

Are the Right People Measuring the Right Things? A Lean Path to Achieving Business Objectives

Paul E. McMahon
PEM Systems

This article provides experiences and guidance applying Lean and Agile techniques together with the Capability Maturity Model® (CMM) Integration (CMMI®) framework to aid measurement and to help in achieving business objectives. While many believe the CMMI produces non-Lean practices, an underlying premise of the article is that both Lean and Agile techniques can be CMMI compliant, and when used together with selective CMMI Level 4 and 5 practices can help organizations achieve business objectives faster. This article goes beyond measurement by addressing the more comprehensive question: Are the right people measuring the right things and taking the right actions at the right time? A case study, six lessons, two process improvement insights related to business objectives, and a commonly held myth about the CMMI framework are included.

The effectiveness of the CMMI framework at helping organizations achieve business objectives is currently a hotly debated topic [1]. One of my clients responded to my recent suggestion to consider using the CMMI and Agile techniques together by stating: *We do not place much value in the CMMI because we have not seen a difference between Level 4 and 5 organizations and those rated lower.* At a recent conference in a presentation about high process-maturity project failures, we also heard some failures were due to *non-process effects such as people* [2]. To help us understand what might be going on, I would like to share a case study.

Case Study Background

I have a client who has been using the CMM/CMMI framework for many years to aid process improvement. According to traditional literature, they appear to be doing the right things.

They have discussed and captured their business objectives using the Goal-Question-Metric technique, and they have aligned their measures with these objectives. They have standardized their processes and trained their people [3]. This client has multiple product lines. Their processes and training include a strong emphasis on product baseline management with disciplined change approvals. Their established and documented business objectives include increasing off-the-shelf product sales, reducing unique customer customization, and meeting cost and schedule commitments.

To help achieve these objectives, the client has been striving to increase management visibility of work through the use of standard product and process metrics. This all seems to make sense. So a natural question arises: Are they achieving their business objectives?

While isolated success stories exist,

most senior managers – as well as the company metrics – indicate the organization has fallen far short of its goals.

While the organization continues to propose and bid off-the-shelf solutions, standard metrics show significant unplanned product changes and cost and schedule overruns. A frequent refrain heard at senior management reviews is the following: “Why are we making all these unplanned and un-bid changes?”

So what happened? While this organization appears to have followed the prevailing wisdom in deploying sound processes, why have they failed to achieve their objectives?

This case is not unique. Over the past 10 years, I have observed variants of this pattern in multiple organizations. A closer look at this case may provide insight that can help answer why many organizations seeking higher process maturities are failing to enjoy the promised results of their investments. Let us start with measurement fundamentals.

Measurement Fundamentals

Few, if any, would disagree with the need to measure. But before any organization starts a measurement program, clear objectives and a plan are needed. A fundamental purpose of measurement is to guide management decision making [4]. But how do you manage these measurements to facilitate their effective use?

There is a great deal of literature available today describing the importance of employing a *project management database as the basis for retaining process measurements for process management* [5]. Within the Quantitative Project Management (QPM) Process Area of the CMMI, Specific Practice 2.4 identifies the expectation of recording statistical and quality management data in the organization’s measurement repository [1].

Another fundamental question faced is

deciding what to measure. Watts Humphrey describes three categories of measures: product, process, and resource [3]. Examples of each are:

- Process measure: Defects found by phase responsible.
- Product measure: Defects found by product component.
- Resource measure: Hours per defect fix.

These categories are referred to as *foundation* measures, and it is expected that organizations use these as a *starting point* to derive more meaningful and useful measures *specific* to their business needs [3].

Measurement in Case Study

In our case study, measures in all three of the categories described previously were collected for years, retained in an organizational measurement repository, and the purpose of the measurement program was documented and measurement training deployed.

Independent Analysis and Findings

Due to concerns related to the non-achievement of business objectives, I was asked by my client to conduct an independent analysis. My analysis began with an examination of the data that had been collected for years in the organizational measurement repository. I observed from the *Defects Found By Phase Responsible* data a high percentage of defects being injected late in the development cycle (e.g., test and integration phases). But when I talked to developers, I heard that the majority of their problems were due to vague requirements that did not receive appropriate analysis during the early phases (e.g., requirements and analysis phases).

