University Honors Program Research Project

APPROVAL PAGE

Student Name: Jonathan Winstead

Department: Computer Science

College: Science

Degree: Bachelor of Science

Project Advisor: Dr. Richard Coleman

Full title of project as it should appear in Graduation Program and transcript:

Evaluation of Senior Project Group Processes Against CMMI Standards

Abstract:
(include near the beginning of your thesis also)

In order to better understand the quality and character of the methods used by students in the department's senior project course, I have conducted a study of the processes of my senior project group to evaluate how they conform or fail to conform to the applicable processes required for CMMI maturity levels two and three. As both a participant and an observer in my project group, I have gathered relevant data pertaining to the group's various actions and processes. I have utilized CMMI standard evaluation practices to analyze our group processes insofar as said practices are applicable to the context of the project. In addition to discussing and evaluating the methods used by my group, I have also made recommendations for how we could have better met the given standards.

Approved by:

Project Advisor: Richard L Coleman Date: 5-12-2011
(signature)

Department Chair: H-S. Ramparsad Date: 5-12-2011
(signature)

University Honors Program Director: _____ Date: _____
(signature)

Evaluation of Senior Project Group Processes Against CMMI Standards

Jonathan Winstead
5/13/2011

In order to better understand the quality and character of the methods used by students in the department's senior project course, I have conducted a study of the processes of my senior project group to evaluate how they conform or fail to conform to the applicable processes required for CMMI maturity levels two and three. By functioning as both a participant and an observer in my project group, I have gathered relevant data pertaining to the group's various actions and processes. I have utilized CMMI standard evaluation practices to analyze our group processes insofar as said practices are applicable to the context of the project. In addition to discussing and evaluating the methods used by my group, I have also made recommendations for how we could have better met the given standards.

Project Overview

As the basis for my honors research project, I was assigned to observe the collective processes used by my group in our team software design course and compare them to the standards laid out by the CMMI. More specifically I was tasked with examining our practices against those defined as belonging to the CMMI's maturity levels two and three. In addition to comparing group processes to standards, I was also asked to make recommendations on how our compliance could have been improved.

The senior project course itself consisted of four teams of four to five seniors in the university's computer science program. The instructor divided the course into six distinct phases marked by the development of a given document or software product and the presentation of said work products to the class. The six phases, as defined by their assigned work products, were as follows: planning, requirements definition, requirements specification, design, test design, and implementation. Each group operated within this framework to produce a different software product. My group was assigned to develop a data management system for a theoretical trucking company. From the given statement of work, we were able to develop a web application that allowed employees of the company to view data specifically pertinent to their positions.

CMMI Breakdown

The CMMI, short for Capability Maturity Model Integration, was developed by the Software Engineering Institute at Carnegie Mellon University. Since the model's initial release, the SEI has released multiple versions of the official CMMI manual. For the sake of clarification, my research utilized the CMMI for Development version 1.2, and that is the version I will be speaking to in this section. The first essential point to understand regarding the CMMI is that it is a model for guiding business process improvement. Process improvement is not to be

confused with the actual methods for creating or deploying a product. As clarified by the independently published CMMI FAQ, "the assumption with all CMMIs is that the organization has its own standards, processes and procedures by which they actually get things done" (CMMI FAQ).

The second critical point to understanding the CMMI is to grasp the structure of the model. The highest level conceptual item is the maturity level. Maturity levels mark milestones in the quest for process improvement by providing a general statement of an organization's processes. The most commonly sought maturity levels and the ones that I examine in detail are levels two and three. The initial maturity level, at which all organizations begin in regards to the CMMI, is characterized by processes that lack the abilities to be predicted and managed due to their chaotic nature. An organization at level two has improved upon these problems by stabilizing its processes across individual projects. At level three the organization has provided standard processes to all departments in order to stabilize projects across the entire organization (CMMI for Development, 36-37).

Of course these vague descriptions of how organizations operate are not the basis upon which clients are evaluated to determine their level of process improvement. Evaluation of an organization's compliance with a given maturity level is determined by its work in several key process areas. These process areas are further divided into specific and generic goals, which are respectively subdivided into specific and generic practices. Process areas and their specific goals and practices contain the bulk of the model's process improvement standards. In order to be certified for a given maturity level, an organization must show compliance with each process area associated with that level down to the specific and generic goals. Each level contains multiple process areas that must be observed for a successful appraisal. Each of the process

areas for maturity levels two and three are discussed in detail later in the document (CMMI FAQ).

