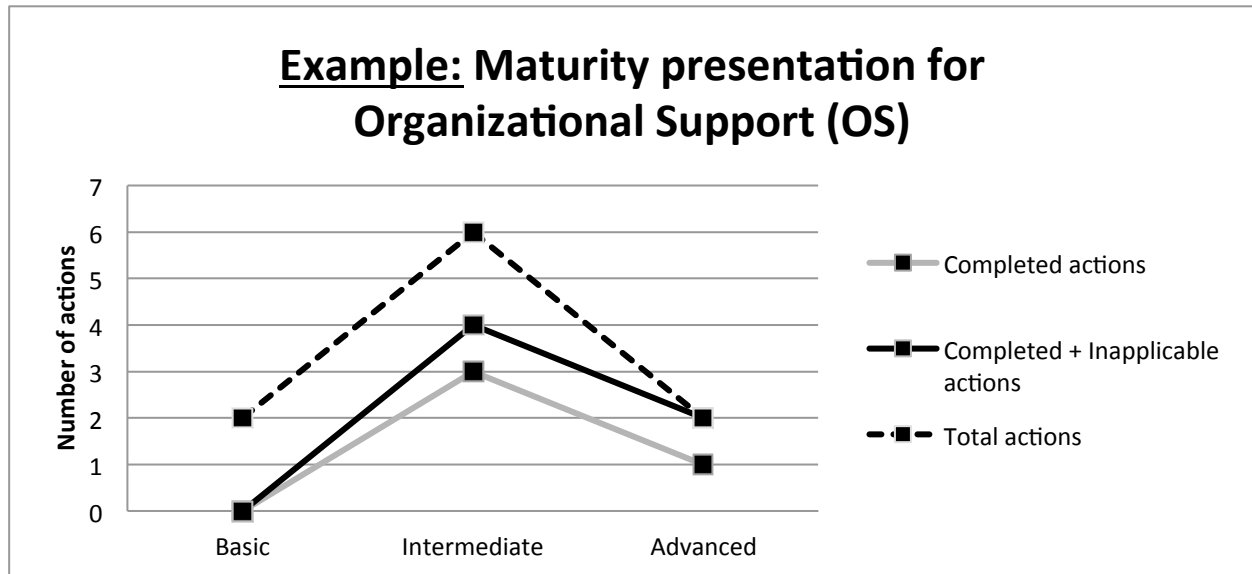


Figure 3. Graphical presentation of assessment results.

The grey line presents actions which were completed in the real process. In this case, no action was done at the lowest level, 3 actions were completed in Intermediate level and 1 action in the highest level. The black line presents actions completed together with actions were not performed due to unnecessary or inapplicable reasons in real process of the assessing organization. The distance between the grey line and black line is called the model lag, which represents the number of inapplicable actions. Hence, the model lag shows the applicability of the model in the real setting. In this case, the model lag is fairly small with only two inapplicable actions. This means a high applicability of the model.

The dash line in the graph presents the total actions that should be completed at 3 levels in “Organizational Support” MPA. For example, at Basic level, there are 2 actions that should be finished. The difference between the black line and the dash line is important because it denotes the improvement area of the process. It shows how many additional actions that should be done in order to attain a certain level of maturity.

Overall, the graph denotes that, in this MPA, the process has not completed all the actions at Basic level. Hence, according to the above rule, the MPA resides on Level 0. In order to reach the Basic level, two more actions have to be done. If the company aims for Intermediate level, it has to perform two Basic actions and another 2 Intermediate ones. Similar work can be done with other MPAs to achieve the result for the whole process.

5. References

- [1] Somerville I., “Software Engineering”, Addison-Wesley, 1995.
- [2] Wohlin C., Aurum A., “Engineering and Managing Software Requirements”, Springer, 2005.
- [3] Karlsson, L., Dahlstedt, A.G., Regnell, B., Natt och Dag, J., Persson, Requirements engineering challenges in market-driven software development - An interview study with practitioners, In the Journal of Information and Software Technology, 49, 6, pp. 588-604, 2007.

- [4] Regnell B., Beremark P., and Eklundh O., "A Market-driven Requirements Engineering Process - Results from an Industrial Process Improvement Programme", Springer, pp. 121-129, 1998.
- [5] Juristo N., Moreno A.M, Silva A., "Is the European Industry Moving Toward Solving Requirements Engineering Problems?", IEEE Software, vol.19, no.6, pp. 70-77, Nov/Dec 2002.
- [6] Beecham, S., Hall, T., & Rainer, A., Software process problems in twelve software companies: An empirical analysis, Empirical Software Engineering, 8, 7–42, 2003
- [7] Niazi, M., An empirical study for the improvement of requirements engineering process, The 17th International Conference on Software Engineering and Knowledge Engineering , pp. 396–399, 2005.
- [8] Hall T., Beecham S., Rainer A., "Requirements Problems in Twelve Companies: An Empirical Analysis", IEE Proceedings for Software, October, vol.149, no.5, pp.153-160, 2002.
- [9] Gorschek T., "Requirements Engineering Supporting Technical Product Management", Karlskrona : Blekinge Institute of Technology, 2006.
- [10] Villalón C., Agustín C., Gilabert S., Seco D., Sánchez G., and Cota P., "Experiences in the Application of Software Process Improvement in SMES," Software Quality Journal, vol. 10, pp. 261 – 273, 2002.
- [11] Paulk M.C., Curtis B., Chrissis M.B. and Weber C., Capability Maturity Model TM for Software Version 1.1, 1993.
- [12] CMMI for Development, Version 1.2, CMMI-DEV v1.2, CMU/SEI-2006-TR-008, Technical Report, Software Engineering Institute, August 2006, URL: <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf>.
- [13] The TickIT Guide – Using ISO 9001:2000 for Software Quality Management System, Construction, Certification and Continual Improvement, Issue 5.0, 2001.
- [14] Ian Somerville and Pete Sawyer, Requirements Engineering – A Good Practice Guide, John Wiley & Sons, Chichester UK, 2000.
- [15] Gorschek T., Tejle K., "A Method for Assessing Requirements Engineering Process Maturity in Software Projects", Blekinge Institute of Technology, Master Thesis Computer Science no. MSC-2002:2, 2002.
- [16] Wiegers K., Software Requirements, Microsoft Press. Redmond, Washington, 2003.

Part III. Model Description

This part provides the description of the Uni-REPM in two views: Process Area and Maturity Level.

1. Process Area View

In this section, the model will be presented by process area. The process area view helps organizations to focus on practices within a specific requirements engineering area e.g. Elicitation.

ID	Title	Level	Page no.
OS	Organizational Support		15
OS.GA	General Actions		15
OS.GA.a1	Create a Product-wide Glossary of Terms	1	15
OS.GA.a2	Train personnel in Requirements Development and Management Processes	2	15
OS.RR	Roles and Responsibilities		16
OS.RR.a1	Assign Owner(s) of Requirements Development and Management Processes	1	16
OS.RR.a2	Define Roles and Responsibilities for Requirements Development and Management Processes	2	16
OS.RR.a3	Define Roles and Responsibilities for Release Planning	2	16
OS.RR.a4	Define Roles and Responsibilities for Change Control	2	16
OS.RR.a5	Define Roles and Responsibilities for Product Management	3	17
OS.S	Strategies		18
OS.S.a1	Define Product Strategies	2	18
OS.S.a2	Define Product Roadmaps	2	18
OS.S.a3	Communicate Strategies in Organization	3	18
PM	Requirements Process Management		20
PM.GA	General Actions		20
PM.GA.a1	Define and Maintain Requirements Development and Management Processes	1	20
PM.GA.a2	Introduce Tool Support for Requirements Development and Management	1	20
PM.GA.a3	Involve various perspectives in Requirement Development and Management Process	2	21
PM.CM	Configuration Management		22
PM.CM.a1	Manage Versions of Requirements	1	22
PM.CM.a2	Baseline Requirements	1	22
PM.CM.a3	Define a Process for Managing Change and Evolution	2	23
PM.CM.a4	Track change requests	2	23
PM.RT	Requirements Traceability Policy		24
PM.RT.a1	Uniquely Identify each Requirement	1	24
PM.RT.a2	Document Requirements' Source	1	24
PM.RT.a3	Define traceability policies	2	24
PM.RT.a4	Document Requirements' Relation	2	24
PM.RT.a5	Document Impact of Requirement on Other Artifacts	2	25
PM.RC	Requirements Communication and Negotiation		26
PM.RC.a1	Establish Effective Communication With Requirements Issuers	1	26
PM.RC.a2	Obtain common understanding of requirements among different involving roles	3	26