First Key Observation

My first key observation was this disconnect between the objective data in the organizational measurement repository

built up over multiple years and multiple projects, and what I was hearing from talking to people in the trenches. Rather than selecting the phase that was actually responsible for the defect (e.g., requirements) many people had erroneously selected the current project phase (e.g., integration), causing the organizational repository to be inaccurate.

More Findings

Trying to better understand what was going on, I interviewed a number of workers and asked them to describe how they do their jobs. Prior to these interviews, I had reviewed the organization's standard processes. This was a high-process maturity organization that had previously achieved a formal CMM Level 3 rating.

As the workers described how they did their job, I detected more disconnects between how they said they worked and what was written in their processes.

I asked one developer to describe the product baseline management process. I expected to hear about the disciplined change approval process that was described in their standard processes and training material. But he said, *it's not how we really work*. When I asked the developer to explain further, he said, "We propose things that are similar to what we are going to do, but not exactly. We propose based on where we think the product will be in the future when we think the job will come in. Often those assumptions are wrong."

Second Key Observation

My second key observation was this disconnect between the documented processes and what I was hearing was happening in the trenches.

Stepping Back

I have found the two observations described, to varying degrees, as common patterns in many organizations. Let us explore how organizations get into these situations and how this affects the attainment of business objectives.

As-Is Versus To-Be Process

When process improvement efforts are initiated, there are usually two relevant process views referred to as the *as-is* process and the *to-be* process. The usual approach is to first capture the *as-is*, then discuss weaknesses so appropriate resources can be applied to move the organization toward the desired *to-be*.

If we do not know the *as-is* then we do not know how big a *stretch* it is for the

organization to get to the desired *to-be*. The hardest part of process improvement is not defining processes, but deploying and teaching those in the trenches appropriate changes in behavior.

Lesson One: The First Step to the Right Measures Is Capturing the Real As-Is Process

Often, the *as-is* process does not receive appropriate attention. The common argument goes like this: We are looking at getting better, so does it not make sense to focus on where we want to be? The answer is yes and no.

While it is true that we want to create a clear vision of where we are going, we also need to understand what it takes to get there, which first requires an understanding of where we are. By jumping over the *as-is*, we sometimes skip *critical dialogue* that helps us understand what and why we do what we do today. This dialogue helps us understand which weaknesses are highest priority and most in need of addressing now. This will, in turn, affect the right things to measure now.

Digging Deeper: Looking For Candidate Root Causes

I wanted to find the root cause of why the organization was not achieving its business objectives. I had heard a number of projects were currently overrunning cost and schedule, so I asked a number of workers, "Does the company underestimate when it bids?" I received a mix of answers.

One project engineer responded, "No, our bids are okay. But we often do not get the hardware ordered and installed in time to meet the software integration schedule."

Another said, "... the bid was okay given the assumptions we made at the time of proposal, but when we find that the assumed baseline product functionality is not there after contract award, we do not adjust the schedule or resources for the additional work we now know we have to do."

Another said, "Yes, we sometimes underestimate because the people who do the bids do not always understand all the pieces of the products they are proposing. After we win, we find out there are impacts that were never planned."

I now had some areas to investigate looking for candidate root causes which included the following:

- Hardware procurement and installa-

tion processes.

- Plans and schedule update processes.
- But I did not have quantitative data to back up what I was hearing, so I went back to the organization's measurement repository. Unfortunately, when I looked again at the measures being collected, I found it difficult to tie the existing historical data to potential candidate root causes.

This data, which had been collected over many years, were the typical measures found in textbooks (e.g., defects by phase responsible, defects by product component, hours per defect). These traditional measures were not *specific* and context relevant enough to help in identifying and analyzing the root cause candidates.

Lesson Two: Company Standard Metrics are Often Insufficient for Real Process Improvement

In the book, "Measuring the Software Process" the author tells us that when planning for measurement "... experience has shown that it is important to identify the critical factors," and that "... critical factors often arise from concerns, problems or issues that represent levels of risk that threaten your ability to meet your goals ... or commitments [5]."