CMMI for Academics

There is one important question that needs to be addressed before moving on to the results of the data analysis. Is the CMMI, a model intended to improve processes in the business arena, suitable for use in the academic field? While the length of this document is a pretty good indicator of my conclusion, I do have some objective evidence to show for a positive answer. In order to get an answer, I researched businesses that had similar constraints to those that we are bound by in the classroom. One such type of organization is the small business; development teams in this realm typically operate with small teams and work on projects within a tight development cycle. Two development groups working off of a pilot program implemented by the SEI were faced with the problem of trying to apply the CMMI with limited time and resources. These groups faced some expected problems such as implementing process improvements without having personnel specifically assigned to that end, getting executive approval for implementing said improvements, and working around the very tight relationship small businesses tend to have with their customers. In spite of these challenges, the participants were able to find value in implementing pieces of CMMI under the right conditions. One major concern was aligning the process areas to be implemented with the organizations' business models; process improvements in extraneous areas were not of much help. According to the participants, however, even small improvements such as keeping and distributing meeting minutes as added documentation could provide immediate benefit when aligned with the given business context (Cepeda, Garcia, and Langhout).

More evidence for the utility of CMMI process improvements can be found in the realm of developers using the agile model. This model advocates fast, iterative development cycles undertaken by small teams with embedded customers. The tenants of agile development can often lead to the misconception that agile methods and CMMI are mutual exclusives. According to Broadsword, a consultant company that advocates CMMI and agile methods in tandem, this is not the case as the goals of agile development do not preclude the emphases of CMMI. Short, iterative development cycles, for example, don't make the planning prescribed by the CMMI impossible but simply short and iterative just like the cycle itself. Additionally, agile development does not need to preclude the possibility of achieving a maturity level rating simply for lack of produced documentation. The rationale behind this argument is that even though agile methods produce relatively little in the way of formal documentation, they still produce other kinds of artifacts that can be used to show evidence of process improvement compliance (Dalton).

The remaining sections of the document outline the results of my data analysis for each CMMI process area in levels two and three. For each process area I have given a brief explanation of the specific goals and practices followed by an assessment of my group's performance in that area. I have also provided examples of how our group could have better met the given standards as well as some advice on how the senior project course could better facilitate CMMI compliance.

Level Two Key Process Areas

Configuration Management

The first key process area of maturity level two is configuration management. The main idea of configuration management is to break the software product down into a set of attributes through which the product's integrity and traceability can be managed. The CMMI defines satisfactory performance in configuration management according to three specific goals: establish baselines, track and control changes, and establish integrity. The first goal of establishing baselines revolves around identifying and managing configuration items, which are composed of a software product's work assets such as code, documentation, customer products, and work tools. Once all of the applicable configuration items have been identified, a system for managing and tracking changes to these items should be devised. With this infrastructure in place, a specification of all the critical configuration items at a given time during the course of the project can be created; this specification is known as a baseline (CMMI for Development 114-130).

The second goal of configuration management, track and control changes, encourages teams to carefully manage any changes that are made to the product's configuration items. The recommended method for accomplishing this goal begins with tracking all change requests made on the configuration items and analyzing how these potential changes could impact the project. Only those changes which have been well-documented and examined for impact should be allowed. Given that changes are being tracked and controlled, the third goal of configuration management, establish integrity, becomes a straightforward objective. Establishing integrity involves producing configuration management records regarding revisions, change requests, and

changes between baselines and then verifying those records by performing audits to ensure accuracy (CMMI for Development 114-130).

As far as the compliance of the senior project group goes, our biggest strength in configuration management was establishing baselines. Since the senior design course is already broken down into several phases, it's easy to think of our assets going into each phase as the newest baseline for the project. We further complied in this area by presenting each new baseline with the customer, supervisor, or manager role in mind. However, our conformance to the other two goals did not prove quite as solid. While version control software was provided by the instructor at the beginning of the course, we did not make consistent use of it up to the time of writing. Our methods in the absence of said software consisted of achieving group consensus on the most critical of changes given a favorable projected impact. We exercised a degree of control over changes but made no meaningful efforts to track them. Because we neglected to produce this documentation regarding changes to our configuration items, it wouldn't have been possible to establish integrity through audits.

Improving our CMMI conformance in configuration management would largely be a matter of taking some manner of action towards implementing the second and third goals. Utilizing the version control software provided would be a huge step towards tracking and controlling changes to our configuration items since the software can automatically generate comparisons between different versions of the same document. A well-defined, general procedure for documenting changes could also prove to be a strong asset. Given that we were actually producing some of this configuration management documentation, establishing integrity would become a possibility even if only through such simple means as manually checking a configuration item for consistency with documentation. Implementing formal, comprehensive

audits might be a little closer to the true organization-centered heart of CMMI, but it doesn't seem particularly feasible given the academic model we're operating in.