RE	Requirements Elicitation		27
RE.GA	General Actions		27
RE.GA.a1	Elicit Quality Requirements	1	27
RE.GA.a2	Qualify and Quantify Quality Requirements	2	27
RE.GA.a3	Let Business Concern Guide Focus of Elicitation	2	27
RE.GA.a4	Use Appropriate Elicitation Techniques according to Situation	2	28
RE.GA.a5	Use Artifacts to Facilitate Elicitation	2	28
RE.GA.a6	Create Elicitation Channels for Requirements Sources	3	28
RE.GA.a7	Reuse Requirements	3	28
RE.SI	Stakeholder and Requirements Source Identification		29
RE.SI.a1	Identify and Involve Relevant Stakeholders	1	29
RE.SI.a2	Distinguish between Different Types of Stakeholders	1	29
RE.SI.a3	Identify Other Requirements Sources	1	29
RE.DC	Domain Consideration and Knowledge		30
RE.DC.a1	Elicit Information about System Domain Restrictions	1	30
RE.DC.a2	Elicit Information about System's Technical Infrastructure	1	30
RE.DC.a3	Elicit Information about System's Business Process	1	30
RE.DC.a4	Elicit Information about System's Operational Domain	1	30
RE.DC.a5	Elicit Information about System Boundaries	1	31
RE.DC.a6	Consider Sociopolitical Influences on Requirements Sources	2	31
RA	Requirements Analysis		32
RA.GA	General Actions		32
RA.GA.a1	Perform Requirements Risk Analysis	1	32
RA.GA.a2	Perform Systematic Requirements Prioritization at Project-level	2	32
RA.GA.a3	Analyze Requirements Relations	2	33
RA.GA.a4	Identify Irrelevant Requirements for Early Dismissal (in/out scope OR Triage)	2	34
RA.GA.a5	Analyze the Strength of Relations between Requirements	2	34
RA.GA.a6	Perform refinement and abstraction of requirements	3	34
RA.QA	Quality attributes analysis		35
RA.QA.a1	Analyze for Missing and Double Requirements	1	35
RA.QA.a2	Analyze for Ambiguous Requirements	1	35
RA.QA.a3	Analyze for Correctness of Requirements	1	35
RA.QA.a4	Analyze for Testability of Requirements	1	35
RA.PS	Problems and solutions analysis		36
RA.PS.a1	Create Prototype	1	36
RA.PS.a2	Perform Systems Modeling	3	36
RP	Release Planning		37
RP.GA	General Actions		37
RP.GA.a1	Synchronize Release Plan with Product Roadmap	2	37
RP.GA.a2	Involve different perspectives in release planning	2	37
RP.GA.a3	Post Requirement Selection Evaluation	3	38
RP.GA.a4	Plan multiple release at pre-defined interval	3	38
RP.S	Requirements Selection		39

RP.S.a1	Pack Requirements into Releases	1	39
RP.S.a2	Estimate Cost and Value of Requirements	2	39
RP.S.a3	Perform Requirements Prioritization at Pre-project Level based on Various Dimensions	2	39
DS	Documentation and Requirements Specification		41
DS.GA	General Actions		41
DS.GA.a1	Establish Standardized Structure for SRS	1	41
DS.GA.a2	Define Requirements Attributes	1	40
DS.GA.a3	Define Requirements States	2	42
DS.GA.a4	Document Requirements Rationale	2	42
DS.GA.a5	Record Rationale for Rejected Requirements	3	42
DS.DD	Documentation Deliverables		43
DS.DD.a1	Define User Documentation Deliverables	2	43
DS.DD.a2	Define System Documentation Deliverables	2	43
DS.DD.a3	Define Management Documentation Deliverables	3	43
RV	Requirements Validation		44
RV.GA	General Actions		44
RV.GA.a1	Validate requirements with relevant stakeholders	1	44
RV.GA.a2	Use Checklist to Ensure Quality of Requirements	1	44
RV.GA.a3	Review Requirements	2	44
RV.GA.a4	Organize Inspections	3	44
RV.GA.a5	Develop Preliminary Test Case or User Manual	3	44
RV.GA.a6	Use System Model Paraphrasing for QA	3	45
RV.GA.a7	Define Acceptance Criteria and Acceptance Tests	3	45

OS Organizational Support

This main process area evaluates the amount of support given to requirements engineering practices from the surrounding organization. Organizational support is important, since ultimately the success of any time-consuming activity needs to be understood and supported by the organization.

OS.GA General Actions

OS.GA.a1 Create a Product-wide Glossary of Terms **Level 1**

A glossary defines all specialized terms which are both domain-specific and product-specific. It also includes acronyms and terms with multiple meanings. Using a glossary can help to reduce misunderstanding and establish same interpretation among different readers with different backgrounds. Moreover, it helps non-expert readers understand application domain concepts/jargons.

OS.GA.a2 Train personnel in Requirements Management Process and Specialty **Level 2**

At the very beginning of the process, you should establish training to ascertain that all involving members in your project obtain a clear understanding on the Requirements Management process which they should follow as well as the standards with which they should keep their products aligned. In addition to process training, you should also provide stakeholders particular trainings to develop skills/specialty required for performing particular tasks. It could be elicitation skills, prioritization techniques, tool used in organizations and so on. This is because not all the members of your team can be aware of all the required techniques, and in many cases the techniques and tools are customized to adapt to your organizational situation. Along with the training, it is also necessary that the importance of the tasks is specified. This activity may include:

- Define a training program(s)
- Prepare documents
- Prepare personnel with appropriate knowledge
- Specify mechanism for measuring the effectiveness of the training program

Supporting action(s)

- PM.GA.a1 Define and Maintain Requirements Development and Management Process

OS.RR Roles and Responsibilities

In order to be able to produce repeatable and predictable results, it is important to define the roles, and their responsibilities, that deal with different aspects of requirements engineering. This information should be documented in a central place and everyone should know where and how to access it.

OS.RR.a1 Assign Owner(s) of Requirements Development and Management Processes Level 1

The owner of the requirements development and management processes has the responsibility of managing the process, assuring that all the requirements engineering activities are executed properly and supporting tools, training are available when needed. The benefit of having the process owner is that the process will be kept alive and updated to changes.

Supporting action(s)

- PM.GA.a1 Define and Maintain Requirements Development and Management Processes

OS.RR.a2 Define Roles and Responsibilities for Requirements Development and Management Processes Level 2

Roles and responsibilities for requirements development and management processes should be defined explicitly in details. In the case of mass market product, this step is particular important as the process does not follow a phase-oriented development model but an asynchronous fostering of requirements through a life-cycle. Some responsibilities that should be specified are creating, analyzing, specifying, validating and managing requirements.

Supporting action(s)

- PM.GA.a1 Define and Maintain Requirements Development and Management Process

OS.RR.a3 Define Roles and Responsibilities for Release Planning Level 2

Release planning is the activity in which an optimal collection of requirements is selected for implementation in the next version of a software system. The responsibilities in release planning include deciding which prioritization aspects to consider, how to prioritize, selecting requirements into release, just to name a few. The roles involved in release planning can be product managers, marketing managers, technical managers, experts, customers etc.

