INFOTEH-JAHORINA Vol. 9, Ref. E-I-12, p. 469-472, March 2010.

MENADŽMENT PROJEKTNIH ZAHTEVA U SKLADU SA CMMI PRAKSAMA

PROJECT REQIREMENTS MANAGEMENT ACCORDING TO THE CMMI PRACTICES

Vladimir Kovačević, Mirjana Stojilović, *Institut Mihajlo Pupin, Beograad*

Sadržaj – *Jednu od primarnih aktivnosti svakog projekta razvoja softvera ili hardvera predstavlja menadžment projektnih zahteva. Za uspešan ishod takvog projekta neophodno je obezbediti njihovo potpuno i nedvosmisleno ispunjenje, odnosno implementaciju. Da bi se to postiglo potrebno je propisati striktne, ali istovremeno i neopterećujuće procedure, koje bi članovi projektnog tima sprovodili tokom celog projekta. CMMI model predlaže upravo takve ciljeve i prakse, a procedure opisane u ovom radu ih zadovoljavaju i nadograđuju.*

Abstract - *One of the primary tasks in every hardware or software development project is the management of its requirements. In order to proclaim the project as a success their entire and unambiguous fulfillment is demanded. This can be satisfied by introducing the strict but unladed procedures which must be followed by all stakeholders during the entire project lifecycle. CMMI is the model that proposes exactly that kind of the goals and practices. The procedures described in this paper satisfy and upgrade them.*

1. INTRODUCTION

CMMI (Capability Maturity Model Integration) [1] is a process improvement maturity model for the development of products and services. It consists of best practices that address development and maintenance activities covered in the entire product lifecycle, from conception through delivery and maintenance. The model was developed by the members of the American industry, Office of the Secretary of Defense and the Carnegie Mellon Software Engineering Institute (SEI) [2].

A requirement can be defined as a condition or capability to which a system must conform [3]. The Requirements management is the part of CMMI model (process area) [1]. It includes practices concerning receiving, controlling, assigning and tracing of the project requirements. The mandatory part for every development project (consisting of hardware, software or system design), is collection and implementation of its requirements. The procedures and algorithm for managing requirements described in this paper are developed according to the CMMI model and supplemented with experience from the real development projects.

2. REQUIREMENTS MANAGEMENT ROLES AND RESPONSABILITIES

Before describing requirements management processes, the list participating roles and their responsibilities must be defined:

1. Requirements Manager - the person in charge for receiving, defining, managing project requirements and maintaining the documents with requirements definitions (will be described in details in the following chapters).
2. Requirements Providers - the person or the list of persons eligible to issue a new requirement. This could be the client but also the team members qualified for issuing a project requirement.
3. Requirements Master - the person or the list of persons in charge for approving and accepting the requirements (usually from project's ordering party).
4. Requirements Review Team – the list of persons in charge for analyzing the requirements and confirming their dependency in order to estimate the impact on other requirements and overall project execution.

3. REQUIREMENTS ALGORYTHM

The purpose of this chapter is describing the algorithm for accepting and managing project requirements (see Figure 1). The algorithm is uniform and can be applied for issuing initial project requirements, adding new requirements and changing the existing requirements. All of the algorithm steps as well as the required conditions and resulting actions are described in the following text.

According to Requirements Lifecycle Algorithm, every requirement under management can be in one of the following states (statuses):

1. Raw – a requirement is entered in a raw form, as is received from Requirements Provider (optional)
2. Defined – a requirement and its features are clearly defined by the Requirements Manager
3. Analyzed – a requirement is analyzed by the Requirements Review Team
4. Approved – a requirement is approved by the Requirements Master
5. Implemented – a requirement is implemented by the Development Team
6. Verified - a requirement is verified by using adequate

