The Software Requirements Management Practices: A Review of CMMI Level 2 as a Solution

Enas Alzidi, Master [1]   Loai Alamro Master [2]   Ahmed Talib Abdulameer, PhD [3]

Computer Science dept., Al-Turath University College

[3] IT dept., Technical college of management, Middle Technical University

Abstract

Requirements engineering is a very important software engineering process which focuses on the development and management of different user requirements. Requirements management is a significant aspect of the requirements engineering process, which plays a vital role in supporting product development. However, though requirements management is considered to be an important process, a majority of the software process development projects do not implement it. The software markets face many problems related to requirements management activities like identifying user requirements, inadequate requirements, and the quality of these requirements, which limits the studies addressing these issues. In this study, the researchers explored the different software issues which cause software failure. Furthermore, they also explored the various requirements management activities that reduce software failure during the developmental processes. This study investigated the current requirements management processes and presented CMMI level 2 as a solution for software failure. In this empirical study, the researchers have presented many theoretical implications and contributed to the fields of software engineering and requirements management. This study consists of valuable information that can be used for implementing the requirements management practices.

المستخلص

هندسـة المتطلبـات هـي عمليـة رئيسية فـي هندسـة البرمجيـات والتـي تركـز علـى تطـوير وإدارة متطلبـات المستخدم. واحدة من الأنشطة الرئيسية لهندسة المتطلبـات هي إدارة المتطلبـات، وهي تلعب دورا هلما عندما يتعلق الأمـر بـدعم فرق تطـوير البرمجيـات. وعلـى الـرغم مـن ذلـك، هنـاك نقص فـي المـمارسـة (استخدام) نشاطات إدارة المتطلبات في تطوير المشاريع البرمجية. ولذلك، لا تزال أسواق البرمجيات تواجه عدة مشاكل

211

في ممارسات إدارة المتطلبات مثل جودة المتطلبات، عدم توافق المتطلبات، وتحديد الاحتياجات؛ مع عدم وجود دراسات تتناول الموضوع بشكل كافي. هذه الدراسة تستعرض مشاكل البرمجيات التي تؤدي إلى فشلها. بالإضافة الى ذلك، تستعرض الدراسة ممارسات إدارة المتطلبات التي تستخدم للحد من فشل البرمجيات أثناء عملية التطوير. وعلاوة على ذلك، تستقصي في الوضع الحالي لإدارة متطلبات وتصف بتعمق المستوى الثاني من (CMMI) باعتباره أحد الحلول المستخدمة للتخلص من مشكلة فشل تطوير البرمجيات. وسوف تشمل هذه الدراسة العديد من الآثار النظرية والمساهمات في هندسة البرمجيات وإدارة المتطلبات. وبصورة أدق، ستشمل هذه الدراسة معلومات قيمة عن إدارة المتطلبات والممارسات المنفذة.

1. Introduction

Currently, the software engineering is one of the most rapidly growing industries in the world, owing to the significant rise in the software applications. Any system is developed based on its standards, need for application and finally, the company circumstances.

Software engineering refers to the application of a quantifiable and systematic approach for the creation and maintenance of any software product [1-3]. This field has to face several challenges for preserving and supporting the complexities in the field. Though software development is carried out using many different approaches, many studies have criticised its management practices.

Requirements Engineering (RE) refers to the management and development of the user requirements in the field of software engineering [4]. The development of any software application is based on the success of the RE. Requirement Management (RM) relies on the management of the user requirements compared to the overall software development. In other words, RM is an essential component of the RE process [5]. Zainol [6] stated that most of the RM activities carried out during the software development go unnoticed. A similar argument was presented by [7]. In one study, Sommerville [8] specified that the RM must guarantee the development of good-quality software products and must be favourably considered during the software development. Furthermore, it was seen that the quality and the specification of the software requirements gets compromised, which leads to the incorporation of the RE practices into the RM activities. This is carried out by defining the best available practices and techniques.

In the past few years, many researchers have presented good techniques and tools which can deliver improved processes [8, 9]. Furthermore, many studies have also verified that incorporating the RE processes and techniques can significantly affect the software quality and the general project cost, in comparison to the other RE processes which are not properly defined. Here, the researchers have described the advantages in exploring the present RM practices that could be implemented in the Software Development (SD) projects.

Recently, there has been an increasing interest in the software systems which adapt the modifications in the user requirements for maintaining a continuity and satisfying the software-based objectives [10]. Several researchers have considered RE to be pertinent for SD [11-13].