OS.RR.a4 Define Roles and Responsibilities for Change Control Level 2

As change happens all the time throughout the product lifecycle it is necessary to manage changes effectively by defining who is responsible for what in change control process. Some of the possible roles are change control board, change submitter, evaluator, modifier, and

verifier. The change control board is in charge of making decisions whether to approve proposed changes. The change control board should comprise people from different perspectives e.g. project management, product management, marketing, and development in order to have well-rounded and accurate decisions. The evaluator is responsible for analyzing the impact of the requirements change. The modifier executes the approved change on affected artifacts whereas the verifier checks if the change was implemented correctly.

Supporting action(s)

- PM.CM.a3 Define a Process for Change Control

OS.RR.a5 Define Roles and Responsibilities for Product Management

Level 3

Product management is related to managing requirements, defining releases, and defining products in a context where many internal and external stakeholders are involved. The roles involved in product management can be contributors, controllers and distributors. The contributors possess future-oriented tacit knowledge regarding the market. The controller responsibility is to combine contributors' knowledge into product strategies and roadmaps. The distributor disseminates the product strategy and roadmap knowledge into the organization by identifying who depend heavily on it. These roles present not only internal perspective but also external one including sale and customers. Depending on each company, there can be many more roles and responsibilities.

OS.S Strategies

Certain strategic decisions and/or practices influence the requirements engineering process, especially in a market-driven product.

OS.S.a1 Define Product Strategies Level 2

Product strategies can be defined by identifying where a company wants to go (direction of movement), how it will get there (means), what need to be done (tactics) and why it will be successful (rationale). The direction of movement can be determined in terms of profit, growth and market share. The means to reach the goals is by defining the customer targets, competitive targets and differentiated advantage. The tactics cover product, pricing, promotion, distribution, and service. Documenting the rationale is important because it enables replicating the success of the product.

The benefits of the product strategy are that it not only provides the long-term view of the product in the company but also drives the elicitation and analysis processes.

The strategies should be documented in a central place and updated regularly.

Supporting action(s)

- OS.RR.a5 Define Roles and Responsibilities for Product Management

OS.S.a2 Define Product Roadmaps Level 2

The basic purpose of roadmapping is to explore and communicate the dynamic linkages between markets, products, and technologies over a period of time. It also helps requirements engineers to make business-oriented decisions in release planning, elicitation and analysis. Out of many types of roadmaps, the product-technology roadmap defines what a product tends to achieve over the time in terms of its evolvement and the technology trends.

The roadmaps should be documented in a central place and updated regularly.

Supporting action(s)

- OS.RR.a5 Define Roles and Responsibilities for Product Management Organization

OS.S.a3 Communicate Strategies in Organization Level 3

Strategies are nothing without implementation. Strategies have to be disseminated to those who need to act on them. The first line consumers of strategic knowledge are people in development or productization teams who take the knowledge as input to their activities. Because of the immediate impact of the strategies on their work, a distributor is needed to communicate the strategies directly to the first line consumers. The second line consumers are those whose inputs are from the first line works, such as customers, partners, sales and

technical support. For these people, a documented form of strategies is enough.

Supporting action(s)

- OS.RR.a5 Define Roles and Responsibilities for Product Management
- OS.S.a1 Define Product Strategies
- OS.S.a2 Define Product Roadmaps

PM Requirements Process Management

The requirements process management covers all the activities to manage, control requirements change as well as to ensure the organization of the process and coherence among team members.

PM. GA General Actions

PM.GA.a1 Define and Maintain Requirements Development and Management Processes **Level 1**

It has been clear the benefit of having pre-defined processes to develop and manage requirements. This is to ensure a well-organized way to control the whole requirements process, and to guide the stakeholders of what to do next and How should it be done in a structured way. At project level, it is quite common to follow the phase-oriented process model in which requirements should be managed in phases such as elicitation, analysis (and negotiation), and documentation before being passed to another process within the project development. However, at pre-project level (product level), there is usually continuous stream of huge amount of requirements. Hence, concurrent approach models are preferable.

Example

You can follow one of the process model studied in research REQUEST [1], REPEAT [2], MDRE [3] or can tailor one based on these model to your organization.

Supporting action(s)

- OS.RR.a1 Assign Owner (s)of Requirements Development and Management Processes

PM.GA.a2 Introduce Tool Support for Requirements Development and Management **Level 1**

The tool support for the whole requirements process should be considered and chosen early. You can consider the following types of tools:

- Storage tools: with a huge number of requirements, especially in Market-driven requirements engineering, it is crucial to have a database to store them. It is more advantageous if you can have a centralized repository for requirements so that all the changes will be applied in real time and different stakeholders can have the same view at a set of requirements.
- Version tools: provide automatic assignment of versions.
- Prioritization support tools: support prioritization.
- Elicitation tools: support elicitation

Example

You can find below the table of the support tools which are used quite popularly nowadays [4].

Tool name – Producer name	Purpose
Vital link - Compliance Automation	Database-centric system
RTM system	Provide repository for requirements management
RM Trak - RM Trak	Requirements management at entry level
Caliber RM – Borland	Lifecycle oriented, for large systems, provide traceability
CARE	Database-like view, requirement-centric system
Vital link - Compliance Automation	Database-centric system
DOORS – Telelogic	Integrated management, large projects, API available, High, XML support; PLM, UML tools, MS Project
IRRV (Integral Requisite Analyzer) - TCP Sistemas & Ingeniería	Requirements classification, OO analysis and entity relationship method for database design, traceability, test support, XML support; MS Office
ReqSimile	Finding and linking similar requirements
cost-value tool support	Prioritization using AHP
PARSEQ tool	Tool support for Post-release validation
Reqtify TNI-Valiosys	Traceability and impact analysis; text processing, office tools
Requisite Pro - IBM Rational	Change management, traceability, XML support; work with MS Word, Rational Rose, TeamTest, MS Project
Truereq	Lifecycle-oriented management, team-centric, entry level, XML support;

Supporting action(s)

- OS.GA.a2 Train personnel in Requirements Development and Management Processes

PM.GA.a3 Involve various perspectives in Requirements Development and Management Process Level 2

It is very important to get different perspective involve appropriately in the Requirements Engineering process. They could be relevant engineers, customers or experts collaborating with the ones responsible for specific tasks. This is to prevent subject views in developing and managing requirements and the RE process.

Supporting action(s)

- RE.SI Stakeholder and Requirements Source Identification

PM. CM Configuration Management

When it comes to working with a large number of or continuously changed requirements, you should always manage the configuration of them to support traceability and avoid confusion.

PM.CM.a1 Manage Versions of Requirements

Level 1

Change happens along the requirements process. Therefore, it is necessary to control the version of your requirements in order to avoid confusion and support traceability. You can choose to use documents to version the requirements or tool support. However, if you work with a huge number of requirements, it is recommended to use a version control system. The historical information of requirements version will help you to trace back when necessary (e.g. when uncovering some mistakes performed on a requirement(s)), and to ensure that the requirements you are working on are the right ones (e.g. the latest requirements instead of an obsolete ones).

Example

You can use CVS, Subversion to support version control [4].

Supporting action(s)

- PM.GA.a2 Introduce Tool Support for Requirements Development and Management

PM.CM.a2 Baseline Requirements

Level 1

This is especially important when you work with a huge number of requirements for the system, and sometime they happen to change continuously. The idea of this activity is that, once your team (and customers) has reached an agreement on a set(s) of requirements, you should capture and save this state of the set(s) as a baseline. This baseline will be served as a stable point for other activities, e.g. implementation, testing, etc... This activity is preferably performed in more stable stages such as after analysis (and negotiation) or when release planning is done.

Supporting action(s)

- PM.GA.a2 Introduce Tool Support for Requirements Development and Management
- PM.CM.a1 Manage Versions of Requirements

PM.CM.a3 Define a Process for Change Control

Level 2

Change has been agreed to be the nature of requirements. Since there is no way to avoid it, you had better define a process to control it. The procedure of the change process should be clearly specified. It can cover certain steps a change request must follow and requirements mentioned in the request should be considered. It is also important to specify which factors and which technique to use during re-analyzing process.

Example

You can use tool to support the change process [4]. This will give involved stakeholders a real time view of the change request.