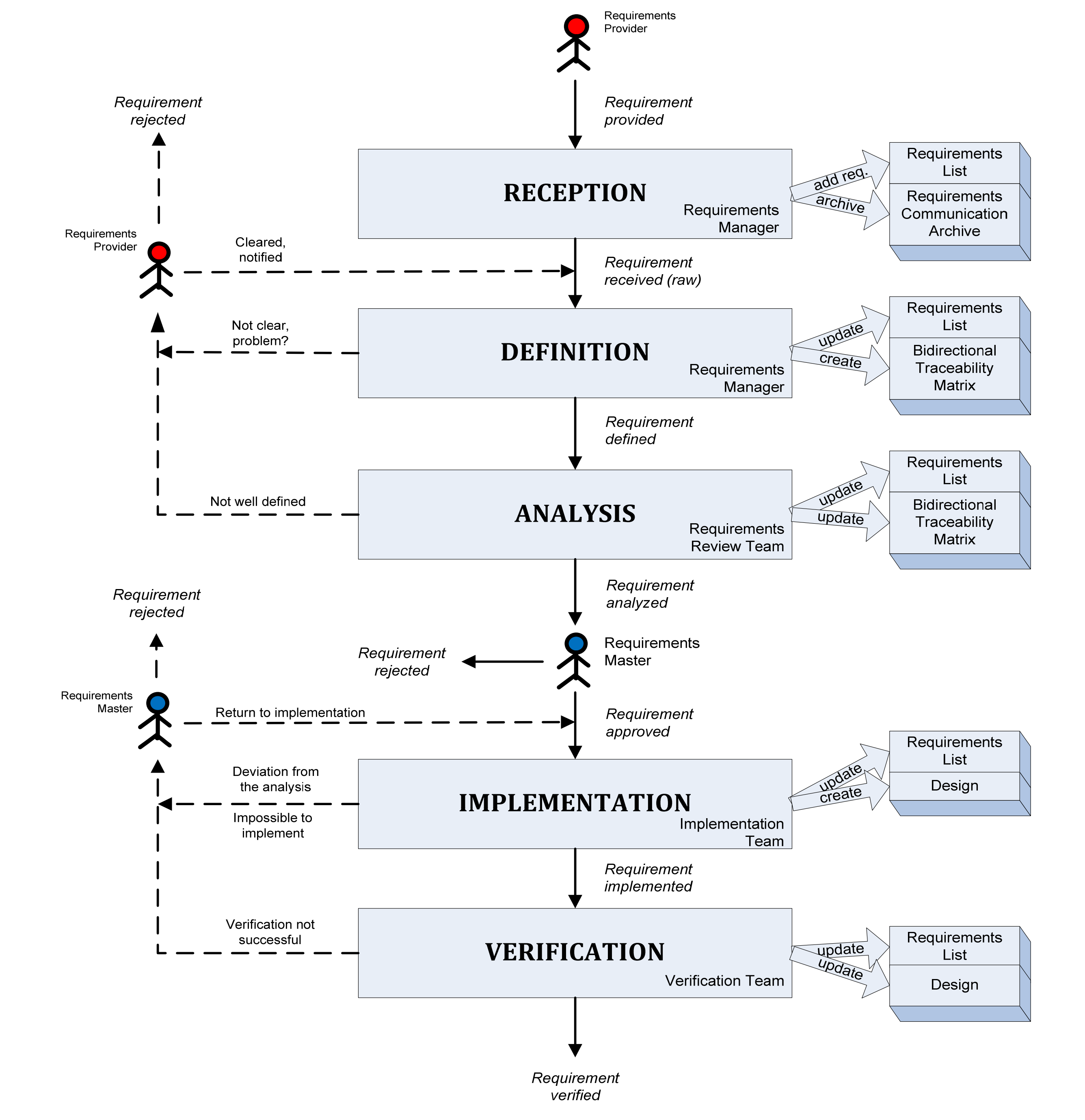

Figure 1: Requirements Lifecycle Algorithm

verification procedure

7. Rejected – a requirement is disapproved by Requirements Master

The features of each requirement are listed in the Requirements List Document [4]. Its main purpose is to help project team members to obtain the better understanding and control of the requirements that they are related to. Features include the following:

