

Quantitative CMMI Assessment for Offshoring Through the Analysis of Project Management Repositories

Thanwadee Sunetnanta¹, Ni-On Nobprapai¹, Olly Gotel²

¹ Mahidol University, Department of Computer Science, Bangkok, Thailand
ccth@mahidol.ac.th, g4637267@student.mahidol.ac.th

² Pace University, Department of Computer Science, New York, NY, USA
ogotel@pace.edu

Abstract. The nature of distributed teams and the existence of multiple sites in offshore software development projects pose a challenging setting for software process improvement. Often, the improvement and appraisal of software processes is achieved through a turnkey solution where best practices are imposed or transferred from a company's headquarters to its offshore units. In so doing, successful project health checks and monitoring for quality on software processes requires strong project management skills, well-built onshore-offshore coordination, and often needs regular onsite visits by software process improvement consultants from the headquarters' team. This paper focuses on software process improvement as guided by the Capability Maturity Model Integration (CMMI) and proposes a model to evaluate the status of such improvement efforts in the context of distributed multi-site projects without some of this overhead. The paper discusses the application of quantitative CMMI assessment through the collection and analysis of project data gathered directly from project repositories to facilitate CMMI implementation and reduce the cost of such implementation for offshore-outsourced software development projects. We exemplify this approach to quantitative CMMI assessment through the analysis of project management data and discuss the future directions of this work in progress.

Keywords: CMMI appraisal, CMMI assessment, offshore software development, project management, quantitative analysis, SCAMPI, software process improvement.

1 Introduction and Motivation

Software process improvement (SPI) is often promoted to ensure more disciplined project management, better project control and (ultimately) enhanced quality in software development products. In practice, SPI efforts are commonly guided and appraised through process improvement approaches such as the Capability Maturity Model Integration (CMMI), as governed by the Software Engineering Institute (SEI) [1]. In the particular context of offshore software development, there can be a strong need to attain a CMMI quality rating at a particular maturity level as it is perceived as

an important factor that contributes to the likely success of the work delivered. Such maturity level ratings further play a very important role in defining how the software development process will be structured and applied to each offshore development unit [2]. From the view of offshore business centers or suppliers, appraisal at a certain CMMI maturity level can maximize their chances of winning business from companies that are pursuing offshore outsourcing services. From the view of onshore business centers or headquarters, working with offshore partners who have gained an appraisal at a requisite CMMI maturity level can help them minimize risk in their outsourced projects.

A quick path to CMMI adoption and implementation in offshore settings can be to impose or transfer a turnkey solution and the best practices from an onshore headquarters to its offshore units. This is often done by providing the offshore units with a collection of procedures, checklists, templates and standards to which the offshore units must conform in their development processes. In so doing, successful project health checks and monitoring for sustained quality on software processes requires strong project management skills, well-built onshore-offshore coordination and regular onsite visits by the SPI consultants from the headquarters' team. A case study has demonstrated the importance of such knowledge transfer from multinational corporations to their subsidiaries so as to build up a knowledge base, improve capabilities, accelerate management localization, and survive intense competition, thereby generating good returns for their parent companies [3]. However, the nature of distributed software development teams and the existence of multiple sites in offshore software development arrangements still pose a challenging setting in which to perform all the necessary management, coordination and consultation tasks.

Though the benefits of attaining a high CMMI maturity level rating are obvious, CMMI adoption and implementation can be costly, time-consuming and complex, especially for low-cost development countries which are the emerging markets for IT and business process offshoring. It is clearly stated by the SEI that the CMMI is a model, not a process standard [4]. The CMMI suggests what to do and what work products should be produced to achieve process quality, but it does not say how to do the process and its essential activities. When systems software engineering is outsourced, providing guidance on how to interpret the CMMI is desirable [5]. Therefore, one factor in successful CMMI interpretation and implementation for offshoring contexts is to overcome any cross-border culture and language barriers in the transition of CMMI practices. Fast-growing offshore outsourcing software development industries in countries like India and those in Central and Eastern Europe may be able to adopt CMMI without the barriers or with fewer barriers of language and culture. However, for other offshoring countries like China, three major obstacles to CMM/CMMI implementation have been reported. The obstacles are over-complex and dogmatic CMMI implementation processes (65%), the high cost of the implementation (52%) and other issues such as the lack of automated supporting tools (<5 %) [6].