In our case study, cost and schedule commitments were apparently being threatened by late hardware and unplanned, un-bid work. Projects were implementing the standard organizational metrics that had been in place for years, and more context-relevant measures had not been derived. These standard measures were inadequate and could not provide the specific objective data required to address the current bottlenecks in the organization.

Lesson Three: Derive Specific Measures for Needed Insight

Examples of specific measures that could potentially help the organization include:

- Cycle time to get critical path hardware on order.
- Cycle time to install and test critical hardware.
- Counts and cost of unplanned changes to product components.

We derived these measures by asking questions related to the organization's business objectives and what we heard from talking to the people in the trenches such as the following:

- Why are we not getting the hardware installed on time?
- Why do the schedules not reflect all the real work?

How Did This Company Get Into This Situation?

Many of us were taught we need to gather large volumes of data before analyzing and using it. The argument often goes like this: We should not use the data for analysis until we have collected a sufficient amount to ensure it is statistically significant. But the flip side is that the value of data erodes over time. Today's projects use shorter development cycles and environments, technology, tools, and people change fast causing the data to become less relevant faster.

Referring back to the first key observation: Why did the data tell a different story than the people in our case study? The answer to this question is because the people were telling us what was happening today from the context of the projects they were currently working on. The data had been collected over many years, and over many diverse projects and environments, and was not well monitored for accuracy.

In "Understanding Variation," Don Wheeler tells us:

Much of the managerial data in use today consists of aggregated counts. Such data tends to be virtually useless in identifying the nature of problems ... The work of process improvement requires specific measures and contextual knowledge [6].

Lesson Four: Someone Needs to Care About the Data

In our case study, people had been trained in the importance of the company standard metrics. Nevertheless, with schedule pressures it had become commonplace for the data to be entered quickly and often without adequate consideration for accuracy. Since no one had an immediate reason to care about the data, there was no motivator for people to take the time to ensure they were entering accurate information.

Lesson Five: Use Small Project Teams to Derive Meaningful Measures and Review and Refine in Short Cycles

In the book, "Measuring the Software Process," an example of data collection using the Personal Software ProcessSM (PSPSM) is provided [5]. The point is made that *because of a short feedback cycle, the engi-*

neers realize the effect of the PSP, and use their own performance data to gauge their improvements. With PSP, developers save their own performance data and, therefore, have immediate access to it.

One of the reasons PSP works is because the engineer gets that immediate and personal feedback concerning their own behavior. If the data reflected their behavior from years earlier, or reflected another engineer's behavior from a different project, they would be less likely to take action to improve.

The same measurement principles that work for individuals, such as PSP, can be effective for small teams. The closer the

“The same measurement principles that work for individuals, such as PSP, can be effective for small teams. The closer the data is in time and environment to the current team situation, the greater the likelihood it will be used for real improvement.”

data is in time and environment to the current team situation, the greater the likelihood it will be used for real improvement.

Why Involve People Through Small Teams?

Experience has shown that using small teams works better than large teams because they tend to focus faster and take ownership of real project issues. This leads to identifying the right specific measures and, ultimately, the right process improvements to help an organization meet its objectives faster by addressing key bottlenecks. This is particularly true when the issues faced cross department boundaries, as was the situation in our case study.

This is consistent with Lean principles such as *empower the team* and *eliminate waste* [7]. For related information, refer to Lean manufacturing experiences [8, 9, 10]. This is also consistent with what Agile teams do through their retrospective or reflections workshops [11, 12].

It may be surprising to learn that these techniques also support the recommendations provided through the CMMI guidelines within the Level 4 QPM Process Area. A CMMI guidelines tip states that, "... the specific practices of QPM are best implemented by those who actually execute the project's defined process – not by management or consulting statisticians only." Another tip in the guideline states that, "... when effectively implemented, QPM empowers individuals and teams by enabling them to accurately estimate and make commitments to these estimates with confidence [1]."