Measurement and Analysis

Measurement and analysis revolves around collecting relevant and useful data during the lifecycle of the given project as well as specifying processes for analyzing, reporting, and storing said data. The CMMI provides two specific goals for this process area: align measurement and analysis activities and provide measurement results. The former goal is concerned with defining what data will be collected, why it is being collected, how it will be collected, and how it will be analyzed. Ideally documentation should be provided to answer each of these questions. The latter goal requires that the data be collected, analyzed, and used to produce useful results. The results should be communicated to the relevant personnel in a manner that facilitates their use in the decision making process (CMMI for Development 178-197).

There is relatively little to be said about our group's compliance in the area of measurement and analysis. No particular phase of the project directly encourages the creation or use of data collection processes, so it shouldn't come as much of a surprise that we didn't have much to show. There are however viable ways of reconciling this goal with the format of the course. One simple metric that could be introduced into the project's phases would be actual vs. expected performance. The planning phase already requires us to project our expected performance in terms of requirements and deadlines, so it wouldn't be a huge strain to look back later in the project and evaluate how well those expectations were met. In addition to simply recording this kind of data, we could further examine it to analyze how well our processes are working and bring ourselves closer to the processes described in the CMMI.

Project Monitoring and Control

The primary purpose of project monitoring and control is to provide a solid, up-to-date understanding of the project's progress. This visibility into the state of the project should be utilized as a control measure in the event of unplanned deviations from the given project plan. The first specific goal prescribed to accomplish these tasks is to monitor the project against the plan. The CMMI describes several practices to the end of accomplishing this goal including monitoring planning parameters, commitments, risks, data management, and stakeholder involvement. It also recommends conducting periodic progress and milestone reviews in order to further ensure that the project is proceeding as expected. The second goal, manage corrective action to closure, expects issues that arise to be not only analyzed for solutions and corrected but also monitored to ensure that the given corrective actions are effective and appropriate (CMMI for Development 313-326).

In the area of project monitoring and control, our group was relatively compliant. We monitored the project against the various documents that we were required to produce, and changes to project parameters were reflected in the documents produced subsequently. We also had a suitable amount of stakeholder involvement in the form of the presentations that mark the end of each phase. We did not conduct formal reviews of our own volition, but the format of the project did require us to frequently look back on our previous work in order to incorporate it in the next document or presentation. This format encourages groups to look back and re-evaluate their previous decisions against a different context, which essentially realizes the idea behind performing formal reviews. Managing corrective action proved to be one of our more ad-hoc dialogue-based processes. No particular efforts were made to document the various fixes or to watch how well these fixes fit the problems.

Our moderate level of compliance in project monitoring and control could easily be improved by expanding upon two of the processes already built into the course. To solidify our standing in monitoring the project against the plan, we could simply incorporate a brief assessment of our progress into each presentation. Each group could provide a status report detailing prominent schedule changes in relation to a set of relevant future subtasks. Improving how we manage corrective actions would be a matter of using the tools that were provided to us at the beginning of the course, those being Tortoise and the Trac system. With Tortoise handling our version control and Trac integrating versioning into its own change control system, it would be a much more straightforward matter to exert control over required changes.

Project Planning

The purpose of project planning is exactly what one would expect: to produce and maintain plans that describe the project's activities. The CMMI prescribes three specific goals to accomplish this task. The first goal, establish estimates, requires teams to consider in advance many project parameters such as scope, size, complexity, effort, and cost. Estimations of scope should be informed by a work breakdown structure, which is a document containing sets of individual tasks that must be performed as part of the project. Size and complexity should be determined in terms of work products and tasks, and these items should also inform estimates of cost and effort together with process models and any prior data gathered by the organization or group. Once the appropriate estimates have been gathered, the next goal is to develop the project plan. The CMMI recommends including the following to build a complete project plan: budget, schedule, risk analysis, data management plan, resources needed, skills and knowledge needed, and involvement of relevant personnel. Each of these items should be described in as much detail as is reasonable for the planning phase. After the complete project plan has been

assembled, the final specific goal, obtain commitment to the plan, can be completed. Commitment to the plan should be obtained once all personnel involved have reviewed the plan and resource levels are appropriate for the given work (CMMI for Development 327-352).