Supporting action(s)

- OS.RR.a3 Define Roles and Responsibilities for Change Control
- PM.GA.a2 Introduce Tool Support for Requirements Development and Management

PM.CM.a4 Track Change Requests

Level 2

Since change requests are usually passed through and processed among different stakeholders, you should always keep track on them. The most common and easy way to do so is defining the change request status and keep it up-to-date. Moreover, you should provide a mechanism to ensure issuers can easily and accurately determine the status and disposition of their change requests.

Example

You can use the following status to track the change requests: New, Selected, Implemented, Verified, and Rejected [3].

Supporting action(s)

- PM.CM.a3 Define a Process for Change Control

PM.RT Requirements Traceability Policy

Along the requirements process, you will mostly apply number of changes on requirements. In order to ensure the consistency of the system, it is important that you prepare for these cases so that you can always trace from requirements to other artifacts of the project and vice versa to apply necessary changes.

PM.RT.a1 Uniquely Identify each Requirement **Level 1**

Every requirement should have a unique identification. This is especially important when working with a large number of requirements. Having this, the requirements can be easily specified when passing between different stakeholders during the process.

Example

Repository can automatically assign ID for requirements for you [4].

PM.RT.a2 Document Requirements' Source **Level 1**

Requirements' source is valuable for traceability, e.g. when need of clarification occurs. In case there is no specific customer, the requirements' source could be the issuers. You can specify and store this information in one of the attributes of individual requirements so that it could be easily found when other stakeholders access the requirement.

Supporting action(s)

- RE.SI.a1 Identify and Involve Relevant Stakeholders
- RE.SI.a3 Identify other Requirements source

PM.RT.a3 Define Traceability Policies **Level 2**

Define policies for traceability helps to determine the tracing routines and directions. It is important to specify in the policies the relevant information and artifact which are impacted by requirements changes. It is also necessary to identify the directions such as backward or forward tracing from requirements to other artifacts. The documentation of tracing result also needs to be defined here.

Supporting action(s)

- RE.SI.a1 Identify and Involve Relevant Stakeholders
- RE.SI.a3 Identify other Requirements source
- PM.GA.a1 Define and Maintain Requirements Development and Management Processes

PM.RT.a4 Document Requirements' Relations **Level 2**

Requirements' relations are valuable for tracing from requirements to requirements (e.g. when change occurs at requirement A which impacts requirement B). This action should be done together with an analysis of the dependencies between requirements.

Supporting action(s)

- RA.GA.a3 Analyze Requirements Relations
- RA.GA.a4 Identify irrelevant requirements for early dismissal (in/out scope OR Triage)

PM.RT.a5 Document Impact of Requirement on Other Artifacts

Level 2

Requirements are the initial images of the system; hence they impact many other artifacts such as test cases, components, modules and so on. Once change occurs, it is important to apply changes in all related artifacts. Therefore, it is necessary to document the impact of requirements on those artifacts. To do so, you can record the related artifacts for each document together with the importance level of this impact. Although this is a quite expensive activity since it requires a lot of effort from different involvements, it is beneficial to ensure the safety of the whole system and to save your effort in later phases (as well as avoid re-work when impact occurs).

Supporting action(s)

- PM.GA.a2 Introduce Tool Support for Requirements Development and Management
 - DS.GA.a2 Define Requirements Attributes
-

PM.RC Requirements communication and negotiation

One important aspect in requirements process management is to ensure the coherence between teams and team members. It is recommended to establish an adequate communication basis among involving parties to reduce gaps (misunderstanding, conflict...).

PM.RC.a1 Establish Effective Communication With Requirements Issuers Level 1

Requirements are usually gathered from different sources. Therefore, there always exists the need of clarification and verification for them. Apart from that, it is also necessary to observe customers' changes in expectation, especially in MDRE case. In this activity, you should establish an effective communication (i.e. define communication channel, interval...) with the issuers to obtain clear understandings of their desires.

Example

Some means could be used to implement this communication such as: Implementation proposal [5] or prototype or making use of rich communication channels.

Supporting action(s)

- PM.RC.a2 Document requirements source

PM.RC.a2 Obtain common understanding of requirements among different involving roles Level 3

Common understanding on requirements (i.e meanings, estimation values, prioritization rationale...) should be shared between different involving roles to reduce gaps. This activity may include regular meetings, emails or informal discussions to exchange necessary information. You should consider other teams which will later work with your outputs such as implementing or testing teams.

Supporting action(s)

- RE.SI.a1 Identify and Involve Relevant Stakeholders

RE Requirements Elicitation

Elicitation is the process of discovering, understanding, anticipating and forecasting the needs and wants of the potential stakeholders in order to convey this information to the system developers. The potential stakeholders can include customers, end-users and other people who have the stake in the system development. In the process, the application domain and organizational knowledge are necessary among other things.

RE.GA General Actions

While there is much to say about the actual elicitation practices, in this sub-process area we focus on the overall framework under which you conduct the specific elicitation practices.

RE.GA.a1 Elicit Quality Requirements Level 1

Quality requirements, also known as non-functional requirements cover performance, accuracy, reliability, security, usability etc. of the system. Quality requirements are critical because they can affect a large part of the functionality. Not eliciting quality requirements can cause customer disappointment and major rework or product failure.

RE.GA.a2 Qualify and Quantify Quality Requirements Level 2

Use appropriate metrics and value to quantify and specify quality requirements in order to understand, measure and test them correctly.

Supporting action(s)

- RE.GA.a1 Elicit Quality Requirements

RE.GA.a3 Let Business Concerns Guide Focus of Elicitation Level 2

Business concerns are abstract high-level goals which the product should meet in order to be useful. In the mass market context, business concerns are expressed in the form of product strategies. By using the business concerns as means to align the elicitation resources, the time and money spent on elicitation is assured to be aligned with overall goals for the product.

Supporting action(s)

- OS.S.a1 Define Product Strategies

RE.GA.a4 Use Appropriate Elicitation Techniques according to Situation Level 2

Requirements for software-intensive system are complex and varied. Based on each unique case, certain suitable techniques/methods should be chosen and adapted. Some of the method selection criteria are usage context, knowledge types, internal filtering of knowledge and the purpose of requirements.

Example

Some of the useful techniques you can choose to use:

- Observation [6]
- Interview [6]
- Brain storming [6]
- Market survey [7]

RE.GA.a5 Use Artifacts to Facilitate Elicitation Level 2

Additional artifacts like prototypes, scenarios can be used to provide a better understanding of the problems at hand by simulating the interactions of the end-users with the system. By using these artifacts, the end-users can refine their ideas about the system requirements as well as expose their real needs.

Example

You can use scenario analysis [6] and prototype [6] to perform this action.

RE.GA.a6 Create Elicitation Channels for Requirements Sources Level 3

Customer feedback and requests that can be turned into requirements can be captured in many different forms such as incident reports, idea feedback and suggestion. It is recommended to give customers feedback afterwards about their suggested requirements to ensure their continuous contributions. Moreover, company should enable elicitation channels for In-house stakeholders e.g. developers, testers to submit new requirements.

RE.GA.a7 Reuse Requirements Level 3

Reusing requirements is also one of the channels for requirements source. Requirements for a new system can be developed by reusing existing requirements of other systems in the same application domains directly or indirectly. Direct reuse means that minimal modifications will be done to make the existing requirements suitable to the new systems whereas indirect usage means that new requirements are created based on existing ones. In order to effectively reuse requirements, a systematic and planned reuse process has to be defined. Companies can reduce cost, time and risk by using this approach.

RE.SI Stakeholder and Requirements Source Identification

Stakeholders are people who have interests in the product. In order to do successful requirements engineering, it is important to identify whom we would like to listen to and which source of information we can look into in order to elicit requirements for the system.

RE.SI.a1 Identify and Involve Relevant Stakeholders Level 1

Explicitly identify all potential stakeholders, who can be customers, end-users, marketing personnel, managers, developers, testers etc, and consult the relevant ones. The stakeholders will provide requirements or impose constraints on the system. This ensures that all the concern of affected people will be taken into account.