The requirements management and requirement development are important in the RE activities when any organisation wishes to adopt the best techniques in their procedures. Shubhamangala, et al. [14] mentioned that the software projects usually failed because of the implementation of poor RM/ RE practices. Similar thoughts were also expressed by Pandey and Suman [15]. Rikhari and Kumar [16] stated that poor RM and RE developmental practices lead to high costs which could cause financial problems, bad software product quality, poor customer satisfaction, waste of time and efforts, delayed product delivery and a poor market value. These factors, in turn, lead to market failure.

RM has to face many challenges during project development. In their study, Basili, et al. [17], mentioned that requirement quality was the major problem which affected the RM. Mu, et al. [18] observed that despite the issues related to the requirement quality, many other problems affect the RM processes like vague requirement statements, incomplete requirement specification and the redundant requirement statements. However, Pohl [19] stated that many of the software products ineffectively managed their user requirements by collecting and storing the necessary requirements in the spreadsheets. Different requirements are usually stored separately in different media, which makes their access difficult. Furthermore, a majority of this important metadata goes missing or is misplaced based on its priority, status, type, etc. [20]. Many earlier studies focused on the RM activities in the private sector [21-23], while less attention was paid to the public sector, particularly, in the teaching institutions like universities. Abdulrahman [24] stated that the application of good practices in the computer centres in the software markets was essential as it helped the developers to develop a good software product.

Therefore, based on the earlier debate, in this study, the researchers investigated the decisive problems affecting the adoption of the proper RM processes by the software market. Furthermore, this study also explored the relationship between the RM practices and the CMMI level 2, which was an important tool that aides in the success of the RM practices. Samalikova, et al. [25] and Mahmood, et al. [26] stated that many models could be used for improving the processes and practices like the Capability Maturity Model Integration (CMMI).

Literature Review

Recently, the application software models have gained a lot of importance as they support the daily human activities. Some of these services and products include washing machines, car components, academics, healthcare services, and other operational processes. Many studies stated that software development was a complicated process and the final product must satisfy all the daily user requirements. Here, the researchers presented a literature review of the RM processes and discussed the various dimensions of the RE processes in Section 2.1. Section 2.2 presents the requirement development and important activities related to user requirements. In Section 2.3, the researchers highlighted the RM activities while in Section 2.4; they investigated the different maturity model integrations. Section 2.5 discusses the earlier studies related to the RM activities.

213

Requirements Engineering

Software development comprises of many data-intensive activities, which are grouped together and are defined as Requirement Engineering (RE), which also involves the stakeholders. The problems related to RE are investigated after defining the goals [27]. RE involves a critical decision-making process wherein the business objectives are transformed into technical specifications. Furthermore, in this process, the conflicting perspectives are also negotiated according to a pre-defined action plan and the reasoning is revised without sustaining any serious budget or time overturns. Also, the requirements are controlled and a disciplined approach is required for handling the heterogeneous, dynamic and huge information volume [28].

RE is the initial stage of the Systems Engineering (SE) process, which consists of the general requirements that define the application systems at each successive abstraction level. In this phase of the SE, the software engineer analyses and documents the various system requirements methodically and implements the development of the system application [29]. The RE activities incorporated the complete system life cycle. RE is an iterative and incremental process that indicates that RM is an important aspect of the RE process [8, 30, 31]. Generally, RE is carried out prior to the development of any system lifecycle [32]. Studies have shown that the development of any sophisticated system and the stabilisation of the software requirements usually lasts for many months and even years [31]. As a result, RE is performed along with other system developmental activities like the coding and designing of software programs.

