
INFORMATION MANAGEMENT: STRATEGY, SYSTEMS, AND TECHNOLOGY

INTRODUCTION TO PROCESS IMPROVEMENT AND THE CMMI

Margaret K. Kulpa and Kent A. Johnson

INSIDE

What is a Process?; Models; Business Goals and Objectives; Problems

WHAT IS A PROCESS?

A process is a series of steps that helps to solve a problem. The steps must be defined in such a way as to be unambiguous — that is, readily understood and capable of being followed in a consistent manner by anyone using the process. Why do we want to do things consistently? Are we promoting turning the workers into robots? No. What focusing on process does for your organization is to reduce redundant work. Why recreate the wheel every time you begin a new project? If you are required to submit a project plan, why not have a procedure that tells you how to write one, plus an example of one from which you can copy and paste? Is that not easier than just wringing your hands and worrying about it and sweating blood devising a new project plan for your new project? OK — you may not be a project manager, so you do not like the example about the project plan. Suppose you are a programmer. Does your manager ever ask you how difficult a particular program might be? Does he ever ask you how long it will take you to code each program? Or does he just give you a list of programs assigned to you and tell you when they are due? Do you ever have heartburn over his estimates? Do you ever say anything, or do you just work unpaid overtime because of his bozo schedule? Having a process for estimating schedules that you have input to will create a more realistic schedule and help relieve some of the burden on you of having to adhere to something that just does not make sense. Does this scheduling process always work perfectly? Can you just say, “No, this schedule stinks, so my manager must change it?” Of course not. But it does give you some room for negotiation.

PAYOFF IDEA

For those who are just beginning a process improvement program, this article focuses on defining what a process is and why it is important; gives a brief comparison of improvement models; offers suggestions of where and how to start your program; and discusses some of the problems you can expect to encounter.

Processes are like recipes. A recipe tells you the ingredients, how to mix the ingredients, what temperature to use, and how long to cook something. But it does not teach you the techniques of slicing and dicing, mixing and beating, whipping and blanching, grilling and poaching, etc. And recipes also leave room for some experimentation and modification. Some recipes even give you suggestions on how to make changes to the dish.

A process as used in process improvement is usually defined at a somewhat high level, with associated procedures supporting the process. The procedures are written in much more detail than the process. Examples follow.

Assume that an organization is focusing on creating a risk management process. This risk process is being developed because the project managers in the organization have been unable to predict troubling issues that affect the delivery of their products to the customer. Perhaps they have delivered late because of high staff turnover and were unable to get enough people with the proper skills on their projects; or the people they get are pulled onto other projects when those projects run into trouble. Another risk may be that the budgets for the projects are usually overrun. So, all of the project managers get together and determine a risk management process that they feel will cover all (or at least most) of the risks they encounter. The risk management *process* they come up with is:

- Identify the risk.
- Analyze the risk.
- Categorize the seriousness and probability of the risk.
- Mitigate the risk.

Assuming these managers feel that they have done a brilliant job, what is the problem with this process? It is too general. If this process was distributed to all managers to follow, each manager would interpret *how* to do this process differently. We are trying to find a way to document how we do work in the organization in a consistent way so that everyone will do things somewhat similarly, and so that people can benefit from the good work being done by others.

So, now we go back to the managers and say, “Great — you have a process. How do we do the steps in your process?” Well, that is a different problem. The managers are tasked with devising *procedures* for *how* to do *what* they have written as the steps in the process.

Using our process example, the first item says to “Identify the risk.” The managers would need to come up with how they identify risks. One of the ways they might do this identification is to start tracking the problems they have in delivering products, and then find trends. From the list of trends, create a list of the ten most frequently occurring problems on the projects. The procedures would then discuss how to use the information in the list to estimate risks that might occur on another project. The third item in the list of steps in the risk management process says to “categorize the seriousness and probability of the risk.” Maybe your organization defines risks as 1 — most critical and most likely to occur; 2 — critical but work may continue; and 3 — not critical, work may continue, fix this

problem during the next phase or release. The managers would need to come up with procedures for how to determine what would put a risk into category 1, 2, or 3.

These examples are just simplistic examples used to illustrate process versus procedures.

So why is focusing on process important? Why not focus on the product or the people or the technology used? Let me explain.