RE.SI.a2 Distinguish between Different Types of Stakeholders Level 1

As different types of stakeholders have different interests and expectations in the system, it is important to distinguish between them in order to elicit all relevant requirements. Among them, customers, end-users and in-house stakeholders are fairly important. Customers are the people who have the authority to purchase/order the system whereas end-users are the ones actually using the system in their work. In some cases, customers may be end-users as well. In-house stakeholders involving in the development/management of the system are often overlooked in the elicitation process.

Supporting action(s)

- RE.SI.a1 Identify and Involve Relevant Stakeholders

RE.SI.a3 Identify Other Requirements Sources Level 1

Besides stakeholders, other sources of information can also provide requirements for the system. Those sources include regulations, bug reports, market surveys, product reviews, and company standards.

RE.DC Domain Consideration and Knowledge

In order to discover accurately requirements of the systems from various stakeholders, it is required to obtain application domain, organizational as well as other specific knowledge. This SPA consists of many types of knowledge that are necessary to be aware of during elicitation.

RE.DC.a1 Elicit Information about System Domain Restrictions Level 1

Domain experts should be consulted regularly in order to identify the domain constraints imposing on the system. For mass market product, the domain expert should come from inside the organization whereas in customer product, the expert can reside in the customer side. If these constraints are overlooked, it would result in a product failure or legal, organizational, physical obstacles.

Supporting action(s)

- RE.SI.a1 Identify and Involve Relevant Stakeholders
- RE.SI.a3 Identify other Requirements source

RE.DC.a2 Elicit Information about System's Technical Infrastructure Level 1

Technical infrastructure refers to the operating environment in which the system will be installed. It consists of the platform, other hardware and software that interact with the system. Taking into account this information can help to avoid some installation problems.

Supporting action(s)

- RE.SI.a1 Identify and Involve Relevant Stakeholders
- RE.SI.a3 Identify other Requirements source

RE.DC.a3 Elicit Information about System's Business Process Level 1

The system is built because it can contribute to the business of the organization. Therefore, general information about the business process in which the system will function helps to drive the elicitation process forward.

Supporting action(s)

- RE.SI.a1 Identify and Involve Relevant Stakeholders
- RE.SI.a3 Identify other Requirements source

RE.DC.a4 Elicit Information about System's Operational Domain Level 1

The elicitation process should take into account other business processes which are supported by the system being developed in order to reveal process requirements and constraints imposing on the system.

Supporting action(s)

- RE.SI.a1 Identify and Involve Relevant Stakeholders

- RE.SI.a3 Identify other Requirements source

RE.DC.a5 Elicit Information about System Boundaries **Level 1**

System boundaries define the scope of the system being developed. This information can be obtained by working with the customer or by consulting the product strategies. The information will then be used to focus the effort on the requirements residing within the boundaries.

Supporting action(s)

- RE.SI.a1 Identify and Involve Relevant Stakeholders
- RE.SI.a3 Identify other Requirements source

RE.DC.a5 Consider Sociopolitical Influences on Requirements Sources **Level 2**

Organizational and political factors can affect or conceal the real system requirements. People may have different hidden agenda and not all of them are willing to contribute to the system being developed. Being aware of these factors can help to understand the real reason for including the requirements.

Supporting action(s)

- RE.SI.a1 Identify and Involve Relevant Stakeholders
- RE.SI.a3 Identify other Requirements source

RA Requirements Analysis

Requirements gathered from different sources need to be analyzed to detect incomplete or incorrect ones as well as to estimate necessary information for later activities (e.g. risk, priorities...). It is also recommended that you should perform some analysis to dismiss irrelevant requirements to avoid wasting effort in next steps.

RA.GA General Actions

RA.GA.a1 Perform Requirements Risk Analysis Level 1

The requirements will need to be analyzed to estimate possible problems arose in the future; hence the managers can have plans to prepare and overcome those risks. In case of products developed for specific customer(s), risk analysis will also provide necessary information for negotiation activities. In case of products developed for mass market, this analysis will be important as an input for release planning. The engineer should perform risk assessment on individual requirements or sets of them or certain selected requirements. In addition, it is also recommended that the probabilities of risks and the effects as well as the significance levels of these effects could be analyzed.

RA.GA.a2 Perform Systematic Requirements Prioritization at Project-level Level 2

In many cases, requirements could be prioritized before they are sent to certain project (e.g. in market-driven development, requirements are usually prioritized at pre-project level to perform release planning). However, at In-project level, there are often more detail requirements introduced in those cases (e.g. features are broken down into implementable and testable requirements). Apart from that, many projects skip the pre-project steps (e.g. in bespoke development). Hence, it is also necessary to prioritize requirements at In-project level. The information of this step will be valuable for negotiation with customer(s) (eg. To eliminate the unnecessary potential requirements) or schedule the implementation of them. The basic aspects can be considered in this step are requirements importance and implementation effort. The requirements priorities analyzed before at pre-project level if available can also be used in this step.

Example

You can choose to use Prioritization techniques [8] such as

- Pair-wise comparisons
- Prioritization working groups
- Scale of 1-to-10 rankings
- Voting schemes (e.g., give each stakeholder a specific number of votes to distribute amongst the requirements or classes of requirements being prioritized)

- Weightings (e.g., weight the votes of different stakeholders)
- Value-Based Software Engineering
- WIN-WIN
- Quality Function Deployment (QFD)
- Cost-value approach
- Focus-point

Supporting Action(s)

- RP.S.a3 Perform Requirements Prioritization at Pre-project level based on various dimensions

RA.GA.a3 Analyze Requirements Relations

Level 2

It is important during the analysis to consider the relations among requirements. This is based on the fact that requirements are usually not singular. There are requirements that require the others requirements to be implemented before, or exclude the others. Hence, having an overview of these relations will help you in later phase, e.g. in release planning, change control, etc. Besides, it is also very important to consider the relations between quality requirements (also known as non-functional requirements) and functional requirements.

Example

The usual considered relations (mentioned in [9]) are:

- **“Require”** relation: Requirement A depends on requirement B (B need to be implemented before A)
- **“And”** relation: a two-way “require” relation
- **“Or”** relation: when requirement A is similar to requirement B

Apart from the aforementioned requirements dependencies, there are more types of them that should be considered as well. These types of dependencies are more value-related, hence mostly support product planning perspective. Those are:

- **“ICOST”**: A requirement stating that “the system should be able to serve 1000 users concurrently” will typically increase the cost of implementing many other requirements. ICOST relation could cause both negative and positive impacts on other requirements.
- **“CVALUE”**: Requirements A may impact value of requirement B (A CVALUE B). For example, a real-time sharing document may decrease the customer value of exporting documents. CVALUE relation could cause both negative and positive impacts on other requirements.

RA.GA.a4 Identify Irrelevant Requirements for Early Dismissal (in/out scope OR Triage) Level 2

This step is to early dismiss the irrelevant requirements so that the huge amount of initial requirements could be reduced to avoid wasting time and effort for future works. In order to perform this step, the requirements should be aligned with the boundaries of the developing system. Requirements which are not in-scoped should be eliminated. The boundaries of the system can be defined by discussions with customers. In case there is no specific customer, requirements should be compared to the strategies and plans of the product.

Example

Model for Early Requirements Triage and Selection (MERTS) can be used as a tool for requirements early dismissing in case there is no specific customer [10].

Supporting action(s)

- RE.DC.a4 Elicit Information about System Boundaries
- OS.S.a1 Define Product Strategies

RA.GA.a5 Analyze the Strength of Relations between Requirements Level 2

In addition to the types of relations, you should also estimate the strength of the detected relations. This information will be very useful for later activities in negotiation or release planning when you need to consider and make the trade-off.

However, it is always difficult to keep track of all kinds of dependencies and to visualize them. Hence, it is also important that you should choose the necessary aspects that you are interested in and will use in later activities.