Group compliance in project planning was relatively solid where the requirements were actually applicable to the academic setting. Due to the nature of the course and setting, some of the suggested estimates were either extraneous or already in place. Project duration and lifecycle, for example, were already laid out as part of the course's requirements, and cost was simply not a factor. We examined scope somewhat indirectly through the requirements definition and specification phases; effort and complexity, however, were not addressed. Developing a project plan was one of our most compliant areas across the level two process areas thanks to the initial project phases. Our software planning document included a projected schedule, a risk analysis section, and a list of personnel roles based on prior knowledge and experience. Data requirements were not included in the planning document but instead examined in the requirements definition and specification documents. Obtaining commitment was another promising area for our group. We reviewed our plans both amongst our own group members as well as with each of the customer, manager, and supervisor roles primarily through the required presentations. Each document we produced was delivered to the instructor to receive approval and confirm our commitments to the project. Reconciling our resource levels did not prove an especially pressing concern; the most we had to do is to download some free software on our personal machines to accommodate code development in our chosen languages.

We don't require a lot of improvement in the area of project planning, but there are a few small changes that could be made to better satisfy CMMI standards. Our most prominent weakness in this process area is our lack of estimates for effort and complexity in relation to

project tasks. While it probably wouldn't be feasible to incorporate these estimates into the planning phase, it might be more reasonable and useful to examine them in the context of the requirements or the design. A possible improvement to the software planning document itself would be to add small sections to discuss data management strategies and the knowledge and skills that will be needed to complete the project.

Process and Product Quality Assurance

This rather brief process area concerns itself with ensuring that all personnel involved in the project have the necessary information to evaluate whether the organization's standards are being met. The specific goals defined by the CMMI to accomplish this are to objectively evaluate processes and work products and to provide objective insight. Evaluating processes and products largely boils down to understanding and applying organizational standards, but measures should also be provided in order to ensure that analyses remain objective. Providing insight into the project is a similarly straightforward affair. Any problems with noncompliance should be promptly reported to management and proper procedures should be used in their resolution. Additionally, records should be established regarding any quality assurance issues that arise (CMMI for Development 353-363).

Group compliance in this process area was rather vague in that we did not quite produce anything that could be seen as affecting process or product quality assurance. The class presentations did require us to think outside our own frames of reference and examine our work through other perspectives. However, the project format doesn't ever require groups to look back on the processes that they've been using. While visibility into the project was not a problem, we didn't directly address quality assurance issues either. One could certainly argue that these details are not especially important given the rapid pace of the course. That being said,

these same details could be addressed without putting a strain on the already tight project timeline. A simple suggestion would be to implement a mid-semester group evaluation that asks students to submit a brief survey concerning group processes and how they affect the quality of the project. This would be best implemented sometime closely following the requirements specification phase so that each group would have an understanding of how their processes have been proceeding.

Requirements Management

Requirements management as a process area speaks for itself; its purpose is to carefully manage the set of requirements while watching for discrepancies between the project and the given requirements. The CMMI provides only one specific goal for this area, which is appropriately stated as "manage requirements". The first steps in this management process are to obtain understanding of the requirements and then to obtain commitment to them. Understanding and commitment do not only apply to the project group but to all parties that are involved. Project teams are also responsible for managing any changes to the requirements as well as analyzing what impact a given change may have. Requirement traceability should be maintained so that all derived requirements can be traced back to a source requirement or to the statement of work. Finally, inconsistencies between the requirements and the project must be corrected and logged for future reference (CMMI for Development 408-419).

Because the course includes two separate requirements analysis phases, our group had a sufficient amount of work to show for this process area. Through the requirements definition and specification phases, we gained a solid understanding of our project's requirements and likewise obtained commitment from the instructor in the relevant roles. It is recommended as part of the course for groups to update requirements with relevant changes and to maintain

traceability from the requirements phase onwards. These tasks are not mandatory, however, and our group did not make efforts to go back and revise the requirements during subsequent project phases. Inconsistencies in the project were observed over the course of the design phase, but we did not go out of our way to record these details.

Our group's biggest weakness in this area is our lack of maintenance of the requirements after the applicable phases of the project. Rather than burden groups with the responsibility of maintaining any and all changes that may need to be made to the requirements over the course of the project, one could instead suggest that a traceability section be added to the software design document. In addition to including the matrix, groups could provide a concise overview of how their requirements have changed during the design phase. This would encourage students to not only re-examine the requirements but also to draw conclusions regarding how they could have better anticipated future changes.