1. Requirement unique identifier
2. Date of receiving the requirement
3. Name of the source which issued the requirement
4. Description of the requirement
5. Status (state) of the requirement
6. Persons in charge for implementing the requirement
7. Customer imperative is the optional feature that can contain following values: optional (system can obtain full functionality without this requirement, but customer would be more satisfied if it is implemented), required (required system feature) and critical (requirement is critical for system functioning). Customer imperative is important requirement property because it can support decisions regarding requirement status and implementation.
8. Complexity is the optional feature that corresponds to the level of existing knowledge and skills needed for implementation of the requirement. Values can be following: low (knowledge and similar designs exist in the institution), medium (knowledge exist in the institution, but there is no similar designs) and high (inexistent or poor level of knowledge about the subject).
9. Optional Comment

The proposed actions and procedures that must be followed in order to satisfy unambiguous requirements implementation are described in the rest of this chapter, through all of the stages.

3.1 REQUIREMENT RECEPTION

The requirements can be received from official Requirements Provider in any form such as: e-mail, telephone, direct face-to-face conversation, fax, mail, documented contract.

On receiving new requirement (in any form) Requirements Manager opens the new row in the Requirements List Document for that project [4] and fills the following fields: Requirement ID, Date, Source, Description and Status (Raw).

3.2 REQUIREMENT DEFINITION AND TRACEBILITY

The purpose of Requirement Definition Phase is to obtain unanimous understanding of the requirement. In this phase, the Requirements Manager needs to perform the requirement definition in a way that is clear and understandable. If Requirements Manager decides that the received requirement is not understandable or clear enough, it must be returned to it's provider for redefinition.

Requirements Manager also enters preliminary requirement data that defines bidirectional traceability between all requirements and optionally between requirement and test case that will verify it. Basically, bidirectional traceability between the requirements is formed in a way that the requirements constitute the rows and the columns of Bidirectional Traceability Matrix [5]. By using such a matrix, requirements interaction can easily be seen and tracked. The measure of influence that one requirement has on other requirements is formed this way (as a number of requirements that are under its influence). Bidirectional traceability is a very powerful tool for supporting decisions for accepting, adding new and especially changing existing requirements (to see its correlation to the other requirements).

Requirements Manager can set status of the requirement to "Rejected" only after confirmation in written form from the Provider of that requirement or Requirements Master. Products of this stage include Project Requirements Document [4] with all defined fields (Status: Defined) and table with bidirectional traceability [5].

If Requirements Manager can't define some of the fields or dependencies it can be left out for the next stage. Requirements Manager must also organize the meeting of the Requirements Review Team.

3.3 REQUIREMENT ANALYSIS

It is recommended to include in Requirements Review Team all of the relevant Project Team members (stakeholders) that are in the charge of the main product components. For smaller projects even all developers should be included.

The purpose of Requirement Review is to analyze newly defined requirements as well as their impact on other requirements and project execution. It is carried out in a discussion manner where Requirements Manager is presenting newly defined requirements, their features and bidirectional traceability, while the rest of attendees present their comments and suggestions. Following cases can arise from the requirement analysis:

1) Requirement needs to be better defined or clarified.
2) Requirement affects project by raising the need for changing already accepted requirements or adding new requirements.
3) Accepting the requirement will cause significant deviation in project execution. In some cases, requirement can be characterized as "impossible to implement".
4) The requirement can be accepted, as it doesn't affect other requirements and doesn't significantly affect project execution.

3.4 REQUIREMNT APPROVAL

The formal approval (e.g. Documented Contract) from the Requirements Master is necessary in order to exclude the risk of misunderstanding in requirements' interpretation and future implementation. The actions proposed for this stage are dependent of the previous requirement analysis result.