This leads to a question: When should an organization consider implementing higher CMMI Level 4 and 5 practices? Before answering this question, let us look closer at the relationship between Agile, Lean, and CMMI Levels 4 and 5.

Agile, Lean, and CMMI Levels 4 and 5

The CMMI guidelines tell us (through a sidebar tip) with respect to Level 5 Causal Analysis and Resolution (CAR), "Although this PA is commonly used for defects, you also can use it for problems such as schedule overruns and inadequate response times that should not be considered defects" [1].

What I find interesting about this tip is the recognition within the CMMI guidelines that defects and problems may both be addressed through common practices. This may alarm some traditionalists, but a primary focus of Agile is value to the customer rather than being overly concerned with categorizing work as a defect or a requirement. This view is also consistent with *Lean thinking* where a defect is defined to be anything that does not meet the *customer needs* [13].

Lean Six Sigma has evolved from the two initiatives Six Sigma and Lean. The focus of Six Sigma has been on reducing defects and reducing cycle time through measurement [14]. This is also a key focus of CMMI Levels 4 and 5 practices. The methods known as *Lean* focus on improving *process flow and speed* [13]. While the roots of Agile methods are proven small-team techniques, they have also drawn heavily from Lean manufacturing experiences [11, 12].

One argument I have heard by Lean and Agile proponents against the CMMI is *why do I need to wait until Level 5 to analyze and fix problems?* This is in reference to where CAR is placed in the staged representation of the CMMI [1]. The answer: You do not, and you should not.

SM Personal Software Process and PSP are service marks of Carnegie Mellon University.

Returning to Our Case Study

In our case study, we addressed two real problems: Cost/schedule overruns, and late hardware. Both had a direct impact on the customer. We needed specific data to verify we were tackling the correct root cause. As soon as the data was gathered, we needed to analyze it and act on it. Fundamental to Lean thinking is continuously identifying the next critical bottleneck, analyzing it, and then taking action to remove it as rapidly as possible [8, 9].

Experience has shown it is on the *analysis* and *action* side where traditional process improvement efforts often fall short. CMMI Level 2 and 3 practices are of fundamental importance, and a level of competency must be achieved in these practices before taking on any Level 4 and 5 practices. But too often, organizations get overly focused on Level 2 and 3 practices to the exclusion of valuable *business-focused* efforts that can be well supported through the higher level practices. Also, as seen in our case study, many organizations that employ only the Level 2 Measurement and Analysis Process Area to drive their measurement initiatives tend to focus on collecting and storing company standard metrics rather than on the critical *analysis*, *communication*, and *actions* to improve.

Process Improvement Insights Related to Business Objectives

Alistair Cockburn has observed a commonality between engineering and manufacturing. You can observe it, "... once you notice *decisions* as the product that moves through a network of *people*" [15]. This observation can be taken a step further to help us gain process improvement insights related to business objectives.

Returning to the case study where the hardware is late causing the software integration schedule to slip, whenever I have investigated similar problems, often the solution comes down to two possibilities:

- Weakness in process.
- People assigned lack skills or training.

The CMMI framework at Levels 4 and 5 provides two similar categories of causes [1]:

1. Common causes of variation (weakness in process).
2. Assignable causes of variation (process not followed).

It turns out in our case study that most of the time the hardware was ordered and installed on time. But sometimes special circumstances occur which perturbs the normal flow of work requiring a *person* to make a *decision*. Sometimes the people

closest to the situation could make that decision, but sometimes (often due to inexperience) they decide to defer their decision impacting the normal cycle time. Often some variant of this situation is at the root of such a problem.

In our case study, we found that sometimes the hardware was not ordered on time due to missing data on a procurement requisition and an inexperienced procurement specialist who didn't know how to handle the situation. We also found that sometimes projects were under-bid because product impacts were not fully identified by assigned personnel. We also found that sometimes proposal assumptions proved incorrect and plans and schedules were not appropriately updated.

At this point a question arises: Are these *process* problems (common causes), or *people* problems (assignable)?

In my experience and in this case study, the answer most often turns out to be a mix of both – the process is never perfect, nor are the people.