RA.GA.a6 Perform Refinement and Abstraction of Requirements Level 3

Requirements often come from various sources hence they are usually diverse in levels of abstraction. By performing refinement/abstraction to synchronize their abstraction levels, it will be easier for understanding, managing and further activities such as prioritization. It is also recommended to keep multiple levels of abstraction, for instance: high level which consists of more general and goal-like requirements, and low level which contains more specific requirements for implementation.

Example

One stakeholder requires “System must look user-friendly” while another asks for something like “Waiting time does not exceed 5 seconds”. You can make use of Requirements Abstract Model (RAM) to perform this step [11]. This model is validated in industry and very useful for this purpose.

RA.QA Quality attributes analysis

In this step, the elicited requirements will be analyzed in order to ensure their quality aspects such as completeness, correctness and testability.

RA.QA.a1 Analyze for Missing and Double Requirements **Level 1**

After elicitation phase, the raw requirements need to be analyzed to detect missing and overlapped requirements. This step is to uncover the incomplete requirements so that a clarification could be made with the source of requirements to obtain the correct desires or expectations of the stakeholders.

RA.QA.a2 Analyze for Ambiguous Requirements **Level 1**

The requirements will also be analyzed to uncover volatility. At this step, the requirements will be checked whether they are clear enough for readers to understand and to be implemented. If volatility is detected, a clarification could be made with the source of requirements to obtain the clearer demand from the stakeholders.

RA.QA.a3 Analyze for Correctness of Requirements **Level 1**

The requirements also need to be checked in term of correctness since many of those are proposed from “non-it” users. Some may even conflict to the other requirements. Hence it is necessary to investigate the incorrect requirements and clarification can be made if necessary.

RA.QA.a4 Analyze for Testability of Requirements **Level 1**

This step is to uncover the inadequate requirements meaning requirements in which information is not sufficient for testing in next phases. It is usually the case of quality requirements. A clarification could be made with the source of requirements to obtain more detail desires or expectations of the stakeholders.

RA.PS Problems and solutions analysis

RA.PS.a1 Prototyping

Level 1

Additional artifacts like prototypes, scenarios can be used to provide a better understanding of the problems at hand by simulating the interactions of the end-users with the system. By using these artifacts, the end-users can refine their ideas about the system requirements as well as expose their real needs.

Example

You can use Scenario analysis [6] to perform this action.

RA.PS.a2 Perform Systems Modeling

Level 3

System modeling covers models of system specification information, system environment and system architecture. Different parts of the system can be modeled, in the context of business processes that may use the system. The different sub-systems existing within the system and the links between them are also necessary to be described here.

Example

You can make use of data processing models, composition models, classification models, stimulus-response model and process model to demonstrate system models [12].

RP Release planning

Release planning covers crucial steps aiming to determine the optimal set of requirements for a certain release to be implemented at a defined/estimated time and cost to achieve some goals. Performing this step carelessly would lead to high risky situations or fail to achieve planned goals. For example, placing important features at a too late release would make the product miss the right moment to gain the customers' impression.

RP.GA General Actions

RP.GA.a1 Synchronize Release Plan with Product Roadmap

Level 2

Product roadmap is important to support the planners in determining the contents of a release. By aligning the requirements with the product plan (including strategies and time) in the roadmap, the planners could easily consider whether the requirements should be included or excluded in a certain release.

Example: If the roadmap states that the upcoming release should target at Chinese market; requirements investigated from Chinese market such as Chinese language feature, Chinese keyboard feature, etc. should be of higher priorities.

In addition, product-technology road map gives planners an overview of the relationship among product releases and their evolvement along the time axis. Hence, it is beneficial for planners to use product-technology road map along their decision process to decide which requirements need to be postponed or excluded, etc. in a certain release due to technology constraints.

Vice versa, the planners should also consider how new features will impact the existing product.

Supporting action(s)

- OS.RR.a3 Define Roles and Responsibilities for Release Planning Activities
- OS.S.a2 Define Product Roadmaps

RP.GA.a2 Involve Different Perspectives in Release Planning

Level 2

As mentioned above, release planning itself is a trade-off among customer-value, financial value, developing cost, risk, etc. The nature of it clearly shows a need of involving different perspective in the process. The four perspectives: product management, marketing, development and finances preferably participate in this step. Besides, it is also beneficial to involve external customers in the process to achieve external view of the products. It is also recommended that the different perspectives could form a cross-functional team and work together along the product life cycle in order to achieve a mutual understanding and improve the decision making quality.

One way to perform this diverse involvement is to get different perspectives participate in prioritization and give them appropriate weights based on their importance.

Supporting action(s)

- RE.SI.a1 Identify and Involve Relevant Stakeholders
- OS.RR.a2 Define Roles and Responsibilities for Release Planning Activities

RP.GA.a3 Post Requirement Selection Evaluation Level 3

Post-release evaluation is the step to assess the quality of the requirements selection in the previous step. This is to uncover previous mistakes/misunderstanding, gain experiences in decision making, hence ensure the quality of decisions for future releases.

The evaluation can be done by analyzing the measurements such as customer value, market penetration, profit, and revenue etc. of different product releases after they were released. Based on this information, the planners could see whether he/she had made a correct decision at that time and further investigate the mistakes.

Example

PARSEQ (Post- Release Analysis of Requirements Selection Quality) is an industrial-validated method supporting this step [13].

Supporting action(s)

- OS.RR.a2 Define Roles and Responsibilities for Release Planning Activities

RP.GA.a4 Plan Multiple Releases at Pre-defined Interval Level 3

Although it is obvious that not all of the potential requirements could be included in one release, it is always recommended to show the plan to implement them in next few releases. The reason is that, requirements present customers' desires, and excluded requirements indicate that customers might be disappointed. Hence, having few releases ahead enable sale personnel to be able to show that they may be compensated in next releases. This is very important for customer-relationship development.

The planning should be undertaken at pre-defined interval since Market-driven planning is more like a trade-off between current state and future. Therefore, regularly reviewing the plans will give more chances to evaluate the decisions, hence re-plan to adapt to the current situation.

RP.S Requirements Selection

RP.S.a1 Pack Requirements into Release Level 1

Requirements after being prioritized will be selected to certain releases. The selection activity usually requires the involvement of different perspectives from marketing, developing and management, etc. Besides, you should also consider the interdependences when packing requirements into release. There are requirements with low priorities but mandatory for other higher priority ones, hence taking the relations into account is very important. Currently, there is no tool that fully support for this activity even it is considered a very challenging one.

Supporting action(s)

- OS.RR.a3 Define Roles and Responsibilities for Release Planning Activities
- RP.S.a3 Perform Requirements Prioritization at Pre-project level based on various dimensions
- RA.GA.a3 Analyze Requirements Functional Dependencies
- RA.GA.a5 Analyze Value-related Dependencies between Requirements

RP.S.a2 Estimate Cost and Value of Requirements Level 2

The cost for implementing each requirements and their value need to be estimated in order to identify benefit can be obtained on individual requirements and set of them. Besides, this information also helps the decision making in release planning process to make the trade-off between cost and benefit of releases.

RP.S.a3 Perform Requirements Prioritization at Pre-project Level based on Various Dimensions Level 2

Requirements prioritization at pre-project level helps to determine the relative necessity of the requirements. With a huge number of mandatory requirements which are impossible to be implemented all at the same time, it is crucial to specify which are more critical than others.

In addition, requirements need to be prioritized along more than one dimension (related or even opposing ones). And these dimensions can be valued differently by different stakeholders. Usually, customer-value, cost and interdependencies are considered as the basic dimensions. Customer-value present customer preference of the requirements while cost presents how much would be spent to implement the requirements (in finance and man month).

Apart from the aforementioned dimensions, the prioritization can also take into account additional ones such as business value, risk, harm avoidance, legal mandate, etc.

Usually the result of prioritization is served as an input for requirements selection. Hence, the more aspects are considered, the more carefully the selection can be performed which

can result in a better decision.