On the other hand, several studies have recommended using improved RE activities for resolving the challenges occurring in the systems development program [33]. Many organisations have executed all necessary changes in their system acquisition practices, and this assists the SE processes to decrease the negative effects that could influence the requirements volatility of the software programs. Wiegers and Beatty [34] classified the software RE process into 2 categories, i.e., Requirement Development (RD) and Requirements Management (RM). Furthermore, these domains can be subdivided into 4 other subsections described in Figure 1.
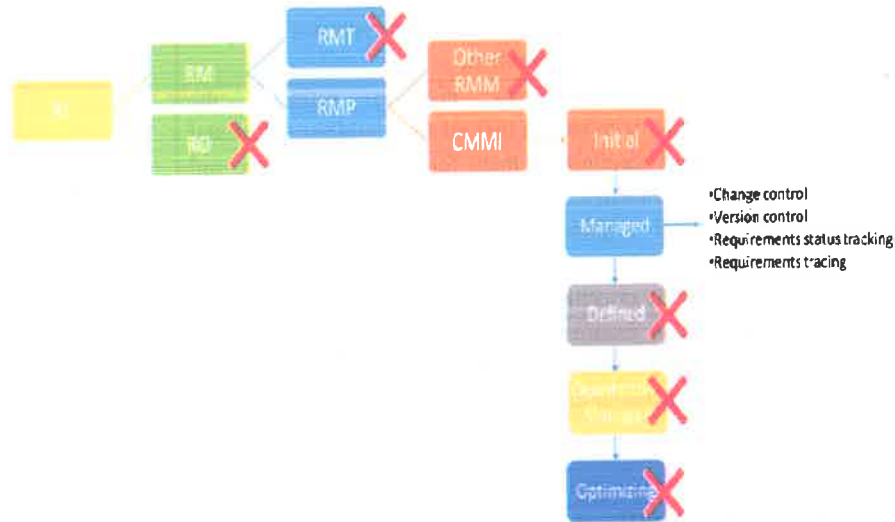
Figure 1: Requirements engineering sub-domains (source: Wiegers and Beatty [34]).

Requirements Development

Requirement Development (RD) of a process is generally executed along with the RM process. Rosenkranz [35] stated that the RD is associated with the best activities that define and document the customer and product requirements using Level 3 processes. This elicits the customer and product requirements and also analysing and validating the activities of the required products. All components of the RD process are described in Table 1.

Table 1: *Requirement development's elements*

REQUIREMENTS DEVELOPMENT	SUMMARY	SOURCES
Requirements Specification	The requirements specification refers to the document requirements of different types in a consistent, accessible, and reviewable way that is readily understandable by the intended audiences.	Wieringa [36]
Requirements Elicitation	The requirements elicitation is the practice of collecting the requirements of a system from users, customers and other stakeholders. Elicitation techniques include both facilitated activities, in which	Wiegers and Beatty [34]

215

	you interact with stakeholders to elicit requirements, and independent activities, in which you work on your own to discover information.	
Requirements Analysis	Requirements analysis involves refining the requirements to ensure that all stakeholders understand them and scrutinizing them for errors, omissions, and other deficiencies. Analysis includes decomposing high-level requirements into appropriate levels of detail, building prototypes, evaluating feasibility, and negotiating priorities.	Pohl [19]
Requirements Validation	Validation ensures that the requirements are correct, demonstrate the desired quality characteristics, and will satisfy customer needs.	Lee, et al. [37]

Requirements Management

The RM process plays a vital role in the development of the software systems. It is the primary step in the SE process and has the potential to affect the direction of the subsequent project activities. Earlier studies have presented a broad definition of the RM process [8, 38, 39].

Dutoit and Paech [40] defined RM as a systematic process used for managing and storing the relevant and important information about the user requirements, while certifying the requirement traceability, and also managing all the changes in the requirements that occur during the lifecycle of this information system. Similar to other management disciplines, the RM process has a simple principle of establishing and meeting expectations. However, the process gets complicated when executed in real-world applications. Table 2 presents a list of tasks related to an effective RM process [41].

Table 2: *List of tasks associated with requirements management*

NUMBER	TASKS
1	Define and communicate what is wanted.
2	Provide traceability to outside documents.
3	Apply requirements to the solution.
4	Optimize the product before commitment.

5	Drive the design and the implementation.
6	Manage change, problem reports, and suggestions.
7	Manage the partitioning of work to specialists.
8	Test and validate the finished product.
9	Control iterative developments.
10	Manage project milestones.
11	Manage interfaces with external systems.

As shown in the table, the RM process is broad-based. Furthermore, it also influences every property of the software project development and includes every individual involved in the developmental process, from the customers to the testers.

Requirements Management Activities

The RM process encompasses all the activities that can maintain accuracy, integrity and the popularity of the requirement agreements through the software project. Wiegers and Beatty [34] described 4 main activities which explained the role played by the RM process in the software development, i.e., version control, change control, requirement status tracking, and requirement tracing. Figure 2 describes each of these activities.
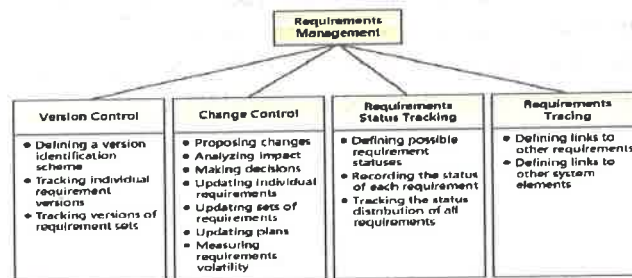