- **Insight One:** When using Lean and Agile techniques, the distinction between causes of variation matters less than solving the problem immediately and meeting the customer needs.
- **Insight Two:** Besides solving the problem immediately, we need to take appropriate action to minimize the likelihood of the problem reoccurring. Most often this action includes both improving the process or environment and providing additional mentoring and/or training to the people.

Lesson Six: Consider Using Selective CMMI Level 4 and 5 Practices Early Together With Agile and Lean Techniques to Address Key Objectives

One reason we distinguish between types of variation at CMMI Level 4 and 5 is to help us take the right actions at the right time. At Level 4, we quantitatively manage sub-processes and at Level 5, we analyze data and take corrective action. However, this doesn't mean you can't do both at the same time. This approach is supported by the CMMI Continuous Representation. In the case study, the right answer was to provide immediate on-the-job mentoring to help workers now and – at the same time – refine the process to help others in the *future*.

A key value in the Level 4 and 5 practices of QPM and CAR is their potential to help an organization focus on critical

sub-processes related to business objectives [1]. These focused efforts allow us to gain critical quantitative data leading to timely actions which, in turn, help achieve business objectives faster.

This same line of reasoning can be employed to help process improvement groups *win the battle of the budget* for continued organizational investment in the higher CMMI level practices [16]. Too often after receiving a Level 3 rating, organizational investment in process improvement dries up. This is unfortunate when the greatest business value may lie just ahead.

A Commonly Held Myth About the CMMI Framework

It is well known that Agile proponents *value individuals and interactions over processes and tools* [12]. It is also well known that often when using the CMMI framework, there is a tendency to try to separate *process issues* from *people issues* [2].

The belief that *people issues* fall outside the CMMI is a myth that rests at the heart of why customers do not see a difference between higher maturity level organizations as well as why many companies are failing to achieve their business objectives when using this model.

This myth is partially rooted in a misunderstanding (and misapplication) of the Generic Practices. I have heard the comment that *the Generic Practices are the same for each process area so there is nothing specific we need to do*. This is incorrect.

As an example, Generic Practices 2.3, 2.4, and 2.5 relate to the activities of providing resources, assigning responsibilities, and providing training [1]. The training of *people* expected by the model is specific to each process area. Therefore, *people issues* fall inside, not outside, the CMMI framework. But keep in mind that nothing says this training cannot be *Lean training* such as *on-the-job* and *just-in-time*.

An Example of the Right People Taking the Right Actions at the Right Time

On a recent Standard CMMI Assessment Method for Process Improvement A (SCAMPI A) appraisal in which I participated, a question was raised with respect to the adequacy of the *organization's measurement repository* which was distributed rather than centralized. When questioned, one of the developers commented, *it works better for us because we carry the measures forward*. His point was that the focus in the company (which uses both Agile and Lean techniques) was less on archiving data in a



Get Your Free Subscription

Fill out and send us this form.

517 SMXS/MXDEA

6022 FIR AVE

BLDG 1238

HILL AFB, UT 84056-5820

FAX: (801) 777-8069 DSN: 777-8069

PHONE: (801) 775-5555 DSN: 775-5555

Or request online at www.stsc.hill.af.mil

NAME: _____

RANK/GRADE: _____

POSITION/TITLE: _____

ORGANIZATION: _____

ADDRESS: _____

BASE/CITY: _____

STATE: _____ ZIP: _____

PHONE: (____) _____

FAX: (____) _____

E-MAIL: _____

CHECK BOX(ES) TO REQUEST BACK ISSUES:

FEB2007 CMMI

MAR2007 SOFTWARE SECURITY

APR2007 AGILE DEVELOPMENT

MAY2007 SOFTWARE ACQUISITION

JUNE2007 COTS INTEGRATION

JULY2007 NET-CENTRICITY

AUG2007 STORIES OF CHANGE

SEPT2007 SERVICE-ORIENTED ARCH.