Example

Several prioritization techniques [8] are available and validated for the engineers to choose, namely:

- Pair-wise comparisons
- Prioritization working groups
- Scale of 1-to-10 rankings
- Voting schemes (e.g., give each stakeholder a specific number of votes to distribute amongst the requirements or classes of requirements being prioritized)
- Weightings (e.g., weight the votes of different stakeholders)
- Value-Based Software Engineering [Boehm 2003]
- WIN-WIN [Boehm 2001]
- Quality Function Deployment (QFD)
- Cost-benefit approach
- Focus-point

DS Documentation and Requirements Specification

Documentation and Requirement specification deal with how a company organizes requirements and other knowledge gathered during requirements engineering process into consistent, accessible and reviewable documents. The software requirements specification (SRS) contains the product's detailed functional and quality requirements.

DS.GA General Actions

DS.GA.a1 Establish Standardized Structure for SRS Level 1

Companies should define a common standard structure which reflects the best practice to organize the requirements document in the companies. The best structures vary among companies as they are influenced by the custom of companies, the type of products developed and the development processes. The common structure helps users to understand the document faster and assure high quality of documents.

DS.GA.a2 Define Requirements Attributes Level 1

Each requirement is specified with a number of attributes associated with it. Attributes are assigned values to reflect what is known about the requirement such as estimated cost, priority, state. Different attributes are specified and utilized depending on the various needs of the companies. The benefit of having attributes is to separate important pieces of information about a requirement from its description. Hence, companies can manage requirements more effectively and efficiently by looking at different properties of the requirements. In case requirements are stored in a database, managers can use tool support to simply query, sort or filter the requirements.

Example

Some of the attributes that can be present are ID, Title, Description, Requirement Source, Status and Rationale [14] [11].

Supporting action(s)

- PM.GA.a2 Introduce Tool Support for Requirements Development and Management

DS.GA.a3 Define Requirements States

Level 2

The states of the requirements represent their refinement levels in the progress towards release. Some possible states are New (requirement is issued), Selected (requirement is analyzed and selected for implementation), Implemented (requirement is successfully realized), Rejected (requirement is excluded). Tracking requirements states help to monitor the requirements and project progress more accurately.

Supporting action(s)

- DS.GA.a2 Define Requirements Attributes
- PM.GA.a2 Introduce Tool Support for Requirements Development and Management

DS.GA.a4 Document Requirements Rationale

Level 2

The reason why a requirement is included should be recorded in order for the readers to understand the requirements. It is extremely useful in case those who initially defined the requirements have left the company. Moreover, it will help problem expert to check if the requirements are consistent with the problem being solved.

Supporting action(s)

- DS.GA.a1 Define Requirements Attributes
- RE.DC Domain Consideration and Knowledge

DS.GA.a5 Record Rationale for Rejected Requirements

Level 3

When requirements are rejected after analysis or negotiation, the reason for rejection and who rejected it should be recorded immediately to avoid being forgotten. This information will be helpful for future reference when dismissed requirements resurface as they can be checked without spending effort in re-analysis.

Supporting action(s)

- DS.GA.a3 Define Requirements States
- RA.GA.a4 Identify irrelevant requirements for early dismiss (in/out scope OR Triage)
- RP.S.a1 Pack requirements into release

DS.DD Documentation Deliverables

The documentation deliverables imply all deliverables the requirements engineering process supposes to produce. You should clearly define the expected deliverables of RE process at the beginning of it as requirements for the project itself. This activity can also be done together with members from other phases that will use the information later.

DS.DD.a1 Define User Documentation Deliverables Level 2

The user deliverables may consist of user manual, user dictionary, etc. This aims to describe the system from user points of view and how to use it.

DS.DD.a2 Define System Documentation Deliverables Level 2

Management deliverables cover all the necessary documents for managing the system such as: system design, technical specification, etc.

DS.DD.a3 Define Management Documentation Deliverables Level 3

Management deliverables cover all the necessary documents for managing the system such as maintenance, administrative manual, etc.

RV Requirements Validation

This process involves checking the documented requirements against defined quality standards and the real needs of various stakeholders. It ensures that the documented requirements are complete, correct, consistent, and unambiguous.

RV.GA General Actions

- RV.GA.a1 Validate requirements with relevant stakeholders **Level 1****
Requirements must be validated with the relevant stakeholders in order to ensure their consistency, completeness and adequacy. Moreover, the intent and interpretation of the requirements can also be verified.
- RV.GA.a2 Use Checklist to Ensure Quality of Requirements **Level 1****
Checklist draws the participants' attention to the certain aspects of the requirements document as well as the frequently encountered problems. Checklist should not be too long to prevent people from referring to the list too often.
- RV.GA.a3 Review Requirements **Level 2****
Reviewing is the technique involving peers (someone other than the author) to examine the requirements and identify defects. The author is then responsible for correcting the found problems.
Supporting action(s)
– RV.GA.a2 Use Checklist to Ensure Quality of Requirements
- RV.GA.a4 Organize Inspections **Level 3****
Inspections are formal meetings in which a small team of inspectors with different perspectives (e.g. customer, analyst, developer, tester) carefully examine the requirements, detect errors and resolve them together. This technique can identify a high percentage of requirements errors but requires all parties to be present at the same time.
Supporting action(s)
– RV.GA.a2 Use Checklist to Ensure Quality of Requirements
- RV.GA.a5 Develop Preliminary Test Case or User Manual **Level 3****
Creating possible test cases or writing a draft user manual can force a detailed look at the requirements and uncover problems with the requirements document related to ambiguities, inconsistencies or usability. The test cases or draft user manual can be used

later as a basis for actual artifacts.

RV.GA.a6 Use system model paraphrasing Level 3

Converting system models into natural language enables general stakeholders to understand these models more clearly and comment on them. In this way, additional requirements or problems will be detected.

Supporting action(s)

- RE.EP.a4 Create Artifacts to Facilitate Elicitation

RV.GA.a7 Define Acceptance Criteria and Acceptance Tests Level 3

An effective technique to validate requirements is by having customers define the acceptance criteria. The acceptance criteria and acceptance test determine if requirements are right and the product satisfies them. They are used to validate the most commonly used and important use cases and requirements but they do not replace system testing.

2. Maturity Level View

In this section, the model can be viewed by maturity level. This view shows the practices from all process areas which the organization should implement in order to achieve a specific maturity level.

Level 1 – Basic		
ID	Title	Level
OS	Organizational Support	
OS.GA	General Actions	
OS.GA.a1	Create a Product-wide Glossary of Terms	1
OS.RR	Roles and Responsibilities	
OS.RR.a1	Assign Owner(s) of Requirements Development and Management Processes	1
PM	Requirements Process Management	
PM.GA	General Actions	
PM.GA.a1	Define and Maintain Requirements Development and Management Processes	1
PM.GA.a2	Introduce Tool Support for Requirements Development and Management	1
PM.CM	Configuration Management	
PM.CM.a1	Manage Versions of Requirements	1
PM.CM.a2	Baseline Requirements	1
PM.RT	Requirements Traceability Policy	
PM.RT.a1	Uniquely Identify each Requirement	1
PM.RT.a2	Document Requirements' Source	1
PM.RC	Requirements Communication and Negotiation	
PM.RC.a1	Establish Effective Communication With Requirements Issuers	1
RE	Requirements Elicitation	
RE.GA	General Actions	
RE.GA.a1	Elicit Quality Requirements	1
RE.SI	Stakeholder and Requirements Source Identification	
RE.SI.a1	Identify and Involve Relevant Stakeholders	1
RE.SI.a2	Distinguish between Different Types of Stakeholders	1
RE.SI.a3	Identify Other Requirements Sources	1
RE.DC	Domain Consideration and Knowledge	
RE.DC.a1	Elicit Information about System Domain Restrictions	1
RE.DC.a2	Elicit Information about System's Technical Infrastructure	1
RE.DC.a3	Elicit Information about System's Business Process	1
RE.DC.a4	Elicit Information about System's Operational Domain	1
RE.DC.a5	Elicit Information about System Boundaries	1
RA	Requirements Analysis	
RA.GA	General Actions	
RA.GA.a1	Perform Requirements Risk Analysis	1
RA.QA	Quality attributes analysis	
RA.QA.a1	Analyze for Missing and Double Requirements	1
RA.QA.a2	Analyze for Ambiguous Requirements	1
RA.QA.a3	Analyze for Correctness of Requirements	1