Figure 2: The Main Activities of *Requirements Management Source: Wiegers and Beatty [34].*

Capability Maturity Model Integration (CMMI)

Many studies have described different models and guidelines which improve the business of the organisations. The most effective approaches concentrate on one particular area and include a

217

systematic technique for solving all the problems affecting the organisation. The CMMI models help to integrate the models by excelling in the particular areas and also further improving them. These models consist of the best practices that could be implemented in the software organisations. This is carried out by addressing the ideas and the delivery of the system products and ensuring their maintenance. This process improves the CMMs. Both the systems and the software engineering processes have to be properly integrated for developing and maintaining the system products [42]. The CMMI model can be represented in 2 different forms, i.e., the Staged and Continuous forms [43]. Regnell, et al. [44] stated that the huge size of the software systems has increased the significance of the RM process in the organisation of the developer's process. Hence, in this study, the researchers have focused on the Level 2 of the RM process, described in the next section.
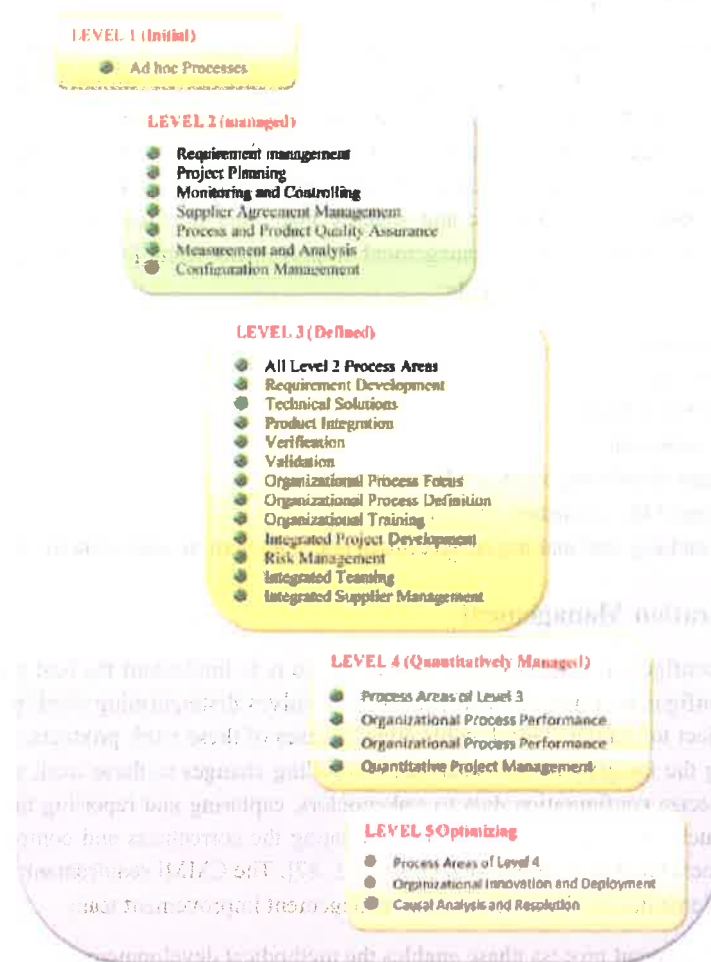
CMMI level 2

The Staged CMMI model comprises of 5 different maturity levels, described in Figure 3, where every layer in the improvement of the current process was designed from Level 1 to 5 [45]. At Level 2, the researchers ensured the project requirements using processes which were pre-planned, measured, executed, managed and controlled. Such processes helped in retaining the current project practices even during stressful periods.

Software Project Planning

The phase where the implementation methodology of CMMI level 2 is elaborated is known as the project plan. In this phase, the resources and the timelines necessary for setting up and imparting training to the different improvement teams are estimated and scheduled [45]. The project plan further includes the introduction of new processes and standards, definition of procedures and processes involved, training materials, tools, pilots, and checklists [45].