This paper proposes to apply a model of *quantitative* CMMI assessment for the benefit of SPI in offshore software development projects to address some of these obstacles. We use the term CMMI 'assessment' to emphasize the fact that what we propose in this paper does not intend to replace a full and traditional CMMI appraisal (as outlined in Section 2). Rather, it seeks to provide complementary and supporting

evidence of process quality and improvement, in an ongoing continual assessment basis and with less need for site visits. Note that this would not eliminate the need for site visits and personnel interviews to gather evidence, but it could help to prepare for such appraisal activities more pointedly. By ‘quantitative’, we mean that the assessment results are calculated from objective evidence that is collected by way of data gathered during the regular course of project development. This is in contrast to traditional CMMI appraisals where project evidence is collected at the time of appraisal, so often in a post-hoc manner.

Tools to support traditional appraisals tend to provide web-based forms or templates to collect project evidence when an appraisal is underway. Such tools typically act more like assistants, providing a navigable structure to help the appraisers gather suitable evidence, a mechanism to record, collate and manage the evidence, and graphical techniques to present the data and resulting maturity profiles for analysis purposes. Example tools include the Interim Maturity Toolkit [9] and CMMiPal [10]. The Interim Maturity Toolkit provides users with spreadsheets to fill in a score from 0 to 10 representing the frequency with which each CMMI practice is conducted, a “?” for ‘I don’t know’, or a “m” for ‘this is not applicable’. CMMiPal uses a database to record objective appraisal evidence. If the evidence is provided for a CMMI practice, then CMMiPal indicates that the practice is done. Other example tools include the Appraisal Assistant (free and general purpose) [7] and the CMMI Appraisal Recorder (commercial) [8]. Given that tool functionality is quite straightforward and similar, many appraisers commonly develop their own proprietary tools to support their tasks. While the results of existing CMMI appraisal tools are generalized in the form of “Do” and “Not-Do” options, our work attempts to classify objective assessment evidence, retrieve this evidence from project repositories, count this evidence, and construct statistical models in an attempt to explain what is observed.

When objective evidence about a project’s activity is collectively stored in a repository for quantification, it allows tasks for local CMMI implementation at an offshore site to be monitored remotely by its headquarters or onshore center. Well-built offshore coordination for CMMI assessment is therefore implicitly achieved through this quantitative CMMI assessment model, thereby reducing the demands of monitoring tasks and the onsite visits associated with CMMI implementation in such settings. Although our first intention in developing this quantitative assessment model was to provide measurable and indicative CMMI assessment results, so as to gauge process quality and guide improvement activities, we also suggest that such a quantitative approach would ease the process of planning and preparing for a fuller appraisal, as well as offer supporting evidence for the appraisal itself.

In Section 2, we describe our quantitative CMMI assessment model and illustrate how it is applied to offshore software development. Section 3 then shows the details of our quantitative CMMI assessment process and rules. We illustrate this approach to quantitative CMMI assessment through the analysis of a project management repository in Section 4, and finish with conclusions and a summary of ongoing work in Section 5.

2 Quantitative CMMI Assessment Model for Offshoring

Our assumption is that the objective evidence for quantitative CMMI assessment can be automatically gathered and measured from an analysis of the existence of key project configuration items (CIs), such as work breakdown structures, requirement reviews, test documents, etc., held in project repositories. While software developers at an offshore site work, the project CIs that are constructed or delivered as part of the development process can be collected into a local project CI repository. Our quantitative CMMI assessment process then makes use of the data in this repository for local CMMI assessment. Considering that a project may be split and outsourced to different offshore units, the local project CI repositories of different offshore sites can be pooled together into an integrated project CI repository for global quantitative CMMI assessment to monitor the quality of the development process as a whole. Fig. 1 depicts this quantitative CMMI assessment concept in a distributed offshore setting.

To conduct the analysis of project CIs for quantitative CMMI assessment, a CMMI appraiser is required to set up rules to identify which project CIs will be suitable as objective evidence of the different sub-practices of the CMMI. This rule setting activity is done as a pre-process to the assessment. To define the same assessment standard for every offshore unit of a single company, a set of assessment rules can be set once and then applied to all the units. Fig. 2 summarizes the main use cases and actors in our assessment model.