RA.QA.a4	Analyze for Testability of Requirements	1
RA.PS	Problems and solutions analysis	
RA.PS.a1	Create Prototype	1
RP	Release Planning	
RP.S	Requirements Selection	
RP.S.a1	Pack Requirements into Releases	1
DS	Documentation and Requirements Specification	
DS.GA	General Actions	
DS.GA.a1	Establish Standardized Structure for SRS	1
DS.GA.a2	Define Requirements Attributes	1
RV	Requirements Validation	
RV.GA	General Actions	
RV.GA.a1	Validate requirements with relevant stakeholders	1
RV.GA.a2	Use Checklist to Ensure Quality of Requirements	1

Level 2 - Intermediate		
OS	Organizational Support	
OS.GA	General Actions	
OS.GA.a2	Train personnel in Requirements Development and Management Processes	2
OS.RR	Roles and Responsibilities	
OS.RR.a2	Define Roles and Responsibilities for Requirements Development and Management Processes	2
OS.RR.a3	Define Roles and Responsibilities for Release Planning	2
OS.RR.a4	Define Roles and Responsibilities for Change Control	2
OS.S	Strategies	
OS.S.a1	Define Product Strategies	2
OS.S.a2	Define Product Roadmaps	2
PM	Requirements Process Management	
PM.GA	General Actions	
PM.GA.a3	Involve various perspectives in Requirement Development and Management Process	2
PM.CM	Configuration Management	
PM.CM.a3	Define a Process for Managing Change and Evolution	2
PM.CM.a4	Track change requests	2
PM.RT	Requirements Traceability Policy	
PM.RT.a3	Define traceability policies	2
PM.RT.a4	Document Requirements' Relation	2
PM.RT.a5	Document Impact of Requirement on Other Artifacts	2
RE	Requirements Elicitation	
RE.GA	General Actions	
RE.GA.a2	Qualify and Quantify Quality Requirements	2
RE.GA.a3	Let Business Concern Guide Focus of Elicitation	2
RE.GA.a4	Use Appropriate Elicitation Techniques according to Situation	2
RE.GA.a5	Use Artifacts to Facilitate Elicitation	2

RE.DC	Domain Consideration and Knowledge	
RE.DC.a6	Consider Sociopolitical Influences on Requirements Sources	2
RA	Requirements Analysis	
RA.GA	General Actions	
RA.GA.a2	Perform Systematic Requirements Prioritization at Project-level	2
RA.GA.a3	Analyze Requirements Relations	2
RA.GA.a4	Identify Irrelevant Requirements for Early Dismissal (in/out scope OR Triage)	2
RA.GA.a5	Analyze the Strength of Relations between Requirements	2
RP	Release Planning	
RP.GA	General Actions	
RP.GA.a1	Synchronize Release Plan with Product Roadmap	2
RP.GA.a2	Involve different perspectives in release planning	2
RP.S	Requirements Selection	
RP.S.a2	Estimate Cost and Value of Requirements	2
RP.S.a3	Perform Requirements Prioritization at Pre-project Level based on Various Dimensions	2
DS	Documentation and Requirements Specification	
DS.GA	General Actions	
DS.GA.a3	Define Requirements States	2
DS.GA.a4	Document Requirements Rationale	2
DS.DD	Documentation Deliverables	
DS.DD.a1	Define User Documentation Deliverables	2
DS.DD.a2	Define System Documentation Deliverables	2
RV	Requirements Validation	
RV.GA	General Actions	
RV.GA.a3	Review Requirements	2

Level 3 -Advanced		
OS	Organizational Support	
OS.RR	Roles and Responsibilities	
OS.RR.a5	Define Roles and Responsibilities for Product Management	3
OS.S	Strategies	
OS.S.a3	Communicate Strategies in Organization	3
PM	Requirements Process Management	
PM.RC	Requirements Communication and Negotiation	
PM.RC.a2	Obtain common understanding of requirements among different involving roles	3
RE	Requirements Elicitation	
RE.GA	General Actions	
RE.GA.a6	Create Elicitation Channels for Requirements Sources	3
RE.GA.a7	Reuse Requirements	3
RA	Requirements Analysis	
RA.GA	General Actions	
RA.GA.a6	Perform refinement and abstraction of requirements	3

RA.PS	Problems and solutions analysis	
RA.PS.a2	Perform Systems Modeling	3
RP	Release Planning	
RP.GA	General Actions	
RP.GA.a3	Post Requirement Selection Evaluation	3
RP.GA.a4	Plan multiple release at pre-defined interval	3
DS	Documentation and Requirements Specification	
DS.GA	General Actions	
DS.GA.a5	Record Rationale for Rejected Requirements	3
DS.DD	Documentation Deliverables	
DS.DD.a3	Define Management Documentation Deliverables	3
RV	Requirements Validation	
RV.GA	General Actions	
RV.GA.a4	Organize Inspections	3
RV.GA.a5	Develop Preliminary Test Case or User Manual	3
RV.GA.a6	Use System Model Paraphrasing for QA	3
RV.GA.a7	Define Acceptance Criteria and Acceptance Tests	3

Reference

- [1] A.C. Yeh, "Requirements Engineering Support Technique (REQUEST) A Market Driven Requirements Management Process," 1992, pp. 211-223.
- [2] B. Regnell, P. Beremark, and O. Eklundh, "A market-driven requirements engineering process: results from an industrial process improvement programme," UK: Springer-Verlag, 1998, pp. 121-9.
- [3] P. Carlshamre and B. Regnell, "Requirements lifecycle management and release planning in market-driven requirements engineering processes," Los Alamitos, CA, USA: IEEE Comput. Soc, 2000, pp. 961-5.
- [4] R. Wieringa and C. Ebert, "RE'03: Practical requirements engineering solutions," IEEE Software, vol. 21, 2004, pp. 16-17.
- [5] Samuel Fricker, Tony Gorschek, Carl Byman, Armin Schmidle, "Handshaking with Implementation Proposals: Negotiating Requirements Understanding," IEEE Software, vol. 27, no. 2, pp. 72-80, Mar./Apr. 2010, doi:10.1109/MS.2009.195
- [6] N.A.M. Maiden and G. Rugg, "ACRE: Selecting methods for requirements acquisition," Software Engineering Journal, vol. 11, 1996, pp. 183-192.
- [7] CMMI for Development, Version 1.2, CMMI-DEV v1.2, CMU/SEI-2006-TR-008, Technical Report, Software Engineering Institute, August 2006
- [8] Firesmith, "Prioritizing requirements," *Journal of Object Technology*, vol. 3, 2004, pp. 35-47.

- [9] P. Carlshamre, K. Sandahl, M. Lindvall, B. Regnell, and J. Natt och Dag, "An industrial survey of requirements interdependencies in software product release planning," Los Alamitos, CA, USA: IEEE Comput. Soc, 2000, pp. 84-91.
- [10] M. Khurum, K. Aslam, and T. Gorschek, "A method for early requirements triage and selection utilizing product strategies", Piscataway, NJ, USA: IEEE, 2008, pp. 97-104.
- [11] T. Gorschek and C. Wohlin, "Requirements abstraction model," *Requirements Engineering*, vol. 11, 2006, pp. 79-101.
- [12] Gorschek T., Tejle K., "A Method for Assessing Requirements Engineering Process Maturity in Software Projects", Blekinge Institute of Technology, Master Thesis Computer Science no. MSC-2002:2, 2002.
- [13] L. Karlsson and B. Regnell, "Introducing tool support for retrospective analysis of release planning decisions," Berlin, Germany: Springer-Verlag, 2006, pp. 19-33.
- [14] P. Sawyer, I. Sommerville, and G. Kotonya, "Improving market-driven RE processes," Espoo, Finland: Tech. Res. Centre Finland, 1999, pp. 222-36.