Writing standards for what a Requirements Specification should look like is a product focus. You are focusing on the template itself. For example, you might want your Requirements Spec to list all of the requirements for the system, and categorize them as to whether they are system-level requirements, software requirements, hardware requirements, safety requirements, performance requirements, etc. Great — this is a good thing to do; but how does anyone know which categories they fit in? This is where a requirements process comes in. It would tell you not only what the Spec should look like, but *how* to write it — how to fill in the blanks for each paragraph in the Spec. A requirements process would also tell you how to elicit requirements from your stakeholders (e.g., customers, end users, requirements analysts) and how to manage the changes to the requirements. The *product* focus would then continue on to what a Design Spec should look like, what coding standards should be followed, what test cases should consist of, etc. The *process* focus would then give guidelines to the people responsible for doing this work on *how* to do it.

Why not focus on the people? Whenever anything goes wrong on a project, the first, visceral reaction is to blame the people doing the work. This is highly unmotivating. This is not to say that people are not important; they are the most important part of any project or any work undertaken. But not everyone can be as brilliant as everyone needs to be everyday. It is not like we wake up one day out of 300 days and say, “Yesterday I was just OK. Today I will be brilliant.” Focusing on process puts the emphasis on having good processes to follow, not on hiring only brilliant people. Rather than having people work harder, have them work smarter. That is what process does for you.

Why not focus on technology? Have you ever heard the saying “garbage in, garbage out?” That is what plastering new technology onto an old problem does for you. Enterprises pay enormous sums to convert existing systems from one language or database to another. There are even tools now that will do that for you (with a lot of work). What do you get? The same system you always had with both the old problems and the new ones. Technology does not provide a quick fix; but it is the one answer that executives are most likely to choose because technology is easily quantifiable and easily budgeted. Look at the dot.com bust. Most of those companies sold quick-fix technologies without any underlying analysis of the problems organizations faced. Most of the dot.coms that operated without this sort of planning are out of business. Technology is our friend — but it is not the only answer to our problems.

Why focus on process? What personal benefit can you gain from all this work? What is in it for you? The following examples are drawn from personal experience in some of the organizations we have helped along this path.

- *Configuration management.* One individual had spent hours trying to find the correct version of source code to make a simple change. He was never sure before whether he had the latest copy of the source code. Now, after following procedures for change control, he is reasonably sure that the code he is using to make updates is actually the code used in production. No more searching for hours for the right program.
- *Testing* Prior to this effort, developers handed the system to the testers and told them, “Here — write a test plan and test this. These are the changes I made.” The test people were never quite sure how thorough their testing was, and they spent hours trying to figure out what the actual requirements for the system were. (You test to ensure the requirements have been met.) Now, with a process for the Requirements Traceability Matrix and the Requirements Specification, the testers spend less time figuring out what to do, and more time actually testing. It has greatly simplified their jobs and greatly improved the testing of the resulting products.
- *Planning* Prior to this effort, the organization could not predict the number of projects that needed to be scheduled ahead of time. Now, with the use of the process for devising a Statement of Work and the focus on the Project Planning process area, the organization is aware of the number of projects requested, what their initial requirements are, the approximate number of staff needed, the approximate size and complexity of the project, and how to prioritize the projects. Requests for support have actually been deferred based on these measures. Time is not wasted on developing systems that will not be used or fixing problems that go away by themselves.
- *Communication.* There is more communication up and down the chain of command as well as across the organization. For example, the director of software engineering is talking to developers, and in some cases, the developers are talking back. This is good. Quality assurance (QA) is reviewing products and processes across several projects. They are seeing the results of these processes and the problems, as well as the differences between the ways project teams perform. QA is also talking to the EPG (the process improvement team) and in some cases, swaying them to change some decisions made, based on how things are actually working (or not working).

So, is process the only answer? No. Process is part of the answer. Process, when supported by training, enough money, enough skilled people, proper tools, and management commitment, can help your organization.

MODELS

What is a model and why do I have to use one? A model is considered a guideline of best practices found by studying other, highly functioning and successful organizations. A model does not contain the steps needed or the sequence of steps needed to implement a process improvement program. The model used simply says, “this is a good thing to do, and this is a good thing to do.” For example, the Project Planning process area suggests that you write a project plan. The Requirements Management process area recommends that you track changes to requirements.

There are many models to choose from, depending on the problems in your organization that you want to solve. Why use a model? We have worked in many organizations that just decided to “improve.” Without using a model as your basis of reference, you have nothing to plan your improvement around, and nothing to measure your results against. Some organizations have decided that they did not like the guidelines in the models used for process improvement in the industry, so they created their own. Most of these organizations failed. It is not easy to write a model. It takes a long time and it costs a lot of money. And remember, the models are summaries of the best practices of effective, successful organizations. So, it would behoove someone to follow most of the practices documented in these models. Most models allow an organization to substitute alternative practices for those practices in the chosen model that do not fit the organization. But beware: the more alternatives you select, the more you deviate from best practices in a model, and the less likely you are of improving the problems in your organization.