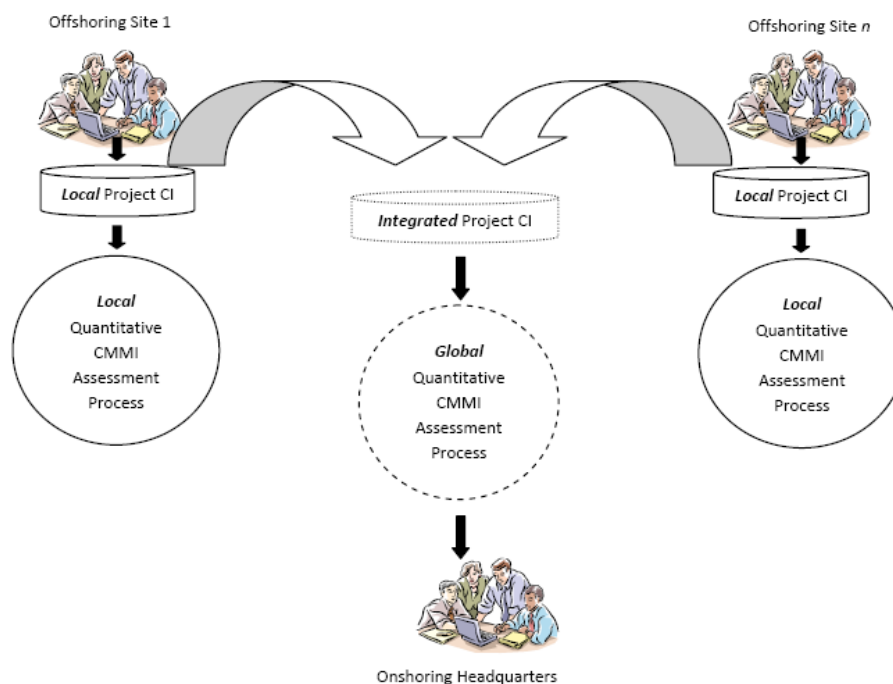


Fig. 1. Overview of Quantitative CMMI Assessment for Offshoring

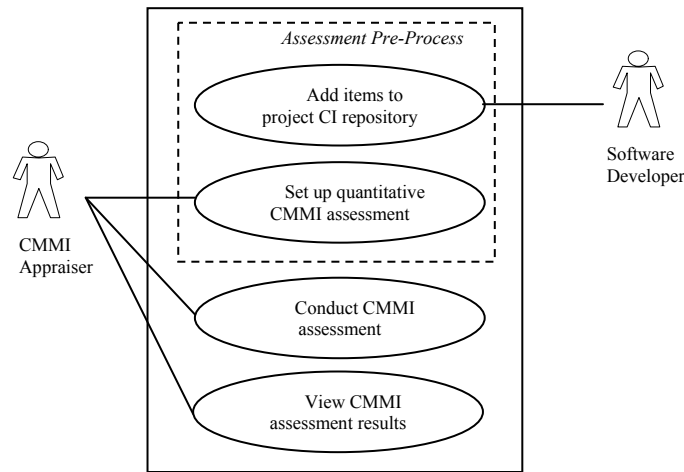


Fig. 2. Use Cases, Actors and Associations in the Quantitative CMMI Assessment Model

The model of our quantitative CMMI assessment aligns with the phases of SCAMPI, the Standard CMMI Appraisal Method for Process Improvement, as designed by the SEI [11]. The SCAMPI phases consist of Plan and Prepare for Appraisal, Conduct Appraisal and Report Results. In the SCAMPI's Plan and Prepare for Appraisal phase, requirements for appraisal are analyzed in order to determine appraisal objectives, scope, constraints, outputs, and to obtain commitment to appraisal input. Following the appraisal plan, team selection and preparation for the collection of objective evidence then proceeds. In the SCAMPI's Conduct Appraisal phase, objective evidence will then be examined, verified, validated and documented, and appraisal results are determined from the analysis of the objective evidence presented. Finally, the appraisal results, lessons learned and feedback are delivered in the SCAMPI's Report Results phase. Our quantitative CMMI assessment model follows such activities as defined by SCAMPI. However, we move part of the data collection activity so as to collect some evidence prior to conducting the appraisal proper. By doing this, we reduce the tasks and time required when the appraisal is actually being conducted. Furthermore, as the results of our quantitative assessment can be calculated from the analysis of the project CIs that exist at any stage during a project, the monitoring of CMMI implementation can be done in near real-time as the implementation progresses. This is particularly appealing for those situations where the cost of an external appraisal may be prohibitive but where awareness of process maturity is beneficial. In the next section, we explain this assessment process and its rules in more detail.

3 Quantitative CMMI Assessment Process and Rules

Fig. 3 shows the architectural view of our CMMI assessment model. It consists of two main components: *Project Configuration Items Repository Manager* and *Quantitative*

CMMI Assessment Engine. A dotted border box in the figure indicates the boundary of each component.