Supplier Agreement Management

Supplier agreement management is not only the most situational process area out of maturity level two but also the area which is the least applicable to our academic model. As the title alludes, supplier agreement management provides goals to help guide teams and organizations towards better managing how they interact with and receive products from suppliers. Because our project format lacks even a loose analog to the supplier role, however, the details of this process area will not be discussed here (CMMI for Development 439-455).

Level Three Key Process Areas

Decision Analysis and Resolution

The first process area of level three, decision analysis and resolution, encourages teams to look closely at their decision making processes and to determine ways of evaluating decisions against viable alternatives. The single specific goal for this area, evaluate alternatives, provides a process beginning with establishing general guidelines for decision analysis, creating criteria for evaluating the available alternatives, and actually identifying the alternate solutions. Once these preliminary details have been gathered, the team should select the most effective method for evaluating the potential alternative solutions, perform the selected evaluation, and finally choose the most effective solution and examine any associated risks (CMMI for Development 131-144).

Our group's performance in this process area mostly consisted of informal, on-the-fly decision making. Because no guidelines were provided for this aspect of the project, we relied on group consensus regarding important decisions and whether or not to pursue alternatives. Similarly, we did not develop any sort of system for evaluating alternatives, nor did we establish any sort of evaluation criteria. In order to bring our decision analysis processes closer to the standard set by this process area, we need to integrate a formal evaluation of alternatives into some part of the project. The design phase would provide an adequate opportunity for this activity because it marks the first time in the project in which teams are given a lot of freedom in the kinds of decisions they must make. Another section could be added to the software design document for the purpose of stating alternatives to certain design decisions and analyzing any

associated risks. In addition, some general decision analysis guidelines provided as part of the course material would provide teams with a foundation to start from in evaluating alternatives.

Integrated Project Management

Integrated project management deals with the ways in which management tasks incorporate issues of personnel and stakeholder involvement and any standard processes provided by the organization. The CMMI provides two specific goals for this process area: use the project's defined process and coordinate and collaborate with relevant stakeholders. The former goal revolves around the concept of the defined process, an overarching sequence of tasks or project phases that is built off of existing organizational assets as a means to effectively complete a project. The defined process must first be defined itself and then used to guide other areas of project planning and management. It should also be used to inform the creation of a work environment that will comply with organizational standards and meet project needs. Finally, the results of working with the defined process should be recorded along with other useful items and added to the organization's pool of resources. The second goal of integrated project management seeks to ensure that the project is coordinated appropriately with stakeholders including customers, suppliers, contractors, and so on. Collaboration with people in these roles should be managed as specified in the defined process. Additionally, dependencies that hinge on interactions with the relevant stakeholders should remain visible to management (CMMI for Development 145-177).

Our group's compliance for this area was very solid thanks to the format of the senior project course. The structure of the class itself served nicely as a foundation for the defined process, and it included enough details to guide the planning and management processes. Since there isn't really any sort of concrete work environment needed to facilitate the project, it wasn't

necessary to establish one. Our one area of shortcoming in utilizing the defined process was our lack of contribution to the course's process assets. It may not be practical to expect teams to produce entire processes or templates, but even a simple survey requesting recommendations or accounts of the course would be a step in the right direction. Our coordination and collaboration with stakeholders was similarly solid. Our defined process included regular interaction with several important stakeholder roles like CEOs, managers, and customers. Major dependencies and coordination issues involving these roles are not problems for the same reason. However, coordination between the individual personnel in our group did prove to be a legitimate concern. Most of our issues in this area could be resolved by employing basic coordination strategies and by making better use of the provided planning and management software.

Organizational Process Definition

The focus of organizational process definition is on building a growing base of process assets and work standards for the organization. The sole specific practice for this area is to establish organizational process assets, a task that encompasses all of the following: standard processes, lifecycle model descriptions, tailoring criteria, measurement repositories, process asset libraries, and work environment standards. Standard processes, which are the foundation for the defined process described in the previous process area, are general sequences of procedures that lead to the completion of a given task or the generation of a given asset. Tailoring criteria provide a sort of interface to standard processes; through this interface a standard process can be applied to real-world projects. The remaining items are relatively self-explanatory. Measures and process assets related to the organization's standard processes should be stored and managed in an organized way, and work environment standards should

additionally be supplemented with tailoring guidelines for individual projects (CMMI for Development 219-240).