In case 1) requirement is returned to previous stage, in which the Requirements Manager can try to better define it or can be send back to its provider for better definition.

In cases 2) and 3) Requirements Manager shall inform Requirements Master about the results of requirement analysis, along with proposed action. After taking in account all presented facts that Requirements Provider or Requirements Master can choose to reject the requirement, to redefine it, or to approve it. If Requirement Provider or Master decides to reject the requirement, Requirements Manager shall designate requirement as "Rejected" and terminate its processing. If the decision is to redefine requirement, Requirements Manager shall designate the requirement as "Raw" and put it back in Requirements Definition stage.

In case 3), Requirements Manager designates requirement as "Approved" after getting the formal approving in written form from Requirements Master. Requirement rejection or approval statement received from Requirements Master must be saved in the Requirements communication archive.

If Requirements review team finds that accepting the requirement induces adding of new requirements, these requirements are treated as new requirements.

Traceability of decisions regarding requirements must exist at all times. If requirement is rejected for some reason the rationale for such decision must be stated in Comment filed of Requirements List Document. If new requirements are added as a result of rejection or approval of some requirement, their Comment field must reflect that fact.

When the requirement is finally approved by Requirements Master, every change in its definition must initiate the rejecting of this requirement and entering new requirement into the process as "Raw" requirement.

After this stage all requirements features and trace abilities must be entered and verified.

3.5 REQUIREMENT IMPLEMENTATION

The Requirement Implementation stage can start only after the requirement is approved by Requirements Master (its status is "Approved"). If during this stage the Implementation Team finds that requirement is impossible to implement or that requirement deviates from the analysis, Requirements Manager must be notified to perform adequate actions.

Implementation Team must also inform Requirements Manager on successful implementation of requirement. Only after getting confirmation the requirement can be designated as "Implemented".

3.6 REQUIREMENT VERIFICATION

Requirement Verification can start only if the requirement is implemented. It is done by Verification Team. Verification is done by performing all test cases that are connected to the requirement. Requirements Manager must be informed about the results of the verification and if it is successful the requirement is designated as "Verified". It is a good practice to implement formal procedures for requirement validation [6].

4. CONCLUSION

The described procedures were implemented on real development projects in the Mihajlo Pupin Institute and verified through formal IT Mark Certification held by SEI. The IT Mark requires the company to have implemented CMMI Maturity Level 2 [7] which includes Requirements Management process area. The procedures described in this paper were highly marked during the certification audit.

The formalizing of the presented procedures enabled complete insight and control of the project requirements and ensured work product completeness with higher time-to-effort ratio and minimized risks. Moreover, it brought less stressful environment for all the employees, especially when working with short and consistent deadlines.

REFERENCES

[1] CMMI Product Team, *CMMI® for Development, Version 1.2*, Pittsburgh, PA 15213-3890, August 2006

[2] Carnegie Mellon Software Engineering Institute, www.sei.cmu.edu, Pittsburgh, USA, founded in 1984.

[3] Requirements Management Using IBM® Rational® RequisitePro®, Chapter 1, Peter Zielczynski, Jun, 2008.

[4] Vladimir Kovacevic, *Requirements List Document Template, http://dl.dropbox.com/u/2594994/CMMI/RM/CMMI_RM _template.xls*, Belgrade, Institute Mihajlo Pupin 2009.

[5] Vladimir Kovacevic, *Requirements Bidirectional Traceability Matrix Template, http://dl.dropbox.com/u/2594994/CMMI/RM/CMMI_RM _template.xls*, Belgrade, Institute Mihajlo Pupin 2009.

[6] Dines Bjøner, Software Engineering 3 - Chapter Requirements Verification and Validation, Springer Berlin Heidelberg, June 29, 2006

[7] Carnegie Mellon Software Engineering Institute, CMMI Maturity Levels, www.tutorialspoint.com/ cmmi/cmmi-maturity-levels.htm