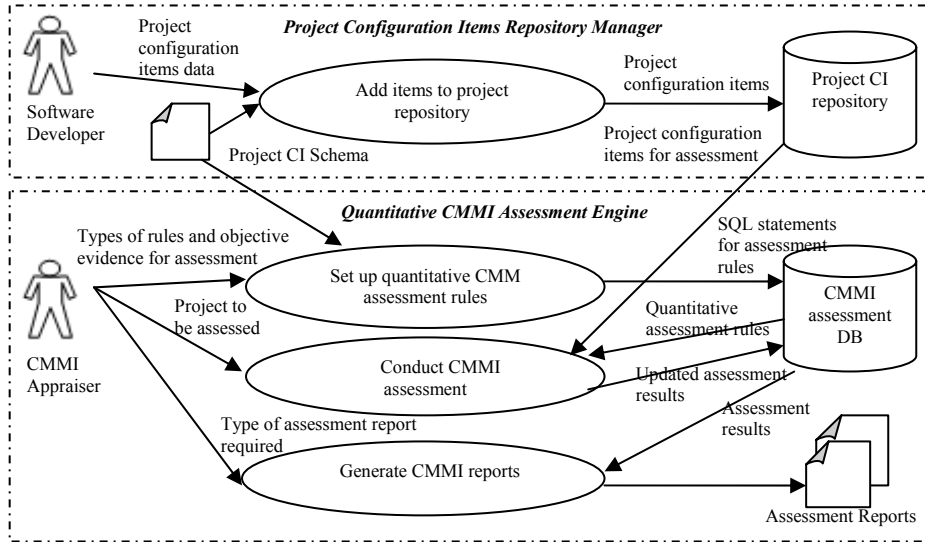


Fig. 3. Architectural View of Quantitative CMMI Assessment

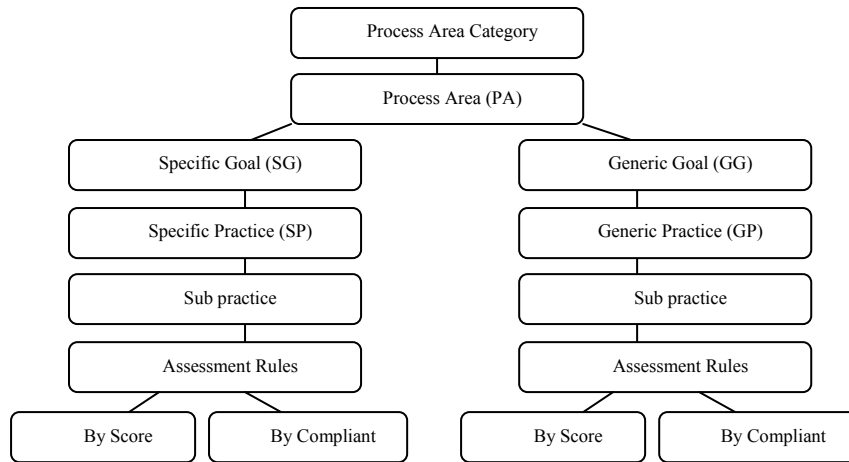


Fig. 4. Hierarchy of Conventional CMMI Components and Associated Assessment Rules

Software developers use the Project Configuration Items Repository Manager to add details of project configuration items as the project development progresses. In our implementation, we design project CI schema representing the data model of configuration items to be collected. An example of the data model is shown in the next section. Details of the project CI will be retained in the project CI repository to be retrieved for further analysis and CMMI assessment.

The Quantitative CMMI Assessment Engine contains the processes that deal with setting up quantitative CMMI assessment rules, conducting CMMI assessment and generating assessment reports. In our model, there are two types of assessment rules to be evaluated: “By Score” assessment rules are for checking on the number of times that the implementation of a certain activity is satisfied with CMMI assessment rules; while “By Compliant” assessment rules are for checking whether an activity is carried out or not, without concern for the number of times the activity is actually performed. Fig. 4 shows how we associate our assessment rules to the hierarchy of conventional CMMI components. The calculation of assessment results is hence derived from sub-practices to process areas in different process area categories.

As mentioned in Section 2, to define a quantitative assessment rule, CMMI appraisers are required to select those types of objective evidence to be measured and types of rules for those measurements. Our Quantitative CMMI Assessment Engine will then automatically transform the specification of assessment rules given by the appraisers into SQL statements. The statements will be subsequently retrieved in the analysis process for CMMI assessment.