A brief overview of some of the more frequently used models follows. Do not consider this overview exhaustive, as we simply summarize the basic thrust of each model. We do not purport to be experts in all of these models. Those of you who are experts in any of the models may take exception to some of our statements. These statements are offered once again only as high-level summarizations.

“In the beginning,” there were ISO and the CMM. ISO stands for International Standards Organization. The ISO 9000/9001 series generates a fundamental quality management framework. The ISO 9000 series is a set of documents that discusses quality systems to use when instituting quality assurance in an organization or enterprise. ISO 9000 itself is a guideline that directs the user as to which set of documents to use and the interrelationship of quality concepts. ISO 9001, 9002, and 9003 deal with external quality assurance pursuits, while ISO 9004 deals with internal quality assurance. ISO 9001 is used to ensure that quality systems are delivered by the supplier during several stages of creation (which may include design, development, production, installation, and servicing). This document is the most pertinent to software development and maintenance. ISO 9000-3 is used when applying ISO 9001 to the development, supply, and maintenance of software. ISO 9001 requires that a documented quality system be implemented, with procedures and instructions. ISO 9000-3 further specifies that this quality system be integrated throughout the entire life cycle.

The CMM stands for the Capability Maturity Model. Most people call it the CMM for Software. The reason for this appellation is that after the CMM was developed, several more CMMs relating to different areas were generated (e.g., Systems Engineering, Acquisition). The CMM was created to help manage organizations that develop software. The CMM was created by analyzing the activities of highly functioning software organizations; that is, those organizations that consistently delivered software systems to their customers on time, within the budget, and that actually worked. These activities became the 316 key practices in the CMM, and the practices themselves were grouped into categories called Key Process Areas. There are 18 Key Process Areas that focus on the best practices found among the organizations reviewed. The Key Process Areas concentrate on such things as managing requirements, managing changes, creating project plans, tracking estimates against actuals, implementing quality assurance activities, instituting peer reviews, and training personnel in processes related to their job duties.

What is the difference between the CMM and ISO? Well, both were developed to improve the quality of systems. ISO was developed in Brussels, Belgium, and related originally to the manufacturing arena. The CMM was developed in the United States for managing the development of software systems. Over the years, the CMM made its journey across the ocean and is now used almost as much internationally as within the United States alone. ISO also “crossed the pond,” that is, made in-roads into the United States in manufacturing businesses, concentrating most often in the Midwest.

ISO focuses mainly on broader issues of quality within an entire enterprise or company. Executives of these organizations are often interviewed. The main product of ISO is the creation of a quality manual that discusses quality initiatives to be implemented throughout the enterprise. The CMM takes a much more limited view. It focuses on only software-intensive projects; it does not look at an entire enterprise or company. It describes an organization (admittedly, ambiguously) as several projects managed under one director (an example). Interviews may include an executive manager or two, but mostly focus on project managers and their team members.

If your organization is ISO certified, does that mean you are automatically CMM Level 3? No — it is like comparing apples and oranges. If you are CMM Level 3, does that mean you are ISO certified? No. Once again, while the two methods have similar goals, they are very different in implementation and scope. Chapter 4 in ISO 9001 is about five pages long, while Sections 5, 6, and 7 of ISO 9000-3 are about eleven pages long. The CMM is over 400 pages long. So, clearly, the two models are different.

The models we are the most familiar with are the Capability Maturity Model (CMM) for Software and the CMMI. CMM and CMMI implement processes that reflect best practices found in industry. The CMM focuses on the software domain of organizations. However, since the problems with software organizations were deemed to fall mostly in the management area, this book can be, and has been, used not only in software organizations, but broadened to include most management situations.

The model behind this article is the Capability Maturity Model Integrated (CMMI). The CMMI officially expanded the scope of the CMM from software to the entire enterprise of organizations. That expansion includes systems engineering as well as software engineering, integrated product and process development (specialized teams that design and develop systems), and acquisition (procuring systems, and monitoring the procurement and management of contracts awarded).

Other models in use include the Federal Aviation Administration's (FAA's) CMM, which builds upon the original CMM and other models as they relate to FAA issues. However, parts of this model can also be extracted and used in other business endeavors (not just aviation).