218

LEVEL 1 (Initial)
- Ad hoc Processes

LEVEL 2 (managed)
- Requirement management
- Project Planning
- Monitoring and Controlling
- Supplier Agreement Management
- Process and Product Quality Assurance
- Measurement and Analysis
- Configuration Management

LEVEL 3 (Defined)
- All Level 2 Process Areas
- Requirement Development
- Technical Solutions
- Product Integration
- Verification
- Validation
- Organizational Process Focus
- Organizational Process Definition
- Organizational Training
- Integrated Project Development
- Risk Management
- Integrated Teaming
- Integrated Supplier Management

LEVEL 4 (Quantitatively Managed)
- Process Areas of Level 3
- Organizational Process Performance
- Organizational Process Performance
- Quantitative Project Management

LEVEL 5 Optimizing
- Process Areas of Level 4
- Organizational Innovation and Deployment
- Causal Analysis and Resolution

Figure 3: CMMI five levels with respective process areas

Monitoring and Controlling

The objective of the project monitoring and control process is to supervise and monitor the project execution and implement corrective actions when the project performance falls short of the expectations. It is a norm to track the progress of the project against the project plan and to implement corrective actions when the progress of the project deviates from expectations [45, 46]. This also includes monitoring the major project performance parameters such as effort, schedule, and budget.

219

Supplier Agreement Management

This process phase plays an important role in managing the acquisition of products from suppliers for cases where there is a formal agreement. The process of supplier agreement management involves establishing best practices for satiating supplier agreements which includes setting up standards and checklists to choose suppliers, creating definitions for agreements with the suppliers, and defining rules for executing the agreements and accepting the suppliers' product. The implementation of the generic and specific goals and practices for this process phase is a responsibility of the supplier management improvement team [45]. The activities related with supplier selection generally include:

Identifying potential suppliers
Forming an evaluation team
Setting up of an evaluation criteria
Issuing of Request for Proposals
Receiving proposals, and shortlisting suppliers for demonstration
Assessing suppliers against the evaluation criteria
Negotiating, decision-making, and making an agreement and statement of work with the supplier

Software Configuration Management

The objective of the configuration management process phase is to implement the best practices for setting up of a configuration management system. It involves distinguishing work products that require to be subject to change control, controlling releases of these work products, creating baselines, maintaining the integrity of the baselines, controlling changes to these work products over time, offering precise configuration data to stakeholders, capturing and reporting the status of configuration products and change requests, and validating the correctness and completeness of configuration products through configuration audits [45, 47]. The CMMI requirements for this process phase are implemented by the configuration management improvement team.

The configuration management process phase enables the methodical development of software. It confirms that the effects of the suggested changes are taken into consideration prior to authorisation. It makes sure that releases are planned in advance and only authorised changes to the software are allowed [47]. The components of the software system and their version numbers are known at all times, and the integrity of the system is maintained.

Process and Product Quality Assurance

This phase focuses on the work products being produced by the projects, and provides visibility to the management on the processes being followed. According to O'Regan [45], the objective of this process phase is to implement the best practices to plan and conduct audits. It involves

planning and conducting audits for quality assurance, delegating audit actions to groups or individuals to tackle detected non-compliance issues, monitoring the audit actions to completion, and documenting and reporting the outcomes to the management and stakeholders. The implementation of this process phase that provides product quality assurance may be carried out by the first teams that are set up in the initiative. However, it is necessary for the CMMI project manager to have a distinct view on how this independent function should be executed.

Measurement and Analysis

The best practice on process implementation includes analysis of measurement information. This measurement analysis involves the need for management information, specification of measures and implementation, identification of the objectives of measurement, and the analysis of the results. The project begins to collect and analyse the measurement data and stores and communicates the results once the measurement objectives, data collection, measures, storage, and analysis processes have been clearly defined. The characteristic or standard deliverables for this process phase include templates, guidelines, a checklist, and a process map. An abstract summary of the activities involved and the guidelines and procedures are provided in the process map.

Related studies

Displayed Several studies were conducted on the requirements management in various disciplines. Table 3 summaries these studies.

Table 3: *Summaries the prior studies related to Requirements management in general and the practices in specific*

Author (s)	Year	Objective	Method (s)	Outcome
Zainol [6]	2008	To gain an insight into the extent to which the requirements management practices have been adopted by the organizations in Malaysia	Quantitative method (questionnaire)	This study shows that, the software industry is lacking of employing good practices in managing requirements.
Solemon, et al. [48]	2010	To identify patterns of the practices of the requirements engineering for some software development companies in Malaysia.	Quantitative Method (Questionnaires)	The overall adoption of the practices in the requirements engineering in these companies is strong. Nevertheless, their study also indicted that, fewer companies use