Upon the completion of the assessment, CMMI appraisers may obtain assessment reports which consist of: (i) a summary of assessment results by process area category and by process area; (ii) a detailed report of the assessment results for all practices and sub-practices; and (iii) an assessment report in the form of an area chart diagram showing the relative strengths and weaknesses of the Specific Goals (SGs) and Generic Goals (GGs). Examples of these reports are illustrated in the next section.

4 Sample Application of Quantitative CMMI Assessment Using the Analysis of Project Management Repositories

To demonstrate how our quantitative CMMI assessment is conducted, we provide a worked example in this section. To limit the scope of this example, we focus on the assessment of the CMMI project management process area category. Fig. 5 represents the data model of the configuration items that we designed as offering objective evidence of the Project Monitoring and Control (PMC), Project Planning (PP) and Supplier Agreement Management (SAM) process areas of the project management process area category. In our implementation, this data model is defined in the form of the project CI schema as mentioned in Fig. 3, and it was derived from the manual inspection of the types of work products required in the assessment of these project management related process areas. The templates for entering actual instances of project CI for these types of work products are provided by the Project Configuration Items Repository Manager, which was described in the previous section. The instances explain relevant attributes of project configuration items. For example, a *project* instance is defined with the attributes of project ID, organization ID, customer ID, project name, project description, project objective, project manager, overall budget and actual cost. Likewise, a work breakdown structure or WBS instance is defined with its ID and details of tasks that are contained in the structure, including task name, task description, task category and phases. The attribute values of an

instance will be checked for the quantified results of an assessment as specified by predefined assessment rules.

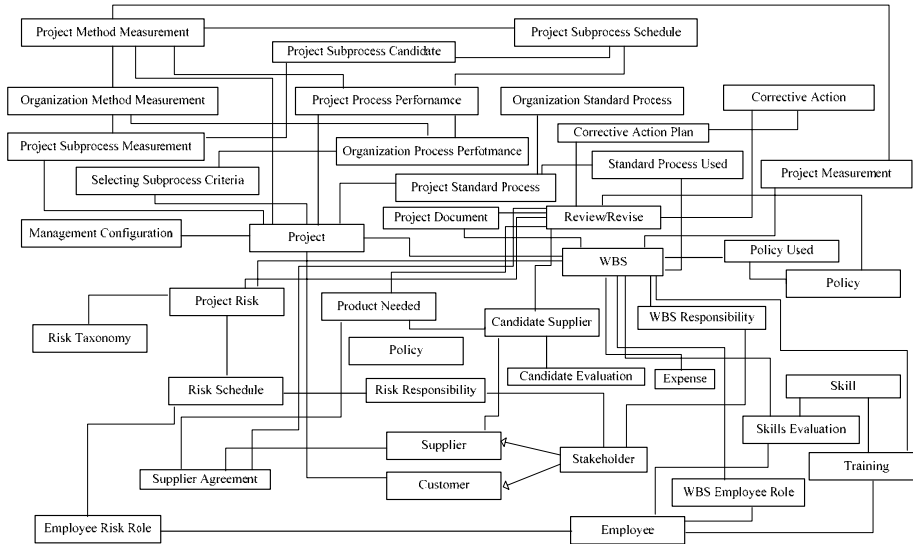


Fig. 5. Logical Data Model of Project CI Repository of Project Management Related Process Areas

Fig. 6 exemplifies some of the rules that have been defined for this example. The description of tasks required for the sub-practices was obtained from the CMMI standard. The assessment rules defined in the figure are samples of the selection of objective evidence by CMMI appraisers for each of the sub-practices. The research methodology for defining the rules is to firstly inspect the assessment requirements specified by CMMI sub-practices, then to select the relevant objective evidence from the project CI repository, and then finally to determine whether the assessment of the sub-practices should be done by score or by compliance as previously defined in Section 3. The SQL statements shown in Fig. 6 are quantitative CMMI rules that were automatically generated by our Quantitative CMMI Assessment Engine in response to the selection of objective evidence and assessment requirements. Note that the specification of quantitative assessment rules can be made at a fine-grained level to define criteria on the attributes of project configuration items. An example can be seen from the grey highlighted rule of sub-practice 1 of SP 1.1 of SG 1 in the Project Planning Practice Area in Fig. 6.

To demonstrate how the Quantitative CMMI Assessment Engine will generate the results of a quantitative CMMI assessment from the analysis of project management repositories, assume that we are conducting a CMMI assessment for the ABC Company whose nature of business is software services and solutions. The ABC Company has outsourced its inventory system development to an offshore unit. The offshore unit maintains records of how they have dealt with suppliers and managed product needs, but they rarely documented details on project tasks.