Evaluating compliance for organizational process definition involves more critiquing of the course format and materials and less examination of group processes. Establishing the organizational assets described by this process area would be tantamount to redefining the course itself, a task that would be far outside the scope of our responsibility. Furthermore looking at this area in terms of the assets presented as part of the course provides a favorable outlook. Standard processes have been provided in the form of the guidelines for each required document, and a standard lifecycle model to follow is laid out from the start. We do have a reasonable level of flexibility in the ways we apply our standard processes to our given projects. The guidelines for each of the required documents contain sections that may or may not be applicable or helpful depending on the details of the project and the projected implementation strategies. The implementation phase itself is relatively open in terms of the technology that can be utilized to complete the project. The establishment of a process asset library is satisfied by the course website, which contains document guidelines, example documents, and many general software engineering resources. The main weak point for this process area is the lack of metrics related to the standard processes. This could be remedied by supplying some basic measurements such as estimates of document sizes as well as product quality and reliability.

Organizational Process Focus

The main goal of organizational process focus is to determine what improvements can be made to organizational processes and to see that these improvements are implemented and contributing positively to projects. The CMMI provides three specific goals: determine process improvement opportunities, plan and implement process improvements, and deploy

organizational process assets and incorporate lessons learned. The first goal requires that the organization determine what kinds of processes are needed to match its business model and analyze those processes for their strengths and weaknesses. With these factors in mind, the actual process improvements can be determined. The next specific goal deals with the preliminary planning needed to facilitate the identified process improvements. Process action plans should be developed to elaborate on what infrastructure and personnel are needed to support the process improvements defined previously. Once these plans have been properly executed, work should begin on the final goal. This should be accomplished by distributing the improved processes and process assets throughout the organization. The improved processes should continue to be monitored and analyzed for effectiveness (CMMI for Development 241-260).

Organizational process focus is another process area that is difficult to evaluate in the context of group compliance but can be applied more closely to the course as a whole. However, process improvements concerning the course are in the hands of the department and instructor; no formal mechanism currently exists to allow project groups to directly contribute to the course's processes or assets. As such there is little to discuss regarding compliance for this area. A suggestion made in a prior section could work towards improving this state. That particular recommendation was to implement a simple survey towards the end of the course requesting comments and suggestions on standard processes and assets. While this level of involvement would not cover the goals of the process area as a whole, it would at least be a start towards involving students in process improvement activities.

Organizational Training

The purpose of organizational training is to invest organizational resources towards equipping personnel with the necessary knowledge and work strategies. The first goal provided by the CMMI towards this end is to establish an organizational training capability. Establishing training capability begins with determining what training will be needed in light of projects in the long-term future and also determining what training should be provided by the organization and exactly who should receive it. Plans should be developed for training that needs to be performed in the more immediate future, and the organization should improve its capacity to deliver the necessary training. Providing the given training is the second goal for this process area. In addition to actually performing the training deemed as most needed, records regarding the training and who received it should be established, and assessments should be performed to evaluate the effectiveness of the training (CMMI for Development 275-292).

Compliance for this process area once again mainly concerns the course format. The lectures provided on software engineering and project lifecycles provide a decent analog to the concept of organizational training in that the information is relevant across the groups and not specific to any one project. Disseminating the training is not an issue and neither is establishing long-term training for obvious reasons. Simply observing group processes and work products should suffice for assessing the effectiveness of the given training.

Product Integration

Product integration revolves around getting a product into a suitable state prior to delivering it to a customer. In general the process involves assembling, testing, and deploying the product in question. The first step in this process area is to prepare for product integration. In order to meet this goal, teams should define an order in which to integrate the various components of a product and provide suitable environments in which to support the integration

sequence. Procedures for integrating the components and criteria for determining whether a component is ready to add should likewise be defined and documented. The second step in product integration is to ensure interface compatibility. This can be accomplished by checking interface descriptions to ensure proper understanding and coherence. All interfaces must be managed to keep track of changes and resolve any incompatibility issues that arise between product components. The final goal, assemble product components and deliver the product, begins with teams confirming that all product components are tested and ready for integration. The components should then be assembled according to the previously defined sequence and evaluated afterwards to check for the proper functionality. Finally, the finished product should be packaged and delivered through an appropriate medium (CMMI for Development 293-312).

Our compliance in product integration proved to be a bit sparse. We did not perform any sort of preparations for product integration or pay specific attention to the ways in which the components of our product interfaced. As per our course requirements, we did perform testing on the product components but only once they had been assembled into a whole. Packaging the product was an issue left up until the last minute that could have benefitted from consideration in advance. Clearly there is a lot of room for improvement in this area. One essential task we would need to perform to better meet the standards would be to have an integration sequence and procedures laid out prior to the beginning of the implementation. This information together with the interface descriptions could be included in the design document as part of the architecture. Additionally a statement of delivery could be provided to the instructor during either the design or testing phase to encourage groups to think ahead about what deliverables need to be included and in what format.