Six Sigma is also being used more and more frequently. Six Sigma attempts to reduce the variation in processes to a very small number. Six Sigma focuses on improvements and measures that will lower the cost of doing business. Six Sigma consists of defining project goals and objectives, measuring narrow ranges of potential causes and establishing a baseline, analyzing data to identify trends and causes of the deviations, improving the processes at the identified cause level, and controlling the problem.

Some organizations are now merging Six Sigma and CMM/CMMI activities into one process improvement effort. Six Sigma focuses heavily on selecting a limited number of issues, measuring the problem (e.g., number of defects in a product line, excessive time to build a product), and then measuring the results (e.g., the effectiveness) of the fix. CMM/CMMI focuses on implementing best practices found in the industry and changing the infrastructure and culture of an organization. The two can complement one another. However, we feel that Six Sigma is better performed on only those organizations that have already been rated at a strong CMM/CMMI Maturity Level 3 or higher. Six Sigma requires a high degree of sophistication in aligning business objectives with Six Sigma techniques, as well as a high degree of sophistication in other areas of management (business, quality, process, and change).

BUSINESS GOALS AND OBJECTIVES

Much is made about aligning your process improvement effort with the business goals that the organization is trying to achieve. This alignment is easier said than done. Most organizations just beginning process improvement do not really have clearly defined business objectives. In fact, what we most often hear from executives when we ask them what they are trying to accomplish by doing process improvement is to reduce the number of people they need to staff projects, and have the people who survive the cut become more productive — that is, have the people do more work in less time. Of course, that does not include the executive doing the talking. Well, process improvement will not allow you to significantly reduce your task force. In fact, especially over the short term, you may have to actually hire more people to structure process improvement efforts and staff them adequately. So this “goal” is simply wrong.

Another most often heard response to “What are your business goals as they relate to process improvement?” is to get Maturity Level 3 for contract awards. If these organizations could buy Level 3 (and there is some discussion as to the ability to buy a level rating), these organizations would be most willing to go that route.

The other problem we run into is when we ask, “What are your business objectives?”, we repeatedly hear one answer — an incredulous, “Why, to make money of course!” The point of this discussion is that most organizations beginning process improvement are simply not sophisticated enough to have clear goals.

There is a technique that can be used to help define business goals and process improvement focus areas. It is called the Goal–Question–Metric technique. In this method, a workshop is held. It should actually be called the Problem–Goal–Question–Metric approach. During the workshop, a list of the most common problems found in the organization is presented. Questions are then asked relating to the problem, and the areas of the CMMI are used to help focus ensuing process improvement efforts. The approach starts with a business goal and works backward to identify improvement actions to achieve that goal. Here is a basic example:

- *Problem:* We cannot keep up with the number of requirements changes.
- *Goal:* Improve our requirements change process.
- *Question:* How can we improve our requirements change process?
- *Metric:* Number of requirements changes submitted, approved, implemented, or cancelled versus the number of original requirements documented; time it takes to implement a change in requirements.
- *Associated Process Areas:* Requirements Management (Level 2), Project Planning (Level 2), Product and Process Quality Assurance (Level 2), Requirements Development (Level 3).

The main difference between this approach and addressing the seven process areas in Level 2 of the CMMI staged representation simultaneously is that structuring your process improvement program to focus on key problems in your organization helps define the scope of initial efforts and their sequence. Some members of your process improvement team and of your organization will find this approach more relevant to their everyday work and will therefore be more enthusiastic about the program.

Although we have discussed this Goal–Question–Metric approach, we do not strongly recommend it for low-maturity organizations because most beginning organizations do not have a clear understanding of business objectives and business goals. They also will not readily admit that there are problems in the organization. The example we used was pertinent to requirements problems in systems. However, most true business objectives are at a much higher level. For example, an organization that clearly understands business objectives would state that one of its business objectives would be to improve customer satisfaction by 10 percent by reducing the number of defects embedded in their systems and

delivered to their customers by 15 percent. To truly be effective, this type of business objective, backed up by this metric, requires a sophisticated approach to identifying problems in organizations; fully defining the problem; relating the problem to existing, documented business goals; measuring the current process and its results; and measuring the expected versus realized outcome. This approach is way beyond most low-maturity organizations.

So, what to do? Because the approach above is used to structure your improvement efforts, why not use an officially approved assessment method? We suggest that if you are using the CMMI, that a SCAMPI be used. The SCAMPI is a method whereby a team is formed that examines any processes (both formal and, in low-maturity organizations, informal) that exist in an organization and rates their strengths and weaknesses as they relate to the CMMI. The organization will then decide which process areas to focus on first.