Category Project Management
 PA: Project Planning
 SG 1: Establish Estimate
 SP 1.1 Establish the scope of the project

Subpractice	Description	Descriptive Assessment Rule	SQL
1	Identify risks and their mitigation tasks	Check relationship of WBS with project risk	SELECT COUNT(X) FROM WBS, Proj_Risk WHERE (Proj_Risk.WBS_ID = WBS.WBS_ID)
	Tasks for skill and knowledge acquisition	Check WBS relationship with skill and training identified	SELECT COUNT(X) FROM WBS, Skill_Need WHERE (Skill_Need.WBS_ID = WBS.WBS_ID)
	Tasks for development of needed support plans, such as configuration management, quality assurance, and verification plans	Check tasks defined in WBS for the task categories of supported plan, configuration management, quality assurance and verification plan	SELECT COUNT(X) FROM WBS WHERE Task_Category IN (Support Plan, Configuration Management, Quality Assurance, Verification Plan)
2	Identify the work packages in sufficient detail to specify estimates of project tasks, responsibilities, and schedule	Check availability of filling information in WBS about planning	SELECT COUNT(X) FROM WBS WHERE Phase="Planning"
3	Identify product or product components that will be externally acquired	Check availability of WBS relationship with product needed	SELECT COUNT(X) FROM WBS, Product_Need WHERE (Product_Need.WBS_ID = WBS.WBS_ID)
4	Identify work products that will be reused	Check availability of reused material in WBS	SELECT COUNT(X) FROM WBS WHERE Reused_Material IS NOT NULL

SP 1.2 Establish estimate of work product and task attributes

Subpractice	Description	Descriptive Assessment Rule	SQL
1	Determine the technical approach for the project	Check availability of project document with document type of technical approach document	SELECT COUNT(X) FROM WBS, Proj_Doc WHERE WBS.WBS_ID = Proj_Doc.WBS_ID AND Proj_Doc.Doc_Type = "Technical Approach"
2	Use appropriate methods to determine the attributes of the work products and tasks that will be used to estimate the resource requirements	Check availability of WBS relationship with project measurement	SELECT COUNT(X) FROM WBS, Proj_Measurement WHERE (Proj_Measurement.WBS_ID = WBS.WBS_ID)
3	Estimate the attributes of the work products and tasks	Count ratio of expected work product in WBS table	SELECT COUNT(X) FROM WBS WHERE Expected_WP IS NOT NULL; SELECT COUNT(X) FROM WBS;

SP 1.3 Define project life cycle

Subpractice	Description	Descriptive Assessment Rule	SQL
1	Define the project lifecycle phases on which to scope the planning effort	Check availability of phase in WBS table	SELECT COUNT(X) FROM WBS WHERE Phase IS NOT NULL

Fig. 6. Sample Quantitative Assessment Rules for the Project Management Process Area Category

Fig. 7 shows the existence (or not) of project configuration items in the project management repository of the ABC Company's offshore unit. For this sample case,

there are some configuration items or work products that the ABC Company produces that provide evidence of project schedule planning and supplier agreement management activities. Nonetheless, the company does not hold evidence that they conduct any activities related to managing risk, resources, documents, skills, training and stakeholders. In other words, the project configuration items related to those activities do not exist in the ABC Company’s project management data repository.

CI Types	Project Scheduling	Risk Management	Resource Management	Document Management	Skill and Training Management	Supplier Agreement Management	Stakeholder Management
Existence	Some	None	None	None	None	Some	None