OCT2007 SYSTEMS ENGINEERING

NOV2007 WORKING AS A TEAM

DEC2007 SOFTWARE SUSTAINMENT

JAN2008 TRAINING AND EDUCATION

FEB2008 SMALL PROJECTS, BIG ISSUES

MAR2008 THE BEGINNING

APR2008 PROJECT TRACKING

TO REQUEST BACK ISSUES ON TOPICS NOT LISTED ABOVE, PLEASE CONTACT <STSC.CUSTOMERSERVICE@HILL.AF.MIL>.

centralized repository, and more on *project personnel* analyzing the current data and *taking action* to carry the measures forward, providing *timely* process improvements to help the current active projects.

Agile practices, such as incremental planning, continuous measurement, and retrospective [11, 12], along with Lean practices of eliminating waste, amplifying learning, and delivering as fast as possible [7], provide proven ways that can help an organization achieve its business objectives faster. These techniques can also comply with CMMI practices including the capture of measurements, causal analysis, and taking action to improve [1] – while avoiding the pitfalls observed in our case study.

Conclusion

Ask yourself this: Is your customer seeing the results of your process improvement efforts? If not, do you understand your real *as-is* process? These are the first questions to ask that will lead to measuring the right things.

If you use the CMMI framework, you can also gain the benefits of Lean and Agile techniques. Involve and listen to your people in the trenches to help find the right things to measure leading to timely improvement actions. Lean and Agile techniques are not only compatible with the CMMI framework, they can facilitate your CMMI implementation and help you achieve your business objectives faster.

Many high-process maturity organizations today are integrating Agile and Lean techniques into their CMMI-compliant processes. If your organization isn't moving in this direction you may soon find yourself trailing the competition. ♦

References

1. Chrissis, Mary Beth, Mike Konrad, and Sandy Shrum. CMMI Guidelines for Process Integration and Product Improvement. 2nd ed. Addison-Wesley Professional, 2006.
2. Hefner, Rick. "How High Process Maturity Projects Fail." Proc. from the 2006 Software and Systems Technology Conference, Salt Lake City, UT.
3. Humphrey, Watts. A Discipline for Software Engineering. Addison-Wesley, 1995.
4. Clements, Paul. Software Product Lines. Addison-Wesley, 2002.
5. Florac, Carleton. Measuring the Software Process. Pearson Education, 1999.
6. Wheeler, Don. Understanding Variation. 2nd ed. 2000.
7. Anderson, Callison. Challenges Faced By Military Programs and Lean

Software Development. Proc. of Systems and Software Technology Conference, Tampa, FL, 2007.

8. Goldratt, Eli. The Goal. 3rd ed, Northriver Press, 2004.
9. Kennedy, Michael. Product Development for the Lean Enterprise. Oaklea Press, 2003.
10. Liker, Jeffrey. The Toyota Way. McGraw-Hill, 2004.
11. Schwaber, Ken. Agile Project Management With Scrum. Microsoft Press, 2004.
12. Cockburn, Alistair. Agile Software Development. Addison-Wesley, 2006.
13. George, Mike, David Rowlands, and Bill Kastle. What Is Lean Six Sigma? McGraw-Hill, 2004.
14. Pande, Holpp. What Is Six Sigma? McGraw-Hill, 2002.
15. Cockburn, Alistair. "What Engineering Has In Common With Manufacturing and Why It Matters." CROSSTALK, Apr. 2007.
16. Reifer, Don. "Profiles of Level 5 CMMI Organizations." CROSSTALK, Jan. 2007.

Did this article pique your interest?

You can hear more at the 2008 Systems and Software Technology Conference April 29-May 2, 2008, in Las Vegas, NV. Paul E. McMahon will present in Track 5 on Tuesday, April 2, at 2:40 p.m.

About the Author



Paul E. McMahon, principal of PEM Systems, helps large and small organizations as they move toward increased agility. He has

taught software engineering, conducted workshops on engineering process and management, published articles on Agile software development, and authored "Virtual Project Management: Software Solutions for Today and the Future." McMahon is a frequent speaker at industry conferences, and is a certified ScrumMaster. He has more than 25 years of engineering and management experience working for companies, including Hughes and Lockheed Martin.

PEM Systems

118 Matthews ST

Binghamton, NY 13905

Phone: (607) 798-7740

E-mail: pemcmahon@acm.org