Requirements Development

The central idea of this process area is, as expected, to create and evaluate customer and product requirements. The first goal covers comprehension and commitment to customer requirements. In addition to understanding the requirements specifically laid out by the customer, teams should also pay attention to implicit needs that the customer does not directly state. The second goal deals with the requirements for the product, which should include the necessary technical details to help inform the design. Requirements also need to be generated for each of the product components and their respective interfaces. Once both sets of requirements have been produced, teams should perform analysis and validation. All requirements should be examined from the perspective of the product's uses and design in order to anticipate gaps in the stated requirements. Aside from analyzing identified requirements to ensure that all necessary items are discussed, teams should also strive to balance out constraints such as cost, schedule, and performance when examining requirements (CMMI for Development 388-407).

Our performance in requirements development was mixed. The areas in which we best complied were those revolving around developing customer and product requirements. The course supported two requirements phases: one for the requirements definition document intended for customer use and one for the requirements specification document intended for developer use. These documents covered the basic requirements needed to form a foundation for the design and also listed requirements specifically related to the individual components of the product. Product component interfaces, however, were not addressed. Analysis and validation were the issues that received the least amount of attention during the requirements phases. Operational scenarios were not deliberately discussed, and we did not write our requirements in consideration of time or product performance. Furthermore, our documentation did not indicate

that any validation had been performed although we did produce prototypes of our product components to present to the customer.

The main areas we need to improve on in requirements development are analysis and validation. The given requirements documents could be extended to include operational scenarios in the form of possible sequences of user interaction with the product or component. Since each of the project groups produced materials related to customer validation for use in presentations, it would not be a large burden to simply include these materials in the documents as well. Integrating requirements for component interfaces is a less straightforward task because not all products will be divided up in the same way. In order to keep the requirements phases from becoming too complex, it may be acceptable to simply recommend teams consider issues of interfacing in their product component requirements.

Risk Management

As expected the purpose of risk management is to identify potential risks and take steps to prevent these risks from negatively impacting the project. The task of managing risk spans the lifetime of the project, and the CMMI enforces this by providing goals and practices to the same end. The first specific goal, prepare for risk management, requires teams to examine the project in advance for potential sources of risk. These sources should be categorized according to the project areas in which they would most likely arise. Parameters should be defined for each projected risk category to establish metrics such as likelihood of occurrence and potential impact. A risk management strategy should be compiled from the previously mentioned data in addition to prevention and mitigation strategies. Once the given preparations are complete, the next goal is to actually identify and analyze the specific risks. While following the methods laid out in the risk management strategy, teams should perform analyses by evaluating, categorizing, and

prioritizing the identified risks. Each of the lower level risks need to be grouped into broader categories so that prevention and mitigation efforts can be concentrated according to project area. The final step in risk management is mitigation. This goal revolves around developing plans for handling the project's most prominent risks. Backups, alternate approaches, and damage control strategies should all be included in the mitigation plans. In order to make effective use of these plans, risk status should be monitored throughout the project. Mitigation strategies should be enacted at the earliest signs of trouble (CMMI for Development 420-438).

Our risk management strategy was a good first effort, but it did not go into the level of detail required by the CMMI. Our documentation did not categorize risks and conflated sources or risk with the actual lower level risks. We did identify probability parameters for each of our risks, but these figures were generated primarily through guesswork as opposed to more scientific means. Risk evaluation was performed on the basis of determining methods for prevention and mitigation. For each identified risk we specified only one prevention strategy and one mitigation strategy; this would be insufficient in most real world cases.

In order to improve on this foundation, we would need a more structured way of approaching risk management. A systematic method for identifying risk sources, categorizing them, and defining their parameters would prove valuable. Resources that clearly explain these items should be provided at the outset of the project planning phase. The required level of detail for the risk management portion of the planning document should be outlined as well.

Technical Solution

As a process area technical solution guides teams in producing complete designs for products and product components based on the previously identified requirements. The CMMI lays out three specific goals: select product component solutions, develop the design, and

implement the product design. For the first goal project teams should draft several potential solutions for each product component along with criteria to select the best solution in light of the organization's business model. The most effective solution should be selected and documented for each product component. The second goal entails creating the designs themselves. The design should be composed of two phases; in the preliminary phase teams should identify the needed capabilities and the architecture of the overall product. The detailed design phase, on the other hand, examines each product component in the level of detail necessary to completely inform the design. Interfaces need to be designed for product components according to the parameters they must operate within. Teams should also weigh the options of creating, purchasing, or reusing product components over the course of the design phase. For the final goal of the technical solution area, the design should be fully implemented, and product support documentation should be developed to comprehensively cover use of the product (CMMI for Development 456-482).