PROBLEMS

The process focus is not without its challenges. That is a nice way of saying, “it ain’t perfect.” If you look at the last line of one of the preceding paragraphs, it says, “Process, when supported by training, enough money, enough skilled people, proper tools, and management commitment, can help your organization.” Most people do not understand process until they have been struggling with it for at least a year. Training is considered a dirty word in some organizations — “If they don’t have the skills, we don’t hire them,” or “We only hire skilled people.” And as most of you know, if you are really good at something, your organization is going to place you on one project full-time, and then pull you off that project and place you on one in trouble (full-time), and then have you “help out” on another project in your “spare” time. And when you do a wonderful job on the project in crisis and bring it in on time, your reward is to get your butt kicked on the other two projects because you are late delivering to them. This example is an example of a dysfunctional, low-maturity, yet commonly found organization.

The same problems occur in process improvement. You seldom get the “right” people. You get a lot of people just waiting to retire, or out of rehab, or summer interns, or “burnouts,” or people who just cannot code well. You staff up with five or six full-time process improvement team members, only to find them pulled off when their former projects run into trouble. Training for process improvement is absolutely mandatory; but once the people get the training and start to do the work, they become overwhelmed and decide to go back to their old jobs, or leave the company for greener pastures. Or, the organization decides to give a “condensed” version of the training because it is too expensive — both in dollars spent and in time spent not doing “real work.” Or, the organization buys a process improvement tool guaranteed to get you your level rating in one year, only to discover that you cannot just plug it in — you have to actually write procedures and do work.

The answer? Well, just like anything else, there is no one answer except maybe the following — tell your boss (or whoever’s idea this whole process improve-

ment thing was) in a tactful way to put his money where his mouth is. If he wants it, it will cost him — in dollars, people, and time, including his time. Do not commit to “Level 2 in two years,” or any other such slogan. Commit to trying to improve your organization. And mention to the boss that most improvement models focus improvement activities on *management* — not worker bees and practitioners. See if he then wants to continue. Educating your boss, as well as the rest of the organization, is key to a smooth transition to the process improvement path.

Process improvement is really about *change*. No one really likes to change, unless the current way of doing things is so onerous and repulsive that they just cannot stand it anymore. Most folks do not get to this state when working for an organization — they simply go elsewhere. So, no one in your organization will be really excited to adopt a program that makes them change the way they do their jobs. Most people have developed over the years their own system or process for doing their jobs. And if you are the process improvement specialist, you will be responsible for making them change. And guess what? You will also have to change the way you do your job. One way to get people to buy into this whole change/process thing is to have them write the procedures. If they cannot or will not write, then get them to at least participate in meetings where they can voice their issues and contribute to creating the process for doing their work.

SUMMARY

This article has introduced the novice process improvement individual to some terms and concepts commonly used, and why process improvement is important. This article is a high-level overview. For more information, please read *Interpreting the CMMI: A Process Improvement Approach*, by Margaret K. Kulpa and Kent A. Johnson. The power of the CMMI lies in that it lets you define your job — how you do it and what gets produced from it. But with power comes responsibility. If you want to control how you do your job, you need to fully participate in the process improvement program.

Mistakes are a good thing. It shows you are trying. It shows you are improving after your mistakes go down. However, the predominant mantra in the industry is that mistakes are bad. If you make a mistake, you get hammered for it. There is no middle ground, no shades of gray. You are either 100 percent right or 100 percent wrong. With process improvement, if no mistakes are made, then the processes are probably not being used. Processes are generally not written “right” or perfectly the first time around; nor are they initially implemented and practiced correctly.

Also remember that although we say that process improvement involves skilled professionals with an objective, unbiased, professional attitude toward their work, everyone on the process improvement team has a large amount of personal investment in this effort. So things do tend to get *personal*. Try to see the “big picture” and try to be flexible.

Margaret Kulpa (margaret.kulpa@agiledigm.com) is currently the Chief Operating Officer of AgileDigm, Incorporated. As such, she provides senior-level management support for SEI Initiatives and Software Process Improvement. Her work includes writing materials and providing training for Software Capability Evaluations (SCEs), Capability Maturity Model training, and software process improvement training, as well as performing these evaluations and improvement activities.

Kent A. Johnson (kent.johnson@teraquest.com) is the Director of Systems Engineering for TeraQuest and an SEI-authorized CBA IPI Lead Assessor, an authorized SCAMPI Lead Assessor, and an authorized CMMI Instructor. Kent's background includes over 25 years experience in development of complex software-intensive systems.