221

				appropriate software or tools to support their requirements engineering.
Gülke, et al. [49]	2012	To understanding the relationship between the requirements management and costs in Automotive Development Projects	Qualitative Method (Documentations)	The result shows that new requirements or changes on the existing requirements lead to three different types of costs: (1) Investment Costs, (2) Direct Costs, and (3) Overhead/ Indirect Costs.
Yu and Shen [50]	2013	To focus on the practices of projects constructed under traditional procurement system.	Qualitative method (Interviews)	They identified that, the processes and limitations of current practice included lack of practical framework, misinterpretation of requirements, difficulties in identifying requirements, conflicts between expectation and constraints, complex hierarchy of client's organization and communication problems in eliciting client requirements.
Shuhud, et al. [51]	2013	To examine the practices of requirements elicitation activities and understanding of how different stakeholders collaborate in the requirements elicitation process	Qualitative Method (Documentation and Case study)	This study found, the activities should be shared among other stakeholders to establish a common understanding of the requirements and that social software has the potential to support this.
Violante and Vezzetti [52]	2014	The main purpose of this study is develop a user-based strategy based on Kano Methodology to define a structured set of guidelines to support the design of the features of an integrated PLM requirement management tool.	Quantitative Method (Questionnaire)	The results obtained show that if some features are included in enhanced RM tools, the user satisfaction increases. While, there are several features decrease the user dissatisfaction.
Khankaew and	2014	To investigate the current	Qualitative Method	The results show that, the

222

		state of RE problems and practices amongst small and medium software companies in Thailand	(Interview)	companies in Thailand encounter common problems such as, clarity, correctness, competiveness, change management and customer communication.
Riddle [7]				
Kakeshita and Yamashita [53]	2015	Propose tool to assist education of the basics of requirement management based on requirements engineering body of knowledge (REBOK)	Qualitative Method (comparative cases)	The results indicated that, the proposed tool (called REMEST) was simple and easy to use. In addition, it is useful for beginner of requirement management.

Conclusion

The current study aims to investigate the current situation for the requirements management practices in software markets. The current study illustrated the situation of requirements management in the software markets based on the related studies. The concludes, that there is a weakness point in requirements management development which led to software failure. This study revealed that hard to identified appropriate approach of managing requirements to the software markets; because of lack of using best requirements management practices between the developer.

There are different models can use helps to improve the requirements management activities, such as CMMI. The study presented CMMI level 2 as a common model used to improve the requirements management activities, thus reducing the failure of software development. The final conclusion of the study on requirements management activities, was that there were two main recommendations should be mentioned: 1. The software developers needs to awareness and truing about the importance of using requirements management activities through holding seminars and workshop, and so on; 2. The software developers should more pay attention the use of successful requirements management activities model, such as CMMI.

REFERENCES

[1]   R. De Lemos, H. Giese, H. A. Müller, M. Shaw, J. Andersson, M. Litoiu, *et al.*, "Software engineering for self-adaptive systems: A second research roadmap," in *Software Engineering for Self-Adaptive Systems II*, ed: Springer, 2013, pp. 1-32.

[2]   B. Fitzgerald and K.-J. Stol, "Continuous software engineering and beyond: trends and challenges," in *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*, 2014, pp. 1-9.

[3]   D. Šmite, C. Wohlin, Z. Galviņa, and R. Prikladnicki, "An empirically based terminology and taxonomy for global software engineering," *Empirical Software Engineering*, vol. 19, pp. 105-153, 2014.

[4]   P. F. Katina, C. B. Keating, and M. J. Ra'ed, "System requirements engineering in complex situations," *Requirements Engineering*, vol. 19, pp. 45-62, 2014.

[5]   M. Shahid, S. Ibrahim, and M. N. r. Mahrin, "An Evaluation of Requirements Management and Traceability Tools," *World Academy of Science, Engineering and Technology, WASET*, 2011.

[6]   A. Zainol, "Investigation into requirements management practices in the Malaysian software industry," in *Computer science and software engineering, 2008 international conference on*, 2008, pp. 292-295.

[7]   S. Khankaew and S. Riddle, "A review of practice and problems in requirements engineering in small and medium software enterprises in Thailand," in *2014 IEEE 4th International Workshop on Empirical Requirements Engineering (EmpiRE)*, 2014, pp. 1-8.

[8]   I. Sommerville, *Software engineering*, 9th ed. ed. Boston :: Pearson, 2011.

[9]   J. Horkoff and E. Yu, "Comparison and evaluation of goal-oriented satisfaction analysis techniques," *Requirements Engineering*, vol. 18, pp. 199-222, 2013.