Fig. 7. Status of Project Configuration Items for ABC Company’s Offshore Unit

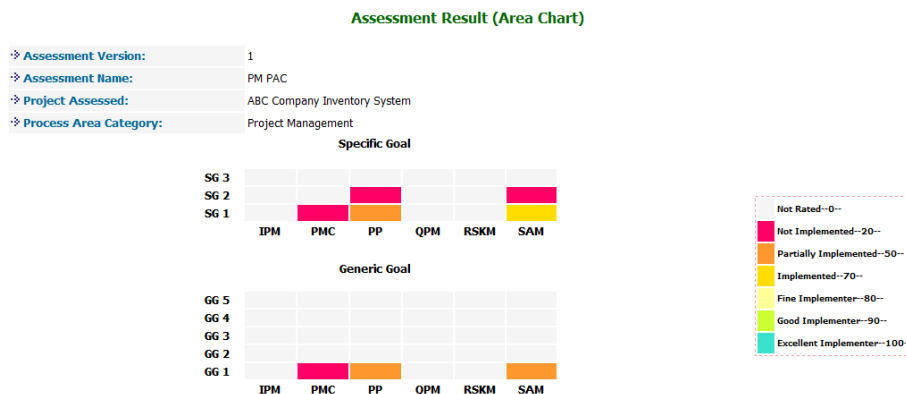
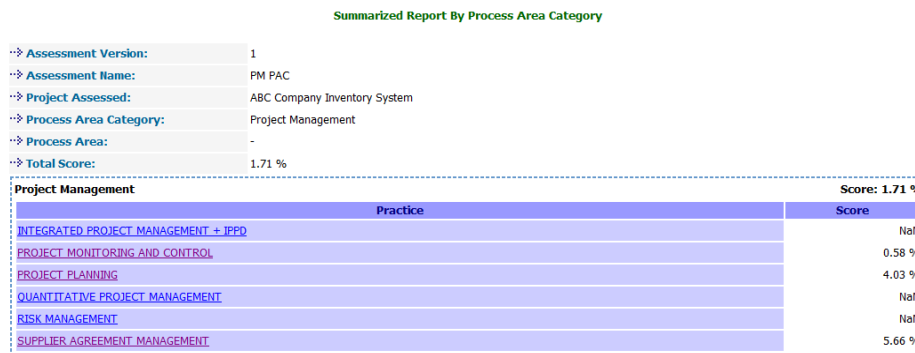


Fig. 8. Sample Quantitative CMMI Assessment Result for ABC Company’s Offshore Unit

Fig. 8 illustrates the assessment results that can be obtained from this example. The numbers that are the results of this assessment for each practice are calculated from the assessment rules that were generated for quantifying the project CI in the project management repository. The existence of CI types for project scheduling and supplier agreement management results in the assessment scores in the areas of Project Monitoring and Control, Project Planning, and Supplier Agreement Management.

From Fig. 8, we can see the assessment results that indicate not only whether the practices are done in the areas but also the numbers representing the percentage of project configuration items that match the objective evidence as defined for the assessment of each practice or process area. It can also be seen that the overall maturity of the project management process area of the offshore unit of the ABC Company is very low (1.71%). Out of PMC, PP and SAM, this offshore unit has the strongest maturity in SAM. However, further process improvement on SAM is still required for the offshore unit as the maturity of its SAM process is still only 5.66%. In other words, the ABC Company's offshore unit has done only 5.66% of all the tasks required for evidence of process maturity in supplier agreement management, as recommended by the CMMI. (Note the NaN results in Fig. 9 reflect process areas that are not yet supported.) The assessment results, in the form of an area chart shown also in Fig. 9, further exemplify the rating of the generic goals (GG) and specific goals (SG) in the project management process area category that this offshore unit satisfies.

5 Conclusions and Ongoing Work

We have presented the design of our quantitative CMMI assessment model in this paper. In order to automate the CMMI assessment process and to collect objective evidence for CMMI assessment, our model is comprised of two parts: a Project Configuration Items Repository Manager and a Quantitative CMMI Assessment Engine. In comparison to SCAMPI phases, our model collects objective SPI evidence prior to the time the appraisal is actually conducted based on examining existing project data that has been collected during the normal course of project activities. Unlike similar tools available from companies doing CMMI consulting or conducting SCAMPI appraisals, we have proposed to separate the collection of objective assessment evidence (where possible) from the participants who will assess the evidence. Instead of combining the collection of all evidence with the appraisal process, we anticipate that some pre-collection of data is possible. This can potentially reduce both the complexity of tasks and the time required to conduct appraisals, particularly in a multi-site setting. Further, it can provide for an up-to-the-moment perspective on both local and integrated levels in such settings.

To evaluate our model, we have developed a CMMI assessment tool and have designed the assessment rules for the Project Monitoring and Control (PMC), Project Planning (PP) and Supplier Agreement Management (SAM) project management process areas of the CMMI. We used VB.NET to implement the Project Configuration Items Repository Manager for the data entry of the project configuration items and we used ASP.NET to implement a web-based application for the CMMI assessment, which is equipped with our Quantitative CMMI Assessment Engine. The tool was exercised with test cases to provide for a preliminary inspection of the correctness of the assessment rules in the form of the SQL statements that were automatically generated by our Quantitative CMMI Assessment Engine, as well as to evaluate the correctness of the assessment results as measured from the analysis of the project management data of the test cases. To fully test our model for quantitative