Our compliance in this process area unsurprisingly stems from the design phase of the project. Stemming from our decisions during the requirements phases, we did split our design into several product components. No alternate solutions were provided for these components; we simply went with our initial ideas. Our process incorporated the idea behind the two-phase preliminary and detailed design, but we did not provide separate documentation for these phases. Because there was no particular need for interfaces between the components of our product, we did not attempt to design any. We also had no need to perform a make, buy, or reuse analysis since all parts of the project had to be developed from the beginning. No product support documentation was produced; we instead opted to design the product to be as straightforward and self-documenting as possible.

Making recommendations for improving compliance during the design phase is particularly difficult because of how tight the last three project phases are in terms of how much content must be developed in the given time. One improvement that shouldn't be too intrusive would be identifying component interface designs as part of the design document. It should be specified that not all detailed designs will require the use of interfaces, and these items should not be included if they are not needed. Another small tweak that could be made to the course format would be to include a statement of the preliminary design prior to the main design phase. Such a statement would simply provide a look at the final feature set of the product and a brief overview of the projected architecture without stating the details of the components.

Validation and Verification

Although validation and verification each have their own process area defined as part of maturity level three, their goals are similar enough to warrant considering them together. The purpose of validation is to ensure that products behave as expected in the end-user environment. Verification tests a given product to show that it complies with its given set of requirements. Both process areas have similar goals; the first goal requires teams to prepare the given tests and testing environment, and the last goal sees the tests actually carried out and analyzed. Verification provides an additional specific goal in which peer reviews are conducted on the given product or component (CMMI for Development 483-513).

The required testing done during the implementation phase mirrored the validation and verification processes fairly well. We were indeed required to design tests both to verify our final software product against our specified requirements and to validate the software's performance in its intended environment. The only real area of concern for our compliance here would be the peer reviews required as part of verification testing. While our group did meet

multiple times to demonstrate the validity of the software, we did not specifically meet to evaluate our finished product in light of the requirements. Requirements were however discussed in an incidental manner during our testing meetings, so the spirit of the peer review was still retained for the most part. The only recommendation for this process area would be to specifically suggest that groups meet to discuss verification in support of the required verification testing.

Conclusions

Compiling and analyzing my research has shown a consistent trend in the application of CMMI process improvements to the academic setting. The idea behind this trend is that, just like the small businesses and agile developers reported, the CMMI can be scaled to fit the context of environments other than mid-to-large sized businesses. The key to applying the CMMI in any setting is to make it work towards your goals instead of working purely towards a given maturity level. Our goal in the senior design course is not to have our work appraised and a maturity level attached to our efforts. Our goal is to simulate a real-world project and environment in order to prepare us to function as well as possible in these settings. When applied to the end of enhancing our education in anticipation of the workplace, the CMMI proves to be a valuable asset. I have shown through my research that much of the senior design course already alludes to the process improvements that would be implemented in a level two or three organization. An allusion rather than strict adherence is the best option to prepare students for the possibility of working with the CMMI in their future careers because the CMMI is first and foremost a tool for enhancement.

Bibliography

Cepeda, Sandra, Suzanne Garcia, and Jacquelyn Langhout. "Is CMMI Useful and Usable in Small Settings? One Example." *CrossTalk* (2008): 14-18. *CrossTalk Online*. Feb. 2008. Web. <http://www.crosstalkonline.org/storage/issue-archives/2008/200802/200802-0-Issue.pdf>.

CMMI FAQ - Brutally Honest, Totally Hip, No Nonsense Answers (a Public Service of Entinex). Entinex, Inc., 31 Oct. 2010. Web. <http://www.cmmifaq.info/>.

CMMI for Development Version 1.2. SEI CMU, Aug. 2006. Web. <http://www.sei.cmu.edu/reports/06tr008.pdf>.

Dalton, Jeff. "AgileCMMI." *Www.upstartsystems.com/*. Broadsword, 2009. Web. < http://www.upstartsystems.com/Portals/1/Agile%20CMMI%20-%20DC%20SPIN.pdf>.

Kasse, Tim. *Practical Insight into CMMI*. Boston, MA: Artech House, 2004. EBook.

Leithiser, Robert, and Drew Hamilton. "Agile Versus CMMI - Process Template Selection and Integration with Microsoft Team Foundation Server." Proc. of ACM Southeast Regional Conference. 2008. Web.