[10]   V. E. Silva Souza, A. Lapouchnian, W. N. Robinson, and J. Mylopoulos, "Awareness requirements for adaptive systems," in *Proceedings of the 6th international symposium on Software engineering for adaptive and self-managing systems*, 2011, pp. 60-69.

[11]   A. Quispe, M. Marques, L. Silvestre, S. F. Ochoa, and R. Robbes, "Requirements engineering practices in very small software enterprises: A diagnostic study," in *Chilean Computer Science Society (SCCC), 2010 XXIX International Conference of the*, 2010, pp. 81-87.

[12]   M. Harman, P. McMinn, J. T. De Souza, and S. Yoo, "Search based software engineering: Techniques, taxonomy, tutorial," in *Empirical software engineering and verification*, ed: Springer, 2012, pp. 1-59.

[13]   M. Kassab, C. Neill, and P. Laplante, "Software Development Practices and Software Quality: A Survey," *Software Quality Professional*, vol. 16, 2014.

[14]   B. Shubhamangala, L. M. Rao, A. Dakshinamurthy, and C. G. L. Singh, "Ability based domain specific training: a pragmatic solution to poor requirement engineering in CMM level 5 companies," in *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*, 2012, pp. 459-464.

[15]   D. Pandey and U. Suman, "The Usefulness of RE Practices in S/W Development," *Journal of Global Research in Computer Science*, vol. 3, pp. 56-69, 2012.

[16]   P. Rikhari and A. Kumar, "Validating the Change Management Process for Managing Changing Requirements with the help of a Real Life Project," *International Journal of Computer Applications,* vol. 58, 2012.

[17]   V. R. Basili, J. Heidrich, M. Lindvall, J. Münch, M. Regardie, D. Rombach, *et al.*, "Linking software development and business strategy through measurement," *arXiv preprint arXiv:1311.6224,* 2013.

[18]   K. Mu, J. Hong, Z. Jin, and W. Liu, "From inconsistency handling to non-canonical requirements management: A logical perspective," *International Journal of Approximate Reasoning,* vol. 54, pp. 109-131, 2013.

[19]   K. Pohl, "The three dimensions of requirements engineering," in *Seminal Contributions to Information Systems Engineering,* ed: Springer, 2013, pp. 63-80.

[20]   W. Li, D. Brown, J. H. Hayes, and M. Truszczynski, "Answer-set programming in requirements engineering," in *International Working Conference on Requirements Engineering: Foundation for Software Quality,* 2014, pp. 168-183.

[21]   Z. Nawaz, S. Aftab, and F. Anwer, "Software Process Model;Agile Modeling;Customized Feature Driven Development Mode," *International Journal of Modern Education and Computer Science (IJMECS),* vol. 9, pp. 53-59, 2017.

[22]   M. Othman, A. M. Zain, and A. R. Hamdan, "A Review on Project Management and Issues Surrounding Dynamic Development Environment of ICT project: Formation of Research Area," *JDCTA,* vol. 4, pp. 96-105, 2010.

[23]   A. A. Rahman, A. Haron, S. Sahibuddin, and M. Harun, "An Empirical Study of the Software Project Requirements Engineering Practice in Malaysia Public Sector-A Perspective from the Stakeholders' Challenges," *International Journal of Computer Theory and Engineering,* vol. 6, p. 52, 2014.

[24]   R. A. Abdulrahman, "The Investigation on the Best Practices of Extreme Programming (Xp) Quality Implementation at UUM IT," 2015.

[25]   J. Samalikova, R. J. Kusters, J. J. Trienekens, and A. Weijters, "Process mining support for Capability Maturity Model Integration-based software process assessment, in principle and in practice," *Journal of Software: Evolution and Process,* vol. 26, pp. 714-728, 2014.

[26]   M. N. Mahmood, S. P. Dhakal, A. Wiewiora, R. Keast, and K. Brown, "Towards an integrated maturity model of asset management capabilities," in *Proceedings of the 7th World Congress on Engineering Asset Management (WCEAM 2012),* 2015, pp. 431-441.

[27]   O. C. Gotel, F. T. Marchese, and S. J. Morris, "The potential for synergy between information visualization and software engineering visualization," in *2008 12th International Conference Information Visualisation,* 2008, pp. 547-552.

[28]   S. Reddivari, "Visual analytics for software requirements engineering," in *2013 21st IEEE International Requirements Engineering Conference (RE),* 2013, pp. 389-392.

[29]   S. Asghar and M. Umar, "Requirement engineering challenges in development of software applications and selection of customer-off-the-shelf (COTS) components," *International Journal of Software Engineering (IJSE),* vol. 1, p. 32, 2010.