CMMI assessment, more work needs to be done to define additional rules for the remaining process areas in the project management process area category, such as Integrated Project Management, Quantitative Project Management and Risk Management. We are currently investigating how to import the project management data from existing project management tools so that the assessment can be done based on the content of project configuration items held in such tools directly, thus easing the process. Future work also involves extending the design of the project's CI repository and its data entry to include additional CIs from other software development phases beyond project management. Although it may require intensive expert experience and human work at the onset to set up the assessment rules, the set up is required once and passed on to multiple offshoring sites where the similar practice of CMMI assessment is required. As with the appraisal process in general, where the selection of appropriate evidence depends upon the skill and integrity of the appraiser, the assessment rules are likewise pivotal. However, ensuring the quality and consistency of the SEI's process appraisal technology is outside the scope of our current work.

Although in its early stages, this work has begun to demonstrate how quantitative CMMI assessment can be achieved through the analysis of project configuration items that can be collected automatically, remotely and cumulatively while a software development project is in progress, irrespective of the global distribution of project units and without the need for software developers to change their everyday practices to explicitly enable such continual assessment. In so doing, we suggest that it will help to reduce the cost and effort of SPI implementation for offshore software development, especially in emerging offshoring countries where the costs of such programs are prohibitive and the guidelines are not so well established. The quantitative assessment results generated from our model are primarily intended to form part of an internal or self-appraisal program, so facilitating and guiding CMMI interpretation and implementation. In this way, our model can be considered as a complementary technique to the wider appraisal activities that are demanded by SCAMPI, not a replacement. However, the benefit is that it does offer precise measurement and analysis of objective assessment evidence by explicit programming queries and statistical models instead of relying solely on an individuals' interpretation of the existence of objective assessment evidence.

Further work is required to put this work into practice with real project data and to evaluate its practicality and appeal to the wider software industry. Its role in supporting SPI in offshoring contexts will be the primary target of our studies. At this current stage of the work, we have demonstrated that the assessment rules proposed in this model can enable the quantified measurement and quality rating of process activities or work products relative to CMMI criteria. The validation of this model, process and supporting tool by CMMI appraisers, and the usefulness of the outputs generated, will need to be further investigated as the next step. The quality of the rules set for CMMI assessment in this model will be validated further, by inspecting the precision and recall of the outputs returned by the rules in comparison to the outputs of manual CMMI assessment. Time factors, such as the time required in setting the rules and analyzing the outputs, in comparison to the time required for manual assessment, will also be studied to examine the time saving that use of this tool may

bring. We plan to make the tool freely available for others to use so as to enable this next validation step.

References

1. Software Engineering Institute (SEI): CMMI for Development Version 1.2. Carnegie Mellon University, Pittsburgh, (2006)
2. Pilatti, L. and Audy, J. L. N.: Global Software Development Offshore Insourcing Organizations Characteristics Lessons Learned from a Case Study. In IEEE International Conference on Global Software Engineering (ICGSE'06), (2006)
3. Wanga P., Tongb T. W., Koh C. P.: An Integrated Model of Knowledge Transfer from MNC Parent to China subsidiary, Journal of World Business 39, pp 168-182, Elsevier (2004)
4. Software Engineering Institute (SEI): CMMI? Or Agile: Why Not Embrace Both!. <http://www.sei.cmu.edu/pub/documents/08.reports/08tn003.pdf>
5. Konrad M., Chrissis M. B., Curtis B., and Paulk M.: A Report on the May 2002 CMMI® Workshop, Adoption Barriers and Benefits for Commercial Software and Information Systems Organizations, Software Engineering Institute, <http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02sr005.pdf>
6. Wu, Z., Christensen, D., Li, M. and Wang., Q.: A Survey of CMM/CMMI Implementation in China, SPW 2005, LNCS, vol. 3840, pp. 507--520, Springer, Heidelberg (2005)
7. Appraisal Assistant - <http://www.sqi.gu.edu.au/AppraisalAssistant/indexFrameset.html>
8. CMMI® Appraisal Recorder - <http://www.se-cure.ch/Products.html>
9. Interim Maturity Evaluation based on Capability Maturity Model Integration for Development (CMMI-DEV), V1.2, Management Information System, http://www.man-info-systems.com/index_files/FreeTools.htm
10. CMMiPal 1.0, Chemuturi Consultants, <http://www.brothersoft.com/cmmipal-63969.html>
11. Software Engineering Institute (SEI): Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method Definition Document, Handbook, CMU/SEI-2001-HB-001, Carnegie Mellon University, Pittsburgh, (2001)