[30]   K. Pohl, *Requirements engineering: fundamentals, principles, and techniques:* Springer Publishing Company, Incorporated, 2010.

[31]   S. Robertson and J. Robertson, *Mastering the requirements process: Getting requirements right*: Addison-wesley, 2012.

[32]   R. Thayer and M. Dorfman, "System and Software Requirements Engineering,(two volumes)," ed: IEEE Computer Society Press, 1990.

[33]   M. W. Grenn, "A Theory of Information Quality and a Framework for Its Implementation in the Requirements Engineering Process," *ProQuest LLC*, 2013.

[34]   K. Wiegers and J. Beatty, *Software requirements*: Pearson Education, 2013.

[35]   C. Rosenkranz, "The Emergence of Shared Understanding: Applying Functional Pragmatics to Study the Requirements Development Process," *Information Systems Journal*, 2012.

[36]   R. J. Wieringa, "Requirements Specification," in *Design Science Methodology for Information Systems and Software Engineering*, ed: Springer, 2014, pp. 51-57.

[37]   K.-H. Lee, P.-G. Min, J.-H. Cho, and D.-J. Lim, "Model-driven requirements validation for automotive embedded software using UML," in *Computing Technology and Information Management (ICCM), 2012 8th International Conference on*, 2012, pp. 46-50.

[38]   D. Leffingwell and D. Widrig, *Managing software requirements: a unified approach*: Addison-Wesley Professional, 2000.

[39]   A. Karim Jallow, P. Demian, A. N. Baldwin, and C. Anumba, "An empirical study of the complexity of requirements management in construction projects," *Engineering, Construction and Architectural Management*, vol. 21, pp. 505-531, 2014.

[40]   A. H. Dutoit and B. Paech, "Supporting evolution: Using rationale in use case driven software development," in *Sixth International Workshop on Requirements Engineering: Foundation for Software Quality" Stockholm*, 2000.

[41]   J. R. Persse and D. J. Persse, *Implementing the capability maturity model*: John Wiley & Sons, Inc., 2001.

[42]   R. C. Gutiérrez and H. F. A. Jiménez, "QualDev process: procesos adaptables de desarrollo de software para proyectos ágiles," *Revista Ingeniería y Competitividad*, vol. 6, pp. 93-104, 2011.

[43]   M. Yadav and K. Kumar, "CMM or CMMI, which is more appropriate for a software industry?," *International Journal in IT & Engineering*, vol. 2, pp. 24-33, 2014.

[44]   B. Regnell, R. B. Svensson, and K. Wnuk, "Can we beat the complexity of very large-scale requirements engineering?," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*, 2008, pp. 123-128.

[45]   G. O'Regan, *Introduction to software process improvement*: Springer Science & Business Media, 2010.

[46]   M. B. Chrissis, M. Konrad, and S. Shrum, *CMMI for development: guidelines for process integration and product improvement*: Pearson Education, 2011.

[47]   C. Kothapalli, S. Ganesh, H. K. Singh, D. Radhika, T. Rajaram, K. Ravikanth, *et al.*, "Continual monitoring of code quality," in *Proceedings of the 4th India Software Engineering Conference*, 2011, pp. 175-184.

[48]   B. Solemon, S. Sahibuddin, and A. A. A. Ghani, "Adoption of requirements engineering practices in Malaysian software development companies," in *International Conference on Advanced Software Engineering and Its Applications*, 2010, pp. 141-150.

[49]   T. Gülke, B. Rumpe, M. Jansen, and J. Axmann, "High-Level requirements management and complexity costs in automotive development projects: a problem statement," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*, 2012, pp. 94-100.

[50]   A. T. Yu and G. Q. Shen, "Problems and solutions of requirements management for construction projects under the traditional procurement systems," *Facilities*, vol. 31, pp. 223-237, 2013.

[51]   M. I. M. Shuhud, A. Richter, and A. Ahmad, "Supporting Requirements Elicitation Practices," in *International Conference on Collaboration and Technology*, 2013, pp. 306-321.

[52]   M. G. Violante and E. Vezzetti, "A methodology for supporting requirement management tools (RMt) design in the PLM scenario: An user-based strategy," *Computers in industry*, vol. 65, pp. 1065-1075, 2014.

[53]   T. Kakeshita and S. Yamashita, "A Requirement Management Education Support Tool for Requirement Elicitation Process of REBOK," in *Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence (ACIT-CSI), 2015 3rd International Conference on*, 2015, pp. 40-45.